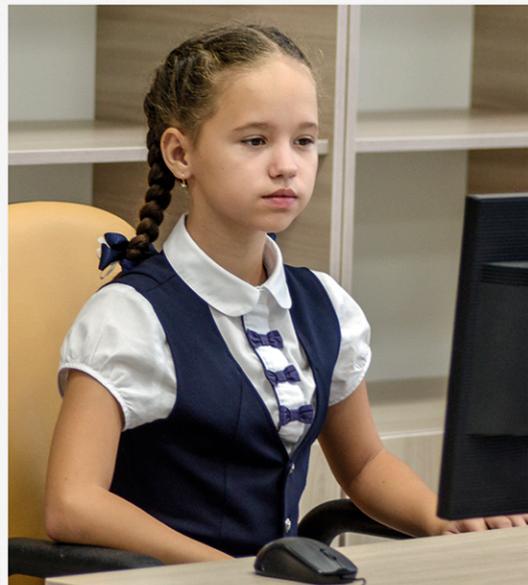
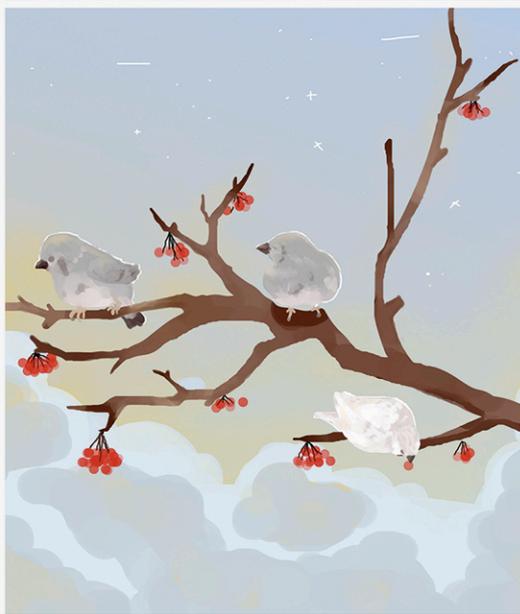
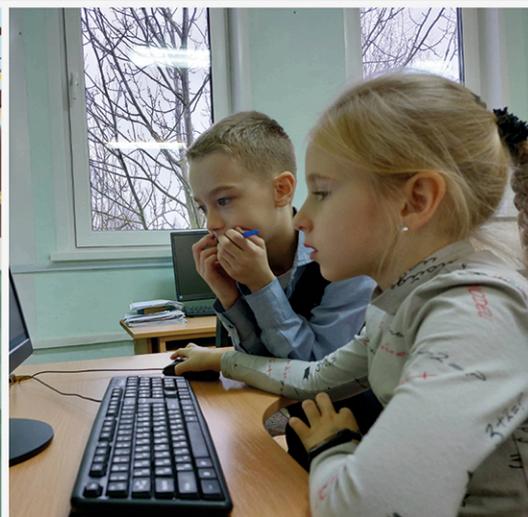


ИНФОРМАТИКА В ШКОЛЕ

№ 1'2022

ISSN 2221-1993

www.infojournal.ru





1С:Образование

Система организации и поддержки учебного процесса онлайн

Онлайн-система предназначена для организации электронного обучения и включения дистанционных образовательных технологий в учебный процесс в школе или колледже.



Функциональные возможности

- Ориентированная на образовательную организацию система администрирования пользователей.
- Учет особенностей организации учебного процесса в конкретной школе или колледже.
- Цифровая библиотека учебных пособий по всем основным общеобразовательным дисциплинам.
- Десятки тысяч интерактивных мультимедийных образовательных ресурсов в составе библиотеки.
- Инструменты для создания собственных цифровых учебных материалов различного дидактического назначения.
- Назначение учащимся групповых и индивидуальных заданий с автоматической проверкой.
- Детальное информирование преподавателя о ходе и результатах самостоятельной учебной деятельности учащегося.
- Совместное использование с любыми системами видеоконференцсвязи для проведения онлайн-занятий.

Преимущества использования

- Отсутствие затрат на развертывание, администрирование и эксплуатацию системы в сети образовательной организации.
- Отдельная база данных для каждой школы или колледжа.
- Неограниченное количество классов и групп, преподавателей и учащихся.
- Регулярно обновляемая цифровая библиотека учебных пособий.
- Низкая стоимость подключения и простота в использовании.

**Заполните заявку на сайте
и получите бесплатный тестовый
доступ на 30 календарных дней.**



ИНФОРМАТИКА В ШКОЛЕ

ИЗДАЕТСЯ С СЕНТЯБРЯ 2002 ГОДА

№ 1 (174) февраль 2022

Содержание

КОНКУРС ИНФО-2021

Итоги XVIII Всероссийского конкурса научно-практических работ ИНФО-2021	4
Климина Н. В. Веб-подход к созданию VR-контента при проведении занятий в VR-квантуме мини-технопарка «Кванториум»	7
Курганова Н. А. Использование технологии цифрового сторителлинга при изучении темы «Компьютерные вирусы» на уроке информатики	16
Фоминых И. А., Довранов А. Р. О разработке курса по выбору «Основы программирования на Python» для предпрофильной подготовки школьников	22
Блинов Д. М. Образовательные анимационные видеоролики в школьном образовании	30
Баракина Т. В. Формирование знаково-символической деятельности у младших школьников в процессе ознакомления с информационными процессами	34

ИНТЕГРИРОВАННЫЕ УРОКИ

Сальникова О. С., Селяничев О. Л. Особенности занятий по геоинформационным технологиям со школьниками в рамках дополнительного образования	40
Букина Е. Ю. Интегрированный урок по литературе и информатике «Мужичок с ноготок» (VI класс)	46

МЕТОДИЧЕСКАЯ КОПИЛКА

Маркушевич М. В. Элементы методики преподавания темы «Создание и обработка звуковой информации» с использованием свободного звукового редактора Audacity	55
Дженжер В. О., Денисова Л. В. Динамические массивы и списки в PascalABC.NET	67

ЗАДАЧИ

Магомедов А. М., Раджабова Н. Ш. Замощение клетчатой полосы шириной 4	81
--	----

КОНКУРСЫ

Итоги XXII Всероссийского конкурса цифровых изображений и фотографий ФОТО 1-2022	85
--	----

Свидетельство о регистрации
средства массовой информации
ПИ № 77-12068 от 11 марта 2002 г.

Издатель ООО «Образование и Информатика»
119261, г. Москва, Ленинский пр-т, д. 82/2, комн. 6
Телефон: (495) 140-19-86
E-mail: readinfo@infojournal.ru
Сайт издательства: <http://infojournal.ru/>
Почтовый адрес: 119270, г. Москва, а/я 15

Подписано в печать 28.02.22.
Формат 60×90^{1/8}. Усл. печ. л. 11,0
Тираж 2000 экз. Заказ № 1670.
Отпечатано в типографии ООО «Принт сервис групп»,
105187, г. Москва, Борисовская ул., д. 14, стр. 6,
тел./факс: (499) 785-05-18, e-mail: 3565264@mail.ru

© «Образование и Информатика», 2022



ISSN 2221-1993

Журнал входит в Перечень российских рецензируемых научных изданий ВАК, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук

Контакты

Главный редактор:
akull@mail.ru
Редакция
readinfo@infojournal.ru
Отдел распространения
info@infojournal.ru
Телефон:
+7 (495) 140-1986
Почтовый адрес
119270, Москва, а/я 15
Сайт журнала
<http://school.infojournal.ru>

ОБЪЕДИНЕННАЯ РЕДАКЦИЯ ИНФО

**Главный редактор журнала
«Информатика и образование»**
ГРИГОРЬЕВ Сергей Георгиевич

**Главный редактор журнала
«Информатика в школе»**
БОСОВА Людмила Леонидовна

Директор издательства
РЫБАКОВ Даниил Сергеевич

Научный редактор
ДЕРГАЧЕВА Лариса Михайловна

Ведущий редактор
КИРИЧЕНКО Ирина Борисовна

Корректор
ШАРАПКОВА Людмила Михайловна

Верстка
ФЕДОТОВ Дмитрий Викторович

Дизайн
ГУБКИН Владислав Александрович

Отдел распространения и рекламы
КУЗНЕЦОВА Елена Александровна

УЧРЕДИТЕЛЬ:
ИЗДАТЕЛЬСТВО
«ОБРАЗОВАНИЕ И ИНФОРМАТИКА»

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ «ИНФОРМАТИКА В ШКОЛЕ»

ГЛАВНЫЙ РЕДАКТОР

БОСОВА Людмила Леонидовна

доктор пед. наук, доцент, заслуженный учитель РФ, лауреат премии Правительства РФ в области образования, Институт математики и информатики Московского педагогического государственного университета, зав. кафедрой теории и методики обучения математике и информатике

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

БАРАКИНА Татьяна Вячеславовна

канд. пед. наук, доцент, Омский государственный педагогический университет, доцент кафедры предметных технологий начального и дошкольного образования

БЕШЕНКОВ Сергей Александрович

доктор пед. наук, профессор, Институт управления образованием РАО, Москва, руководитель Центра информатизации образования

ВОЕВОДИН Владимир Валентинович

чл.-корр. РАН, доктор физ.-мат. наук, профессор, лауреат премии Правительства РФ в области образования, Научно-исследовательский вычислительный центр МГУ имени М. В. Ломоносова, директор

ДЕРГАЧЕВА Лариса Михайловна

канд. пед. наук, доцент, научный редактор издательства «Образование и Информатика»

ЗАСЛАВСКАЯ Ольга Юрьевна

доктор пед. наук, профессор, Институт цифрового образования Московского городского педагогического университета, научный руководитель департамента информатизации образования

ЗЕНКИНА Светлана Викторовна

доктор пед. наук, профессор, Академия социального управления, Московская область, профессор кафедры общеобразовательных дисциплин

ЗУБРИЛИН Андрей Анатольевич

канд. филос. наук, доцент, Мордовский государственный педагогический университет имени М. Е. Евсевьева, г. Саранск, зав. кафедрой информатики и вычислительной техники

КИРИЧЕНКО Ирина Борисовна

заместитель главного редактора журнала «Информатика в школе»

ЛАПТЕВ Владимир Валентинович

академик РАО, доктор пед. наук, канд. физ.-мат. наук, профессор, Российский государственный педагогический университет им. А. И. Герцена, Санкт-Петербург, первый проректор

ЛЕВЧЕНКО Ирина Витальевна

доктор пед. наук, профессор, Институт цифрового образования Московского городского педагогического университета, научный руководитель департамента информатики, управления и технологий

РЫБАКОВ Даниил Сергеевич

канд. пед. наук, доцент, директор издательства «Образование и Информатика»

СЛИНКИНА Ирина Николаевна

канд. пед. наук, доцент, Шадринский государственный педагогический университет, декан факультета информатики, математики и естественных наук

ФЕДОСОВ Александр Юрьевич

доктор пед. наук, доцент, Российский государственный социальный университет, Москва, профессор факультета информационных технологий

ВСЕРОССИЙСКИЙ КОНКУРС «ШКОЛЬНАЯ ИНФОРМАТИКА: ВЧЕРА, СЕГОДНЯ, ЗАВТРА»

Уважаемые коллеги!

Издательство «Образование и Информатика», редакция журнала «Информатика в школе»
объявляют о проведении
Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»,
посвященного 20-летию журнала «Информатика в школе»

Конкурс проводится по двум номинациям:

1. Номинация для педагогов: Школьная информатика: вчера, сегодня, завтра.

В номинации могут быть представлены работы самой разнообразной тематики, отражающие вопросы преподавания информатики и информационных технологий в школе. Приветствуются как материалы, в ретроспективе обзорающие те или аспекты обучения информатике, так и работы, отражающие инновационные подходы к преподаванию дисциплины, а также к использованию информационных технологий в школьном образовании.

2. Номинация для учащихся: Школьная информатика: заглянем в будущее.

В номинации могут быть представлены работы учащихся (статьи, рассказы, эссе, пьесы, стихи, литературные зарисовки), отражающие видение школьниками того, каким будет предмет «Информатика» в будущем, как будут использоваться информационные технологии в разных школьных предметах.

Оргкомитет конкурса

Руководит конкурсом Организационный комитет (Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов редакционных коллегий журналов «Информатика и образование» и «Информатика в школе», сотрудников объединенной редакции журналов.

Цели и задачи конкурса

1. Выявление и поддержка талантливых педагогов.
2. Включение педагогов в деятельность по разработке нового содержания образования, новых образовательных технологий и методик обучения.
3. Выявление и распространение новых образовательных технологий, способствующих развитию интереса школьников к информатике и информационным технологиям.
4. Создание информационно-образовательного пространства на страницах журнала «Информатика в школе» по обмену и распространению опыта преподавания информатики, а также использования в образовательном процессе информационно-коммуникационных технологий.
5. Развитие интереса школьников к информатике и другим учебным дисциплинам, к информационным технологиям.
6. Творческое развитие школьников, повышение их социальной активности, создание условий для самореализации.
7. Повышение информационной культуры и информационно-коммуникационной компетентности всех участников образовательного процесса — учащихся, педагогов, родителей.

Работы на конкурс принимаются с 5 апреля по 25 мая 2022 года включительно. Работы, присланные позже 25 мая 2022 года, к участию в конкурсе допускаться не будут. Подача работ производится только через заполнение формы заявки на сайте ИНФО (необходима предварительная регистрация на сайте или авторизация для зарегистрированных пользователей).

Итоги конкурса будут подведены в № 3-2022 (июнь) журнала «Информатика в школе», а также опубликованы на сайте издательства «Образование и Информатика».

Лучшие работы будут опубликованы в журнале «Информатика в школе».

Победители конкурса получат:

- диплом от издательства «Образование и Информатика»;
- печатный экземпляр журнала «Информатика в школе» № 3-2022 (июнь), в котором будут опубликованы итоги конкурса;
- авторский печатный экземпляр журнала «Информатика в школе» с опубликованной работой.

**Подробную информацию
о требованиях к оформлению конкурсной работы и конкурсной заявки,
а также всю дополнительную информацию о конкурсе
вы можете найти на сайте издательства «Образование и Информатика»:**

<http://infojournal.ru/competition/informatics-2022/>

а также получить в редакции ИНФО

по адресу: readinfo@infojournal.ru

и по телефону: (495) 140-19-86



ИТОГИ XVIII ВСЕРОССИЙСКОГО КОНКУРСА НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ИНФО-2021

Уважаемые коллеги!

В октябре 2021 года издательство «Образование и Информатика» объявило конкурс научно-практических работ ИНФО-2021 по методике обучения информатике и информатизации образования.

Было организовано жюри конкурса, в которое вошли представители Российской академии образования, ведущие методисты, члены редакционных коллегий журналов «Информатика и образование» и «Информатика в школе», сотрудники объединенной редакции ИНФО.

В конкурсе приняли участие как работники образования — учителя, преподаватели вузов, работники учреждений дошкольного образования, педагоги системы дополнительного образования, методисты, — так и студенты педвузов из разных регионов Российской Федерации, а также из стран ближнего зарубежья.

Конкурс проводился по трем номинациям:

1. Цифровая трансформация образования: дошкольное и начальное общее образование.
2. Цифровая трансформация образования: основное и среднее общее образование.
3. Цифровая трансформация образования: высшее образование.

Представляем лауреатов (1-е место) и дипломантов (2-е место) конкурса ИНФО-2021. Все победители представлены в алфавитном порядке.

НОМИНАЦИЯ 1

«ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ОБРАЗОВАНИЯ: ДОШКОЛЬНОЕ И НАЧАЛЬНОЕ ОБЩЕЕ ОБРАЗОВАНИЕ»

По мнению жюри, в данной номинации не было представлено работ, достойных 1-го места.

ДИПЛОМАНТ КОНКУРСА (2-е МЕСТО)

Баракина Татьяна Вячеславовна,

*доцент кафедры предметных технологий начального и дошкольного образования,
Омский государственный педагогический университет, г. Омск, Россия*

НОМИНАЦИЯ 2

«ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ОБРАЗОВАНИЯ: ОСНОВНОЕ И СРЕДНЕЕ ОБЩЕЕ ОБРАЗОВАНИЕ»

ЛАУРЕАТ КОНКУРСА (1-е МЕСТО)

Климина Наталья Владимировна,

*учитель информатики, лицей имени Героя Советского Союза П. И. Викулова,
г. Сызрань, Самарская область, Россия*

ДИПЛОМАНТЫ КОНКУРСА (2-е МЕСТО)

Блинов Дмитрий Михайлович,

*заместитель директора по УВР, средняя общеобразовательная школа № 67,
г. Ижевск, Удмуртская Республика, Россия*

Довранов Атагелди Русланович,

*учитель информатики, Мочалищенская средняя общеобразовательная школа,
п. Мочалище, Звениговский район, Республика Марий Эл, Россия*

Фоминых Ирина Анатольевна,

*доцент кафедры общеобразовательных дисциплин и методики их преподавания,
Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия*

Курганова Наталья Александровна,
*доцент кафедры прикладной информатики и математики,
Омский государственный педагогический университет, г. Омск, Россия*

НОМИНАЦИЯ 3 «ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ОБРАЗОВАНИЯ: ВЫСШЕЕ ОБРАЗОВАНИЕ»

ЛАУРЕАТЫ КОНКУРСА (1-е МЕСТО)

Логинов Анатолий Владимирович,
*старший преподаватель кафедры информационных образовательных технологий и систем,
Луганский государственный педагогический университет, г. Луганск, Луганская Народная Республика*
Панишева Ольга Викторовна,
*доцент кафедры высшей математики и методики преподавания математики,
Луганский государственный педагогический университет, г. Луганск, Луганская Народная Республика*

ДИПЛОМАНТЫ КОНКУРСА (2-е МЕСТО)

Боровская Елена Владимировна,
старший преподаватель кафедры информатики, информационных технологий и методики обучения информатике, Южно-Уральский государственный гуманитарно-педагогический университет, г. Челябинск, Россия
Дмитриева Ольга Александровна,
доцент кафедры информатики, информационных технологий и методики обучения информатике, Южно-Уральский государственный гуманитарно-педагогический университет, г. Челябинск, Россия
Леонова Елена Анатольевна,
доцент кафедры информатики, информационных технологий и методики обучения информатике, Южно-Уральский государственный гуманитарно-педагогический университет, г. Челябинск, Россия

Бурняшов Борис Анатольевич,
*доцент кафедры социально-гуманитарных и естественно-научных дисциплин,
Северо-Кавказский филиал Российского государственного университета правосудия, г. Краснодар;
доцент кафедры социальной работы, психологии и педагогики высшего образования Кубанского государственного университета, г. Краснодар, Россия*

Все представленные выше лауреаты (1-е место) и дипломанты (2-е место) конкурса ИНФО-2021 будут награждены дипломами соответствующего достоинства от издательства «Образование и Информатика». Их работы будут опубликованы в журналах «Информатика в школе» (номинации 1 и 2) и «Информатика и образование» (номинация 3).

В качестве приза победители конкурса ИНФО-2021 получат подписку в печатном виде на 2022 год на журналы «Информатика и образование» и «Информатика в школе».

Уважаемые коллеги! Обращаем ваше внимание, что с 2022 года электронные версии журналов «Информатика и образование» и «Информатика в школе» будут размещаться на сайте издательства в режиме открытого доступа (Open Access), поэтому электронная подписка победителям не оформляется (как это было в предыдущих конкурсах).

Также по результатам конкурса отмечены жюри и рекомендованы к публикации в журналах «Информатика и образование» и «Информатика в школе» работы следующих авторов:

Васева Елена Сергеевна,
доцент кафедры информационных технологий, Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия
Бужинская Надежда Владимировна,
доцент кафедры информационных технологий, Нижнетагильский государственный социально-педагогический институт (филиал) Российского государственного профессионально-педагогического университета, г. Нижний Тагил, Свердловская область, Россия

Диков Андрей Валентинович,

доцент кафедры информатики и методики обучения информатике и математике,
Пензенский государственный университет, г. Пенза, Россия

Зубрилин Андрей Анатольевич,

заведующий кафедрой информатики и вычислительной техники,
Мордовский государственный педагогический университет имени М. Е. Евсевьева,
г. Саранск, Республика Мордовия, Россия

Латышева Любовь Павловна,

доцент кафедры высшей математики и методики обучения математике,
Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Скорнякова Анна Юрьевна,

доцент кафедры высшей математики и методики обучения математике,
Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Черемных Елена Леонидовна,

и.о. заведующего кафедрой высшей математики и методики обучения математике,
Пермский государственный гуманитарно-педагогический университет, г. Пермь, Россия

Лаптева Татьяна Дмитриевна,

студентка магистратуры, математический факультет, Пермский государственный
гуманитарно-педагогический университет, г. Пермь, Россия

Мельникова Екатерина Викторовна,

ассистент кафедры высшей математики и методики обучения математике Пермский государственный
гуманитарно-педагогический университет, г. Пермь, Россия

Мазанова Ольга Юрьевна,

учитель, учебно-консультационный пункт «Республиканская детская больница»
Республиканского центра образования, г. Сыктывкар, Республика Коми, Россия

Махмутова Марина Владимировна,

доцент кафедры бизнес-информатики и информационных технологий,
Магнитогорский государственный технический университет им. Г. И. Носова,

г. Магнитогорск, Челябинская область, Россия

Шевцова Маргарита Александровна,

студентка 5-го курса кафедры бизнес-информатики и информационных технологий,
Магнитогорский государственный технический университет им. Г. И. Носова,

г. Магнитогорск, Челябинская область, Россия

Назаров Дмитрий Михайлович,

профессор, заведующий кафедрой бизнес-информатики, Уральский государственный экономический университет,
г. Екатеринбург, Свердловская область, Россия

Ковтун Денис Борисович,

ассистент кафедры бизнес-информатики, Уральский государственный экономический университет,
г. Екатеринбург, Свердловская область, Россия

Полякова Анна Юрьевна,

аспирант Института математики, естествознания и техники, специалист по учебно-методической работе
отдела планирования, мониторинга образовательной деятельности и оценки качества,
Елецкий государственный университет им. И. А. Бунина, г. Елец, Липецкая область, Россия

Участники конкурса, чьи работы рекомендованы к публикации, получают сертификат об участии в конкурсе и публикации вместе с авторским экземпляром журнала, в котором будет опубликована работа.

Остальные конкурсанты могут получить сертификат об участии, который будет подготовлен по индивидуальному запросу.

Следите за информацией о новых конкурсах в журналах «Информатика и образование» и «Информатика в школе», а также на сайте ИНФО: <http://www.infojournal.ru/>

DOI: 10.32517/2221-1993-2022-21-1-7-15

Н. В. Климина

победитель конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: основное и среднее общее образование», лицей имени Героя Советского Союза П. И. Видулова городского округа Сызрань Самарской области, Россия

ВЕБ-ПОДХОД К СОЗДАНИЮ VR-КОНТЕНТА ПРИ ПРОВЕДЕНИИ ЗАНЯТИЙ В VR-КВАНТУМЕ МИНИ-ТЕХНОПАРКА «КВАНТОРИУМ»*

Аннотация

В статье предложено методическое сопровождение занятий в VR-квантуме мини-технопарка «Кванториум» по созданию VR-контента с помощью веб-технологий. Описание практических работ позволяет учащимся пошагово научиться создавать виртуальные миры, формируя код в простейшем текстовом редакторе Блокнот. Создание виртуального мира напоминает создание веб-страниц на языке HTML. Платформенная независимость и доступность программного обеспечения для создания виртуальной реальности позволяют использовать такой веб-подход как в школе на занятиях мини-технопарка, так и при самостоятельной работе учащихся дома. Данный подход идеально подходит для проведения дистанционных занятий. Проведение занятий в VR-квантуме на базе предлагаемой методики будет способствовать формированию у обучающихся навыков создания VR-контента для шлемов виртуальной реальности и умения применять информационные технологии для решения творческих задач в условиях информационного общества.

Ключевые слова: виртуальная реальность, виртуальный мир, фреймворк, виртуальное пространство, сцена виртуальной реальности, управление сценой виртуальной реальности, событие, «Кванториум».

1. Технология виртуальной реальности. Спецификация WebVR API

Виртуальная, дополненная и смешанная реальности — это активно развивающиеся в настоящее время направления информационных технологий. Если некоторое время назад виртуальная реальность ассоциировалась у нас с играми и кинофильмами, то сейчас применение указанных технологий разнообразно: образование, наука, производство. И можно с уверенностью сказать, что это технологии будущего.

Виртуальная реальность (*англ.* Virtual Reality, VR) — настоящий тренд современности. С ней невероятно удобно и интересно работать. В последнее время наблюдается активный ее перенос в браузеры — современные браузеры включают в себя поддержку виртуальной и дополненной реальностей.

Спецификация **WebVR API** была представлена компанией Mozilla в 2016 году для описания программного интерфейса для работы с VR-устройствами Oculus Rift и им подобными. Разработчиками этой спецификации являются Брэндон Джонс (Brandon Jones) из компании

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2022/

Контактная информация

Климина Наталья Владимировна, учитель информатики, лицей имени Героя Советского Союза П. И. Видулова городского округа Сызрань Самарской области, Россия; *адрес:* 446001, Россия, Самарская область, г. Сызрань, ул. Степана Разина, д. 1; *e-mail:* natklimina@yandex.ru

N. V. Klimina

Lyceum named after the Hero of the Soviet Union P. I. Vikulov of the Syzran Urban District of the Samara Region, Russia

WEB APPROACH TO CREATING VR CONTENT DURING LESSONS IN THE VR QUANTUM OF THE MINI TECHNOPARK "QUANTORIUM"

Abstract

The article proposes methodological support for classes in the VR quantum of the mini technopark "Quantorium" for creating VR content using web technologies. The description of practical works allows students to learn how to create virtual worlds step by step by generating code in the simplest text editor Notepad. Creating a virtual world is like creating web pages in HTML. Platform independence and the availability of software for creating virtual reality make it possible to use such a web-based approach both at school in the mini technopark classes and when students work independently at home. This approach is ideal for distance learning. Conducting classes in a VR quantum based on the proposed methodology will help students develop the skills to create VR content for virtual reality helmets and the ability to apply information technologies to solve creative problems in the information society.

Keywords: virtual reality, virtual world, framework, virtual space, virtual reality scene control, event, "Quantorium".

Google и Джастин Роджерс (Justin Rogers) из Microsoft. WebVR API — это программный интерфейс для работы с устройствами VR: очками виртуальной реальности и прилагаемыми к ним контроллерами. Этот программный интерфейс не работает с 3D-графикой, а лишь предоставляет инструменты для рендеринга сцены, для получения информации о VR-устройстве, о его возможностях и технических характеристиках. Работа с графикой, создание сцены, установка источников света, анимация объектов и прочее реализуются программистами. 3D-мир нужно создавать, используя уже ставшие привычными веб-технологии, такие как HTML, CSS, WebGL, Canvas и т. д. [2, 5, 7].

Поддержка спецификации WebVR реализована в браузерах Microsoft Edge, Mozilla Firefox, Yandex, Google Chrome и в мобильных браузерах.

Для создания VR-контента, воспроизводимого в браузере, можно использовать стандартное приложение Блокнот, фреймворк A-Frame (<https://aframe.io/>), онлайн-сервис Glitch (<https://glitch.com/>) и другие фреймворки [8–10].

2. Реализация веб-подхода к созданию VR-контента на занятиях мини-технопарка «Кванториум»

В VR-квантуме* мини-технопарка «Кванториум», организованного на базе лицея имени Героя Советского

* Квантум — направление, по которому происходит обучение (творческая лаборатория).

Союза П. И. Викулова городского округа Сызрань Самарской области, занимаются ученики VIII—IX классов. Для создания VR-контента на занятиях VR-квантума мини-технопарка «Кванториум» используется приложение Блокнот.

Создание контента виртуальной реальности напоминает создание веб-страниц: в коде прописываем подключение библиотек фреймворка A-Frame (<https://aframe.io/>) и просматриваем готовый результат в браузерах Microsoft Edge и Yandex. Платформенная независимость и доступность программного обеспечения для создания виртуальной реальности позволяют использовать такой веб-подход как в лицее на занятиях мини-технопарка, так и при самостоятельной работе учащихся дома. Данный подход идеально подходит для проведения дистанционных занятий.

На *теоретических* занятиях изучается концепция создания виртуальных сцен с помощью тэгов языка фреймворка A-Frame, анализируются готовые образцы с сайта используемого фреймворка, а на *практических* занятиях происходит отработка навыков создания виртуальных сцен в приложении Блокнот.

План занятий-практикумов в технопарке приведен в таблице.

Каждая практическая работа представляет собой пошаговое руководство по созданию сцены виртуальной реальности.

Каждый модуль завершается созданием творческого мини-проекта.

В разделе 3 приведены примеры двух практических работ [1, 3, 4, 9].

Таблица

План практических занятий

№ п/п	Практические работы	Кол-во часов
Модуль 1 «Создаем виртуальное пространство и окружение»		4
1	Практическая работа 1 «Создание сцены виртуальной реальности». <i>Цель работы:</i> формирование навыков использования шаблона сцены виртуальной реальности и параметров среды для создания виртуальной местности	1
2	Практическая работа 2 «Освещение сцены виртуальной реальности». <i>Цель работы:</i> формирование навыков использования источников света для создания освещенности виртуальной местности	1
3	Практическая работа 3 «Создание панорамных сцен виртуальной реальности». <i>Цель работы:</i> формирование навыков использования панорамных фотографий для создания сцены виртуальной реальности	1
4	Практическая работа 4 «Земная поверхность в сцене виртуальной реальности». <i>Цель работы:</i> формирование навыков использования параметров плоскости для создания реалистичной земной поверхности виртуальной местности	1
Модуль 2 «Населяем виртуальный мир»		5
5	Практическая работа 5 «Базовые примитивы». <i>Цель работы:</i> формирование навыков размещения простейших 3D-объектов в виртуальной сцене и редактирования их свойств	1
6	Практическая работа 6 «Наложение текстур на объекты». <i>Цель работы:</i> формирование навыков наложения текстур на 3D-объекты в виртуальной сцене	1
7	Практическая работа 7 «Размещение трехмерных моделей на сцене». <i>Цель работы:</i> формирование навыков размещения пользовательских трехмерных моделей в виртуальной сцене	1

Окончание таблицы

№ п/п	Практические работы	Кол-во часов
8	Практическая работа 8 «Текстовые эффекты». <i>Цель работы:</i> формирование навыков использования плоских и трехмерных текстовых объектов в виртуальной сцене	1
9	Практическая работа 9 «Параметрические объекты». <i>Цель работы:</i> формирование навыков использования плоских и объемных кривых в сцене виртуальной реальности	1
	Модуль 3 «Анимация и управление сценой виртуальной реальности»	7
10	Практическая работа 10 «Анимация объектов». <i>Цель работы:</i> формирование навыков анимирования объектов в сцене виртуальной реальности	1
11	Практическая работа 11 «Управление объектами с клавиатуры». <i>Цель работы:</i> формирование навыков использования клавиш клавиатуры для управления объектами в сцене виртуальной реальности	1
12	Практическая работа 12 «Управление объектами с экрана». <i>Цель работы:</i> формирование навыков использования экранных элементов для управления объектами в сцене виртуальной реальности	1
13	Практическая работа 13 «Действие законов физики в виртуальной реальности». <i>Цель работы:</i> формирование навыков использования свойств объектов для имитации законов физики в сцене виртуальной реальности	1
14	Практическая работа 14 «Эффекты частиц». <i>Цель работы:</i> формирование навыков использования эффектов взаимодействия частиц в сцене виртуальной реальности	1
15	Практическая работа 15 «Гиперссылки в виртуальном мире». <i>Цель работы:</i> формирование навыков создания гиперсвязей в сцене виртуальной реальности	1
16	Практическая работа 16 «Управление с помощью контроллеров». <i>Цель работы:</i> формирование навыков подключения и использования контроллеров от VR-гарнитуры в сцене виртуальной реальности	1

3. Примеры практических работ VR-квантума

3.1. Практическая работа 1 «Создание сцены виртуальной реальности»

Цель работы: формирование навыков использования шаблона сцены виртуальной реальности и параметров среды для создания виртуальной местности [4, 6, 9].

Ход работы.

1. Открыть приложение Блокнот.
2. В текстовом документе разместить код веб-страницы:

```
<html>
<head>
  <title>Моя первая сцена</title>
  <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-environment-component/dist/aframe-environment-component.min.js"></script>
</head>
<body>
  <a-scene>
```

```
<a-camera>
</a-camera>
</a-scene>
</body>
</html>
```

3. Сохранить страницу в файл с расширением html.
4. Открыть файл в браузере. Признаком присутствия в виртуальном пространстве является отображение курсора мыши в виде ладони в области страницы браузера.
5. Подключить встроенный шаблон местности из библиотеки шаблонов:

```
<html>
<head>
  <title>Моя первая сцена</title>
  <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-environment-component/dist/aframe-environment-component.min.js"></script>
</head>
<body>
  <a-scene environment="preset: starry">
    <a-camera>
  </a-camera>
```

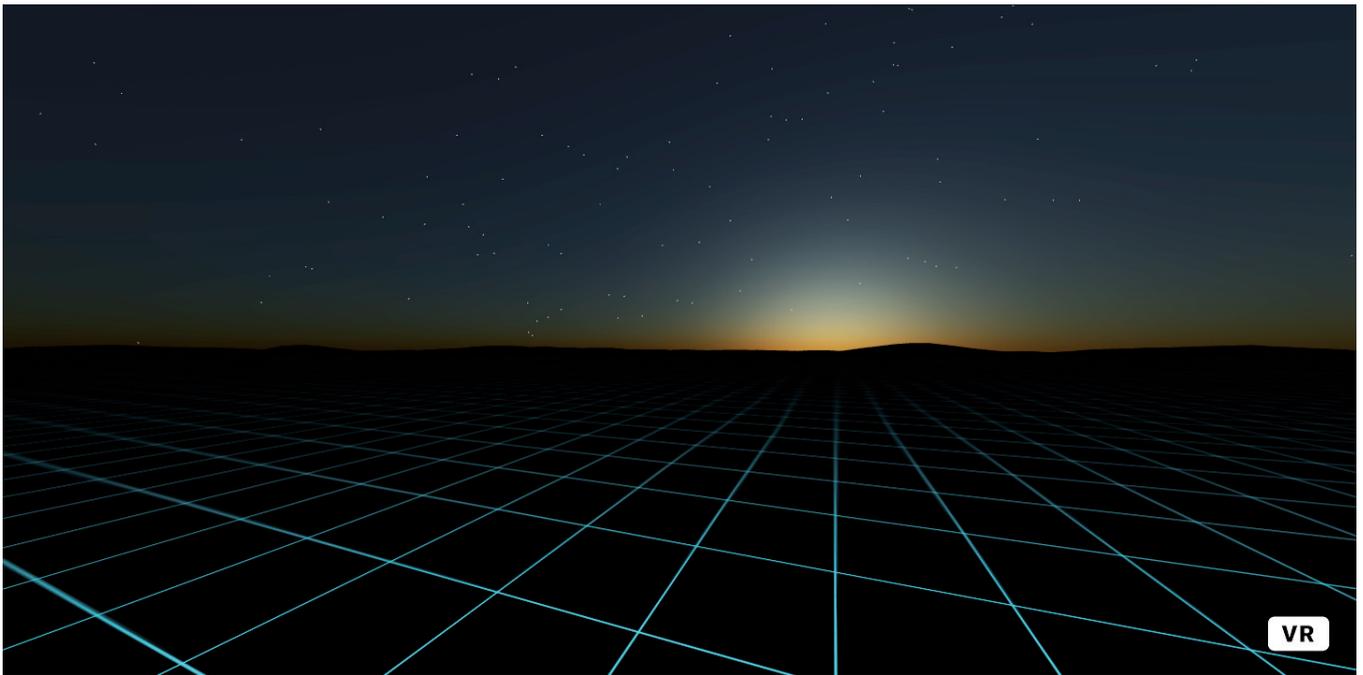


Рис. 1. Начальная виртуальная сцена

```
</a-scene>
</body>
</html>
```

6. Обновить браузер и наблюдать изменения в виртуальной сцене (рис. 1).
7. Применить другие шаблоны местности из библиотеки шаблонов.
8. Установить параметры шаблона сцены **starry**:

```
<html>
<head>
<title>Моя первая сцена</title>
```

```
<script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
<script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
<a-scene environment="preset: starry;
groundTexture: walkernoise; grid: none">
<a-camera>
</a-camera>
</a-scene>
```

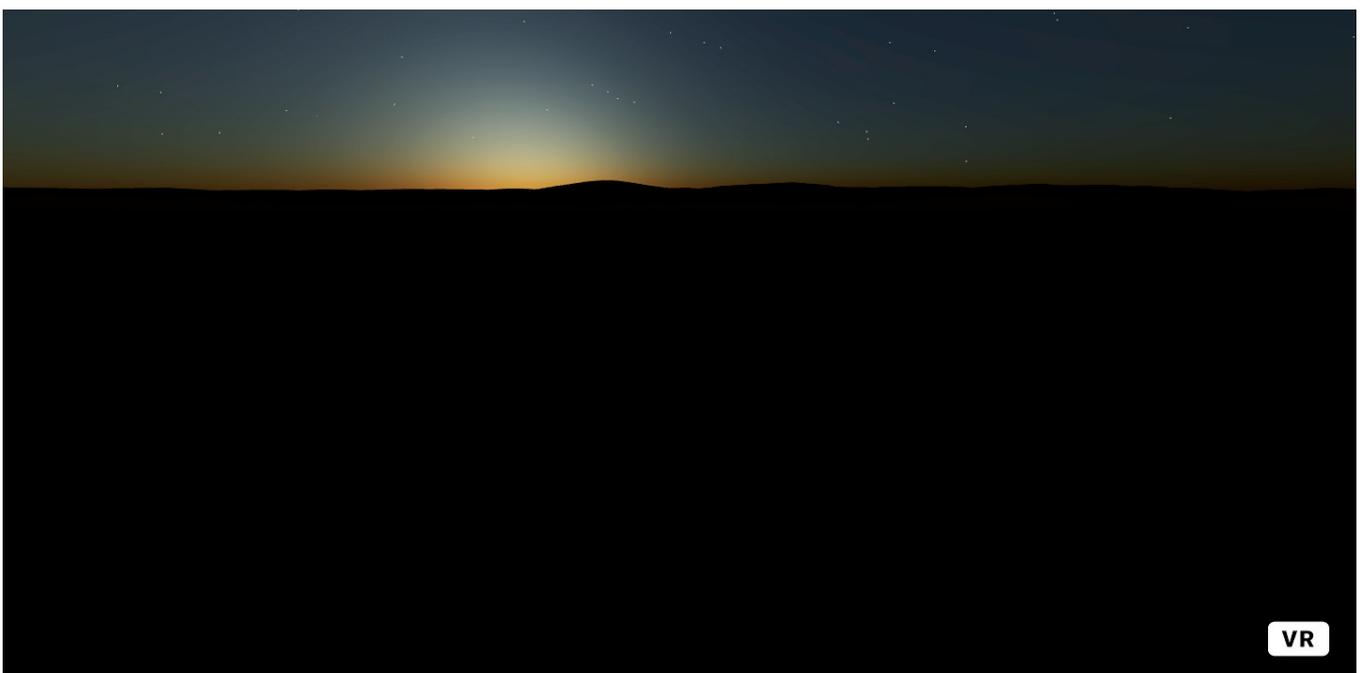


Рис. 2. Модифицированная виртуальная сцена

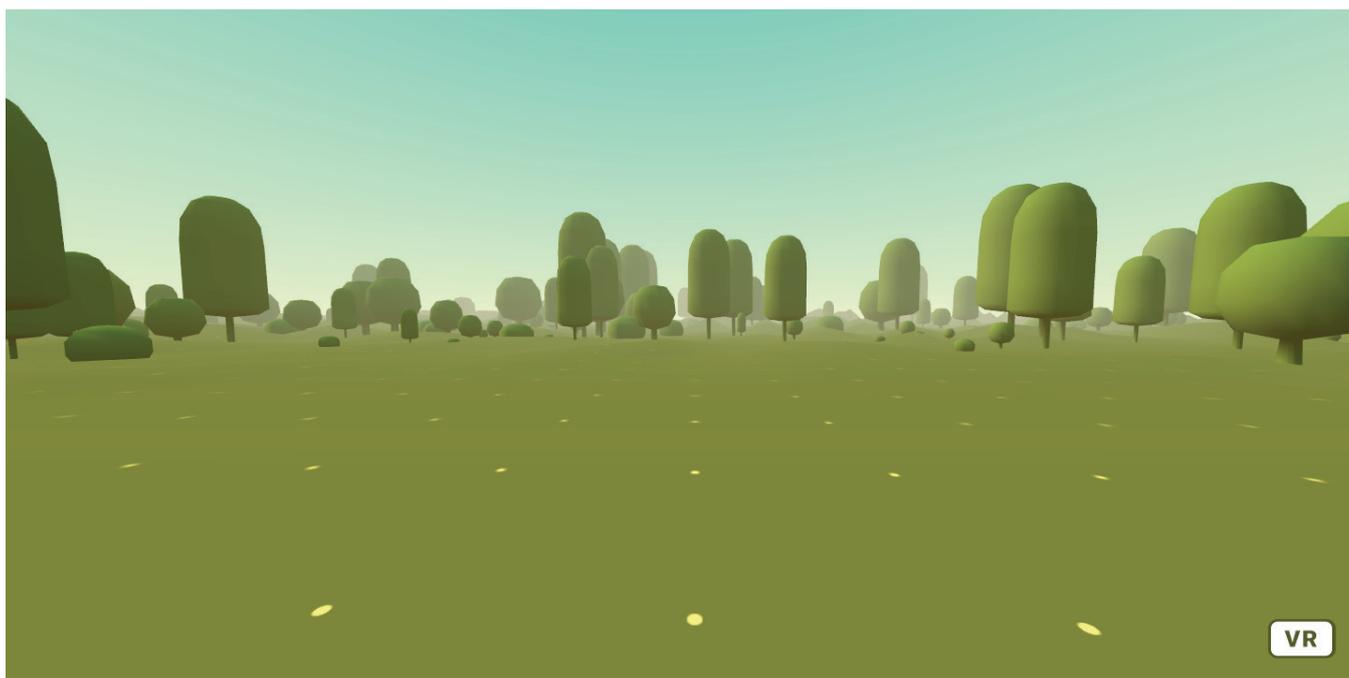


Рис. 3. Установка параметров виртуальной сцены

```
</body>
</html>
```

9. Обновить браузер и наблюдать изменения в виртуальной сцене (рис. 2).

10. Установить параметры шаблона сцены **forest**:

```
<html>
<head>
<title>Моя первая сцена</title>
<script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
<script src="https://unpkg.com/aframe-
```

```
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
<a-scene environment="preset: forest;
groundTexture: walkernoise; dressingAmount:
300; grid: dots">
<a-camera>
</a-camera>
</a-scene>
</body>
</html>
```



Рис. 4. Установка параметров неба виртуальной сцены

11. Обновить браузер и наблюдать изменения в виртуальной сцене (рис. 3).
12. Добавить реалистичное небо:

```
<html>
<head>
  <title>Моя первая сцена</title>
  <script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene environment="preset: forest;
groundTexture: walkernoise; dressingAmount:
300; grid: dots; skyType: atmosphere">
    <a-camera>
  </a-camera>
  </a-scene>
</body>
</html>
```

13. Обновить браузер и наблюдать изменения в виртуальной сцене (рис. 4).

3.2. Практическая работа 6

«Наложение текстур на объекты»

Цель работы: формирование навыков наложения текстур на 3D-объекты в виртуальной сцене [1, 8–10].

Ход работы.

1. Открыть приложение Блокнот.
2. В текстовом документе разместить код веб-страницы:

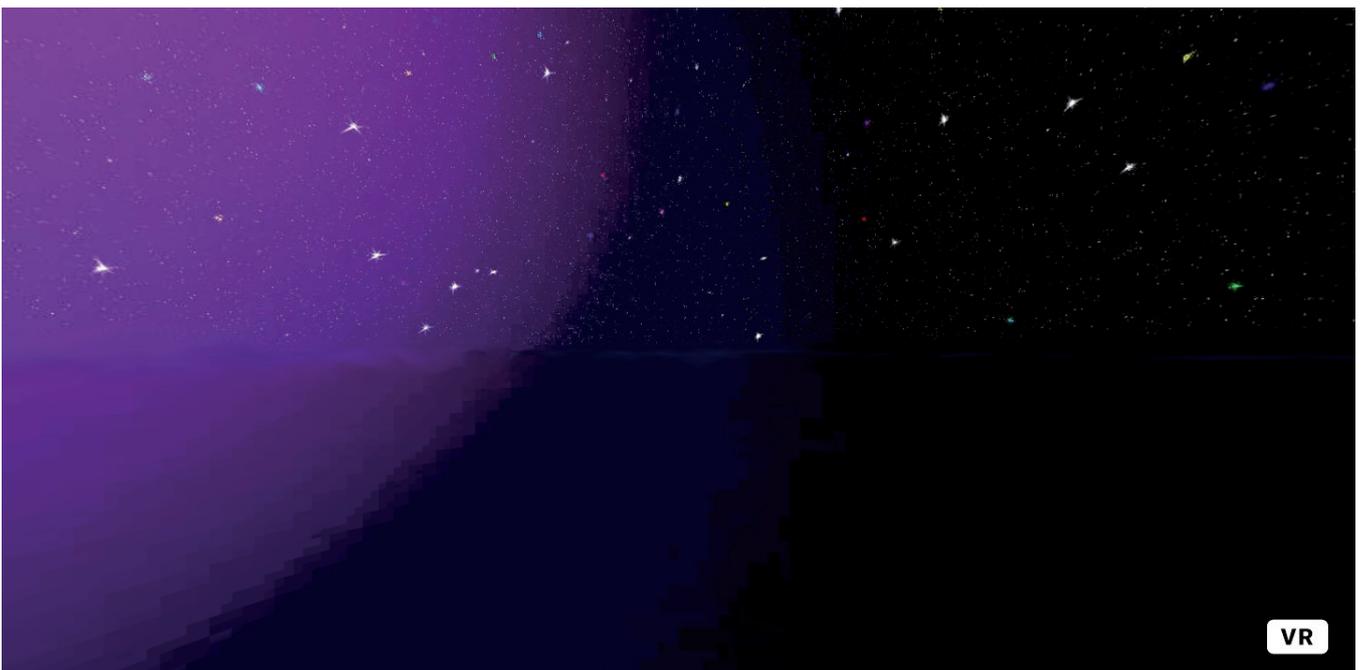
```
<html>
<head>
```

```
<script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
<script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene>
</a-scene>
</body>
</html>
```

3. На сцену добавить фоновое изображение из файла из интернет-хранилища в Glitch:

```
<html>
<head>
  <script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene>
    <a-sky src="https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/
stars.jpg?v=1640024974052">
  </a-sky>
  </a-scene>
</body>
</html>
```

4. Открыть файл в браузере и наблюдать изменения в виртуальной сцене (рис. 5).
5. Добавить к сцене земную поверхность с текстурой из изображения из файла интернет-хранилища в Glitch и установить атрибуты плоскости:



VR

Рис. 5. Фоновое изображение из файла

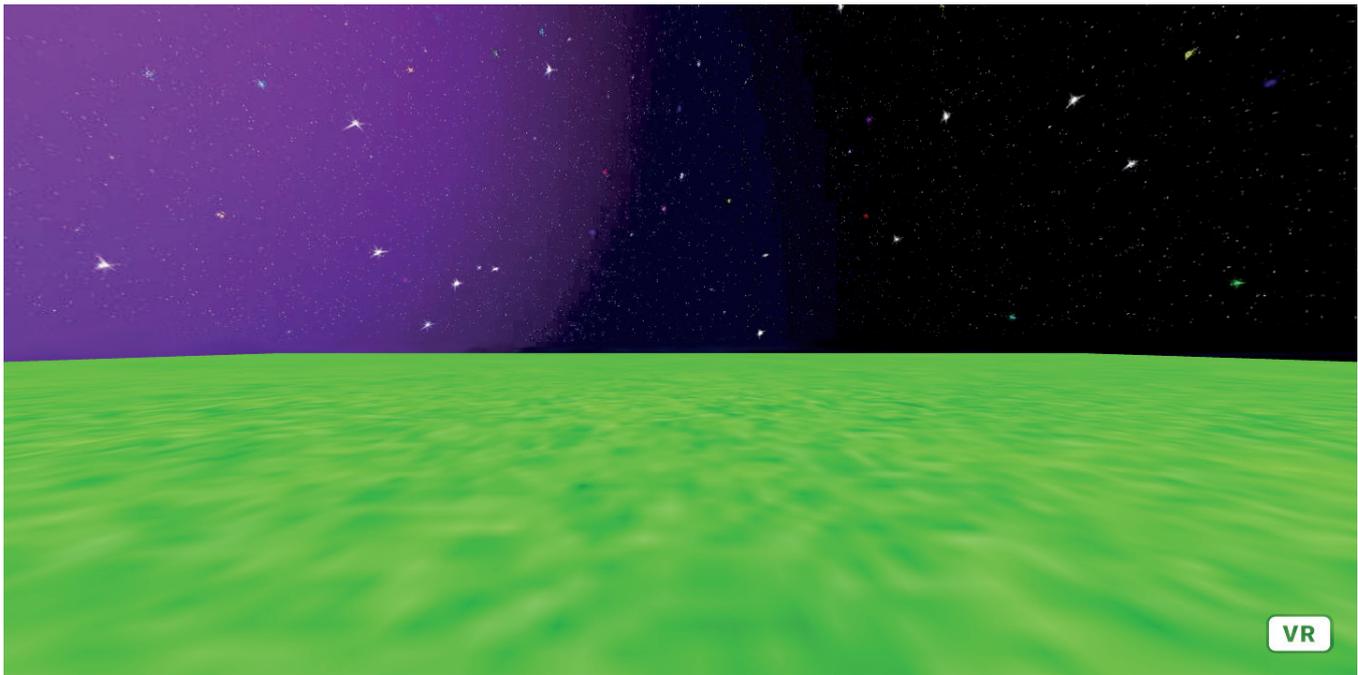


Рис. 6. Текстура травы из файла

```
<html>
<head>
  <script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene>
    <a-sky src="https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/
stars.jpg?v=1640024974052">
  </a-sky>
  <a-plane
width="100"
height="100"
position=" 0.00 0.00 0.00"
rotation="-90 0 0"
material="src: https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/grass.
jpg?v=1640024958852";
repeat: 100 100; ">
  </a-plane>
</a-scene>
</body>
</html>
```

6. Сохранить файл, обновить страницу в браузере и наблюдать изменения в виртуальной сцене (рис. 6).
7. Добавить куб на сцену и наложить на куб текстуру из изображения из файла интернет-хранилища в Glitch и установить атрибуты куба:

```
<html>
<head>
  <script src="https://aframe.io/
```

```
releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene>
    <a-sky src="https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/
stars.jpg?v=1640024974052"></a-sky>
    <a-plane
width="100"
height="100"
position=" 0.00 0.00 0.00"
rotation="-90 0 0"
material="src: https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/grass.
jpg?v=1640024958852";
repeat:100 100; ">
  </a-plane>
  <a-box src="https://cdn.glitch.me/490595e5-
3824-4948-90bd-3815c33733b3/
bubbles.jpg?v=1640024935328"
width = "1" height = "1" depth = "1"
position="-10 2 -10"
rotation="0 60 60"
scale="3 3 3">
  </a-box>
</a-scene>
</body>
</html>
```

8. Сохранить файл, обновить страницу в браузере и наблюдать изменения в виртуальной сцене (рис. 7).
9. Установить источники освещения — фоновый и точечный:

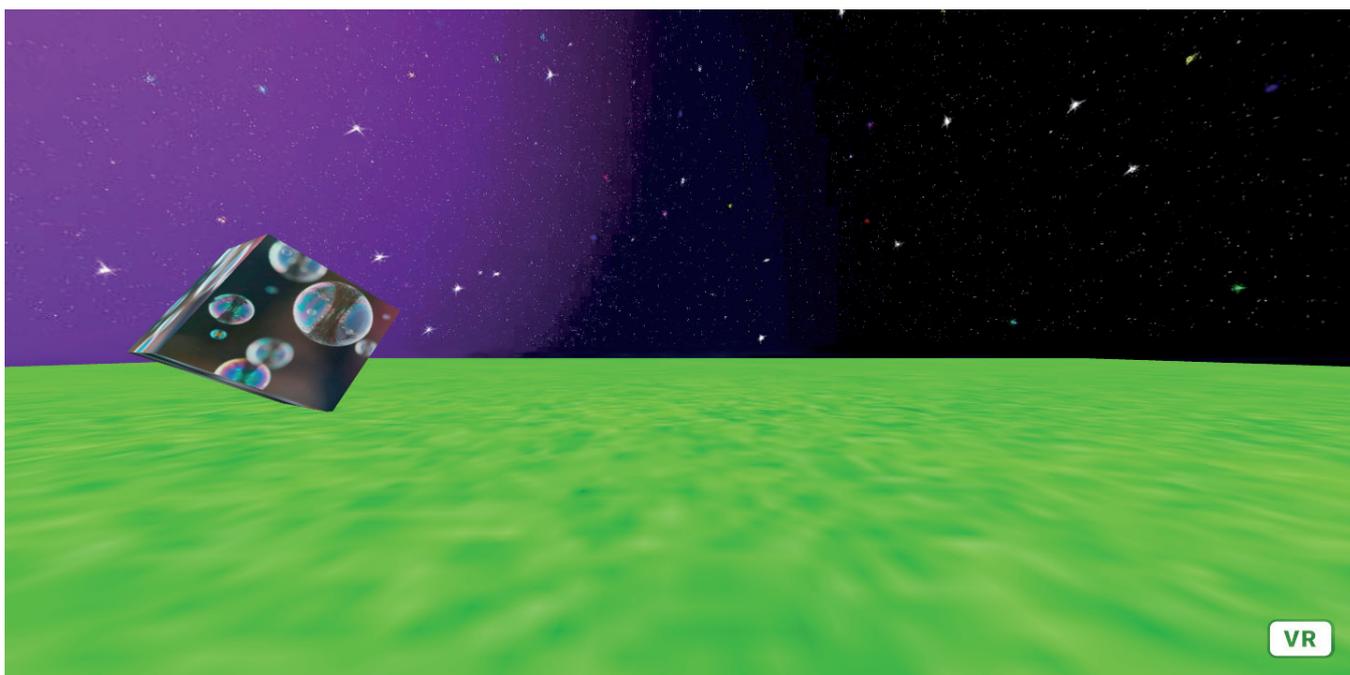


Рис. 7. Текстурированный куб

```

<html>
<head>
  <script src="https://aframe.io/
releases/1.2.0/aframe.min.js"></script>
  <script src="https://unpkg.com/aframe-
environment-component/dist/aframe-
environment-component.min.js"></script>
</head>
<body>
  <a-scene>
    <a-sky src="https://cdn.glitch.me/
490595e5-3824-4948-90bd-3815c33733b3/
stars.jpg?v=1640024974052"></a-sky>
    <a-plane
width="100"
height="100"
position=" 0.00 0.00 0.00"
rotation="-90 0 0"
material="src: https://cdn.
glitch.me/490595e5-3824-4948-90bd-
3815c33733b3/grass.jpg?v=1640024958852";
repeat:100 100; ">
</a-plane>
    <a-box src="https://cdn.glitch.me/

```

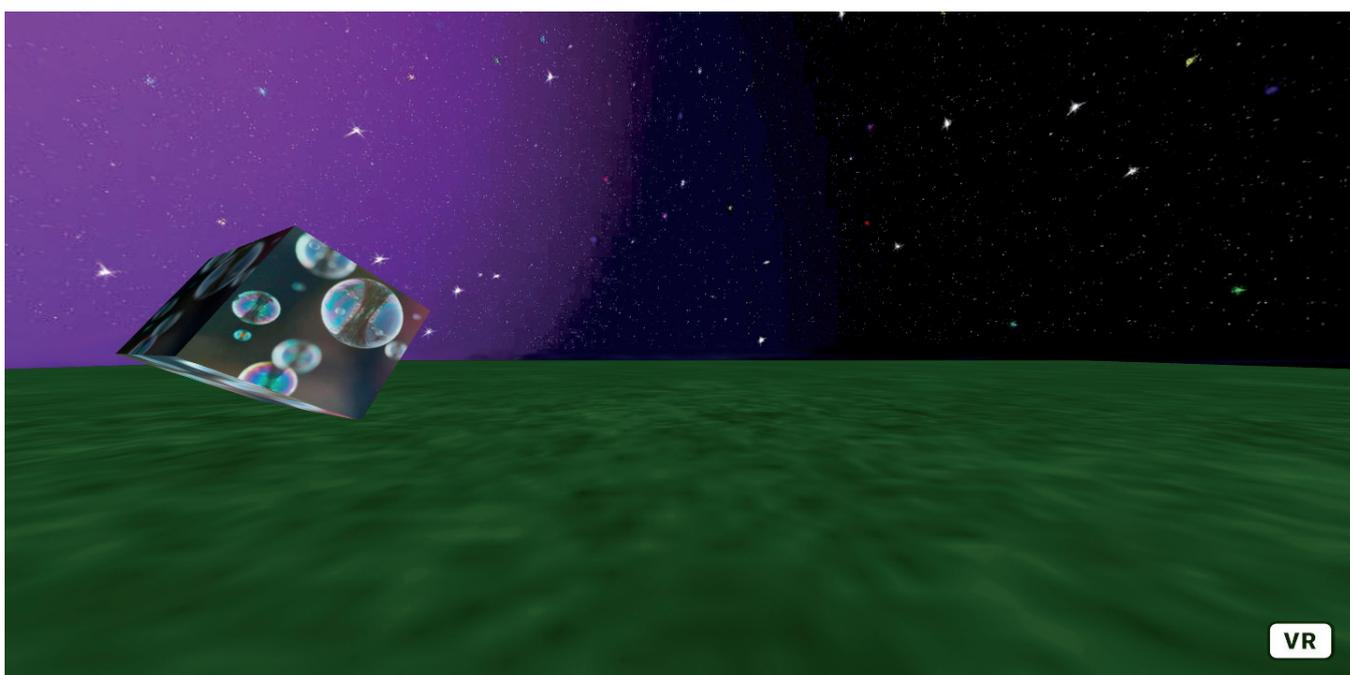


Рис. 8. Итоговая сцена

```

490595e5-3824-4948-90bd-3815c33733b3/
bubbles.jpg?v=1640024935328"
width = "1" height = "1" depth = "1"
position="-10 2 -10"
rotation="0 60 60"
scale="3 3 3">
</a-box>
<a-light type="ambient"
  color="#445451"></a-light>
<a-light type="point" intensity="5"
  position="2 5 5"></a-light>
</a-scene>
</body>
</html>

```

10. Сохранить файл, обновить страницу в браузере и наблюдать изменения в виртуальной сцене (рис. 8).

4. Заключение

С поставкой в школы нового современного оборудования технопарков «Кванториум» обучающиеся получили возможность овладеть перспективными информационными технологиями, одной из которых является технология виртуальной реальности. Создание иммерсивного приложения с WebVR API удобно, универ-

сально и является хорошей альтернативой десктопным приложениям типа Unity.

Список источников

1. Библиотека изображений Flickr. <https://www.flickr.com/>
2. Коловоротный С. В. Виртуальная реальность: манипулирование временем и пространством. <https://cyberpsy.ru/articles/virtual-realityl-time-space/>
3. Настоящая виртуальная реальность на веб-сайтах: живые примеры и обзор инструментов разработки. <https://tproger.ru/articles/web-vr/>
4. Разработка виртуальной реальности. Как стать VR-разработчиком? https://codernet.ru/articles/drugoe/razrabotka_virtualnoj_realnosti_kak_stat_vr-razrabotchikom/
5. Руководство по виртуальной реальности для веб-разработчиков. <https://lpgenerator.ru/blog/2017/11/13/rukovodstvo-po-virtualnoj-realnosti-dlya-veb-razrabotchikov/>
6. Смолин А. А., Жданов Д. Д., Потемин И. С., Меженнин А. В., Богатырев В. А. Системы виртуальной, дополненной и смешанной реальности: учебное пособие. СПб.: Университет ИТМО, 2018. 59 с. <https://books.ifmo.ru/file/pdf/2321.pdf>
7. A-Frame: самый простой способ принести виртуальную реальность в Интернет сегодня. <https://coderlessons.com/articles/veb-razrabotka-articles/a-frame-samyi-prostoi-sposob-prinesti-virtualnuiu-realnost-v-internet-segodnia>
8. A-Frame. Фреймворк. <https://aframe.io/>
9. A-Frame Examples. <https://stemkoski.github.io/A-Frame-Examples/>
10. Glitch. Среда разработки. <https://glitch.com>

DOI: 10.32517/2221-1993-2022-21-1-16-21

Н. А. Курганова

дипломант конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: основное и среднее общее образование», Омский государственный педагогический университет, г. Омск, Россия

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ЦИФРОВОГО СТОРИТЕЛЛИНГА ПРИ ИЗУЧЕНИИ ТЕМЫ «КОМПЬЮТЕРНЫЕ ВИРУСЫ» НА УРОКЕ ИНФОРМАТИКИ

Аннотация

В статье рассматриваются основные особенности организации урока при помощи технологии цифрового сторителлинга. Выделены основные моменты, которые необходимо учитывать перед созданием цифровой истории и в процессе реализации технологии цифрового сторителлинга: место цифровой истории на уроке или в системе уроков; цель и структура цифровой истории; вид истории; цифровой продукт и цифровые инструменты; послайдовый сценарий истории. Приведен пример цифровой истории на основе повествования в сервисе Canva при изучении темы «Компьютерные вирусы», в качестве цифрового продукта выступает мультимедийная презентация. Предложены задания для учащихся, которые целесообразно реализовать при помощи технологии цифрового сторителлинга, с целью их вовлечения в учебный процесс.

Ключевые слова: сторителлинг, цифровой сторителлинг, цифровой продукт, цифровые инструменты, Canva, компьютерные вирусы.

1. Введение

Одной из задач современного образования является создание качественно оформленного цифрового контента по различным дисциплинам, который впоследствии педагог сможет использовать как непосредственно на самом занятии, так и при организации дистанционного взаимодействия с обучающимися.

В то же время отметим, что образовательный контент целесообразно выстраивать на основе современных педагогических технологий.

В качестве одной из таких технологий можно выделить **технологии сторителлинга**. Технология сторителлинга в образовательном процессе предполагает, что раскрытие основных содержательных моментов темы будет представлено при помощи истории, имеющей

определенную цель, структуру, главного героя, при этом история должна быть направлена на решение поставленных задач обучения, воспитания и развития.

Применение информационно-коммуникационных технологий при реализации технологии классического сторителлинга, всех его видов позволяет говорить о **технологии цифрового сторителлинга**. **Цифровая история** — это история, обладающая всеми элементами классической истории, созданная при помощи цифровых инструментов.

Перед созданием цифровой истории и использованием технологии цифрового сторителлинга в классе или в дистанционном взаимодействии учителю необходимо проанализировать и прояснить для себя важные моменты, рассмотрению которых посвящена данная статья.

Контактная информация

Курганова Наталья Александровна, канд. пед. наук, доцент кафедры прикладной информатики и математики, Омский государственный педагогический университет, г. Омск, Россия; *адрес:* 644099, Россия, г. Омск, наб. Тухачевского, д. 14; *e-mail:* kurganovana@gmail.com

N. A. Kurganova

Omsk State Pedagogical University, Omsk, Russia

USING DIGITAL STORYTELLING TECHNOLOGY STUDYING THE THEME "COMPUTER VIRUSES" AT AN INFORMATICS LESSON

Abstract

The article discusses the main features of organizing a lesson using digital storytelling technology. The main points that need to be taken into account before creating the digital story and in the process of implementing digital storytelling technology are highlighted: the place of the digital story in the lesson or in the system of lessons; the purpose and structure of the digital story; the type of the story; the digital product and digital tools; the slide scenario of the story. An example of the digital story based on a narrative in the Canva service is given when studying the theme "Computer viruses", the multimedia presentation acts as the digital product. Tasks for students are proposed, which it is advisable to implement with the help of digital storytelling technology, in order to involve students in the educational process.

Keywords: storytelling, digital storytelling, digital product, digital tools, Canva, computer viruses.

2. Местоположение цифровой истории на уроке или в системе уроков

Перечислим **различные варианты возможного местоположения цифровой истории на уроке.**

- Учитель демонстрирует цифровую историю в классе или при дистанционном взаимодействии в начале урока при изучении нового понятия, правила, новой темы. Ученики выполняют задания по вопросам темы в рабочей тетради.
- Учащиеся сами создают цифровые истории по различным вопросам в ходе изучения новой темы, раскрывая вопросы, предложенные учителем, в формате цифровой истории. Учащиеся самостоятельно подбирают теоретический материал по теме цифровой истории. Демонстрация цифровых продуктов происходит в конце урока.
- Ученики демонстрируют свои цифровые истории как результат работы над проектом. На демонстрацию цифровых историй, на защиту проектов отводится весь урок.
- Учитель демонстрирует цифровую историю в классе или при дистанционном взаимодействии в начале урока при изучении нового понятия, правила, новой темы. Ученики сами создают цифровые истории по различным вопросам в ходе изучения новой темы, раскрывая вопросы, предложенные учителем. При этом у них имеется основной теоретический материал, представленный учителем, на основе которого создается история. Демонстрация цифровых историй учащихся происходит в конце текущего урока или на следующем уроке.

3. Цель и структура цифрового сторителлинга

Перед созданием цифровой истории целесообразно сформулировать ее цель и определиться со структурой истории.

Классическая структура истории выглядит следующим образом:

- вступление;
- развитие события;
- кульминация;
- заключение.

Существуют и другие, неклассические структуры истории, с которыми можно познакомиться в источнике [5].

4. Виды историй

Перечислим **основные виды историй**, которые рекомендуется использовать в образовательном процессе [9].

- *Истории на основе реальных ситуаций.* В основу построения истории положена реальная ситуация, которую необходимо разрешить.
- *Истории на основе повествования.* В данном случае герой (герои) истории предоставляют необходимую для обучения информацию. Данный прием применяется с целью вовлечения учащихся в изучение темы занятия.

- *Истории на основе сценария.* Здесь учащийся становится вовлеченным в историю и достигает различных результатов в зависимости от того, какие решения принимает. Данный прием позволяет сделать процесс обучения более осмысленным.
- *Истории на основе проблемных ситуаций.* Этот вид истории позволяет задействовать учащихся в решении различных проблем, когда они применяют полученный опыт на практике.

5. Цифровой продукт и цифровые инструменты

Цифровая история — это определенный цифровой продукт, оформление которого зависит от вида истории и от программы, выбранной для создания продукта.

Перечислим **основные виды цифровых продуктов** [2, 9].

- *Мультимедийная презентация.* На сегодняшний день самым популярным цифровым продуктом, разрабатываемым в поддержку содержательной составляющей любой дисциплины в школе, в том числе дисциплины «Информатика», позволяющим визуализировать основные моменты изучаемых тем, является презентация. Презентация, созданная на основе использования технологии сторителлинга, обладает всеми его преимуществами. Отметим, что при оформлении образовательного контента при помощи презентаций необходимо учитывать тренды по дизайну презентаций, т. е. создаваемые презентации по информатике должны удовлетворять всем правилам эффективной презентации. Для создания цифровых продуктов можно использовать классическую программу презентационной графики Microsoft PowerPoint, программы Prezi, Sway, Canva и другие онлайн-ресурсы. С основными советами по созданию эффективных презентаций можно познакомиться в [3].
- *Видеопрезентация.* На визуальные элементы, используемые в презентации такого вида, наложены эффекты анимации. Примеры онлайн-сервисов, в которых имеются готовые библиотеки персонажей, фонов, анимационных эффектов: Powtoon, Vyond, Canva и др.
- *Видеоскрайбинг.* Данный цифровой продукт предполагает «рисующую руку», за которой наблюдают обучающиеся. «Рисующая рука» визуализирует картинки, пишет текстовые заметки. Ресурсы для создания такой цифровой истории: VideoScribe, Powtoon.
- *Комикс.* В этом случае рекомендуется использовать сервис Pixton, который содержит множество готовых шаблонов сцен, богатую библиотеку картинок.
- *Веб-страница с мультимедийным контентом.* Для создания цифрового продукта данного вида целесообразно использовать один из конструкторов сайтов, например, Wix или Tilda. При помощи данных конструкторов можно создать мультимедийный лонгрид.

Отметим, что разнообразие программ, онлайн-сервисов предоставляет широкие возможности для создания цифровых историй и их интеграции в учебный процесс.

6. Послайдовый сценарий истории

Перед созданием цифровой истории целесообразно продумать и составить ее сценарий, т. е. выполнить **раскадровку истории**.

Приведем пример использования цифровой истории при изучении темы «Компьютерные вирусы».

Отметим, что при изучении данной темы цифровые истории на уроке будут демонстрировать и учитель, и учащиеся. При этом учитель при помощи цифровой истории объясняет такое новое понятие, как компьютерный вирус (слайды 1–12). Учащиеся также создают небольшие собственные цифровые истории на основе теоретического материала, предоставленного учителем, раскрывают основные вопросы изучаемой темы.

Цель цифровой истории учителя — знакомство учащихся с понятием компьютерного вируса и его признаками. Представленная ниже цифровая история относится к историям на основе повествования [5]. Цифровой продукт в данном случае — мультимедийная презентация. Цифровая история имеет классическую структуру (см. раздел 3) и выполнена в сервисе Canva. Данный сервис является эффективным инструментом для создания презентаций по информатике на основе использования технологии цифрового сторителлинга.

Canva — это онлайн-сервис для графического дизайна, позволяющий создавать визуальные материалы различного типа [8].

Перечислим достоинства этого цифрового инструмента:

- интерфейс сервиса представлен на русском языке;
- интерфейс сервиса разработан на основе принципов простоты и юзабилити;

- Canva представляет собой кроссплатформенный сервис;
- все материалы в сервисе создаются на основе технологии конструктора, т. е. дизайн всех продуктов Canva представлен в виде уже готовых шаблонов, которые представляют собой различные композиционные сочетания — графики и текста, текста и видео, диаграмм, графиков, — на основе изменения или замены которых пользователь и создает свой продукт.

Среди многочисленных возможностей онлайн-сервиса Canva в нем имеется и возможность создания презентаций. Профессиональными дизайнерами для сервиса разработаны шаблоны презентаций различного типа. Данные шаблоны разделены на определенные категории: образовательные, маркетинговые, бизнес-презентации и т. д. В свою очередь, разработанные шаблоны содержат различные макеты для слайдов презентаций.

Особое место среди готовых шаблонов занимают шаблоны, разработанные таким образом, что их использование позволит представить образовательный контент на основе использования технологии сторителлинга.

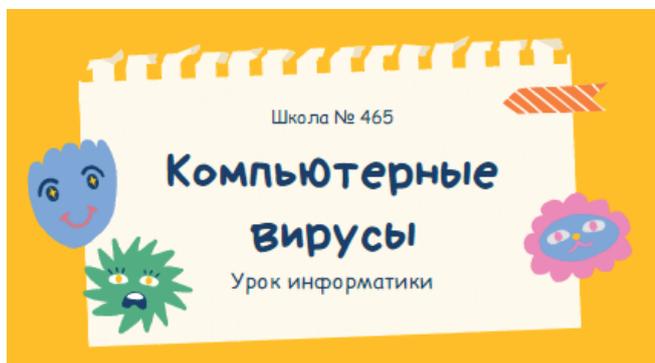
На слайдах 1–12 приведен пример цифровой истории, подготовленной учителем.

Данная цифровая история демонстрируется учащимся в начале урока. С электронной версией презентации вы можете познакомиться по ссылке, указанной в [6].

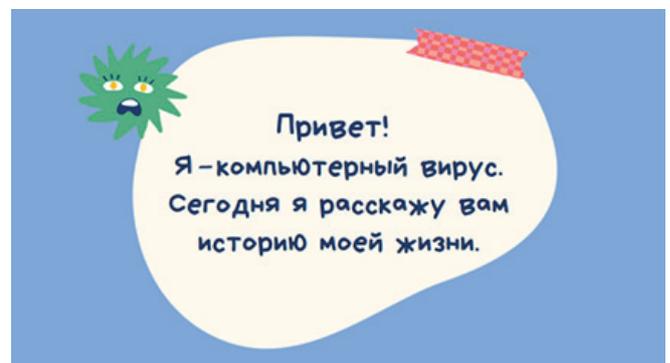
В этой же презентации представлен слайд с заданием по созданию цифровой истории для учащихся (слайд 12). Учитель еще раз формулирует задание:

Задание.

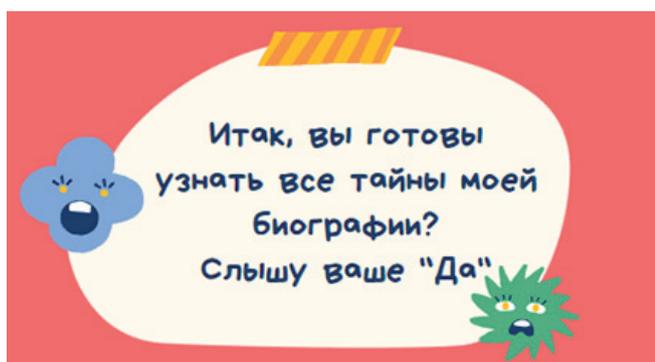
Создайте цифровую историю — мультимедийную презентацию в Canva. Тема цифровой истории — «История одного опасного вируса».



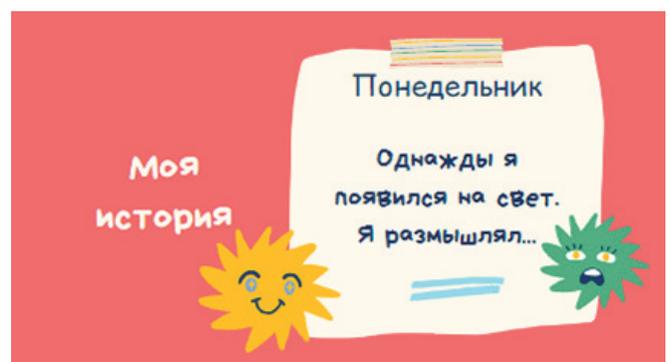
Слайд 1



Слайд 2



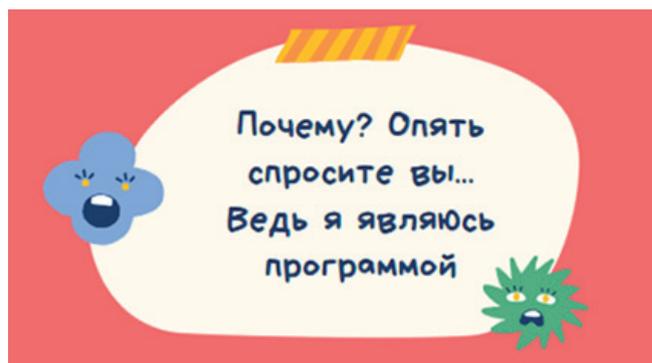
Слайд 3



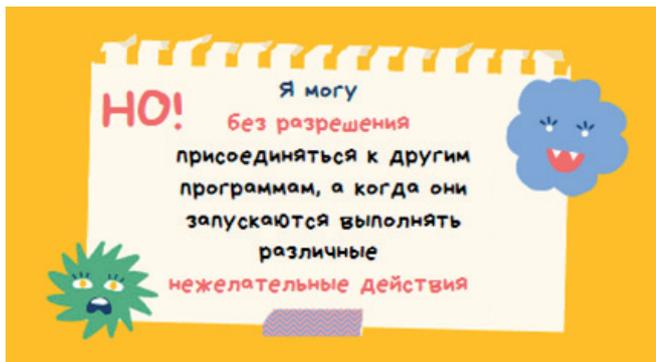
Слайд 4



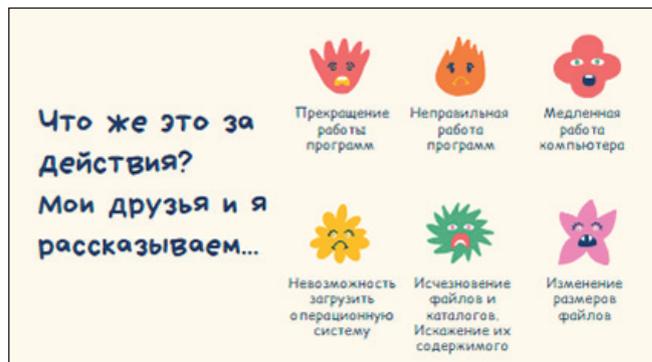
Слайд 5



Слайд 6



Слайд 7



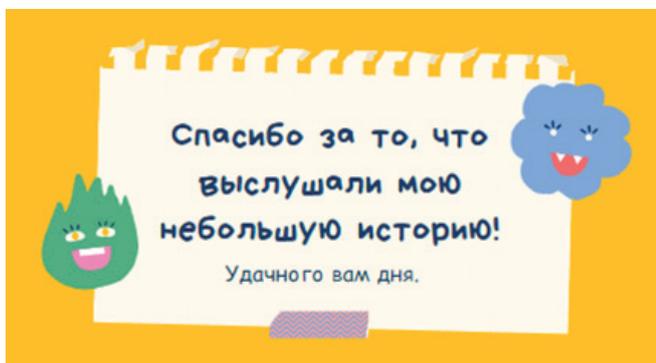
Слайд 8



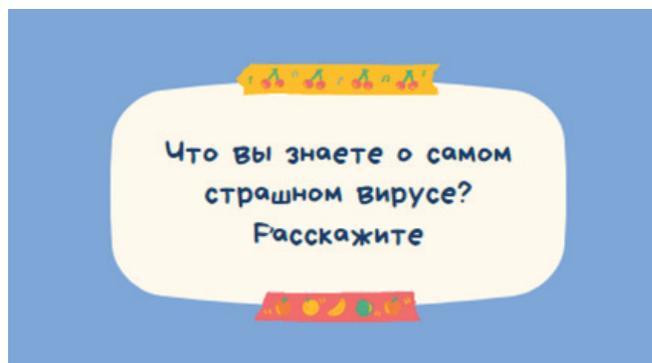
Слайд 9



Слайд 10



Слайд 11



Слайд 12

Для успешного создания учащимися цифровой истории учителю целесообразно:

- разработать рабочий лист с алгоритмом выполнения задания;
- подготовить теоретические материалы по предлагаемому заданию;
- подобрать шаблоны в Canva для последующей их рекомендации учащимся.

Пример рабочего листа

№ п/п	Алгоритм выполнения задания	Отметка о выполнении
1	<p>Перейдите по ссылке при помощи QR-кода и познакомьтесь с теоретическим материалом о самых страшных компьютерных вирусах за последнее время [1].</p> 	
2	<p>Выберите из списка один из опасных компьютерных вирусов. Запишите его название в рабочий лист.</p>	
3	<p>Создайте поккадровый сценарий по заданию — цифровую историю — мультимедийную презентацию в Canva, тема — «История одного опасного вируса». Речь должна идти о том вирусе, который вы выбрали в пункте 2. Воспользуйтесь советами по созданию истории, представленными ниже [4, 7]:</p> <ul style="list-style-type: none"> • На титульном слайде (слайд 1) рекомендуется написать название истории, например, «История одного опасного вируса». • На следующем слайде (слайд 2) желательно представить своего героя и поздороваться с аудиторией. • Далее воспользуйтесь фразами-клише для создания истории (слайды 3–6). <ul style="list-style-type: none"> – «Однажды я ... появился на свет». – «Я появился на свет в ... году». – «Родители меня называли так, потому что ... ». – «Я считаюсь очень опасным среди других вирусов, потому что ... ». • Напишите заключение, финальную фразу (фразы) от имени главного героя (слайды 7, 8). 	
4	<p>Выберите в Canva шаблон, подходящий для реализации технологии сторителлинга, или воспользуйтесь рекомендацией учителя.</p>	
5	<p>Создайте цифровую историю, опираясь на пункты 3 и 4.</p>	
6	<p>Скачайте свою презентацию и поместите ее в общей сетевой папке класса.</p>	

Пример рабочего листа представлен в таблице.

После создания собственной цифровой истории учащиеся демонстрируют, «рассказывают» свои истории классу [11]. Учитель подводит итоги о разнообразии компьютерных вирусов.

В качестве домашнего задания учащимся предлагается самостоятельно создать еще одну цифровую историю в сервисе Canva.

Домашнее задание.

Создайте цифровую историю — мультимедийную презентацию в Canva — по одной из следующих тем:

- «Один день из жизни компьютерного вируса». О поведении компьютерного вируса.
- ««Семья» компьютерного вируса». О различных видах компьютерных вирусов.
- «Программы-супергерои». О различных анти-вирусных программах.

На следующем уроке учащиеся в начале урока рассказывают свои истории, демонстрируя их классу на проекторе.

7. Заключение

Исходя из вышеизложенного, можно сделать вывод, что применение технологии цифрового сторителлинга на уроке позволяет учащимся создавать самостоятельные проекты [10]. В то же время данная технология не только направлена на повышение вовлеченности учеников в процесс обучения, но и позволяет им глубже и быстрее осмыслить какую-либо тему или проблему.

Список источников

1. 18 самых опасных компьютерных вирусов за все время. <https://www.comss.ru/page.php?id=2853>
2. Canva. Сторителлинг — 17 правил интересной истории. https://www.canva.com/ru_ru/obuchenie/storitelling-17-pravil/
3. Microsoft. Советы по созданию и представлению эффективных презентаций. <https://support.microsoft.com/ru-ru/office/советы-по-созданию-и-представлению-эффективных-презентаций-f43156b0-20d2-4c51-8345-0c337cefb88b#bml>
4. Галло К. Искусство сторителлинга. Как создавать истории, которые попадут в самое сердце аудитории. М.: Эксмо, 2020. 350 с.

5. *Ермолаева Ж. Е., Ланухова О. В.* 5 сюжетов педагогического сторителлинга // Eduneo. <https://www.eduneo.ru/5-syuzhetov-v-pedagogicheskom-storitellinge/>

6. *Курганова Н. А.* Цифровая история «Компьютерные вирусы». https://www.canva.com/design/DAEw1Hx626g/nsk6aLftCVha89rDo62Rww/view?utm_content=DAEw1Hx626g&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

7. *ЛеФевр Л.* Искусство объяснять. Как сделать так, чтобы вас понимали с полуслова. М.: Манн, Иванов и Фербер, 2013. 241 с.

8. Онлайн-сервис Canva. <https://www.canva.com/>

9. *Пастухова С. Н.* Сторителлинг — современная педагогическая технология. <https://multiurok.ru/index.php/files/statia-storitelling-sovremennaia-pedagogicheskaia.html>

10. *Симмонс А.* Сторителлинг: как использовать силу историй. М.: Манн, Иванов и Фербер, 2013. 263 с.

11. *Челнокова Е. А., Казначеева С. Н., Калинкина К. В., Григорян Н. М.* Сторителлинг как технология эффективных коммуникаций // Перспективы науки и образования. 2017. № 5 (29). С. 7–12. https://pnojournalf.files.wordpress.com/2017/08/pdf_170501.pdf

DOI: 10.32517/2221-1993-2022-21-1-22-29

И. А. Фоминых

дипломант конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: основное и среднее общее образование», Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия

А. Р. Довранов

дипломант конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: основное и среднее общее образование», Мочалищенская средняя общеобразовательная школа, п. Мочалище, Звениговский район, Республика Марий Эл, Россия;
Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия

О РАЗРАБОТКЕ КУРСА ПО ВЫБОРУ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON» ДЛЯ ПРЕДПРОФИЛЬНОЙ ПОДГОТОВКИ ШКОЛЬНИКОВ

Аннотация

В статье рассматриваются некоторые аспекты предпрофильной подготовки школьников, приводятся компоненты предпрофильной подготовки. Определяется роль предпрофильных курсов по выбору в профессиональном самоопределении школьников в условиях действия федеральных государственных образовательных стандартов общего образования. Акцентируется внимание на актуальности предпрофильных курсов по программированию. Выбор языка программирования Python для этих курсов объясняется его универсальностью и возможностью увеличения функционала за счет библиотек, а также достаточной простотой освоения языка из-за особенностей написания кода. Представлена программа курса по выбору «Основы программирования на Python», в том числе цели и задачи курса, планируемые результаты обучения, содержание обучения, пояснения по реализации курса, тематическое планирование, материально-техническое и учебно-методическое обеспечение. Представляет интерес реализация процесса обучения в контексте работы над реальным проектом. Также описываются структура, содержание и оформление разработанного для преподавания курса образовательного сайта. Отмечается возможность использования сайта на уроках и для самостоятельной работы школьников дома.

Ключевые слова: предпрофильная подготовка школьников, предпрофильный курс по выбору, программирование на Python, образовательный сайт.

Контактная информация

Фоминых Ирина Анатольевна, канд. пед. наук, доцент кафедры общеобразовательных дисциплин и методики их преподавания, Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия; *адрес:* 424002, Россия, Республика Марий Эл, г. Йошкар-Ола, пл. Ленина, д. 1; *e-mail:* foir@yandex.ru

Довранов Атагелди Русланович,

учитель информатики, Мочалищенская средняя общеобразовательная школа, пос. Мочалище, Звениговский район, Республика Марий Эл, Россия; *адрес:* 425051, Россия, Республика Марий Эл, Звениговский район, пос. Мочалище, ул. Школьная, д. 15; студент 5-го курса факультета общего и профессионального образования, Педагогический институт, Марийский государственный университет, г. Йошкар-Ола, Республика Марий Эл, Россия; *адрес:* 424002, Россия, Республика Марий Эл, г. Йошкар-Ола, пл. Ленина, д. 1; *e-mail:* tm.yigit.97@gmail.com.

I. A. Fominykh

Mari State University, Yoshkar-Ola, The Republic of Mari El, Russia

A. R. Dovranov

Mochalishenskaya Secondary School, Mochalishche Village, Zvenigovsky District, The Republic of Mari El, Russia;
Mari State University, Yoshkar-Ola, The Republic of Mari El, Russia

ABOUT THE DEVELOPMENT OF THE COURSE OF CHOICE "BASICS OF PROGRAMMING IN PYTHON" FOR PRE-PROFILE TRAINING OF SCHOOLCHILDREN

Abstract

The article discusses some aspects of pre-profile training of schoolchildren, provides components of pre-profile training. The role of pre-profile elective courses in the professional self-determination of schoolchildren is determined under the conditions of the Federal State Educational Standards of General Education. Attention is focused on the relevance of pre-profile programming courses. The choice of the Python programming language for these courses is explained by its versatility and the ability to increase functionality through libraries, as well as the ease of learning the language due to the peculiarities of writing code. The program of the elective course "Basics of Programming in Python" is presented, including the goals and objectives of the course, planned learning outcomes, learning content, explanations for the implementation of the course, thematic planning, logistical and educational support. It is of interest to implement the learning process in the context of working on a real project. The structure, content and design of the educational site developed for teaching the course are also described. The possibility of using the site in the classroom and for independent work of schoolchildren at home is noted.

Keywords: pre-profile training of schoolchildren, elective course, pre-profile course, programming in Python, educational site.

1. Введение

Одними из важнейших личностных результатов освоения основной образовательной программы основного общего образования являются «готовность и способность обучающихся к осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учетом устойчивых познавательных интересов, а также на основе формирования уважительного отношения к труду, развития опыта участия в социально значимом труде» [9]. Данный результат обусловлен необходимостью осуществления выпускниками IX классов выбора учебного заведения среднего общего или среднего профессионального образования, а также конкретного профиля.

Действительно, в Российской Федерации уже на старшей ступени общего образования предусмотрена возможность профильного обучения. В ФГОС указано: «Организация, осуществляющая образовательную деятельность: обеспечивает реализацию учебных планов одного или нескольких профилей обучения (естественно-научный, гуманитарный, социально-экономический, технологический, универсальный), при наличии необходимых условий профессионального обучения для выполнения определенного вида трудовой деятельности (профессии) в сфере технического и обслуживающего труда» [10]. Профилизация обеспечивается включением в учебный план трех-четырёх предметов на углубленном уровне изучения, а также дополнительных учебных предметов, курсов по выбору обучающихся, предлагаемых организацией, осуществляющей образовательную деятельность. Кроме того, существует множество направлений подготовки в среднем профессиональном образовании.

Помощь в профессиональном самоопределении как «процессе формирования личностью своего отношения к профессиональной деятельности и способе его реализации через согласование личностных и социально-профессиональных потребностей» [2] оказывается обучающимся VIII—IX классов посредством предпрофильной подготовки.

К составляющим предпрофильной подготовки можно отнести [1, 5]:

- включение элементов профориентационной работы в учебный процесс;
- тематические профориентационные классные часы;
- внеклассные мероприятия профориентационной направленности (экскурсии на предприятия, встречи с представителями разных профессий и др.);
- проведение психологических диагностик и тестов профориентации;
- предпрофильные курсы по выбору (ранее — элективные курсы);
- построение и реализация индивидуальных образовательных маршрутов обучения.

Следует отметить, что особенно полезны предпрофильные курсы по выбору в плане пробы себя в той или иной профессиональной области [6]:

- предметно-ориентированные пробы, дающие возможность апробировать разное предметное

содержание с целью самоопределения; проверяющие готовность и способность ученика осваивать выбранный предмет на углубленном уровне; создающие условия для подготовки к экзаменам по выбору (по наиболее вероятным предметам будущего профиля);

- профессиональные пробы, дающие возможность поучаствовать в различных видах профессиональной деятельности, заложенных в содержании соответствующих профилей обучения. Профессиональные пробы могут проводиться на базе школ или предприятий (организаций, учреждений среднего профессионального и высшего образования) — социальных партнеров школы.

В ФГОС основного общего образования отмечено, что «в целях обеспечения индивидуальных потребностей обучающихся в основной образовательной программе основного общего образования предусматриваются: учебные курсы, обеспечивающие различные интересы обучающихся, в том числе этнокультурные; внеурочная деятельность» [9].

Учитывая востребованность в условиях цифровой трансформации экономики страны профессионалов по ИТ-направлениям (особенно программистов), мы задались целью разработать курс по выбору для предпрофильной подготовки обучающихся по программированию.

2. Выбор языка программирования

Предполагается, что основы алгоритмизации и программирования уже изучены школьниками в VII—VIII классах. На данный момент большинство школьных учителей преподают программирование в основной школе на базе строго типизированного и уже привычного языка Pascal. Следует отметить, что существует современная модификация языка Pascal — PascalABC.NET, сочетающая простоту классического языка Pascal, ряд современных расширений и огромный потенциал платформы Microsoft.NET. Однако дополнительные возможности PascalABC.NET рассматриваются лишь при углубленном изучении предмета или в рамках проектной деятельности.

Существуют и альтернативные варианты языков программирования для изучения в школе, среди которых все большую популярность приобретает высокоуровневый язык программирования общего назначения Python.

Python является универсальным языком, который поддерживает несколько технологий программирования и допускает увеличение функционала за счет библиотек или расширений. Поэтому он может быть применен для самого широкого круга задач программирования.

Программирование на Python считается достаточно простым из-за особенностей написания кода. Команды вводятся в интерактивном режиме по одной строке за один раз, очень напоминая управление операционной системой.

Python определяет вложенность группы операторов с помощью отступов. Такую группу именуют блоком. В большинстве языков программирования для этого используются некоторые операторные скобки (служебные слова или знаки препинания). Отступы делают код более наглядным, а также уменьшают число ошибок. Некото-

рые алгоритмические конструкции могут быть записаны в соответствии с их словесной интерпретацией.

Python оперирует объектами, предоставляет возможности динамической типизации, содержит типы данных высокого уровня (списки, кортежи, словари и др.), которые могут хранить и другие данные. С такими типами данных предусмотрен большой набор операций и функций. Кроме того, типы данных могут добавляться из библиотек.

Указанные достоинства языка Python и обусловили его выбор для предпрофильного курса по программированию.

3. Программа курса по выбору «Основы программирования на Python»

Курс рассчитан на учащихся IX классов.

3.1. Цели и задачи курса. Планируемые образовательные результаты

Цели курса:

- сформировать у обучающихся представление о программировании на языке Python;
- дать возможность обучающимся попробовать себя в роли разработчика программного продукта.

Задачи курса:

- развивать у учащихся интерес к информатике в целом и к программированию в частности;
- формировать у школьников представление о профессии программиста;
- развивать алгоритмическое мышление обучающихся;
- закрепить знания и умения школьников по структурному программированию;
- сформировать у учащихся представление об особенностях проектирования программ, оформлении алгоритмических конструкций, типизации на языке Python;
- сформировать у обучающихся представление об облачных вычислениях, интернет-сервисах для программирования;
- разработать проект «Чат-бот».

При изучении курса по выбору формируются следующие результаты:

личностные:

- ответственное отношение к учению, готовность и способность к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений с учетом устойчивых познавательных интересов;
- целостное мировоззрение, соответствующее современному уровню развития науки;

метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи в учебе и познавательной деятельности;

- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение оценивать правильность выполнения учебной задачи;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

предметные:

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе;
- знакомство с языком программирования Python;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в интернете.

Курс ориентирован на предпрофильную подготовку учащихся по информатике. Он расширяет базовый курс по информатике, является практико-ориентированным и дает учащимся возможность познакомиться с основами программирования на языке Python.

Курс по выбору «Основы программирования на Python» включается в индивидуальный учебный план, формируемый в соответствии с интересами и способностями школьников.

3.2. Содержание обучения

1. Введение. Первая программа (1 ч).

Краткая история создания языка Python. Пояснения по работе с сайтом. Постановка задачи создания чат-бота. Функция print(). Строки. Знакомство со средой программирования. Первая программа.

2. Типы данных. Ввод данных (1 ч).

Комментарии. Правила использования комментариев. Типы данных (строковый тип, числовые типы). Функция type(). Переменные. Понятие о динамической типизации. Функция ввода input().

3. Подключение библиотек. Составление линейных вычислительных программ (2 ч).

Арифметические операции. Приоритет операций. Именованные переменных. Соглашение о стиле кода PEP8. Преобразование ввода. Оператор присваивания. Подключение библиотек. Библиотека Math.

4. Ветвления. Цикл с условием (2 ч).

Логический тип данных. Логические операции and, or и not. Операции сравнения. Оператор if. Вложенный оператор if. Цикл while. Переменная-счетчик. Бесконечный цикл.

5. Списки. Функции (2 ч).

Понятие списка. Способы создания списков. Пустые списки. Функция len. Положительные и отрицательные индексы. Обработка списков с помощью цикла while. Функции. Встроенные функции. Создание функции через def. Параметры и аргументы функции. Возврат функцией результата.

Содержание курса рассматривается в контексте разработки проекта «Чат-бот». Постановка задачи создания виртуального собеседника (чат-бота) осуществ-

вляется на первом уроке, а работа над проектом продолжается в течение всего курса. В итоге по окончании курса у каждого обучающегося должен быть разработан проект «Чат-бот» со следующими функциями:

- отображение приветствия бота, имени и даты его создания;
- приветствие ботом пользователя по введенному имени;
- угадывание ботом возраста пользователя по некоторым введенным данным;
- демонстрация ботом умения считать по порядку;
- проведение ботом викторины с пользователем.

Изучение и закрепление учебного материала осуществляется с использованием образовательного сайта «PyStudy». По каждой теме курса работа осуществляется в следующей последовательности:

- 1) изучение теоретического материала с параллельным выполнением, обсуждением и модификацией имеющихся примеров;
- 2) закрепление изученного материала посредством выполнения тестовых заданий и составления небольших программ;
- 3) самостоятельная домашняя работа либо по составлению и решению задач по теме, либо по проекту «Чат-бот».

3.3. Тематическое планирование

№ урока	Тема урока	Основное содержание урока	Практика на уроке	Домашнее задание	Кол-во часов
1	Введение. Первая программа	Краткая история создания языка Python. Пояснения по работе с сайтом. Постановка задачи создания чат-бота. Функция print(). Строки. Знакомство со средой программирования. Первая программа	Тестовые задания на правильность написания строк, работу функции print, составление мини-программ на отображение заданного текста	Практическая работа над проектом 1 «Вас приветствует чат-бот» (отображение приветствия бота, имени и даты его создания)	1
2	Типы данных. Ввод данных	Комментарии. Правила использования комментариев. Типы данных (строковый тип, числовые типы). Функция type(). Переменные. Понятие о динамической типизации. Функция ввода input()	Тестовые задания на правильность написания комментариев, типы данных, работу с функцией input(), составление комментированных мини-программ на ввод строковых данных	Практическая работа над проектом 2 «Представьте чат-боту» (приветствие ботом пользователя по введенному имени)	1
3	Составление линейных вычислительных программ	Арифметические операции. Приоритет операций. Именованные переменные. Соглашение о стиле кода PEP8. Преобразование ввода. Оператор присваивания	Тестовые задания на стилистически правильное написание имен переменных, вычисление значений выражений, составление мини-программ вычислительного характера	Составление и решение двух задач вычислительного характера	1
4	Подключение библиотек. Составление линейных вычислительных программ	Подключение библиотек. Библиотека Math	Решение математических задач	Практическая работа над проектом 3 «Чат-бот угадывает возраст» (угадывание ботом возраста пользователя по некоторым введенным данным)	1
5	Ветвления	Логический тип данных. Логические операции and, or и not. Операции сравнения. Оператор if. Вложенный оператор if.	Тестовые задания на проверку правильности написания и вычисление значений логических выражений, использование оператора if; составление программ на ветвления	Составление и решение двух задач на ветвления	1
6	Цикл с условием	Цикл while. Переменная-счетчик. Бесконечный цикл	Тестовые задания на проверку правильности написания и определение результатов работы программ с оператором while; составление циклических программ	Практическая работа над проектом 4 «Чат-бот умеет считать» (демонстрация ботом умения считать по порядку)	1
7	Списки	Понятие списка. Создание списков с помощью квадратных скобок и функции list. Пустые списки. Функция len. Положительные и отрицательные индексы. Обработка списков с помощью цикла while	Тестовые задания на способы создания списков, правильность обращения к элементам списка; составление программ на обработку списков с помощью цикла while	Составление и решение двух задач с использованием списков	1

Окончание таблицы

№ урока	Тема урока	Основное содержание урока	Практика на уроке	Домашнее задание	Кол-во часов
8	Функции	Функции. Встроенные функции. Создание функции через def. Параметры и аргументы функции. Возврат функцией результата. Полный оператор условия	Тестовые задания на правильность именования и объявления функций, выбор лучшего способа реализации функции, использование полных условий; составление функций вычислительного характера	Практическая работа над проектом 5 «Чат-бот проверит ваши знания» (проведение ботом викторины с пользователем)	1

3.4. Учебно-методическое и материально-техническое обеспечение образовательного процесса

Рекомендуемая литература:

1. *Златопольский Д. М.* Основы программирования на языке Python. М.: ДМК Пресс, 2017. 284 с.

2. *Поляков К. Ю.* Программирование. Python. C++. Ч. 1: учебное пособие: 4-е изд., стер. М.: Просвещение, 2021. 144 с.

3. *Северенс Ч.* Введение в программирование на Python: учебное пособие. 2-е изд. М.: ИНТУИТ, 2016. 231 с. <https://e.lanbook.com/book/100703>

4. *Шуман Х.* Python для детей: руководство / пер. с нем. М. А. Райтман. М.: ДМК Пресс, 2019. 344 с. <https://e.lanbook.com/book/131700>

Аппаратное обеспечение, используемое в курсе:

- компьютеры с выходом в интернет.

4. Разработка образовательного сайта для преподавания курса по выбору «Основы программирования на Python»

Одним из средств обучения основам программирования на Python в рамках предметного курса по выбору (элективного курса) может быть образовательный сайт [4]. Для создания сайта можно использовать любой доступный конструктор сайтов. Мы использовали платформу Wix.com.

Адрес сайта:

<https://tmyigit97.wixsite.com/pythonstudy>

Структура сайта полностью соответствует программе курса по выбору. Для формирования содержания были использованы учебные пособия по языку Python [3, 7, 8, 11].

При посещении сайта обучающиеся сразу могут увидеть, чему они научатся и что они создадут по окончании курса, т. е. конечный продукт — это будет консольное

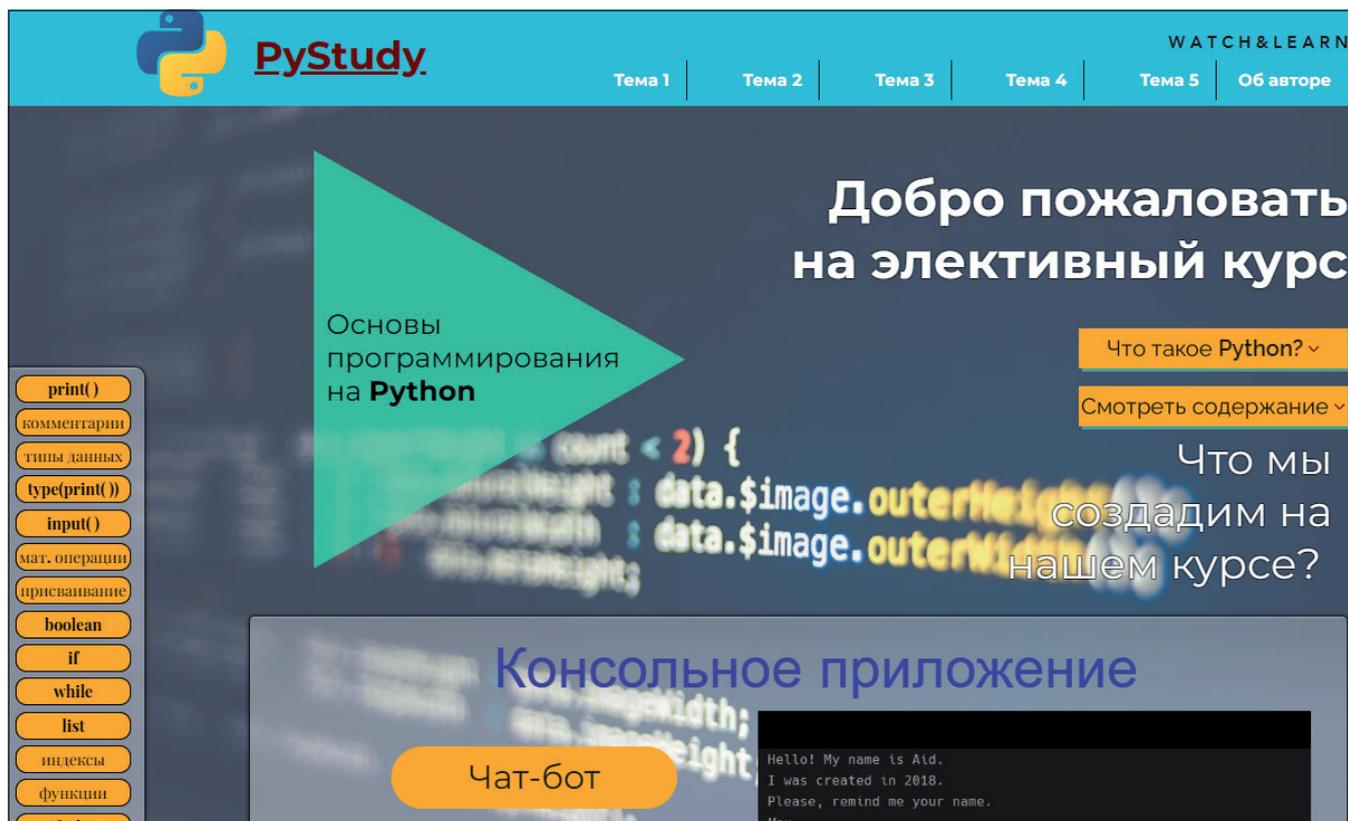


Рис. 1. Главная страница сайта PyStudy

The screenshot shows the 'PyStudy' website interface. At the top, there is a navigation bar with the Python logo and the text 'PyStudy'. To the right, it says 'WATCH & LEARN' and has links for 'Тема 1', 'Тема 2', 'Тема 3', 'Тема 4', 'Тема 5', and 'Об авторе'. The main content area is titled 'Тема 3. Подключение библиотек. Составление линейных вычислительных программ'. Below the title, there is a paragraph explaining that in real life, we often perform **arithmetic operations** to calculate changes, areas, queue lengths, etc. It lists basic operations: addition (+), subtraction (-), multiplication (*), division (/), and integer division (//). A code block shows examples: `print(10 + 10) # 20`, `print(100 - 10) # 90`, `print(10 * 10) # 100`, `print(77 / 10) # 7.7`, and `print(77 // 10) # 7`. A note explains the difference between / and //. A terminal window on the right shows a `ZeroDivisionError: division by zero` error.

Рис. 2. Теоретическая часть занятия

приложение «Чат-бот» (рис. 1). Кроме того, на сайте представлено содержание курса, которое информирует школьников, какие именно действия будут добавлены к функционалу чат-бота в каждой теме.

По каждой теме имеется теоретическая и практическая части. Практика предусматривает выполнение тестовых заданий и составление небольших программ по применению нового учебного материала. Далее предлагается работа над проектом «Чат-бот».

Рассмотрим, например, тему 3. Теория насыщена примерами кодов, в которых для удобства подсвечен синтаксис (рис. 2). Сайт удобен для освоения языка обучающимися, так как содержит онлайн-интерпретатор.

Тестовые задания созданы в Google-форме, которую обучающиеся заполняют на занятии и могут сохранить, если зайдут в свой аккаунт (рис. 3). Затем можно завершить практическую часть самостоятельно дома. По завершении работа отправляется преподавателю.

The screenshot shows the 'PyStudy' website interface for the practical part of the lesson. The title is 'Тема 3. Практика'. Below the title, it says 'Выполните тестовые и практические задания и отправьте на проверку.' and 'Чтобы сохранить изменения, войдите в аккаунт Google. Подробнее...'. There are two quiz questions: '1. Что произойдет, если в Python любое число разделить на ноль?' with options 'InvalidOperation exception', '0', 'ZeroDivisionError exception', and 'undefined'; and '2. Каков результат следующего выражения Python?'. A code editor on the right shows a file named 'main.py' with a cursor on line 1.

Рис. 3. Практическая часть занятия

Тема 3. Работа над проектом

Описание
Продолжайте улучшать своего бота, развивая для него новые навыки. Мы предлагаем простую игру в угадывание, которая предскажет возраст пользователя. Он основан на простом математическом трюке. Сначала взгляните на эту формулу:

```
age = (remainder3 * 70 + remainder5 * 21 + remainder7 * 15) % 105
```

Числа `remainder3`, `remainder5` и `remainder7` представляют собой остатки от деления на 3, 5 и 7 соответственно. Получается, что для каждого числа от 0 до 104 вычисление приведет к самому числу. Это идеально подходит для обычного возраста, не так ли? Спросите у пользователя остатки и используйте их, чтобы угадать возраст!

Цель
На этом этапе вы познакомитесь с ботом. Он будет приветствовать вас по имени, а затем попытается угадать ваш возраст с помощью арифметических операций. Ваша программа должна напечатать следующие строки:

```
Hello! My name is Aid.
I was created in 2020.
Please, remind me your name.
What a great name you have, Max!
Let me guess your age.
Enter remainders of dividing your age by 3, 5 and 7.
Your age is {your_age}; that's a good time to start programming!
```

Прочтите три числа из стандартного ввода. Предположим, что все числа будут указаны в отдельных строках. Вместо `{your_age}` этого бот напечатает возраст, определенный по специальной формуле, описанной выше.

Рис. 4. Задание по проекту «Чат-бот»

На странице по работе над проектом имеются описание новой функции бота, конкретные задачи по составлению кода и поясняющий пример (рис. 4). Представлен шаблон, который содержит весь имеющийся функционал бота и задает обучающимся ориентир в предстоящем программировании. Также присутствуют интерпретатор и форма для отправки кода на проверку учителю.

Данный сайт позволит школьникам, пропустившим по какой-то причине некоторые уроки, попытаться самостоятельно освоить предлагаемый материал и навестать упущенное.

По завершении курса (кнопка «Закончить курс») пользователь может скачать полный код консольного приложения «Чат-бот». Таким образом, каждый школьник сможет сравнить собственное приложение с представленным на сайте и сделать выводы о своих возможностях.

Кроме того, на основе данного сайта создан промо-ролик курса, который с успехом может быть применен для привлечения школьников к освоению курса по выбору «Основы программирования на Python».

5. Выводы

1. Наряду с профильной подготовкой обучающихся старших классов необходимо уделять должное внимание предпрофильной подготовке учащихся основной школы. Именно на этапе выпуска из основной школы обучающимся предстоит сделать первый непростой выбор дальнейшего пути.

2. В предпрофильной подготовке также «необходима ориентация на прикладные аспекты школьного образования, т. е. на ценность школьных знаний для обучающихся в возможности их применения в настоя-

щем времени для достижения значимого практического результата — последовательная подготовка к взрослой жизни посредством практического опыта знакомства с профессиями на протяжении школьного обучения» [12]. В этом могут помочь предпрофильные курсы по выбору, дающие возможность осуществить профессиональные пробы.

3. На этапе цифровой трансформации экономики Российской Федерации востребованными становятся ИТ-специалисты, особенно программисты. Многие школьники задумываются о программировании как перспективной будущей деятельности. Поэтому необходимо дать им возможность испытать себя в этой сфере, например, в рамках предпрофильного курса по выбору.

4. При разработке предпрофильных курсов по выбору необходимо учитывать следующие требования:

- доступность;
- краткосрочность;
- оригинальность содержания;
- деятельностный подход в организации.

Тогда у школьников поддерживается зарождающийся интерес и проверяются способности к определенной профессиональной деятельности.

5. Для поддержки процесса обучения широко применяется цифровой образовательный контент. Для реализации оригинальных курсов по выбору больше всего подойдут авторские цифровые образовательные ресурсы, которые можно разработать с помощью цифровых сервисов, например, с помощью платформ для конструирования сайтов, онлайн-курсов, для разработки интерактивных упражнений, тестов и др.

Курсы по выбору представляют собой инструмент реализации одной из основных задач, стоящих перед системой образования, а именно задачи подготовки

члена общества к самостоятельному выбору индивидуальной траектории развития в соответствии со своими способностями и возможностями, к ответственному принятию решений и эффективным действиям в современном меняющемся мире. Разработанный нами курс по выбору «Основы программирования на Python» может помочь школьникам в их профессиональном самоопределении, а созданный нами сайт может стать подспорьем в процессе обучения — как под непосредственным руководством учителя, так и при самостоятельном освоении курса.

Список источников

1. *Абаскалова Н. П.* Предпрофильная подготовка обучающихся на современном этапе развития образования // Вестник психологии и педагогики Алтайского государственного университета. 2021. № 3. С. 1–9.
2. *Бим-Бад Б. М.* Педагогический энциклопедический словарь. М.: Большая Российская энциклопедия, 2002. С. 224–225.
3. *Златопольский Д. М.* Основы программирования на языке Python. М.: ДМК Пресс, 2017. 284 с.
4. *Кочеткова О. А., Долгополов И. В.* Разработка электронных средств учебного назначения по курсу «Программирование на языке Python» // Университетское образование (МКУО2016). Сборник статей XX Международной научно-методической конференции (Пенза, 7–8 апреля 2016 г.). Пенза: Изд-во Пензенского государственного университета, 2016. С. 104–105.
5. *Мамонтова Т. С., Ермакова Е. В., Кашич И. Ф.* Организация предпрофильной и профильной подготовки старшеклассников // Вестник Южно-Уральского Государственного университета. Серия: Образование. Педагогические науки. 2016. № 1. С. 34–43.
6. Письмо Минобрнауки РФ от 04.03.2010 № 03-413 «О методических рекомендациях по реализации элективных курсов». https://monm.rk.gov.ru/file/19_Minobrnavki_03-413.pdf
7. *Поляков К. Ю.* Программирование. Python. C++. Ч. 1: учебное пособие: 4-е изд., стер. М.: Просвещение, 2021. 144 с.
8. *Северенс Ч.* Введение в программирование на Python: учебное пособие. 2-е изд. М.: ИНТУИТ, 2016. 231 с. <https://e.lanbook.com/book/100703>
9. Федеральный государственный образовательный стандарт основного общего образования. <https://fgos.ru/fgos/fgos-ooo/>
10. Федеральный государственный образовательный стандарт среднего общего образования. <https://fgos.ru/fgos/fgos-soo/>
11. *Шуман Х.* Python для детей: руководство / пер. с нем. М. А. Райтман. М.: ДМК Пресс, 2019. 344 с. <https://e.lanbook.com/book/131700>
12. *Яковлев В. А.* Зарубежный опыт и инновации в предпрофильной подготовке школьников // Гуманитарный научный вестник. 2020. № 9. С. 71–75. DOI: 10.5281/zenodo.4084701

DOI: 10.32517/2221-1993-2022-21-1-30-33

Д. М. Блинов*дипломант конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: основное и среднее общее образование», средняя общеобразовательная школа № 67, г. Ижевск, Удмуртская Республика, Россия*

ОБРАЗОВАТЕЛЬНЫЕ АНИМАЦИОННЫЕ ВИДЕОРОЛИКИ В ШКОЛЬНОМ ОБРАЗОВАНИИ

Аннотация

В статье даются понятия анимации, образовательной анимации и анимационного видео. Раскрывается актуальность использования анимационных видеороликов в образовательном процессе, в частности, описывается их использование в тех случаях, когда невозможно показать реальный объект. Описываются преимущества использования образовательных анимационных видеороликов как одного из способов наглядного представления учебной информации. Описываются форматы использования обучающих анимационных видеороликов в процессе обучения, в том числе дистанционный формат. В качестве примера описан обучающий анимационный видеоролик «Четырехтактный двигатель внутреннего сгорания», созданный для использования на уроках физики, рассмотрены цель его использования и некоторые особенности. Даны рекомендации по эффективному использованию анимационных видеороликов в образовательном процессе на основе личного опыта автора.

Ключевые слова: физика, анимация, анимационное видео, электронное обучение, компьютерная модель, дистанционное обучение, электронный урок, дидактический материал.

1. Актуальность использования анимационных видеороликов

Повышение эффективности усвоения учащимися учебного материала, определяющее улучшение качества образования, — первостепенная задача каждого общеобразовательного учреждения. Школа должна формировать у обучающихся целостную систему универсальных знаний, умений и навыков, а также опыт самостоятельной деятельности и личной ответственности, что и определяет современное качество содержания образования [1].

Несомненно, эффективность обучения напрямую зависит от мотивации учеников, которая имеет важное значение не только при классическом формате обучения, но и при электронном обучении (e-learning). Эффек-

тивность обучения напрямую влияет на вовлеченность учеников в процесс обучения, на их результаты обучения. Недостаточная мотивация учащихся или даже отсутствие мотивации к учению — это проблема, с которой педагоги сталкиваются ежедневно.

С целью повышения мотивации обучающихся педагоги используют различные средства развития познавательного интереса, к которым относятся в том числе и наглядные методы обучения, позволяющие изложить материал в образной форме. Наглядные методы обучения используются педагогами постоянно и имеют свои преимущества перед другими методами. В настоящее время данные методы начинают превалировать в обучении. Самый действенный способ пробудить внутреннюю мотивацию ученика — дать ему возможность приобрести новые знания и применить их на

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2022/

Контактная информация

Блинов Дмитрий Михайлович, зам. директора по УВР, средняя общеобразовательная школа № 67, г. Ижевск, Удмуртская Республика, Россия; *адрес:* 426008, г. Ижевск, пер. Широкий, д. 73; *e-mail:* dibli@yandex.ru

D. M. Blinov

School 67, Izhevsk, The Udmurt Republic, Russia

EDUCATIONAL ANIMATED VIDEOS IN SCHOOL EDUCATION

Abstract

The article gives the concepts of animation, educational animation and animated video. The relevance of the use of animated videos in the educational process is revealed, in particular, their use is described in cases where it is impossible to show a real object. The advantages of using educational animated videos as one of the ways of visual presentation of educational information are described. The formats for using educational animated videos in the learning process, including the distance format, are described. As an example, an educational animated video "Four-stroke internal combustion engine", created for use in physics lessons, is described, the purpose of its use and some features are considered. Recommendations are given for the effective use of animated videos in the educational process based on the author's personal experience.

Keywords: physics, animation, animation video, e-learning, computer model, distance learning, e-lesson, didactic material.

практике. Но не всегда практика возможна в условиях учебного кабинета. И вот тут помогает обучающее видео. Видеоматериалы, в том числе образовательное анимационное видео, занимают особое место в процессе обучения.

Что такое образовательное анимационное видео?

Например, профессор Curtin University (Австралия) Р. К. Лоу (R. K. Lowe) считает, что понятие «образовательная анимация» может относиться к любому анимированному изображению, которое используется для обучения и изучения. «Это может быть так же просто, как сделать стрелку движущейся на диаграмме, или так же сложно, как создать собственный полный мультфильм на определенную тему» [11].

Анимация определяется как создание иллюзии движения путем последовательной съемки изображений — кадр за кадром — с использованием любых материалов и технических средств [7]. Слово «анимация» (*англ.* animation) в переводе с английского означает «одушевление», «оживление». В толковом словаре Т. Ф. Ефремовой анимация — это киносъемка последовательных фаз движения рисованных или объемных фигур, в результате которой при показе на экране у зрителя создается иллюзия их движения [5].

Термин «**обучающая анимация**» означает любое динамическое изображение, которое может быть использовано в образовательных целях. Это может быть изображение движения автомобиля по кривой траектории или полноценный анимационный фильм, например, о каком-либо физическом явлении.

Сегодня **образовательные анимационные фильмы** становятся неотъемлемой частью процесса обучения. Основная задача такого фильма — привлечение внимания обучающихся с целью раскрыть им определенную учебную тему в доступной и простой форме.

Динамические изображения, в отличие от статических картинок, более эффективны в плане привлечения внимания учащихся и позволяют сделать обучение более интересным и увлекательным. Ричард Лоу утверждает, что «образовательная анимация в виде анимационного фильма или же небольшой анимации объекта особенно эффективна, когда она заменяет статические изображения, которые пытаются показать движение и изменение» [11]. Например, анимация может быть с успехом использована, когда надо продемонстрировать физические процессы, такие как ядерные реакции, движение планет Солнечной системы или процесс деления живой клетки. Такие процессы традиционно изображались в учебниках с помощью статических изображений, но статические изображения не могут передать всю суть явления. К тому же не все ученики имеют хорошее воображение, и потому у большинства из них, после того как они безуспешно пытались расшифровать статические изображения, просто пропадает интерес к дальнейшему изучению темы. Анимационные компьютерные модели физических явлений и процессов, используемые на уроках, позволяют учащимся усвоить материал более глубоко [3, 6].

Но, если анимация создана некачественно, она может оттолкнуть ученика. Поэтому эффективность анимации, используемой на учебных занятиях, в значительной степени зависит от ее качества.

Для эффективной анимации характерны следующие особенности:

- гармоничное сочетание текста и изображений;
- не слишком большой объем информации;
- удобная для восприятия скорость движения объектов;
- акценты на самых важных элементах;
- голосовое и звуковое сопровождение.

Использование обучающих анимационных видеороликов в преподавании школьных предметов позволяет справиться с такой задачей, как объяснение новых понятий, процессов и явлений. Этому способствуют присущие образовательным видеороликам наглядность, различные каналы передачи информации (изображения, звук и текст), последовательность в повествовании и эмоциональные акценты. Объяснение нового материала является определенным «встраиванием нового знания» в картину мира ученика. Поэтому создание обучающего анимационного видео должно начинаться с исследования аудитории и определения грани «знания — незнания», что позволит соблюсти соотношение между объяснением новой информации и ее применением [8]. Главное правило: учебный материал видеоролика должен быть адаптирован к тому уровню сложности, которым владеют ученики, и к тому опыту, который у них есть [2].

Таким образом, использование обучающего видео в преподавании школьных предметов — это один из способов вовлечь в образовательный процесс чувства ученика, заинтересовать его и побудить к тому, чтобы он сам начал задавать вопросы. Усвоение нового материала невозможно без собственных усилий учащихся, для которых необходима мотивация.

Обучающее анимационное видео учитель может подобрать для своей работы на открытых онлайн-платформах в сети Интернет. Но не всегда можно найти подходящий по конкретной теме качественный видеоролик. Можно видеоролики создавать самостоятельно. Но это очень трудоемкий процесс. Лично для меня создание образовательного контента — это определенный способ саморазвития и совершенствования своих компетенций.

2. Анимационный видеоролик «Четырехтактный двигатель внутреннего сгорания»

Один из созданных автором данной статьи обучающих анимационных видеороликов — «Четырехтактный двигатель внутреннего сгорания». Этот анимационный видеоролик (рис. 1) предназначен для использования на уроках физики в VIII классе при изучении темы «Двигатель внутреннего сгорания» [4, 9, 10].

Длительность видеоролика — 3 мин 50 с.

Цель видеоролика: объяснить и продемонстрировать устройство и работу двигателя внутреннего сгорания (ДВС).

Актуальность данного анимационного видеоролика и целесообразность его использования на уроках объясняется тем, что невозможно продемонстрировать работу двигателя внутреннего сгорания в реальности и показать двигатель изнутри. Анимационная компьютерная модель ДВС позволяет решить эту проблему.



Рис. 1. Анимационный видеоролик «Четырехтактный двигатель внутреннего сгорания»



Рис. 2. Двигатель внутреннего сгорания



Рис. 3. Принцип работы ДВС

Видеоролик сопровождается аудиообъяснением и текстовым описанием, что позволяет учащимся получать информацию всеми способами восприятия (рис. 2).

Представленная в видеоролике анимация позволяет доступно объяснить принцип работы ДВС. Компьютер-

ная анимация модели ДВС передает все особенности работы реального двигателя. Каждый такт рабочего цикла ДВС представлен анимацией с его подробным объяснением (рис. 3).

В заключении видеоролика рассказывается о применении ДВС, что дает полную картину по данной теме.

Возможные формы работы с образовательным видеороликом:

- *Фронтальная работа с классом* — объяснение нового материала учителем с демонстрацией видеоролика на интерактивной доске: учитель может при необходимости останавливать видеоролик, дополнительно комментируя его для полноты объяснения; при необходимости видео можно перематывать назад для повторной демонстрации.
- *Самостоятельная работа учащихся на уроке* — ученикам дается время для самостоятельного просмотра материала; перед просмотром учитель ставит задачу или озвучивает вопросы, на которые нужно будет дать ответы после просмотра видеоролика; после просмотра видеоролика учащимся предлагается ответить на ряд вопросов для самоконтроля, помогающих закрепить пройденный материал, или обсудить видеоролик.
- *Самостоятельная работа учащихся дома* — ученикам предлагается дома в качестве повторения или самостоятельного изучения (при дистанционных или домашних формах обучения) просмотреть видеоролик, при этом каждый ученик может это сделать в собственном темпе и при необходимости посмотреть видеоролик многократно.

3. Опыт использования образовательных видеороликов

На основе личного опыта можно сформулировать краткие **рекомендации по эффективному использованию образовательных видеороликов:**

- содержание видеороликов должно быть связано с темой урока;
- новый видеоролик учитель должен просмотреть до урока, выбрать фрагменты для показа на уроке, составить вопросы к беседе по содержанию видеоролика;
- необходимо определить время и место видеоролика на уроке; в зависимости от дидактических задач урока видеоролик может быть показан отдельными фрагментами на различных этапах урока;
- при необходимости акцентирования внимания учеников на отдельных моментах учитель может прокомментировать кадры видеоролика, поставив видео на паузу;
- если видеоролик планируется демонстрировать полностью, то перед просмотром обучающимся ставятся познавательные задачи, которые должны быть решены в процессе просмотра видеоролика; после просмотра проводится беседа или учащиеся отвечают на вопросы;
- в ряде случаев учитель может заранее познакомить учеников с реальными объектами или их моделями, которые будут демонстрироваться в видеоролике. Например, перед просмотром видеоролика «Четырехтактный двигатель внутреннего сгорания» демонстрируется натуральная уменьшенная модель ДВС;
- при просмотре учащимися видеоролика вне школы (дома, при дистанционном обучении) учителю

необходимо подготовить ряд вопросов, на которые ученики должны ответить письменно или по которым они должны пройти онлайн-тест.

Опыт использования анимационного видео в образовательном процессе также позволил выявить **ряд преимуществ видеороликов перед другими средствами обучения** (плакат, презентация):

- возможность более оперативно и доступно донести до обучающихся информацию;
- высокая степень наглядности;
- возможность многократно просматривать материал;
- возможность организации индивидуального подхода к учащемуся;
- возможность просмотра материала с паузами;
- возможность проводить обучение в любом формате (офлайн, онлайн);
- возможность размещения видеороликов в глобальной сети.

В целом использование видеороликов в учебном процессе позволяет учителям решать не только задачу повышения эффективности усвоения учебного материала учащимися, но и еще несколько задач:

- активизация деятельности обучающихся;
- повышение у них интереса к изучаемому предмету;
- наглядное представление материала;
- компактное хранение информации.

Список источников

1. Бердникова И. А. Повышение качества высшего образования // Проблемы управления качеством образования в вузе. Сборник статей II Международной научно-практической конференции. Пенза: РИО ПГСХА, 2007. 310 с.
2. Берлёв С. В. Особенности применения видеоматериалов и учебных презентаций в преподавании технических дисциплин // Проблемы и перспективы развития образования. Материалы Международной научной конференции: в 2 т. Т. 2. Пермь: Меркурий, 2011. С. 184–186.
3. Губернаторова Л. И., Потехин К. А. Новые информационные технологии в процессе преподавания физики: учебное пособие. Владимир: Собор, 2005. 124 с.
4. Дик Ю. И., Коровин В. А., Орлов В. А. Программы для общеобразовательных учреждений: Физика. Астрономия. 7–11 кл. 4-е изд., перераб. М.: Дрофа, 2004. 256 с.
5. Ефремова Т. Ф. Новый словарь русского языка. Толково-словообразовательный. М.: Русский язык. 2000. 1615 с.
6. А. Ф. Кавтрев. Опыт использования компьютерных моделей на уроках физики в школе «Дипломат» // Сборник РГПУ им. А. И. Герцена «Физика в школе и вузе». СПб.: Образование, 1998. С. 102–105.
7. Кривуля Н. Г. Эволюция художественных моделей в процессе развития мировых аниматографий: дис. ... д-ра искусствоведения. М., 2009. 494 с.
8. ЛеФевьер Л. Искусство объяснять. Как сделать так, чтобы вас понимали с полуслова. М.: Манн, Иванов и Фербер, 2013. 290 с.
9. Перышкин А. В. Физика. 8 класс: учебник для общеобразовательных учреждений. М.: Дрофа, 2019. 240 с.
10. Пурышева Н. С., Вежеевская Н. Е. Физика. 8 класс: учебник для общеобразовательных учреждений. М.: Дрофа, 2020. 249 с.
11. Lowe R. K., Schnotz W. (Eds.). Learning with animation: Research implications for design. New York: Cambridge University Press, 2008. P. 304–356.

DOI: 10.32517/2221-1993-2022-21-1-34-39

Т. В. Баракина

дипломант конкурса ИНФО-2021 в номинации «Цифровая трансформация образования: дошкольное и начальное общее образование», Омский государственный педагогический университет, г. Омск, Россия

ФОРМИРОВАНИЕ ЗНАКОВО-СИМВОЛИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ У МЛАДШИХ ШКОЛЬНИКОВ В ПРОЦЕССЕ ОЗНАКОМЛЕНИЯ С ИНФОРМАЦИОННЫМИ ПРОЦЕССАМИ

Аннотация

Сегодня мы наблюдаем цифровую трансформацию в самых разных сферах жизни, в том числе в образовании. Образовательные учреждения укомплектовывают новым оборудованием, подключают к высокоскоростному интернету, создается цифровая система документооборота и т. д. Разрабатываются новые и модернизируются уже существующие образовательные технологии с учетом возможности использования цифрового оборудования, способствующие развитию новых личностных качеств у обучающихся, «цифрового» мышления.

В статье рассматривается влияние цифровой трансформации образования на особенности восприятия информации обучающимися, на развитие знаково-символической деятельности. Описаны функции знаково-символических средств в деятельности (коммуникативная, познавательная, замещающая), виды знаково-символической деятельности человека (моделирование, кодирование, схематизация, замещение), влияние социальных сетей и мессенджеров на восприятие информации обучающимися, процесс изменения в восприятии человеком письменной (цифровой) информации. Представлены две серии упражнений: 1) по ознакомлению младших школьников с информационными процессами с использованием языка эмодзи, 2) по ознакомлению с редактированием информации в «искаженных текстах».

Ключевые слова: цифровая трансформация, информация, информационные процессы, редактирование, кодирование, моделирование, схематизация, замещение, знаково-символическая деятельность, информатика, младшие школьники.

1. Введение

В 2020 году президент России В. В. Путин, выступая на международной онлайн-конференции Artificial Intelligence Journey (AI Journey), отметил важность цифровой трансформации России, которая должна затронуть «каждого человека, каждую семью, каждую отрасль экономики и социальной сферы, каждую организацию

и каждый уровень власти, всю систему государственного управления» [6].

Что же понимается под *цифровой трансформацией*? Это трансформация системы управления путем пересмотра стратегии, моделей, операций, продуктов, маркетингового подхода и целей, обеспечиваемая принятием цифровых технологий. Она призвана ускорить продажи и рост бизнеса и увеличить эффективность

Контактная информация

Баракина Татьяна Вячеславовна, канд. пед. наук, доцент, доцент кафедры предметных технологий начального и дошкольного образования, Омский государственный педагогический университет, г. Омск, Россия; *адрес:* 644099, Россия, г. Омск, наб. Тухачевского, д. 14; *e-mail:* barakina77@mail.ru

T. V. Barakina

Omsk State Pedagogical University, Omsk, Russia

FORMATION OF SIGN AND SYMBOLIC ACTIVITY IN JUNIOR SCHOOLCHILDREN IN THE PROCESS OF ACQUAINTANCE WITH INFORMATION PROCESSES

Abstract

Currently, we are witnessing a digital transformation in various areas of life, including education. Educational institutions are equipped with new equipment, connected to high-speed Internet, a digital document management system is being created, etc. New educational technologies are being developed and existing ones are being modernized, taking into account the possibility of using digital equipment, which contribute to the development of new personal qualities in students, "digital" thinking.

The article discusses the impact of the digital transformation of education on the peculiarities of the perception of information by students, on the development of sign-symbolic activity. The functions of sign-symbolic means in activity (communicative, cognitive, substitute), types of sign-symbolic human activity (modeling, coding, schematization, substitution), the influence of social networks and instant messengers on the perception of information by students, the process of change in the perception of written (digital) information are described in the article. Two series of exercises are presented: 1) to familiarize younger students with information processes using the emoji language, 2) to familiarize them with editing information in "distorted texts".

Keywords: digital transformation, information, information processes, editing, coding, modeling, schematization, substitution, sign-symbolic activity, informatics, junior schoolchildren.

деятельности организаций, не относящихся к чисто коммерческим (например, университетов и других образовательных учреждений) [8].

Цифровая трансформация касается любой сферы, в том числе образования. В настоящее время образовательные учреждения укомплектовывают новым оборудованием, подключают к высокоскоростному интернету, создается цифровая система документооборота и т. п. Разрабатываются новые и модернизируются уже существующие образовательные технологии с учетом возможности использования цифрового оборудования, способствующие развитию новых личностных качеств у обучающихся, «цифрового» мышления.

Рассмотрим влияние цифровой трансформации в образовании на особенности восприятия информации ребенком, ее переработку и передачу, на развитие умения работать с различными знаково-символическими средствами.

2. Знаково-символическая деятельность обучающихся в начальной школе

В настоящее время одной из задач обучения в начальной школе является формирование знаково-символической деятельности у обучающихся, обеспечивающей конкретные способы преобразования учебного материала.

Под **знаково-символической деятельностью** понимается отражение и преобразование действительности, ее объектов и интеробъектных связей при помощи специальных средств, основной характеристикой которых является нетождественность отражаемому объекту [7].

Н. Г. Салмина и А. С. Турчин выделяют следующие **основные функции знаково-символических средств в деятельности** [цит. по: 7, 9]:

- *коммуникативная* — передача информации от одного человека к другому;
- *познавательная* — отражение реальности с выделением новых ее признаков;
- *замещающая* — функциональное замещение объекта знаковым средством.

В соответствии с функциями знаково-символической деятельности выделяют следующие ее **виды** (Н. Г. Салмина, А. С. Турчин) [цит. по: 9]:

- *моделирование* — высший уровень работы со знаково-символическими средствами; предполагает четкое разделение работы в реальном и символическом планах, с использованием материализованных или мысленных моделей, в которых отражены структурные, функциональные или генетические связи объекта;
- *схематизация* — знаковым средствам отводится ориентировочная роль, заключающаяся в структурировании реальности при помощи «активной наглядности» (т. е. материализации), специфической чертой является раскрытие пространственных характеристик как объекта, так и знаково-символических средств;
- *кодирование* — сжатие информации с целью возможно более точной передачи информации при общении при помощи разных типов связи знака и объекта (изображение, выражение отношения к нему, обозначение и т. д.);

- *замещение* — часто используется как терминологический эквивалент знаково-символической деятельности вообще, в более узком значении — перенесение функций объекта на знак.

Все указанные виды знаково-символической деятельности являются объектом изучения предметной области «Математика и информатика» в начальной школе при рассмотрении таких понятий, как «информация» и «информационные процессы».

Рассмотрим возможность ознакомления младших школьников с информационными процессами через изучение влияния социальных сетей и мессенджеров на особенности восприятия, передачу и хранение информации человеком.

3. Влияние социальных сетей и мессенджеров на восприятие информации обучающимися. Язык эмодзи

Вы не замечали, что люди все реже и реже стали звонить друг другу, предпочитая написать сообщение?.. А еще лучше — просто отправить смайлик или стикер.

На смену вербальному общению пришло знаковое, символьное. Появился целый язык эмодзи — графический язык идеограмм и смайлов, который используют для выражения эмоций в интерактивном общении, на веб-сайтах.

Смайлики, с помощью которых мы можем продемонстрировать разные эмоции (грусть, радость, злость и т. д.), помогая собеседнику понять смысл сообщения, отличить серьезные разговоры от смешных, в 1982 году предложил профессор Университета Карнеги — Меллона (Питтсбург, США) Скотт Эллиот Фалман (Scott Elliot Fahlman) — улыбку он обозначил с помощью двоеточия и скобки [10].

Благодаря японскому оператору связи DoCoMo в 1995 году смайлики распространились среди огромного числа пользователей. В набор значков добавили сердечко, и это так понравилось людям, что компания резко повысила свой рейтинг популярности. Но так как эти символы существенно нагружали сеть, в 1999 году японский дизайнер Синетака Курита и другие сотрудники компании разработали набор символов-картинок размером 12×12 пикселей — **эмодзи**. Для того чтобы стать популярными во всем мире, им потребовалось еще немного времени — пока такие платформы, как Google, Apple, Android, Facebook не внедрили их поддержку.

Картинки желтого цвета, выражающие эмоции, возникли именно в Японии, что выглядит вполне логичным. Дело в том, что японский язык достаточно сложный, в нем не просто передать суть беседы исключительно в письменной форме. Случается, что понять разговор полностью можно только с использованием жестов и интонации. Жители страны испытывали трудности в общении через интернет и sms, потому что, общаясь такими способами, невозможно увидеть и услышать своего собеседника. Создание символов-эмоций решило эту проблему.

Когда эмодзи стали популярными во всем мире, западная культура изменила некоторое традиционное

значение смайлов из-за неполного понимания японской культуры. Многие пиктограммы пользователи воспринимают и передают не так, как это было задумано в оригинале, а по своему собственному усмотрению. В итоге могут возникнуть некоторые трудности в общении. Поэтому, отправляя смайлик представителю другой культуры, стоит быть внимательным [10].

Наших учеников необходимо подготовить к жизни в мире эмодзи, научить понимать, редактировать, передавать информацию с помощью данного графического языка.

4. Серия упражнений по ознакомлению младших школьников с информационными процессами с использованием языка эмодзи

Задание 1.

Установи соответствие между символом и его значением (рис. 1).



Странно!	Это очень смешно!	Браво!	Все хорошо!	Громко плачу!	Я взбешен!
----------	-------------------	--------	-------------	---------------	------------

Рис. 1

Задание 2.

Существуют основные смайлики, которые используются в социальных сетях и мессенджерах (рис. 2).

Найди в интернете смайлики, представленные на рисунке, и запиши их значения.



Рис. 2

Задание 3.

Выделенные в предложении слова замени на смайлики — нарисуй их.

«Мне сегодня очень **грустно** _____. Все валится из рук. Это какой-то **кошмар** _____. Но вчера было все так **прекрасно** _____. **Почему?** _____.

Сравни свое сообщение с сообщениями других учеников. Совпали ли ваши символы?

Задание 4.

Определи смысл сообщения, данного с помощью языка эмодзи (рис. 3). Запиши его вербально — предложением.



Рис. 3

Сравни свой вариант ответа с вариантами других учеников.

В каком значении использованы символы в сообщении?

Задание 5.

Какие еще символы кроме смайликов используются для передачи сообщений в социальных сетях и мессенджерах? Приведи примеры изображений и укажи их значения. Ответ оформи в виде таблицы (табл. 1).

Таблица 1

Символ	Значение символа	Символ	Значение символа

Задание 6.

Раздели символы, изображенные на рисунке (рис. 4), на группы. Дай названия группам.

По какому принципу ты выполнил классификацию? Можно ли выделить группы иначе?

Задание 7.

Заполни таблицу эмоций с помощью языка эмодзи. Нарисуй символы в соответствующих ячейках таблицы (табл. 2). Обрати внимание, что первоначально эмодзи — изображения человеческого лица — отражали такие эмоции, как горе и радость, нейтральную реакцию. Сейчас появились символы, изображающие животных,

растения, жесты и т. д. С их помощью также можно показать эмоцию, которую условно называют «нетрадиционная эмоция», ее значение можно понять только из контекста сообщения.

Таблица 2

Положительная эмоция	Отрицательная эмоция
Нейтральная эмоция	Нетрадиционная эмоция



Рис. 4*

* <https://avatars.mds.yandex.net/get-pdb/1813431/f7ee2263-7c74-4a8b-8b27-2bfa7cbd4064/s1200?webp=false>

 <p style="text-align: center;">МЯЧИК</p> <p>Наша Таня громко плачет: Уронила в речку мячик. — Тише, Танечка, не плачь! Не утонет в речке мяч.</p>	<p style="text-align: center;">Мячик</p> <p>Наша Таня громко плачет: Уронила в речку мячик. — Тише, Танечка, не плачь! Не утонет в речке мяч!</p>	
--	--	--

Задание 8.

Передай с помощью символов, представленных в задании 6, смысл стихотворения. Нарисуй последовательность символов в пустой ячейке таблицы (табл. 3).

Задание 9.

В сообщениях очень часто используются скобки, которые придают ему соответствующий тон и настрой:

-) — позитивный или веселый тон;
- (— негативный, грустный или печальный тон.

Но могут использоваться сразу несколько скобок. Попробуй определить их значение. Заполни пропуски в таблице (табл. 4).

Таблица 4

Символ	Значение символа	Символ	Значение символа
)	Улыбка	(Грусть
))		((
)))	Громкий смех	((((

Задание 10.

Существуют комбинации скобок и других символов. Рассмотрим примеры, приведенные в таблице (табл. 5), найди в интернете подобные комбинации и укажи их в правой части таблицы.

Таблица 5

Символ	Значение символа	Символ	Значение символа
;—)	Подмигивание		
:(Грусть со слезами		
%)	Головокружение, шок		
8—)	Смайлик в черных очках (скрытность)		
:)	Улыбка		

5. Изменения в восприятии человеком письменной (цифровой) информации

Еще одно последствие цифровой трансформации — изменения в восприятии письменной информации человеком.

Примерно в 2003 году в Сети «гуляла вирусная информация», в которой сообщалось, что неважно, в каком порядке идут буквы в слове, лишь бы первая и последняя буквы были на своих местах, а остальное мозг сам скомпонует и вычленил смысл текста. И приводился такой пример текста: «По рзезульаттам илссеовадний одонго английсокто унвиертисета, не иеemt занчнейя, в кокам пряокде рсапожолены бквуы в солве. Галвоне, чотбы преавя и пслоендя бквуы блыи на мсете. Осатлыине бквуы мгоут селдовтаь в плоонм беспорядке, все рвано ткест чтаитсея без побрелм. Пичрионий эгото ялвятеся то, что мы чиатем не кдажю бквуу по отдльенотси, а все солво цликееом» [1].

И в этом есть определенная доля правды. Но в русском языке не все так просто. Стак называемыми «длинными словами» это не работает.

Можно отметить, что для письменных сообщений современного человека (в частности, сообщений в Сети) характерны следующие особенности [5]:

- отказ от знаков пунктуации;
- лояльное отношение к автозамене, опечаткам, ошибкам;
- дробление текста на малые части;
- краткие и однозначные формулировки;
- отказ от распространенных предложений;
- использование сокращений и т. п.

Заметим, что лояльное отношение к подобным особенностям современной письменной речи не означает принятие нарушений существующих норм и правил использования языка, поэтому в процессе ознакомления обучающихся с таким информационным процессом, как редактирование, важно включать задания на исправление орфографических, синтаксических и стилистических ошибок. Кроме того, необходимо учить детей работать как с распространенными, так и нераспространенными предложениями, преобразуя одни в другие.

6. Серия упражнений по ознакомлению младших школьников с редактированием информации в «искаженных текстах»

Задание 1.

Восстанови сокращения (табл. 6).

Таблица 6

Сокращение	Полное слово
СПС	Спасибо!
СПКН	
ВТ	
КСТ	
М-ДУ	
МТД	Молодец! Так держать!
ОС	Очень смешно!
ПТН	
ДР	
НЗЧ	Не за что!
ИНЕТ	
ЛЮ	
НЗЧ	
!ОС	Не очень смешно!

Задание 2.

Придумай свои сокращения слов или приведи примеры, которые тебе встречались в сообщениях (табл. 7).

Таблица 7

Полное слово	Сокращение

Задание 3.

Восстанови текст. Вставь пропущенные буквы.

«Во всем дме есть тоько оно не всем обкновнне сщество — Крлсн, ктрй жвёт на крш. Да, он жвт на крше, и одно это уже нбыкнвнно. Быть мжт, в дрхх грдах длго обстт инче, но в Сткглъме пчти нкгда не слчтсся, чтбы кто-нбудь жл на крше, да ещё в тдльном маленьком дмке. А вт Крлсн, прдставьте сбе, жвёт имнно там» [2].

Задание 4.

Расставь знаки препинания в тексте.

«Карлсон это маленький толстенный самоуверенный человек и к тому же он умеет летать На самолётах и вертолётах летать могут все а вот Карлсон умеет летать сам по себе Стоит ему только нажать кнопку на животе

как у него за спиной тут же начинает работать хитроумный моторчик С минуту пока пропеллер не раскрутится как следует Карлсон стоит неподвижно но когда мотор заработает вовсю Карлсон взмывает ввысь и летит слегка покачиваясь с таким важным и достойным видом словно какой-нибудь директор конечно если можно себе представить директора с пропеллером за спиной» [2].

Задание 5.

Преврати распространенные предложения в нераспространенные (табл. 8).

Таблица 8

Распространенное предложение	Нераспространенное предложение
Малыш был очень рад, что познакомился с Карлсоном [3]	
Как только Карлсон прилетал, начинались необычайные приключения [3]	

7. Заключение

Цифровая трансформация — это не только обеспечение различных сфер услуг и деятельности новым оборудованием, создание цифровых платформ и продуктов, но и те изменения, которые происходят с человеком и обществом.

Ребенок должен быть готов к жизни в цифровом обществе, он должен уметь работать с огромным потоком информации, который на него обрушивается. Поэтому уже с дошкольного и младшего школьного возраста необходимо знакомить обучающихся с информацией, информационными процессами и их видоизменением под влиянием интернета, социальных сетей, мессенджеров и пр. [4].

Список источников

1. Алиев Р. О том, как мы ворпсиманием тескт // Хабр. <https://habr.com/ru/post/148896/>
2. Линдгрэн А. Три повести о Малыше и Карлсоне. Красноярск: Универс, 1993. 415 с.
3. Мячик. Стихи Агнии Барто // Аудиосказки.нет. <http://audioskazki.net/archives/1130>
4. Никулина Т. В., Стариченко Е. Б. Информатизация и цифровизация образования: понятия, технологии, управление // Педагогическое образование в России. 2018. № 8. С. 107–113.
5. Поздеева С. И. К проблеме формирования у младших школьников умений работать с информационными текстами // Вестник ТГПУ. 2016. № 5 (170). С. 28–31.
6. Путин заявил о необходимости цифровой трансформации России // ТАСС. 4 декабря 2020 года. <https://tass.ru/ekonomika/10172635>
7. Цветков А. В. Об универсальной структуре знаково-символической деятельности // Известия Российского государственного педагогического университета им. А. И. Герцена. 2008. № 75. С. 266–271.
8. Цифровая трансформация // Википедия. https://ru.wikipedia.org/wiki/Цифровая_трансформация
9. Шулекина Ю. А. Специфика развития знаково-символической деятельности младших школьников // Системная психология и социология. 2019. № 3 (31). С. 46–54.
10. Эмодзи — что это такое, зачем нужны и как ими пользоваться. <https://zen.yandex.ru/media/sharaut/emodzi-chto-eto-takoe-zachem-nujny-i-kak-nimi-polzovatsia-60b86705872b3c5c43d3a8c5>



DOI: 10.32517/2221-1993-2022-21-1-40-45

О. С. Сальникова, О. Л. Селяничев

Череповецкий государственный университет, г. Череповец, Вологодская область, Россия

ОСОБЕННОСТИ ЗАНЯТИЙ ПО ГЕОИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ СО ШКОЛЬНИКАМИ В РАМКАХ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

Аннотация

В статье раскрываются вопросы организации занятий по геоинформационным технологиям со школьниками в рамках дополнительного образования. Дополнительное образование рассматривается как деятельность Дома научной коллаборации на базе Череповецкого государственного университета. Целью работы является описание системы заданий в рамках программы «Геоинформационные технологии» через призму мотива путешествия. С опорой на анализ психолого-педагогической литературы в статье выделены следующие положения для системы занятий: освоение понятия «геоинформационные технологии»; учет основных характеристик понятия «путешествие»; опора на личный опыт учеников и эмоциональную составляющую; использование активных и интерактивных методов обучения. В работе описаны творческие кейсы по образовательной программе «Геоинформационные технологии», которые направлены на изучение новой информации, анализ полученной информации, фиксацию и представление полученных результатов. В основу планирования занятий заложены межпредметные связи. Комплекс кейсовых заданий направлен на формирование личностных, метапредметных и предметных результатов, а также позволяет развивать интерес к изучению географии и истории планеты Земля с использованием современных информационных технологий. В качестве инструментов для выполнения кейсовых заданий выступают Google Карты и Яндекс.Карты.

Ключевые слова: геоинформационные технологии, геоинформационные системы, дополнительное образование, Дом научной коллаборации.

1. Актуальность программы дополнительного образования «Геоинформационные технологии»

Значение дополнительного образования для развития и самоопределения личности ребенка подчеркнута

в психолого-педагогической литературе: дополнительное образование «может стать незаменимой составной частью процесса самопознания, самоопределения и личностного развития каждого ребенка» [10, с. 47]. Важным является выбор направления или кружка по интересам с учетом индивидуальных особенностей

Контактная информация

Сальникова Ольга Сергеевна, канд. филол. наук, доцент кафедры математического и программного обеспечения электронно-вычислительных машин, Череповецкий государственный университет, г. Череповец, Вологодская область, Россия; *адрес:* 162600, Россия, Вологодская обл., г. Череповец, пр-т Луначарского, д. 5; *e-mail:* ossalnikova@chsu.ru

Селяничев Олег Леонидович, канд. тех. наук, доцент, доцент кафедры математического и программного обеспечения электронно-вычислительных машин, Череповецкий государственный университет, г. Череповец, Вологодская область, Россия; *адрес:* 162600, Россия, Вологодская обл., г. Череповец, пр-т Луначарского, д. 5; *e-mail:* olselianichev@chsu.ru

O. S. Salnikova, O. L. Selyanichev

Cherepovets State University, Cherepovets, Vologda Region, Russia

FEATURES OF CLASSES ON GEOINFORMATION TECHNOLOGIES WITH SCHOOLCHILDREN IN THE FRAMEWORK OF ADDITIONAL EDUCATION

Abstract

The article reveals the issues of organizing classes on geoinformation technologies with schoolchildren in the framework of additional education. Additional education is considered as an activity of the House of Scientific Collaboration on the basis of Cherepovets State University. The purpose of the work is to describe the system of tasks within the framework of the program "Geoinformation technologies" through the prism of the travel motive. Based on the analysis of psychological and pedagogical literature, the article highlights the following provisions for the system of classes: mastering the concept of "geoinformation technologies"; taking into account the main characteristics of the concept of "travel"; relying on the personal experience of students and the emotional component; the use of active and interactive teaching methods. The paper describes creative cases on the educational program "Geoinformation technologies", which are aimed at studying new information, analyzing the information received, fixing and presenting the results obtained. Inter-subject connections are the basis for planning classes. The complex of case tasks is aimed at the formation of personal, meta-subject and subject results, and also allows to develop interest in studying the geography and history of the planet Earth using modern information technologies. Google Maps and Yandex.Maps act as tools for solving case tasks.

Keywords: geoinformation technologies, geoinformation systems, additional education, House of Scientific Collaboration.

и предпочтений ребенка. Так, в работе О. С. Щербиной рассмотрены возможности дополнительного образования для одаренных детей: «Благодаря свободе выбора вида деятельности, отсутствию жестких рамок, организации дополнительного образования позволяют одаренным детям реализовать свои способности, раскрыть свои потенциалы» [21, с. 419]. Дополнительное образование включает не только знания, но, что более ценно, практический опыт в той или иной области: «Результативность образования сегодня оценивается способностью успешно решать актуальные проблемы, с которыми ежедневно сталкивается человек» [20, с. 313].

В работах педагогов и методистов описаны особенности организации дополнительного образования на базе образовательных центров [15, 18], общеобразовательной школы [10, 13, 20] и высших учебных заведений [11, 15, 20, 21].

Мы будем рассматривать дополнительное образование как деятельность центра «Дом научной коллаборации им. академика Ивана Павловича Бардина» на базе Череповецкого государственного университета, а именно образовательную программу «Геоинформационные технологии».

Программа «Геоинформационные технологии» направлена на формирование у школьников понимания взаимосвязи между информационными технологиями и изучением поверхности планеты Земля, на развитие умения собирать, анализировать и визуализировать информацию об окружающем мире. Данное направление входит в блок «Урок технологии» и реализуется в рамках договора о сетевом взаимодействии со школами города; программа рассчитана на один учебный год. Данная работа проводится во исполнение Указа Президента Российской Федерации от 7 мая 2018 г. № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года», в котором выражена необходимость обновления содержания и совершенствования методов обучения предметной области «Технология» [19].

Внимание к изучению школьниками географии и геоинформационных технологий обусловлено современными тенденциями в образовании к интеграции и междисциплинарности [1, 8, 11, 14, 16], а также особенностями образовательной области «Технология» [6, 7]. Подчеркнем, что «дополнительное образование по образовательной области “Технология” призвано развивать технологическую культуру, самостоятельность, творчество, способности, интересы и склонности учащихся в области науки и техники» [1, с. 7]; «дисциплина “Технология” является междисциплинарным предметом, где объединены различные дисциплины: физика, химия, информатика, русский язык, черчение, математика, экономика и др.» [11, с. 496].

Роль активности самого обучающегося неоспорима и неоднократно подчеркивалась в психолого-педагогической литературе, а в условиях VUCA-мира* она становится определенным требованием: «Обучающийся

принимает активное осознанное участие в обучении, он выступает инициатором индивидуальных образовательных траекторий и приобретает необходимые компетенции с учетом своих потребностей» [4, с. 32].

Образовательные программы Дома научной коллаборации реализуются через выполнение обучающимися кейсовых заданий и участие в проектной деятельности. Последняя играет важную роль в образовательном процессе, так как «центры ДНК нацелены на создание среды ускоренного развития для детей на площадках вузов, что позволяет им получать знания и экспертизу от научных сотрудников и выстраивать собственную образовательную траекторию, главным образом, при вовлечении в проектную деятельность» [20, с. 314].

Цель данной статьи — описать систему заданий в рамках программы дополнительного образования «Геоинформационные технологии» через призму мотива путешествия.

2. Методические особенности занятий по геоинформационным технологиям

В качестве основы для системы занятий выступают следующие методические ориентиры:

- 1) освоение понятия «геоинформационные технологии»;
- 2) учет основных характеристик понятия «путешествие»;
- 3) опора на личный опыт учащихся и эмоциональную составляющую;
- 4) использование активных и интерактивных методов обучения.

2.1. Введение понятия «геоинформационные технологии»

В качестве пропедевтической подготовки слушателей курсов (разновозрастные учащиеся среднего звена общеобразовательных школ г. Череповца) в решении задачи приобщения школьников к геоинформационным технологиям отправной точкой стало введение самого понятия через определение инструмента, реализующего эти технологии, — **географическая информационная система (геоинформационная система, ГИС)**: совокупность технических, программных и информационных средств, обеспечивающих ввод, хранение, обработку, математико-картографическое моделирование и образное интегрированное представление пространственных и соотнесенных с ними атрибутивных данных для решения задач территориального планирования и управления [2]. Оставляя за рамками занятий обсуждение истории возникновения и развития ГИС, их классификации, необходимо перечислить области применения ГИС (местное самоуправление, коммунальное хозяйство, охрана окружающей среды, здравоохранение, транспорт, розничная торговля, финансовые услуги и др.), тем самым указать их востребованность в широком диапазоне видов человеческой деятельности и этим обеспечить заинтересованность обучающихся в освоении предмета. Учитывая возрастные особенности детей (V–VIII классы средней школы), важно в доступных формах описать архитектуру, структуру ГИС и их функциональные возможности, что позволит в обсуждении

* VUCA: Volatility — нестабильность, Uncertainty — неопределенность, Complexity — сложность, Ambiguity — неоднозначность, т. е. VUCA-мир — это новая реальность: нестабильная, хаотичная, неоднозначная. <https://teachline.ru>

назвать круг задач, для решения которых ГИС подойдут лучшим образом.

2.2. Характеристика понятия «путешествие»

Перечисляя возможные применения ГИС, первоочередным дети всегда называют путешествия: называются туристические поездки — семейные с целью отдыха или знакомства с новыми интересными регионами и объектами в мире, стране, регионе. Важно расширить представление ребенка о значении понятия «путешествие». По определению, *путешествие* является движением к чему-то новому, неизвестному, при этом задачами путешествия являются изучение нового, обогащение имеющегося опыта, приобщение к культуре (ср.: Путешествие — 1. Поездка или передвижение пешком куда-либо далеко за пределы постоянного местожительства с научной, общеобразовательной, спортивной и другими целями. 2. Жанр художественной или научной литературы, содержащей географические и этнографические сведения, описания путевых впечатлений и размышлений автора; произведение такого жанра [17]).

2.3. Личный опыт учащихся и эмоциональная составляющая

По мере понимания глубины понятия «путешествие» школьники вслед за описанием путешествий называют примеры прокладывания маршрутов по картам — собственных поездок, экспедиций известных путешественников, а также приключений, описанных в литературе. Выбор урока-путешествия как формы организации изучения нового материала или обобщения уже полученных знаний в рамках обучения вызван возможностями данной формы для формирования и развития интереса у школьников [3, 5, 16], при этом в основу планирования занятия заложены межпредметные связи — востребованными оказываются знания по всем предметам школьной программы. Правильно заданные поисковые цели подталкивают обучающихся к расширению своих знаний: они обращаются к справочникам по географии, вооружаются формулами для расчета пути, прокладывают безопасные маршруты, читают историческую и художественную литературу для погружения в детали путешествий.

2.4. Активные и интерактивные методы обучения

Под *активными методами обучения* подразумеваются «методы, которые побуждают учащихся активно мыслить и практиковаться в процессе усвоения учебного материала» [9, с. 16]. На этапе опроса об известных детям маршрутах, героях путешествий, способах перемещения называются разнообразные варианты, что позволяет из предлагаемого списка подготовить набор заданий для практического освоения инструментов ГИС. При определенном разнообразии этих инструментов считаем неоправданным в ограниченные сроки проведения занятий в рамках дополнительного образования приобщать школьников к знанию возможностей многих программ, достаточно сделать первый шаг к их освоению — познакомиться с наборами приложений Google Карты и Яндекс.Карты. Первый инструмент обеспечит воз-

можности планирования путешествий по всему земному шару, а также по другим планетам. Использование отечественного продукта для изучения преимущественно своей страны поможет развитию гражданского, патриотического самосознания.

При изучении **темы «Карты»** (какие карты были раньше, какие есть сейчас, сравнение вариантов карт мира, используемых в разных странах) на примере изменения представлений о географии Земли и их фиксации на картах дети осознают важность развития и саморазвития; при изучении проекции происходит соревнование при составлении «Пазла Меркатора».

В теме «Работа с ресурсом Яндекс.Карты» исследуются пространство около дома, школы, выделение территории, работа с площадью, построение маршрута, работа в конструкторе карт и др. Основной акцент при обучении сделан на стремлении приблизить познавательную деятельность к окружающей действительности.

В содержание **темы «Сравнение Яндекс.Карт и Google Карт»** входит выделение плюсов и минусов ресурсов (критериев для анализа), анализ удобства работы для совершения виртуального кругосветного путешествия. Важным является наличие этапа рефлексии.

3. Система кейсовых заданий

По опыту проведенных занятий были собраны творческие кейсы:

- 1) «Вокруг света за...»;
- 2) «Путешествие по Швеции»;
- 3) «На какую планету я попал?»;
- 4) «Будущее сегодня».

В качестве кейсовых заданий обучающимся предлагаются связанные с путешествиями задания:

- изучение новой информации о выбранном путешествии;
- анализ полученной информации (в основу которого заложен педагогический прием «исследование»);
- фиксация и представление полученных результатов.

3.1. Кейс «Вокруг света за...»

В основу одного из заданий легло путешествие, описанное в литературном произведении. Самый известный из готовых маршрутов кругосветного путешествия, описанный в романе Ж. Верна «Вокруг света за 80 дней», был адаптирован для кейса «Вокруг света за...». Задание кейса — совершить и описать кругосветное путешествие по маршруту англичанина Филеаса Фогга и его слуги-француза Жана Паспарту по маршруту: Лондон (Великобритания) — Суэц (Египет) — Мумбаи (Индия) — Калькутта (Индия) — Гонконг — Иокогама (Япония) — Сан-Франциско (США) — Нью-Йорк (США) — Лондон (Великобритания). Для выполнения задания необходимо воспользоваться ресурсом Google Карты для прокладывания маршрута и просмотра его деталей, достопримечательностей по маршруту. Результаты путешествия нужно отразить в таблице. На основе этих данных обучающиеся делают вывод о том, что самый быстрый способ передвижения по маршруту «Лондон — Суэц» — на самолете (см. табл.).

Результаты путешествия по маршруту Лондон — Суэц

Средство передвижения	Маршрут	Путь (км)	Время в пути (часы/дни)
Пешком	Лондон — Дартфорд — Кентербери — Дувр — Дюнкерк — Брюссель — Дюрен — Бонн — Гисен — Клатови — Вена — Дьор — Секешфехервар — Тимишоара — Лугож — Стамбул — Эскишехир — Конья — Ливан — Дамаск — Суэц	5035	964 / 41
Велосипед	1) Лондон — Кале — Брюссель — Кёльн — Франкфурт-на-Майне — Нюрнберг — Будапешт — Белград — София — Стамбул — Анкара — Адана — Хомс — Дамаск — Суэц 2) Лондон — Кале — Мец — Штутгарт — Мюнхен — Любляна — Будапешт — Белград — София — Стамбул — Анкара — Адана — Хомс — Дамаск — Суэц	5547 5535	370 / 16
Автомобиль	1) Лондон — Кале — Брюссель — Кёльн — Франкфурт-на-Майне — Нюрнберг — Будапешт — Белград — София — Стамбул — Анкара — Адана — Хомс — Дамаск — Суэц 2) Лондон — Кале — Мец — Штутгарт — Мюнхен — Любляна — Будапешт — Белград — София — Стамбул — Анкара — Адана — Хомс — Дамаск — Суэц	5547 5535	60 / 2,5
Поезд	Лондон — Дувр — София — Стамбул — Суэц	5904	57,5 / 2,5
Самолет	Лондон — Дувр — София — Стамбул — Каир — Суэц	4071	7,5 / 0,3
Теплоход	Лондон — Суэц	3670	367 / 16

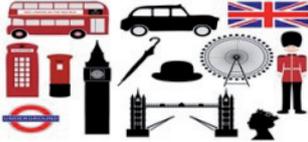


Великобритания – столица Лондон



Полное официальное название страны — *Соединенное Королевство Великобритании и Северной Ирландии.*

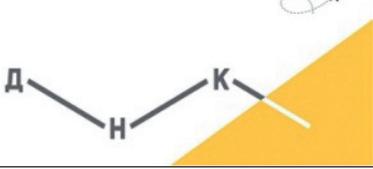
Территория — 242,5 тыс. кв.км.
Население — 57 826 000: англичане (80%), шотландцы, ирландцы, валлийцы.






Соединенное Королевство Великобритании и Северной Ирландии состоит из четырех национальных областей: *Англии, Уэльса, Шотландии и Северной Ирландии.*

Великобритания – это горы, озера, уголки дикой природы и звуки волынки Шотландии; удивительные замками и национального парка Уэльса; архитектурные памятники, театры, музеи, художественные галереи, магазины, бары и рестораны Англии; уникальные природные памятники Северной Ирландии.





ОПОРНЫЙ ВУЗ ВОЛОГОДЧИНЫ

Рис. Пример решения кейса «Вокруг света за...»

Остальная часть маршрута самостоятельно оформляется школьниками в аналогичных таблицах. Далее обучающимся предлагается ответить на вопрос, сколько дней потрачено на кругосветное путешествие; определить, какие трудности встретились при совершении виртуального путешествия.

Далее подбирается краткая информация о точках маршрута (Лондон, Суэц, Мумбаи, Калькутта, Гонконг,

Иокогама, Сан-Франциско, Нью-Йорк) и решение кейса оформляется в формате презентации MS PowerPoint (см. рис.).

3.2. Кейс «Путешествие по Швеции»

В этом кейсовом задании обучающимся предложено совершить путешествие по местам, которые посетил Нильс Хольгерссон, герой произведения С. Лагерлёф

«Удивительное путешествие Нильса Хольгерссона с дикими гусями по Швеции»: используя ресурс Google Карты, необходимо найти по указанным координатам места путешествия и отметить их на карте (используется растровый графический редактор MS Paint). Результаты работы обучающиеся сохраняют в формате PNG/JPEG и представляют другим участникам в докладе.

3.3. Кейс «На какую планету я попал?»

Данное кейсовое задание ориентирует учеников на проверку достоверности информации, а также направлено на развитие воображения.

Ситуация кейса: Маленький принц, герой повести-сказки Антуана де Сент-Экзюпери, странствуя с перелетными птицами, посетил шесть астероидов с номерами 325–330.

Обучающимся необходимо проверить, есть ли астероиды с такими номерами; как они называются сегодня и что известно о них.

После этого ученики представляют себя на месте путешественника по Солнечной системе и совершают виртуальное путешествие с помощью ресурса NASA Solar System Exploration. Результаты путешествия фиксируются в виде интеллект-карты, в которой центральным понятием является «Солнечная система».

3.4. Кейс «Будущее сегодня»

Еще одной интересной для развития учащихся является такая тематика, как путешествие во времени. Человечество существовало и будет жить в том же пространственном измерении, что и сегодня, но границы, контуры, координаты объектов на поверхности Земли были и будут другие, отличные от сегодняшних. Предлагая школьникам перенестись во времени, мы сталкиваемся с необходимостью представлять себе и новое пространство, соответствующее другому времени.

Для привязки событий другого времени к территориям, на которых они происходили или произойдут, возможностей популярных электронных карт может оказаться недостаточно. Поисковые порывы подтолкнут детей к поиску решений: через исследование созданных человечеством бумажных карт они придут к обнаружению современных компьютерных их реализаций с возможностью перемещаться во времени смещением экранной кнопки на полосе прокрутки или необходимости самому спроектировать такую карту — для времени, которое еще никем не описано. Отметим, что для развития ребенка «принципиальную роль играют формирование учениками своего образа будущего, конкретные усилия по воплощению своей мечты» [12, с. 3].

К таким исследованиям относится творческое задание «Будущее сегодня», в котором рассматривается воображаемое путешествие в будущее. Отталкиваясь от истории, изложенной в романе Герберта Уэллса «Машина времени», детям предлагается представить, что они могут отправиться в ближайшее или далекое будущее. Ключевыми становятся ответы на вопросы:

- На что будет похожа планета Земля?
- Что нового и интересного появится в будущем?
- Как изменится наша жизнь?
- Сможем ли мы путешествовать на другие планеты?

- Как будут выглядеть наши дома, наш город?
- Какой будет транспорт?

Результаты работы и ответы на вопрос, как будет выглядеть будущее, обучающиеся отражают в виде рисунков, макетов карт.

4. Выводы

Комплекс рассмотренных кейсовых заданий направлен на формирование личностных (готовность к самостоятельной познавательной деятельности; владение способами взаимодействия с окружающими людьми, навыками работы в группе), метапредметных (способность самостоятельно работать с информацией из различных источников: искать, анализировать и отбирать необходимую информацию; организовывать, преобразовывать, сохранять и передавать информацию; осведомленность о национальной, общечеловеческой культуре) и предметных (представление и визуализация пространственных данных для непрофессиональных пользователей; понимание устройства современных картографических сервисов) результатов обучения.

Рассмотренные занятия позволяют формировать интерес школьников к изучению географии и истории планеты Земля с использованием современных информационных технологий. В центре занятий находится мотив путешествия, так как с древности путешествие является одним из основных источников получения знаний о Земле.

Список источников

1. Белова И. А. Дополнительное образование школьников в образовательной области «Технология» // Научный поиск. 2014. № 2. С. 7–8.
2. Бугаевский Л. М., Цветков В. Я. Геоинформационные системы. М.: Златоуст, 2000. 222 с.
3. Гаврилова Л. Г. Присоединение Сибири. Урок-путешествие в 7 классе // Образование в современной школе. 2008. № 2. С. 26–30.
4. Гоголева Н. А., Карпачева Н. В. «Вытягивающая модель» в образовании // Сахалинское образование — XXI век. 2019. № 3. С. 31–34.
5. Калашиник В. Н., Хозивалиева М. П. Сказка о сказке. Урок-путешествие. 5 класс // Литература в школе. 2012. № 7. С. 33–35.
6. Китайгородский М. Д. Soft skills как преемники универсальных учебных действий в цифровизации технологического образования // Информатика в школе. 2021. № 2. С. 28–31.
7. Китайгородский М. Д., Новикова Н. Н., Муртазин И. А., Истомин Ю. Н. Сетевое взаимодействие при реализации проекта «Урок технологии» в ключевом центре «Дом научной коллаборации им. В. А. Витязевой» // Информатизация образования и методика электронного обучения: цифровые технологии в образовании. Материалы V Международной научной конференции. В 2-х ч. / под общ. ред. М. В. Носкова Красноярск, 2021. С. 525–529.
8. Локшина А. Урок-путешествие с использованием Scapex Web GEOMIXER // Земля из космоса: наиболее эффективные решения. 2015. № 3. С. 34–37.
9. Лунева О. А., Симонова И. В. Активные методы обучения // Педагогический опыт: от теории к практике. Материалы Всероссийской научно-практической конференции (Чебоксары, 24 июля 2020 г.) / редкол.: О. Н. Широков [и др.]. Чебоксары: ЦНС «Интерактив плюс», 2020. С. 16–17.
10. Макарова Л. Н. Роль педагогов учреждений дополнительного образования в профессиональном самоопределении

учащихся в условиях школы // Психолого-педагогический журнал «Гаудеамус». 2018. № 17. С. 45–51.

11. *Муртазин И. А., Истомин Ю. Н.* Развитие ключевого центра «Дом научной коллаборации» в условиях цифровизации образования // Информатизация образования и методика электронного обучения: цифровые технологии в образовании. Материалы IV Международной научной конференции (Красноярск, 6–9 октября 2020 г.): в 2 ч. Ч. 2 / под общ. ред. М. В. Носкова. Красноярск: Сиб. федер. ун-т, 2020. С. 494–498.

12. *Пастухова Л. С., Малинецкий Г. Г.* Педагогика мечты // Вестник Московского университета. Серия 20. Педагогическое образование. 2019. № 3. С. 3–17.

13. *Перевалова М. Н.* Дополнительное образование в области ИТ как средство профессиональной ориентации школьников // Вестник ТОГИРРОС. 2018. № 2. С. 52–53.

14. *Потупчик Е. Г.* Организация и проведение интегрированного урока «Геоинформационные системы в нашей жизни» в условиях дистанционного обучения // Информатика в школе. 2021. № 2. С. 41–54.

15. *Сергеева М. А., Воронина М. В., Раскладкина М. К.* Доступный космос: геоинформационные технологии для школьников // Мир новой экономики. 2020. № 1. С. 6–14.

16. *Смирнова А. Ф.* Интегрированный урок «Путешествие с наречием в историю моды» // Русский язык в школе. 2012. № 2. С. 30–33.

17. Справочно-информационный портал ГРАМОТА.РУ. <http://gramota.ru/>

18. *Тураева З. О.* Lean training — вытягивающая модель обучения // Педагогический опыт: от теории к практике. Материалы Всероссийской научно-практической конференции. (Чебоксары, 24 июля 2020 г.) / редкол.: О. Н. Широков [и др.]. Чебоксары: ЦНС «Интерактив плюс», 2020. С. 18–19.

19. Указ Президента РФ от 7 мая 2018 г. № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года» (с изменениями и дополнениями). <https://base.garant.ru/71937200/>

20. *Шатунова О. В., Иванов С. В.* STEAM-образование: формирование универсальных навыков и новой грамотности // Технологическое образование в системе «Школа — Колледж — Вуз»: традиции и инновации. Материалы Всероссийской научно-практической конференции (Воронеж, 25 марта 2021 г.) / ред. Ю. Б. Ащеулов. Воронеж: Воронежский государственный педагогический университет, 2021. С. 312–315.

21. *Щербинина О. С.* Дополнительное образование одаренных школьников в условиях вуза // Социальное воспитание как целевая функция дополнительного образования детей. Сборник научных статей международной научно-практической конференции (Курск, 24–25 января 2019 г.) / под общ. ред. Т. А. Антопольской. Курск: Университетская книга, 2019. С. 418–422.

DOI: 10.32517/2221-1993-2022-21-1-46-54

Е. Ю. Букина

средняя общеобразовательная школа № 1, г. Зеленокумск, Ставропольский край, Россия

ИНТЕГРИРОВАННЫЙ УРОК ПО ЛИТЕРАТУРЕ И ИНФОРМАТИКЕ «МУЖИЧОК С НОГОТОК» (VI КЛАСС)*

Аннотация

В статье рассматривается проведение интегрированного урока литературы и информатики. Данный урок посвящен 200-летию со дня рождения Н. А. Некрасова, которое отмечалось 10 декабря 2021 года. На уроке ученики знакомятся с произведением великого русского поэта и писателя Н. А. Некрасова «Крестьянские дети», узнают о том, как жили дети в прошлом. В среде программирования Scratch учащиеся создают анимацию (мультфильм) по отрывку из поэмы «Крестьянские дети» «Мужичок с ноготок». Урок направлен на формирование у учащихся представления о жизни и творчестве Н. А. Некрасова, развитие интереса к проектной деятельности и программированию, стимулирование использования школьниками среды программирования Scratch для создания мультимедийных проектов. На уроке созданы условия для овладения основными универсальными учебными действиями информационного характера, такими как постановка и формулирование проблемы; поиск и выделение необходимой информации, применение методов информационного поиска; структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий; самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.

Ключевые слова: проектная технология, стихотворение, Некрасов, программирование, Scratch, Скретч.

Актуальность использования проектной методики в урочной деятельности

Современная школьная информатика — это дисциплина, направленная на формирование широкого спектра метапредметных образовательных результатов, отвечающая требованиям времени и непрерывно изменяющаяся в соответствии с этими требованиями. Сегодня основные изменения в содержании школьного курса информатики связаны с пересмотром содержания общего образования в целом, с развитием самой информатики как области знания, с широким использованием средств информационных и комму-

никационных технологий в образовательном процессе [1]. Информатика имеет большое и постоянно возрастающее число междисциплинарных связей, причем как на уровне понятийного аппарата, так и на уровне инструментария. Многие предметные знания и способы деятельности (включая использование средств ИКТ), освоенные обучающимися на базе информатики, находят применение как в рамках образовательного процесса при изучении других предметных областей, так и в иных жизненных ситуациях, становятся значимыми для формирования качеств личности, т. е. ориентированы на формирование метапредметных и личностных результатов [3].

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2022/

Контактная информация

Букина Елена Юрьевна, учитель информатики, средняя общеобразовательная школа № 1, г. Зеленокумск, Ставропольский край, Россия; *адрес:* 357910, Россия, Ставропольский край, г. Зеленокумск, ул. Урицкого, д. 26; *e-mail:* lenok20102010@mail.ru

E. Yu. Bukina

School 1, Zelenokumsk, Stavropol Krai, Russia

THE INTEGRATED LESSON IN LITERATURE AND INFORMATICS “POCKET-SIZED MAN” (VI CLASS)

Abstract

The article describes the conduct of the integrated lesson in literature and informatics. This lesson is dedicated to the 200th anniversary of the birth of N. A. Nekrasov, which was celebrated on December 10, 2021. At the lesson, students get acquainted with the work of the great Russian poet and writer N. A. Nekrasov “Peasant children”, learn about how children lived in the past. In the Scratch programming environment, students create an animation (cartoon) based on an excerpt from the poem “Peasant children” “Pocket-sized man”. The lesson is aimed at forming students’ ideas about the life and work of N. A. Nekrasov, developing interest in project activities and programming, encouraging students to use the Scratch programming environment to create multimedia projects. The lesson provides conditions for mastering the basic universal educational activities of an informational nature, such as posing and formulating a problem; search and selection of necessary information, application of information retrieval methods; structuring and visualization of information; selection of the most effective ways of solving problems depending on specific conditions; independent creation of activity algorithms in solving problems of a creative and exploratory nature.

Keywords: project technology, lesson, poem, Nekrasov, programming, Scratch.

Современные требования к образовательному процессу ставят новые задачи перед учителем: не пассивное прочтение учениками художественного текста, а формирование у учащихся умения самостоятельно работать с текстом, развитие творческого мышления, способности прогнозировать свою деятельность. Одна из важнейших задач учителя — включение учащихся в творческую деятельность, увеличение доли их самостоятельности в учебном процессе [9].

Практика показывает, что использование проектной методики в образовательном процессе обеспечивает формирование ключевых компетенций: исследовательской, коммуникативной, информационной. Проектная деятельность на уроке направлена на формирование у школьников общеучебных умений и навыков, универсальных способов деятельности. Проектная деятельность учащихся — это учебно-познавательная, творческая или игровая деятельность, результатом которой становится решение какой-либо проблемы, представленное в виде подробного описания (проекта) [8].

Способность создавать компьютерные программы является в современном обществе важной частью общей грамотности. Среда визуального программирования Scratch — это бесплатное приложение, которое позволяет легко создавать собственные графические анимации (мультфильмы). Элементы управления Scratch просты в использовании, и детям очень нравится создавать свои собственные проекты в этой среде. Анимации создаются на основе различных визуальных элементов (фонов, изображений и графики, сгенерированных в программе), которые необходимо запрограммировать. Средой программирования Scratch можно пользоваться без загрузки программы, работая на сайте Scratch: <https://scratch.mit.edu/> [12].

Среда программирования Scratch помогает учиться мыслить творчески, рассуждать системно и работать совместно — это необходимые навыки для жизни в XXI веке.

В среде Scratch обучающиеся в полной мере могут раскрыть свои творческие таланты, создавая мультфильмы, игры, анимированные открытки, презентации, обучающие программы, тренажеры, интерактивные тесты. Они могут придумывать различные объекты, определять, как эти объекты будут выглядеть в разных условиях, перемещать объекты по экрану, устанавливать способы взаимодействия между объектами; сочинять истории, рисовать и оживлять на экране своих придуманных персонажей, осваивая при этом технологии обработки графической и звуковой информации, анимационные технологии, мультимедийные технологии. Scratch позволяет работать над проектами совместно [6].

Педагогический потенциал среды программирования Scratch позволяет рассматривать ее как перспективный инструмент организации междисциплинарной проектной учебно-познавательной деятельности обучающегося, направленной на личностное и творческое развитие ребенка и позволяющей ему воссоздать единую картину мира.

В настоящей статье представлена методическая разработка интегрированного урока по литературе и информатике, посвященного 200-летию со дня ро-

ждения Н. А. Некрасова, которое отмечалось 10 декабря 2021 года. На данном уроке учащиеся знакомятся с фрагментом поэмы Н. А. Некрасова «Крестьянские дети» — стихотворением «Мужичок с ноготок» («Однажды, в студеную зимнюю пору») и в среде программирования Scratch создают анимацию (мультфильм) по этому стихотворению. Урок направлен на формирование у учащихся представления о жизни и творчестве Н. А. Некрасова, развитие интереса к проектной деятельности и программированию, стимулирование использования школьниками среды программирования Scratch для создания мультимедийных проектов.

Сценарий интегрированного урока по литературе и информатике

Цели урока:

- знакомство учащихся со стихотворением Н. А. Некрасова «Мужичок с ноготок», расширение их знаний о жизни детей в прошлом;
- развитие умения учащихся анализировать поэтическое произведение;
- развитие устной речи учеников, обогащение их словарного запаса;
- создание учащимися анимации (мультфильма) в среде программирования Scratch;
- развитие умения учеников работать в парах (группах), создание условий для воспитания культуры общения.

Задачи урока:

образовательные:

- совершенствование навыков анализа текста, применения приемов работы с текстом и словарем;
- совершенствование навыков анализа произведения;
- развитие умения понимать авторскую позицию и позицию героя;
- изучение основ блочного программирования;

развивающие:

- развитие навыка выразительного чтения текста и чтения наизусть;
- развитие способности эстетического восприятия и оценки художественной литературы и воплощенных в ней явлений жизни;
- совершенствование речевых навыков и умений;
- развитие ассоциативного и логического мышления;
- формирование познавательного интереса к литературе и компьютерным технологиям;
- развитие алгоритмического мышления и применения полученных знаний для решения практических задач;
- развитие умения сопоставлять полученную информацию и делать выводы;
- развитие умения представлять компьютерные проекты перед аудиторией (самопрезентация);
- развитие навыков работы в группах;
- развитие коммуникативных навыков, внимания, памяти;

воспитательные:

- воспитание нравственных качеств, умения оценивать свои поступки, окружающую действительность, умения сочувствовать, сопереживать, отстаивать нравственную позицию, воспитание основ нравственного поведения;
- повышение ответственности не только за собственные знания, но и за успехи всего коллектива;
- воспитание умения внимательно слушать и слышать, уважать другое мнение;
- привитие навыков самостоятельности в работе;
- повышение интереса к изучению литературы, информатики, информационных технологий;
- содействие обретению учащимися нравственных ценностей, осознанию ответственности за свои поступки, воспитание активной жизненной позиции;
- повышение уровня толерантности к ошибкам в обучении и жизни;
- повышение уровня любознательности и самостоятельности в решении задач;
- воспитание чувства такта при работе в группе, дисциплинированности.

Планируемые образовательные результаты:**предметные:**

- умение читать вдумчиво, понимать значение незнакомых слов из контекста произведения, быть внимательным к авторскому слову, способным творить свой читательский мир, самостоятельно оценивать художественное произведение;
- применение теоретических знаний для решения практических заданий;
- применение терминов и понятий в контексте заданий;

метапредметные:

- **регулятивные:**
 - умение формулировать учебную задачу, оценивать свои речевые высказывания и высказывания сверстников;
 - овладение навыками анализа, сравнения;
- **познавательные:**
 - умение находить ответы на вопросы в тексте, иллюстрациях;
 - умение устанавливать аналогии, делать обобщения, выводы;
- **коммуникативные:**
 - умение участвовать в диалоге с одноклассниками, бесконфликтно дискутировать, передавать чувства героев и свое отношение к ним;
 - развитие коммуникации с учителем, сверстниками;

личностные:

- умение усвоить главный урок, извлеченный из произведения, высказывать стремление руководствоваться выводами, сделанными на уроке, в собственных жизненных ситуациях;
- умение расширять кругозор, проявлять желание читать и размышлять;
- формирование мировоззренческих установок;

- умение формулировать собственную точку зрения;
- развитие осознанного отношения к учению.

Возрастная категория: VI класс.**Вид урока:**

по литературе: урок внеклассного чтения;

по информатике: урок закрепления полученных знаний.

Тип урока: интегрированный урок.**Технологии, использованные в ходе урока:**

- информационно-коммуникационные технологии с использованием визуальной среды программирования Scratch;
- проектная технология;
- здоровьесберегающий элемент (атмосфера на уроке, благоприятный эмоциональный фон, смена видов деятельности, физкультминутка, гимнастика для глаз).

Методы обучения:

- **словесные:** работа с текстом, выразительное чтение, чтение наизусть, художественные приемы;
- **наглядные:** иллюстрации, презентация;
- **практические:** групповая работа;
- **проблемно-поисковые:** ответы на вопросы;
- **индуктивные (синтез):** умение обобщить, связать материал с современностью;
- **дедуктивные (анализ):** анализ текста;
- **самостоятельная работа:** работа в группах, элемент мини-проекта.

Организационные формы работы: фронтальная, групповая, индивидуальная.

Учебное оборудование (оснащение) занятия:

- проектор;
- компьютеры с установленной программной средой визуального программирования Scratch или с выходом в интернет, в частности на сайт Scratch: <https://scratch.mit.edu/>

Программное обеспечение:

- операционная система Microsoft Windows;
- браузер Google Chrome;
- среда программирования Scratch (ссылка для скачивания: <https://scratch.ru.uptodown.com/windows/>);
- презентация к уроку, подготовленная в Microsoft PowerPoint*.

Литература:

- *Курдюмова Т. Ф.* Литература. 6 класс: учебник-хрестоматия. В 2 ч. М.: Дрофа, Российский учебник, 2019. Ч. 1. 272 с.; Ч. 2. 240 с.
- *Босова Л. Л., Босова А. Ю.* Информатика: учебник для 6 класса. М.: БИНОМ. Лаборатория знаний, 2021. 224 с.

Предварительная подготовка: создать аккаунты на сайте: <https://scratch.mit.edu/> или установить программную среду Scratch на компьютеры.

* Презентацию можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2022/

План урока.

№ этапа	Содержание этапа урока	Время, мин	Виды и формы работы
1	Организационный момент	2	Приветствие. Мотивация и стимулирование деятельности учащихся
2	Актуализация знаний	4	Беседа, создание проблемной ситуации
3	Изучение нового материала	12	Фронтальная работа
4	Закрепление изученного материала	12	Работа в группах по карточкам
5	Физкультминутка	4	Разминка, упражнения для глаз
6	Выполнение проекта «Мой любимый литературный герой» к стихотворению Н. А. Некрасова «Мужичок с ноготок» в среде программирования Scratch	37	Работа в группах в среде Scratch
7	Представление проекта	15	Презентация проекта
8	Подведение итогов урока	3	Рефлексия настроения и эмоционального состояния
9	Домашнее задание	1	

Ход урока**1. Организационный момент**

Учитель.

Сегодня особенный, творческий час.

В мир классики я приглашаю всех вас.

Кто автор? Что пишет? Узнаем мы на уроке...

Кому посвящаются автора строки?

Читать и исследовать — наша задача.

Надеюсь, что ждет вас, конечно, удача! [11]

2. Актуализация знаний

Учитель. Мир русской поэзии богат и разнообразен. Поэты XVIII—XIX веков в своих произведениях описывали красоту русской природы, события и жизнь людей, выражали свое отношение к происходящему.

Задание.

Расшифруйте слово:

15 – 6 – 12 – 18 – 1 – 19 – 16 – 3

и вы узнаете, с произведением какого поэта мы сегодня познакомимся. Слово запишите заглавными буквами. Каждой букве соответствует ее порядковый номер в алфавите.

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	

Демонстрируется слайд презентации с портретом Н. А. Некрасова (рис. 1).

Учитель. На предыдущем уроке вы познакомились с биографией поэта. Давайте вспомним сведения о нем.

Вопросы и задания:

- Назовите полное имя и отчество Некрасова. (*Николай Алексеевич Некрасов.*)

- В каком году родился великий русский поэт? (*В 1821 году. В 2021 году отмечалось 200-летие со дня его рождения.*)
- Каким по счету ребенком он был в семье? (*Третьим.*)
- Кто занимался воспитанием Некрасова? (*Мама.*)
- Кем был отец Некрасова? (*Армейским офицером.*)
- Во сколько лет Некрасов написал свои первые стихи? (*В семь лет.*)
- Кому было посвящено первое стихотворение Некрасова? (*Маме.*)
- О чем мечтала Елена Андреевна? (*Чтобы сын стал образованным человеком.*)
- На какой реке прошло детство Некрасова? (*На Волге.*)
- Где проводил время Некрасов, когда отца не было дома? (*Он играл с крестьянскими детьми.*)
- Сколько лет прожил Некрасов? (*56 лет.*)
- О чем писал в своих произведениях Некрасов? (*О природе, о людях, о детях.*)

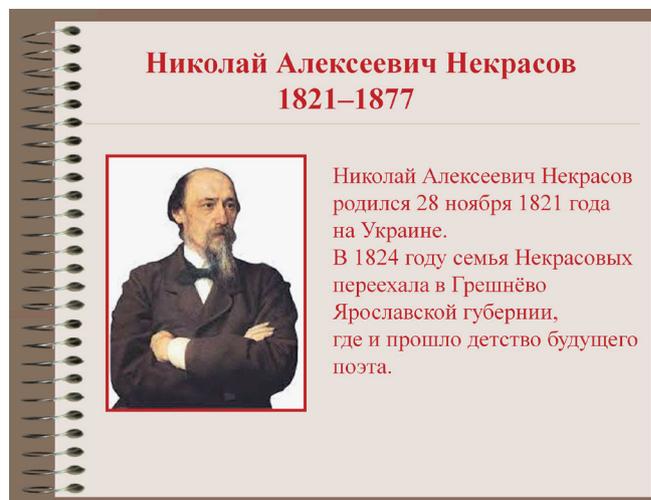


Рис. 1

3. Изучение нового материала

Учитель. Ребята, на предыдущем уроке информатики вы познакомились с видами информационных моделей. По способу представления различают следующие виды информационных моделей: знаковые, образные и смешанные. Все эти виды информационных моделей используются в образовании (иллюстрации в учебниках, наглядные пособия и учебные плакаты) [4].

Давайте рассмотрим более подробно знаковые информационные модели. Разновидностью знаковых информационных моделей являются словесные описания. Множество словесных описаний содержится в ваших школьных учебниках: в учебнике истории представлены модели исторических событий, в учебнике географии — модели географических объектов и природных процессов, в учебнике биологии — модели объектов животного и растительного мира, в учебнике литературы — произведения художественной литературы.

В словесных описаниях ситуации, события, процессы приводятся на естественном языке (русском, английском, немецком и др.). Словесные описания могут быть выполнены в разных стилях. Различают разговорный и книжный стили. Книжный стиль имеет такие разновидности, как научный, официально-деловой, публицистический, художественный.

Произведения художественной литературы — это словесные модели. Они фиксируют внимание читателя на определенных сторонах человеческой жизни. Анализируя литературное произведение, вы выделяете в нем объекты и их свойства, отношения между героями, связи между событиями, проводите параллели с другими произведениями.

Пожалуй, ни у кого из русских поэтов не найдешь столько стихов, посвященных детям, сколько у Некрасова. У него есть поэма, посвященная крестьянским детям. Она так и называется — «Крестьянские дети».

Сегодня мы познакомимся с отрывком из поэмы «Крестьянские дети» — стихотворением «Мужичок с ноготок».

Прослушайте стихотворение. (*Звучит стихотворение Н. А. Некрасова «Мужичок с ноготок» («Однажды, в студеную зимнюю пору») [10].*)

...Однажды, в студеную зимнюю пору,
Я из лесу вышел; был сильный мороз.
Гляжу, поднимается медленно в гору
Лошадка, везущая хворосту воз.
И, шествуя важно, в спокойствии чинном,
Лошадку ведет под уздцы мужичок
В больших сапогах, в полушубке овчинном,
В больших рукавицах... а сам с ноготок!
«Здорово, парнище!» — «Ступай себе мимо!»
— «Уж больно ты грозен, как я погляжу!»
Откуда дровишки?» — «Из лесу, вестимо;
Отец, слышишь, рубит, а я отвожу».
(В лесу раздавался топор дровосека.)
«А что, у отца-то большая семья?»
— «Семья-то большая, да два человека
Всего мужиков-то: отец мой да я...»
— «Так вон оно что! А как звать тебя?»
— «Власом».
— «А кой тебе годик?» — «Шестой миновал...
Ну, мертвая!» — крикнул малюточка басом.
Рванул под уздцы и быстрее зашагал...

Понравилось вам стихотворение? Чем? (*Ответы учащихся.*)

От чьего лица написано это стихотворение? (*От лица автора.*)

Прочитайте самостоятельно вполголоса стихотворение.

Давайте дадим объяснение значения слов из стихотворения «Мужичок с ноготок».

Словарная работа [2].

- Что значит «студеная пора»? (*Очень холодно.*)
- «Хворост»? (*Сухие сучья, ветви.*)
- «В спокойствии чинном»? (*Спокойно, важно.*)
- «Под уздцы»? (*Уздцы — это часть конской сбруи, надеваемые на голову лошади ремни с удилами и поводьями.*)
- «В полушубке овчинном»? (*Полушубок — короткая, до колен, верхняя одежда на меху.*)
- «С ноготок»? (*Очень маленький.*)
- «Парнище»? (*Мальчик, парень.*)
- «Больно он грозен»? (*Очень грозен.*)
- «Шестой миновал»? (*Исполнилось шесть лет.*)
- «Шествуя важно» (*Идя важно, не торопясь.*)
- «Вестимо»? (*Конечно, разумеется.*)

4. Закрепление изученного материала

Работа по карточкам.

Учитель предлагает выбрать направление (роль), к которому ученик проявляет интерес: сценарист, художник, программист, музыкант. Ребята делятся на группы по три-четыре человека. Каждая группа получает от учителя задание на карточках. Ребята находят ответы на вопросы в тексте стихотворения в учебнике литературы [7] и устно отвечают.

Команда 1.

1. Как зовут мальчика? («Власом».)
2. Сколько ему лет? («Шестой миновал».)
3. Как выглядит мальчик? Найдите в тексте его описание. («В больших сапогах, в полушубке овчинном, В больших рукавицах... а сам с ноготок!»)

Команда 2.

1. Подумайте, почему вся одежда велика Власу? (*В бедной крестьянской семье одежда передавалась от одного ребенка к другому.*)
2. Где поэт встретил мальчика? (*В лесу.*)
3. В какое время года произошла встреча? (*Зимой.*)

Команда 3.

1. Найдите в тексте и прочитайте описание времени, когда происходят события. («Однажды, в студеную зимнюю пору, Я из лесу вышел, был сильный мороз.»)
2. Почему мальчик оказался в лесу в такой мороз? (*Мальчик пришел с отцом в лес за дровами. Он хотя и маленький, но уже помогает взрослым.*)
3. Найдите в тексте и прочитайте слова Власа о его семье. («Семья-то большая, да два человека Всего мужиков-то: отец мой да я.»)

Команда 4.

1. Что значит, что мальчик шествует «в спокойствии чинном»? (*Не суетится, не резвится,*

не шалит. Он во всем подражает взрослым. Делает все неторопливо, уверенно, без лишних движений.)

2. Найдите слова, с которыми Влас обращается к лошади. («*Ну, мертвая!*» — крикнул мальюточка басом».)
3. Почему он так говорит? (*Лошадь остановилась. Мальчик кричит так, как кричат взрослые на лошадей, которые замедлили шаг.*)

Команда 5.

1. Как разговаривает Влас? Найдите подтверждение своему ответу в тексте. (*Коротко и грубо. «Здорово, парнище!» — «Ступай себе мимо!»*)
2. Что подчеркивает Некрасов словом «парнище»? (*Некрасов подчеркивает, что мальчик мал и ростом, и по годам. Поэтому мальчик отвечает сердито на это приветствие поэта.*)
3. Как вы думаете, как автор относится к мальчику? Почему вы так решили? (*С уважением, с любовью. Он называет мальчика ласково: «мужичок», «малюточка».*)

Команда 6.

1. «Семья-то большая, да два человека Всего мужиков-то: отец мой да я». Подумайте, почему мальчик так говорит? (*Он понимает важность своей работы в семье.*)
2. Как идет мальчик? Найдите в тексте и прочитайте нужные строки. (*Он идет гордо, важно: «шествуя важно, в спокойствии чинном». Ему некогда разговаривать с поэтом, ему нужно работать.*)
3. Некрасов говорит, что мальчик «шествует важно». Не идет, не бредет, не шагает, а «шествует». Почему поэт подобрал именно это слово? (*Мальчик уважает себя за то, что ему доверили такое важное дело. Он гордится этим.*)

Задание командам.

Составить синквейн, описать лирического героя по следующей памятке:

1. Имя героя.
2. Два слова — признака героя («Какой?»).
3. Три слова — действия героя («Что делает?»).

4. Одно предложение о герое.
5. Кто он.

Учитель. Если вы еще не знаете, что такое синквейн, то я вам сейчас объясню.

Синквейн (от фр. *cinquains*, англ. *cinquain*) — это творческая работа, которая имеет короткую форму стихотворения, состоящего из пяти нерифмованных строк. Синквейн — это не простое стихотворение, а стихотворение, написанное по следующим правилам:

- 1-я строка — одно существительное, выражающее главную тему синквейна.
- 2-я строка — два прилагательных, выражающих главную мысль.
- 3-я строка — три глагола, описывающих действия в рамках темы.
- 4-я строка — фраза, несущая определенный смысл.
- 5-я строка — заключение в форме существительного (ассоциация с первым словом).

Пример работы по памятке:

1. Влас.
2. Маленький, важный.
3. Ведет, шествует, отвозит.
4. Влас чинно ведет под уздцы лошадь.
5. Помощник.

5. Физкультминутка

Разминка «Зимние забавы».

Мы зимой в снежки играем, мы играем.
(Имитируют лепку снежков.)
По сугробам мы шагаем, мы шагаем.
(Шагают, высоко поднимая колени.)
И на лыжах мы бежим, мы бежим.
(Прыгают на месте.)
На коньках по льду скользим, мы скользим.
(Махи руками — одна вперед, вторая назад, и на оборот.)
И снегурку лепим мы, лепим мы.
(Имитируют лепку.)
Гостью-зиму любим мы, любим мы.
(Разводят руки в поклоне и ставят на пояс.)

Упражнения для глаз (рис. 2).



Рис. 2 [5]

6. Выполнение проекта «Мой любимый литературный герой» к стихотворению Н. А. Некрасова «Мужичок с ноготок» в среде программирования Scratch

Учитель. Ребята, вы разделились на команды, когда выполняли задания по карточкам. В каждой команде у вас есть сценарист, программист, музыкант и художник. Сейчас вы будете работать над проектом, назовем его «Мой любимый литературный герой». Вам нужно создать анимацию (мультфильм) к стихотворению Н. А. Некрасова «Мужичок с ноготок».

Этапы проекта.

1. Изучение стихотворения, выбор концепции анимации.
2. Выбор спрайтов и фонов из библиотек среды программирования Scratch (также можно нарисовать самостоятельно).
3. Редактирование спрайтов и фонов в графическом редакторе.
4. Озвучание персонажей.
5. Создание программы для анимации персонажей.
6. Тестирование мультфильма и его доработка.
7. Публикация проекта.
8. Презентация проекта.

Scratch позволяет работать над проектами совместно. Для выполнения данного проекта собраны команды. Каждому члену команды поручено то, что получается у него лучше всего.

Распределение ролей в команде.

- сценаристы — разрабатывают сценарий проекта;
- художники — подбирают (или рисуют) фоны, спрайты;
- музыканты — записывают звуковое сопровождение;
- программисты — создают скрипты для спрайтов, пишут программный код.

7. Представление проекта

Представление готового проекта и его демонстрация происходят в классе с использованием мультимедийного проектора. Представители от каждой команды демонстрируют проекты классу. Учитель и ученики обсуждают и оценивают проекты. Учителю помимо собственно проекта и его представления следует также оценить умения учащихся работать в группе:

- наличие определенных обязанностей у каждого участника команды и их функциональное разделение;
- взаимодействие членов команды, направленное на результат работы;
- вклад каждого члена команды.

К оцениванию проектов можно подключить приглашенных участников мероприятия: классного руководителя, представителя администрации школы, членов школьного научного общества, родителей. Тогда оценивание будет более объективным.

Пример реализованного проекта «Мой любимый литературный герой» к стихотворению Н. А. Некрасова «Мужичок с ноготок» в среде программирования Scratch (рис. 3) размещен на сайте Scratch:

<https://scratch.mit.edu/projects/604578399/>
(над проектом работали ученики 6 «Б» класса МОУ СОШ № 1 г. Зеленокумска Луиза Алавердян, Екатерина Алейникова, Геннадий Нахрачёв; руководитель проекта — автор данной статьи). Для запуска проекта нажмите на зеленый флажок. Не забудьте включить звук. Программные коды проекта представлены в *Приложении*.

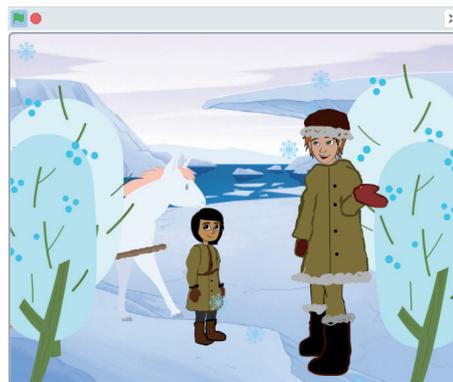


Рис. 3

8. Подведение итогов урока

Учащиеся по очереди высказываются одним предложением, выбирая начало фразы из «экрана рефлексии», представленного на доске:

- Сегодня я узнал...
- Я выполнял задания...
- Я понял, что...
- Теперь я могу...
- Я научился...
- У меня получилось...
- Я смог...
- Меня удивило...
- Урок дал мне для жизни...
- Мне захотелось...
- Было интересно...
- Было трудно...

9. Домашнее задание

Задание. Выучить отрывок из поэмы «Крестьянские дети» наизусть.

Учитель. Как будете учить наизусть? (*С выражением, с интонацией, с соблюдением пауз.*)

Список источников

1. Авторская мастерская Л. Л. Босовой // Издательство «БИНОМ. Лаборатория знаний». <https://lbz.ru/metodist/authors/informatika/3/>
2. Анализ стихотворения Некрасова «Мужичок с ноготок». <https://ostihe.ru/analiz-stihotvoreniya/nekrasova/muzhichok-s-nogotok>
3. Босова Л. Л., Босова А. Ю. Информатика: методическое пособие для 5–6 классов. М.: БИНОМ. Лаборатория знаний, 2019. 354 с.
4. Босова Л. Л., Босова А. Ю. Информатика: учебник для 6 класса. 7-е изд., стер. М.: БИНОМ. Лаборатория знаний, 2021. 224 с.
5. Гимнастика для глаз. http://дюстц.рф/uploads/posts/2021-10/1635146668_img18.jpg
6. Голиков Д. В. 40 проектов на Scratch для юных программистов. СПб.: БХВ-Петербург, 2019. 192 с.

7. Курдюмова Т. Ф. Литература. 6 класс: учебник-хрестоматия. В 2 ч. М: Дрофа, Российский учебник, 2019. Ч. 1. 167 с.

8. Медведева Е. Е. Проектная деятельность учащихся как средство формирования ключевых компетенций. <https://urok.1sept.ru/articles/596218>

9. Скарлыгина М. Развитие функциональной грамотности на уроках литературного чтения. <https://www.maam.ru/detskijasad/razvitiye-funkcionalnoi-gramotnosti-na-urokah-literaturnogo-chtenija.html>

10. Стихотворение Н. А. Некрасова «Однажды, в студеную зимнюю пору» (Стихи Русских Поэтов) // YouTube. https://www.youtube.com/watch?v=RVnRDgOMxL0&ab_channel=СтихиРусскихПоэтов

11. Требесова М. А. Разработка открытого урока по литературному чтению. 3 класс. Н. А. Некрасов «Мужичок с ноготок». <https://multiurok.ru/files/razrabotka-otkrytogo-uroka-po-literaturnomu-chte-2.html>

12. Scratch — Imagine, Program, Share. <https://scratch.mit.edu/>

Приложение

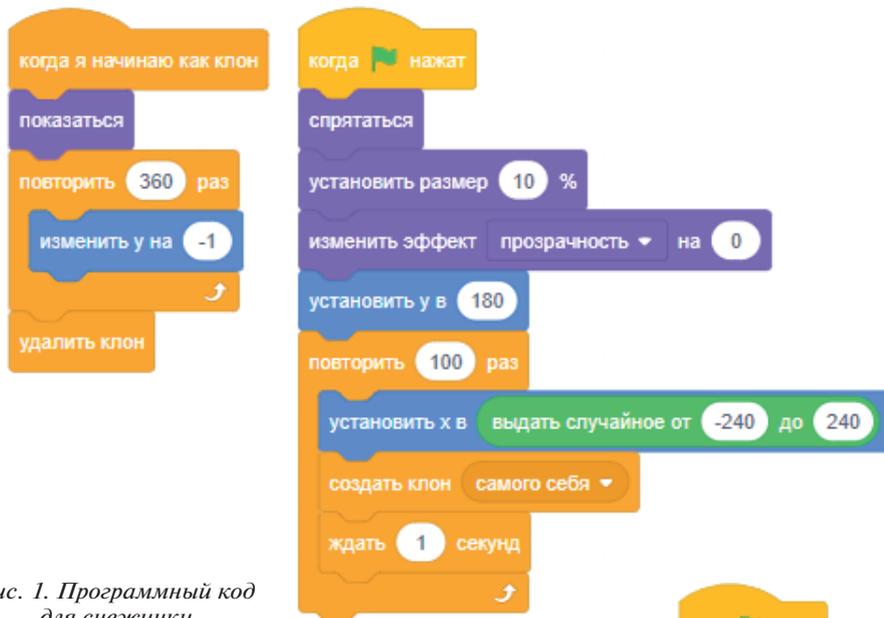


Рис. 1. Программный код для снежинки

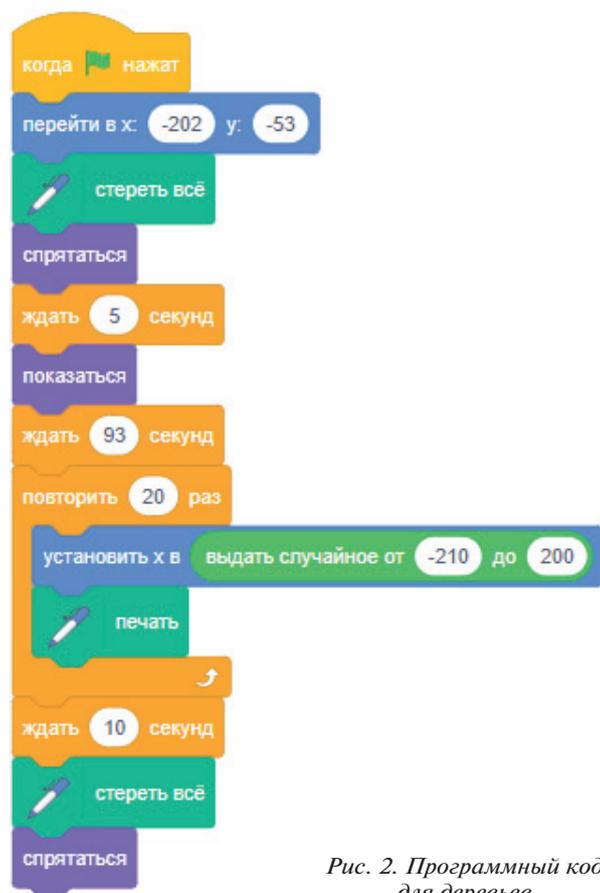


Рис. 2. Программный код для деревьев



Рис. 3. Фрагмент программного кода для мальчика

Scratch code for the author character. The script starts with a 'when green flag is clicked' event, followed by a 'go to front layer' block, a 'go to x: 126 y: 13' block, a 'hide' block, a 'wait 6 seconds' block, a 'show' block, a 'send message: вниз' block, a 'play sound: рор to end' block, a 'wait 2 seconds' block, a 'play sound: автор to end' block, a 'wait 1 second' block, a 'play sound: автор2 to end' block, a 'wait 1 second' block, a 'turn on sound: Я' block, a 'say: Здорово парнище! 2 seconds' block, and a final 'wait 3 seconds' block.

Рис. 4. Фрагмент программного кода для автора

Scratch code for the carriage character. The script starts with a 'when green flag is clicked' event, followed by a 'go to x: -105 y: -140' block, a 'go to back layer' block, a 'hide' block, a 'wait 17 seconds' block, a 'show' block, a 'when I receive: лошадь' block, a 'move 3 seconds to point x: 407 y: -113' block, and a 'hide' block.

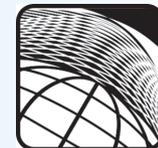
Рис. 5. Программный код для повозки

Scratch code for the horse character. The script starts with a 'when green flag is clicked' event, followed by a 'hide' block, a 'wait 17 seconds' block, a 'set x to: -228' block, a 'go to back layer' block, a 'show' block, a 'send message: анимация лошадь' block, a 'move 2 seconds to point x: -153 y: -29' block, a 'wait 69 seconds' block, a 'hide' block, a 'when I receive: лошадь' block, a 'show' block, a 'move 3 seconds to point x: 261 y: -16' block, a 'when I receive: анимация лошадь' block, a 'repeat 15 times' loop containing 'next costume' and 'wait 0.1 seconds' blocks, an 'change costume to: unicorn running-d' block, another 'when I receive: лошадь' block, another 'repeat 15 times' loop containing 'next costume' and 'wait 0.1 seconds' blocks, and a final 'change costume to: unicorn running-d' block.

Рис. 6. Программный код для лошади

Scratch code for the poem recitation. The script starts with a 'when background changes to: Light' event, followed by a 'wait 15 seconds' block, a 'show' block, a 'wait 13 seconds' block, a 'hide' block, a 'when green flag is clicked' event, a 'hide' block, and a 'go to front layer' block.

Рис. 7. Программный код для стихотворения



DOI: 10.32517/2221-1993-2022-21-1-55-66

М. В. Маркушевич
Школа № 1352, г. Москва, Россия

ЭЛЕМЕНТЫ МЕТОДИКИ ПРЕПОДАВАНИЯ ТЕМЫ «СОЗДАНИЕ И ОБРАБОТКА ЗВУКОВОЙ ИНФОРМАЦИИ» С ИСПОЛЬЗОВАНИЕМ СВОБОДНОГО ЗВУКОВОГО РЕДАКТОРА AUDACITY

Аннотация

В статье представлен каркас методики обучения теме «Создание и обработка звуковой информации» курса информатики на уровне основного общего образования с использованием свободного кроссплатформенного офлайн-звукового редактора Audacity. Под понятием «методика обучения» понимается процесс передачи от преподавателя обучающемуся некоего необходимого и достаточного набора алгоритмов создания тех или иных информационных объектов, соответствующих изучаемой информационной технологии.

По результатам анализа возможных вариантов программного обеспечения для поддержки преподавания рассматриваемой темы курса информатики делается вывод о том, что оптимальным вариантом являются офлайн-свободные кроссплатформенные звуковые редакторы, обладающие невысокими минимальными системными требованиями. Один из наиболее удачных вариантов программного обеспечения — аудиоредактор Audacity, который имеет целый ряд преимуществ по сравнению со своими аналогами.

В статье подвергается сомнению принцип инвариантности методики обучения информационным технологиям, в частности технологии создания и обработки звуковой информации, относительно программного и аппаратного обеспечения, используемого в учебном процессе, и обосновывается вывод о частичной несостоятельности данного принципа.

Приводится возможный вариант каркаса методики обучения рассматриваемой теме курса информатики, включающий в себя разбиение содержания обучения на семь этапов. Каждый из этапов включает описание изложения соответствующего учебного материала, в том числе технологию выполнения практических работ, направленных на овладение учащимися основными элементами функционала звукового редактора Audacity. Описание авторской методики обучения также содержит

Контактная информация

Маркушевич Михаил Владимирович, учитель информатики, школа № 1352, г. Москва, Россия; *адрес:* 107497, Россия, г. Москва, ул. Байкальская, д. 44; *e-mail:* mihael11@yandex.ru

M. V. Markushevich
School 1352, Moscow, Russia

ELEMENTS OF THE METHODOLOGY FOR TEACHING THE THEME "CREATING AND PROCESSING OF SOUND INFORMATION" USING THE FREE SOUND EDITOR AUDACITY

Abstract

The article presents the framework of the methodology for teaching the theme "Creating and processing of sound information" of the informatics course at the level of basic general education using the free cross-platform offline sound editor Audacity. The concept of "teaching methodology" refers to the process of transferring from the teacher to the student a certain necessary and sufficient set of algorithms for creating certain information objects that correspond to the information technology being studied.

As a result of the analysis of possible software options to support the teaching of the theme under consideration in the informatics course, it is concluded that offline free cross-platform sound editors with low minimum system requirements is the best option. One of the most successful software options is the Audacity audio editor, which has a number of advantages over its counterparts.

The article questions the principle of invariance of the method of teaching information technology, in particular technology for creating and processing sound information, with respect to the software and hardware used in the educational process, and substantiates the conclusion about the partial failure of this principle.

A possible variant of the framework of the methodology for teaching the theme of the informatics course under consideration is given, which includes the division of the learning content into seven stages. Each of the stages includes a description of the presentation of the relevant educational material, including the technology for performing practical work aimed at mastering the basic elements of the functionality of the Audacity sound editor by students. The description of the author's teaching methodology also contains the lesson planning corresponding to it on the theme "Creating and processing of sound information" of the informatics course for the basic school.

The advantages of the proposed teaching methodology include the possibility of its use for both full-time and distance learning, as well as the low cost of its use in the educational process of a general education school. In addition, the methodology under consideration ensures a high resistance of the educational process to the destructive impact on the information structure of Russian educational organizations, which includes, among other things, an unmotivated refusal of licenses to use foreign-made proprietary software.

Keywords: digital sound, Audacity, free software, teaching methodology.

соответствующее ей поурочное планирование по теме «Создание и обработка звуковой информации» курса информатики для основной школы.

К преимуществам предлагаемой методики обучения можно отнести возможность ее использования как для очной, так и для дистанционной форм обучения, а также низкий уровень стоимости ее применения в учебном процессе общеобразовательной школы. Кроме того, рассматриваемая методика обеспечивает высокую устойчивость учебного процесса к деструктивному влиянию на информационную структуру российских образовательных организаций, заключающемуся в том числе в немотивированном отказе в лицензиях на использование проприетарного программного обеспечения иностранного производства.

Ключевые слова: цифровой звук, Audacity, свободное программное обеспечение, методика обучения.

1. Введение

Данная статья является продолжением предыдущих работ автора [15, 16] (в соавторстве с А. Ю. Федосовым), представляющих последовательное построение каркаса методики обучения школьников информатике на уровне основного общего образования. Предлагаемая методика базируется на применении в учебно-воспитательном процессе исключительно *свободного отечественного* программного обеспечения. В своих статьях мы, прежде всего, рассматриваем методику обучения различным информационным технологиям. Так, в настоящей статье будет рассмотрена технология создания и обработки цифрового звука.

Среди преподаваемых в основной школе информационных технологий технология обработки звуковой информации является, наверно, одной из наименее разработанных в плане методики обучения. Причина этого кроется в отсутствии ее непосредственного «офисного» приложения и очень редком обращении к данной тематике в контрольно-измерительных материалах ОГЭ и ЕГЭ по информатике. Действительно, в ОГЭ версии 2022 года нет ни одного задания, связанного с темой обработки звуковой информации, а в ЕГЭ ранее в задании 7 были задачи на расчет информационного объема звукового файла, но в последних версиях диагностических работ МЦКО в формате ЕГЭ присутствуют только задачи на кодирование растровой графической информации.

Весьма скромное отражение нашла данная тема также в наиболее распространенных учебниках информатики для уровня основного общего образования, что отмечает, в частности, Н. Ю. Королева в своей работе «Формирование умений работы с цифровым звуком в школьном курсе информатики» [9]. Однако умение обрабатывать звуковую информацию в настоящее время является достаточно существенным цифровым навыком для учащихся основной школы. В связи с этим в настоящей статье мы постараемся частично восполнить некоторый дефицит, существующий в области методики преподавания технологии обработки цифрового звука в основной школе.

Многочисленные преимущества использования свободного программного обеспечения отечественной разработки многократно подчеркивались нами в работах, написанных в соавторстве с А. Ю. Федосовым (см., например, [15]). К основным преимуществам можно отнести, прежде всего, *обеспечение безопасности информационной инфраструктуры российских образовательных организаций на фоне постоянного ужесточения экономических санкций со стороны правительств ряда западных государств*, направленных в том числе против отечественных вузов [11], а также существенную экономию бюджетных средств на закупку лицензий на использование программного обеспечения.

Что касается понятия *методика обучения некоторой информационной технологии*, то, с нашей точки зрения, его можно приближенно рассматривать как *процесс передачи от преподавателя обучающемуся некоего необходимого и достаточного набора алгоритмов создания тех или иных информационных объектов, соответствующих изучаемой информационной технологии, включая знание и умение использовать стандартный инструментарий той программной среды (сред), которая является оптимальной для данных целей на основе сравнительного анализа доступного программного обеспечения.*

Таким образом, в качестве показателя того, что учащийся приобрел необходимые цифровые навыки и овладел рассматриваемой информационной технологией на некотором минимально необходимом уровне, будет факт создания им достаточно сложного, комплексного информационного объекта, соответствующего типу данной информационной технологии.

В случае технологии создания и обработки звуковой информации в качестве такого информационного объекта, с нашей точки зрения, можно рассматривать *аудиоклип*, при создании которого используются как самостоятельно записанные, так и скаченные из легальных источников звуковые фрагменты, которые были в дальнейшем собраны в единое целое в звуковом редакторе.

2. Выбор оптимального варианта программного обеспечения для поддержки преподавания темы «Создание и обработка звуковой информации» на уровне основного общего образования

При выборе оптимального типа программного обеспечения в первую очередь рационально было бы исключить те группы ПО, которые в принципе можно не рассматривать для поддержки преподавания темы «Создание и обработка звуковой информации» в основной школе. К ним необходимо отнести все **мобильные приложения** в связи с тем, что в новых санитарно-эпидемиологических правилах СП 2.4.3648–20, которые уже вступили в силу и будут действовать до 1 января 2027 года, говорится, что для образовательных целей мобильные средства связи не могут использоваться [12].

Данная норма была введена в связи с тем, что мобильные устройства имеют чаще всего маленький размер экрана (меньше 10 дюймов), что в свою очередь влечет за собой напряжение органов зрения и нарушение осанки учащихся.

Еще один тип программного обеспечения для обработки цифрового звука — **онлайн-редакторы зву-**

ка, которые, с одной стороны, имеют некоторые преимущества перед офлайн-версиями — например, возможность их использования не зависит от типа операционной системы, установленной на учебном компьютере. Однако, с другой стороны, их применение в учебном процессе основной школы достаточно рискованно в связи с тем, что скорость и стабильность интернет-соединения в различных образовательных организациях неоднородны и преподаватель, использующий на уроке эти редакторы, может столкнуться с тем, что в текущий момент онлайн-сервис недоступен или скорость обработки звука очень низка из-за нестабильного соединения, что ставит под угрозу нормальное проведение урока.

Таким образом, нам представляется, что *оптимальным типом программного обеспечения для поддержки преподавания темы «Создание и обработка звуковой информации» являются свободные офлайн-звуковые редакторы, которые должны быть кроссплатформенными и обладать достаточно низкими минимальными системными требованиями к конфигурации компьютеров, на которых планируется осуществлять учебный процесс.* В данном случае мы говорим и о домашних компьютерах учащихся, так как было бы неплохо активно применять практические домашние задания в ходе обучения по данной теме курса информатики для повышения качества учебного процес-

са. Требование кроссплатформенности звуковых редакторов также связано с тем, что на учебных компьютерах в классе информатики и на домашних компьютерах преподавателя и учащихся могут использоваться различные типы операционных систем (Windows, MacOS, Linux).

Очевидно, что проприетарные платные звуковые редакторы, среди которых можно выделить такие, как Adobe Audition, Acoustica, FL Studio, Amadeus Pro, Sound Forge Pro, для использования в учебном процессе основной школы также будут далеки от оптимума ввиду довольно высокой стоимости лицензии, часто избыточного для непрофессиональных целей функционала и весьма сложного интерфейса пользователя. Тем не менее среди достоинств проприетарных звуковых редакторов можно отметить наличие учебно-методической литературы и самоучителей по работе с данными программными средствами, таких как, например, книга С. В. Глушакова, А. В. Харьковского по использованию программы для обработки звука Sound Forge [5].

Таким образом, нам предстоит выбирать из списка свободных кроссплатформенных офлайн-звуковых редакторов с невысокими требованиями к конфигурации аппаратного обеспечения. Наиболее известные из этих редакторов представлены в таблице 1.

В результате анализа информации, приведенной в таблице 1, мы выделили **звуковой редактор Audacity**

Таблица 1

Сравнительные характеристики свободных кроссплатформенных офлайн-звуковых редакторов

№ п/п	Название звукового редактора	Официальный сайт проекта	Плюсы звукового редактора	Минусы звукового редактора
1	Audacity	https://www.audacityteam.org	<ul style="list-style-type: none"> • Кроссплатформенное ПО; • мощный функционал; • простой интерфейс; • интерфейс русифицирован; • низкие минимальные системные требования; • возможность экспорта файлов в популярные форматы MP3, WAV, OGG; • большое количество встроенных звуковых эффектов 	<ul style="list-style-type: none"> • Некоторый дефицит учебной литературы и справочных интернет-источников
2	Ardour	https://ardour.org/	<ul style="list-style-type: none"> • Кроссплатформенное ПО; • широкий функционал 	<ul style="list-style-type: none"> • Перегруженный инструментами пользовательский интерфейс; • бесплатная версия имеет некоторые ограничения (возникает тишина в записи через 10 минут); • отсутствие русскоязычных обучающих материалов
3	Wave Editor	https://www.wave-editor.com/ru/	<ul style="list-style-type: none"> • Простой интерфейс пользователя; • есть все основные операции; • низкие системные требования 	<ul style="list-style-type: none"> • Небольшое количество функций; • подходит только для самых элементарных операций со звуком; • иногда нормализация громкости с первого раза срабатывает некорректно; • отсутствие русскоязычных обучающих материалов
4	MusineKit	https://www.brainmodular.com/motionkit-en.php	<ul style="list-style-type: none"> • Наличие подробной инструкции пользователя на сайте проекта; • для загрузки доступны дистрибутивы для операционных систем семейства Windows и MacOS 	<ul style="list-style-type: none"> • Нет русифицированной версии, только английская и французская; • отсутствие русскоязычных обучающих материалов

как возможный оптимальный вариант программного обеспечения для поддержки учебного процесса по теме «Создание и обработка звуковой информации» в основной школе. Далее мы рассмотрим подробнее его основные пользовательские характеристики, а также обеспеченность учебно-методической литературой и справочными интернет-сайтами.

Официальный сайт проекта Audacity содержит ознакомительное руководство пользователя, включающее описание основных операций (запись, воспроизведение, редактирование и сохранение звуковых файлов, изменение громкости звучания, устранение шумов и т. п.) [19]. Более подробное описание функционала аудиоредактора Audacity можно найти в учебном пособии В. А. Горюнова, А. Н. Стася «Обработка и монтаж аудиозаписей с использованием Audacity» [6].

Кроме того, существует несколько устаревший на сегодняшний день учебный курс по Audacity авторского коллектива под руководством А. Прокудина [13].

Надо отметить, что описание аудиоредактора Audacity встречается в очень малом количестве учебной и учебно-методической литературы. Так, например, краткое описание присутствует в учебном пособии для преподавателей по отечественной операционной системе «Линукс Юниор» В. Б. Волкова [4]. В этом пособии автор подчеркивает, что Audacity в отличие от многих своих бесплатных аналогов (таких как, например, программа KRes, имеющая сборку только для операционных систем семейства Linux) является профессиональным звуковым редактором. В качестве примера использования редактора Audacity В. Б. Волков предлагает рассмотреть технологию подавления низкоуровневого шума в звукозаписи между словами.

Интересно, что аудиоредактор Audacity редко упоминается даже в литературе, посвященной свободным операционным системам семейства Linux, хотя программа Audacity устанавливается по умолчанию в различных дистрибутивах Linux в качестве редактора звуковых файлов, например, в отечественном дистрибутиве «Альт Образование 10», и руководство пользователя для данного дистрибутива включает в себя краткое описание функционала Audacity [2]. Кроме того, краткое описание Audacity можно обнаружить в переводном учебном пособии «Linux на практике» американцев К. Бреснахэна, Р. Блума [3].

Программа Audacity упоминается также в учебнике по информатике для вузов А. С. Грошева, П. В. Залякова [7], базирующемся на использовании свободного программного обеспечения, но, к сожалению, Audacity в этом учебнике рассматривается только в разделе, посвященном теоретическим вопросам кодирования цифрового звука.

Что касается минимальных системных требований аудиоредактора Audacity, то они вполне скромные и позволяющие установить данное программное обеспечение практически на любой из используемых в настоящее время в учебном процессе персональных компьютеров [14]:

- операционная система: Windows XP, Vista, 7, 8, 10, Linux, MacOS, UNIX-подобные операционные системы;
- тактовая частота процессора: от 1 ГГц;

- оперативная память: от 512 Мб;
- место на жестком диске: от 51 Мб;
- язык интерфейса: русский, английский;
- размер дистрибутива: 25 Мб.

Звуковой редактор Audacity обладает весьма внушительным функционалом, который позволяет пользователю решать следующие задачи при создании и обработке звуковой информации [18]:

- импорт и экспорт файлов WAV, MP3 (с использованием кодировщика LAME MP3), Vorbis, FLAC и других форматов;
- запись с микрофона, линейного входа и других источников;
- запись с одновременным прослушиванием имеющихся дорожек;
- запись до 16 каналов одновременно (необходима многоканальная звуковая карта);
- эффекты и расширения как в комплекте поставки, так и устанавливаемые отдельно (LADSPA либо на функциональном языке Nyquist);
- индикаторы уровня записи и воспроизведения;
- изменение темпа с сохранением высоты тона;
- изменение высоты тона с сохранением темпа;
- удаление шума по образцу;
- спектральный анализ с использованием преобразования Фурье с различными формами окна;
- воспроизведение множества дорожек одновременно (без поддержки многоканального звука — при воспроизведении используются только два канала, в которые микшируются все дорожки);
- сведение дорожек с разными качественными характеристиками с автоматическим преобразованием к заданным характеристикам проекта в режиме реального времени;
- результаты могут сохраняться во множество звуковых форматов.

Таким образом, можно отметить наличие минимально достаточного числа учебно-методической литературы и справочных интернет-сайтов, посвященных звуковому редактору Audacity, при их почти полном отсутствии у свободных аналогов, а также очень внушительный функционал при весьма скромных минимальных системных требованиях. Вышеуказанные преимущества, с нашей точки зрения, делают именно звуковой редактор Audacity оптимальным программным средством для поддержки преподавания темы «Создание и обработка звуковой информации» на уровне основного общего образования.

3. Еще раз о принципе инвариантности методики обучения относительно программного и аппаратного обеспечения

Принцип инвариантности методики обучения какой-либо информационной технологии, в том числе технологии создания и обработки цифрового звука, многими авторами (см. например, работу Н. Ю. Королевой [9]) воспринимается как нечто незыблемое и очевидное само по себе. Однако нам представляется, что данный вопрос является значительно более сложным и требует

дальнейших исследований. Тем не менее постараемся рассмотреть его в первом приближении на материале изучаемой нами информационной технологии.

В разделе 2 данной статьи мы уже показали, что для основной школы реально можно использовать только свободные кроссплатформенные офлайн-звуковые редакторы. Под кроссплатформенными мы подразумеваем аудиоредакторы, имеющие сборки под настольные операционные системы (Windows, Linux, MacOS), но не под мобильные ОС (Android, iOS), так как использование абсолютного большинства мобильных устройств в учебном процессе основной школы в настоящее время запрещено законодательно. Кроме того, интересующее нас программное обеспечение очевидно должно обладать невысокими минимальными системными требованиями к аппаратному обеспечению персональных компьютеров, на которых планируется его установка, так как методика обучения должна быть рассчитана в том числе на использование в школах, не оснащенных современными мощными учебными компьютерами. Таким образом, говорить об инвариантности методики обучения относительно аппаратного обеспечения, используемого в учебном процессе, уже не получается.

Далее, рассматривая функционал и пользовательский интерфейс аудиоредакторов, хотелось бы отметить, что для использования в учебном процессе основной школы оптимальным является, с одной стороны, достаточно простой, интуитивно понятный интерфейс, не перегруженный различными сложными в понимании их использования инструментами, так как в основной школе информационные технологии изучаются с ознакомительными целями. С другой стороны, функционал программного обеспечения должен быть все-таки не примитивным, иначе возможна пониженная мотивация учащихся к изучению данной технологии и соответствующего ей программного обеспечения ввиду того, что, когда слишком просто — это неинтересно для подростка.

Таким образом, список подходящего для поддержки преподавания темы «Создание и обработка звуковой ин-

формации» программного обеспечения все более и более сужается, и, с нашей точки зрения, *представляется рациональным, чтобы разработчик методики обучения некоей информационной технологии рекомендовал конкретный вариант программного обеспечения, которое предполагается использовать в его методике*. Данная позиция определяется тем, что у учителей информатики, особенно у молодых специалистов, использующих определенную методику обучения в своей работе, явно не хватает времени на эксперименты с применением различных аналогичных программ и на выбор оптимального варианта. Применение неудачного программного обеспечения, а также непродуманного стека практических заданий по освоению изучаемой информационной технологии может привести к духовно-нравственным деформациям у школьников, например к неоправданному пораженческому опыту. В связи с вышесказанным принцип инвариантности методики обучения информационным технологиям относительно программного обеспечения, используемого в учебном процессе, также представляется весьма сомнительным и определенно нуждающимся в дальнейшем изучении и уточнении.

4. Поурочное планирование по теме «Создание и обработка звуковой информации» на уровне основного общего образования

В таблице 2 представлен возможный вариант поурочного планирования по теме «Создание и обработка звуковой информации» для основной школы. Отметим, что полный вариант преподавания рассматриваемой нами информационной технологии займет 10 учебных часов, на что не всегда есть время в учебном плане в рамках преподавания непосредственно предмета информатики. В таком случае существует возможность включить изучение цифрового звука по предлагаемой

Таблица 2

Поурочное планирование по теме «Создание и обработка звуковой информации» для основной школы

№ урока	Содержание урока	Вид урока
1	Теория звука. Цифровое кодирование аналогового звукового сигнала. Дискретизация. Квантование	Лекция
2	Различные цифровые звуковые форматы. Форматы MP3, WAV, OGG. Решение задач на нахождение информационного объема звукового файла двумя способами — расчетным и экспериментальным	Лекция, семинар
3	Первичное знакомство с звуковым редактором Audacity. Решение задач на нахождение информационного объема звукового файла двумя способами — расчетным и экспериментальным	Практическая работа, семинар
4	Обзор основных программных средств для создания и обработки цифровой звуковой информации. Критерии выбора оптимального программного средства для обработки цифрового звука	Лекция, семинар
5	Официальный сайт проекта Audacity и установка дистрибутива программы под основными настольными операционными системами (Windows, Linux, MacOS). Подробное изучение интерфейса пользователя звукового редактора Audacity	Лекция, практическая работа

№ урока	Содержание урока	Вид урока
6	Поиск в открытых источниках и последующий импорт в звуковой редактор Audacity звуковых файлов, содержащих как отдельные звуки, так и музыку. Обзор законодательства в области защиты авторских и смежных прав на музыкальные произведения	Лекция, практическая работа
7	Параллельное использование нескольких звуковых дорожек. Основные операции редактирования звуковых файлов (выделение, вырезание, копирование, вставка звукового фрагмента). Последующее сохранение готового проекта во внутреннем формате редактора Audacity — AUP	Практическая работа
8	Самостоятельная запись звука (голоса) с помощью редактора Audacity и последующее наложение на него фоновой мелодии	Практическая работа
9	Применение различных эффектов к фрагментам звукозаписи	Практическая работа
10	Защита индивидуальных проектов по созданию собственного аудиоклипа учащимися средствами звукового редактора Audacity	Конференция

нами методике в материал элективного курса или объединения дополнительного образования по теме, например, «Современные информационные технологии».

5. Краткое содержание обучения по теме «Создание и обработка звуковой информации» на уровне основного общего образования

5.1. Этап 1. Теоретические основы цифрового кодирования звуковой информации.

Экспорт цифрового звука в различные звуковые форматы — MP3, WAV, OGG

В настоящей работе мы не ставим цель подробно разобрать теоретические аспекты оцифровки звуковой информации, так как это сделано ранее авторами многих учебных пособий по информационным технологиям, например, А. С. Грошевым, П. В. Закляковым в [7, с. 96–108].

В данном разделе в рамках разрабатываемой методики можно предложить возможный вариант проведения практической работы по измерению информационного объема звукового файла, записываемого с неким заданным значением дискретизации. Идея проведения лабораторной работы связана с задачами на кодирование звуковой информации, аналогичными задаче 7 из КИМ ЕГЭ по информатике.

Возьмем, например, следующее условие задачи [8]:

Производится двухканальная (стерео) звукозапись с частотой дискретизации 48 кГц. Запись длится 2 минуты 15 секунд, ее результаты записываются в файл без сжатия данных, причем каждый сигнал кодируется минимально возможным и одинаковым количеством бит. Максимальная битовая глубина кодирования звука, которая была использована в этой записи, — 24 бита. Определите информационный объем полученного файла в мегабайтах. В ответе запишите только число.

С одной стороны, задача может быть решена с помощью вычислений, а с другой стороны, результат вычислений может быть проверен проведением звукозаписи

с указанными в задаче параметрами и последующим определением объема полученного звукового файла.

Вычислим информационный объем звукового файла теоретически по формуле 1 с точностью до первого знака после запятой и получим 37,1 Мбайт.

$$I = \frac{2 \cdot 48000 \cdot 135 \cdot 24}{8 \cdot 1024 \cdot 1024} = 37,1 \text{ Мбайт.} \quad (1)$$

Далее проведем звукозапись в программе Audacity с указанными в задаче параметрами и затем экспортируем эту звукозапись в доступные форматы. Результаты данного эксперимента приведены в таблице 3.

Таким образом, на основе данных таблицы 3 можно сделать вывод, что расчетное значение информационного объема звукового файла совпадает с экспериментально полученным значением для файла, экспортированного в «сырой» звуковой формат WAV без потери качества звука.

5.2. Этап 2. Обзор основных программных средств для создания и обработки цифровой звуковой информации. Критерии выбора оптимального программного средства для обработки цифрового звука

Материал данного этапа обучения полностью представлен в разделе 2 настоящей статьи.

5.3. Этап 3. Официальный сайт проекта Audacity и установка дистрибутива программы под основными настольными операционными системами (Windows, Linux и MacOS). Подробное изучение интерфейса пользователя звукового редактора Audacity

Интерфейс пользователя аудиоредактора Audacity включает в себя следующие основные элементы (рис. 1):

1. Панель меню программы.
2. Панель основных инструментов (пауза, воспроизведение, стоп, запись и т. д.).
3. Панель «Инструменты» позволяет выбрать текущий инструмент для таких задач, как выделение, регулировка громкости или масштабирование звукового сигнала по горизонтали.

Результаты эксперимента по оцифровке звука в разных форматах

№ п/п	Формат аудиоданных для выгрузки	Характеристика формата	Информационный объем выгруженного файла, Мбайт
1	MP3	В формате MP3 используется алгоритм сжатия с потерями, разработанный для существенного уменьшения размера данных, необходимых для воспроизведения записи и обеспечения качества воспроизведения звука, близкого к оригинальному [21]	1,96
2	OGG (Vorbis)	Открытый стандарт формата мультимедиакодекса, являющийся основным файловым и потоковым форматом для мультимедиакодексов фонда Xiph.Org. OGG является всего лишь контейнером. Звук или видео сжимаются кодеками, например, Vorbis. Vorbis — это кодек для сжатия звука с потерями на средних и высоких битрейтах (~16–500 (Кбит/с)/канал) [22]	5,44
3	WAV	Формат файла-контейнера для хранения записи оцифрованного аудиопотока. Этот контейнер, как правило, используется для хранения несжатого звука в импульсно-кодовой модуляции [23]	37,1

4. Панель инструментов индикатора записи.

5. Панель инструментов индикатора воспроизведения.

6. Панель инструментов микшера. Здесь настраиваются громкость записи (амплитуда сигнала, с которой будет выполняться запись) и громкость воспроизведения.

7. Панель инструментов *Редактирование звука*. Содержит инструменты, которые выполняют те же функции,

что и те, которые доступны через меню *Правка*, меню просмотра, меню дорожек и сочетания «горячих» клавиш.

8. Панель инструментов *Управление скоростью*. Позволяет воспроизводить или заикливать звук с более низкой или более высокой скоростью, чем обычно, что также влияет на высоту тона. Более низкая, чем обычно, скорость может облегчить расшифровку речи или текстов песен.

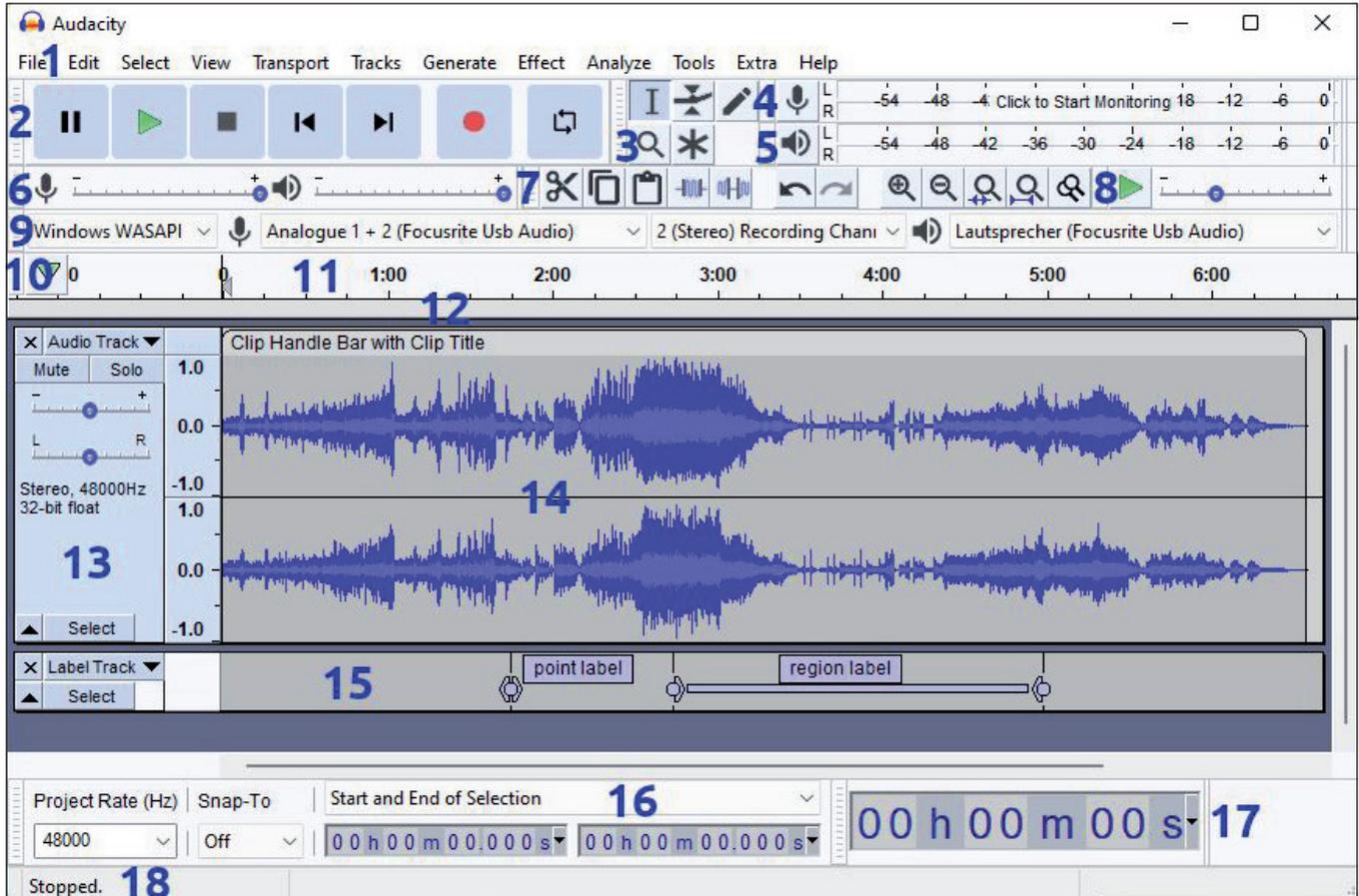


Рис. 1. Интерфейс пользователя аудиоредактора Audacity

9. Панель звуковых устройств, зарегистрированных на компьютере. Позволяет выбрать микрофон и динамики, которые будут применяться при записи и воспроизведении в Audacity.

10. Режим закрепленной/незакрепленной головки воспроизведения/записи. Существует возможность изменить режим воспроизведения и записи с фиксированной головкой, закрепленной на временной шкале. В этом режиме головка остается статической, а профиль звукового сигнала будет двигаться по мере воспроизведения или записи звука. Переключение режимов поведения головки воспроизведения/записи производится с помощью кнопки слева от временной шкалы. По умолчанию данная кнопка отображается в виде зеленого треугольника, направленного вниз. В этом режиме головка воспроизведения или записи будет перемещаться горизонтально, а волновая форма звукового сигнала будет прокручиваться при необходимости (в случае выхода за границы видимости участка звукового сигнала).

11. Временная шкала. Располагается над звуковыми дорожками в виде горизонтальной линейки, отсчитывающей время от нуля (начала звуковой дорожки). В зависимости от масштабирования линейка может как охватывать минуты записанного звука, так и отображать звуковой сигнал «крупным планом» в интервале нескольких секунд или долей секунды.

12. Скраб-линейка. По умолчанию отключена, включается с помощью команды *Правка, Настройки, Интерфейс, Показать скраб-линейку*. Инструмент используется для перемотки и точного поиска какого-либо определенного места в звукозаписи.

13. Вертикальная панель управления звуковой дорожкой. Позволяет удалять звуковую дорожку, отключать воспроизведение данной звуковой дорожки или, наоборот, включать только данную дорожку, регулировать относительную громкость звука данной дорожки, изменять баланс между каналами (левым и правым) звуковой дорожки, использовать меню настройки свойств данной дорожки.

14. Звуковая дорожка, содержащая цифровой звук. Новые звуковые дорожки создаются всякий раз, когда импортируется аудиофайл или пользователь записывает звук с помощью микрофона. Существует возможность создать новую пустую моно- или стереофоническую дорожку с помощью команды *Треки, Добавить новый*. Проект в Audacity может содержать несколько звуковых дорожек, в таком случае звуки, расположенные на них, будут воспроизводиться параллельно.

15. Трек (дорожка) меток. Дополнительный трек, который можно создать в разрабатываемом проекте. Метки в дорожке меток можно использовать для опорных точек или областей в звуковых дорожках проекта. Для создания метки необходимо выделить участок записи, который предполагается пометить, затем используется команда *Правка, Метки, Добавить метку в выделение*. Созданную таким образом на дорожке меток метку можно отредактировать (изменить имя) с помощью контекстного меню *Правка метки*.

16. Поля выбора позиции. Используются для отображения, например, начала и конца выделения фрагмента звукозаписи в формате «часы, минуты, секунды». Данный инструмент удобно использовать в том случае, когда необходимо выделить звук определенной продолжительности.

17. Панель инструментов времени. Панель, которая отображает текущую позицию звука. При воспроизведении или записи данный инструмент динамически показывает текущее положение головки воспроизведения или записи. Поскольку эта панель инструментов доступна только для отображения информации, ее нельзя использовать для сброса положения звуковой головки.

18. Строка состояния Audacity. Серая горизонтальная полоса, расположенная в нижней части окна редактора. Строка состояния не перемещается в интерфейсе программы. Функция строки состояния состоит в том, чтобы отображать статус программы в настоящий момент времени, а именно о записи или воспроизведении, об остановке воспроизведения и т. п. Выше строки состояния находится переключатель частоты дискретизации звукозаписи. Частота дискретизации для проекта по умолчанию установлена на 44 100 Гц.

5.4. Этап 4. Поиск в открытых источниках и последующий импорт в звуковой редактор Audacity звуковых файлов, содержащих как отдельные звуки, так и музыку. Обзор законодательства в области защиты авторских и смежных прав на музыкальные произведения

После ознакомления учащихся с основными панелями инструментов звукового редактора Audacity и их назначением, можно переходить к следующему этапу изучения темы «Создание и обработка звуковой информации», а именно к скачиванию некоторых звуковых фрагментов и последующей их обработке. Под звуковыми фрагментами мы подразумеваем как различные звуки, так и фрагменты музыкальных произведений.

При использовании учащимися при создании собственных проектов различных звуковых фрагментов, скаченных в том числе из открытых источников в сети Интернет, необходимо обратить внимание на соблюдение законодательства Российской Федерации в области защиты авторских и смежных прав. Согласно статье 1259 Гражданского кодекса РФ, объектами авторских прав являются произведения науки, литературы и искусства независимо от достоинств и назначения произведения, а также от способа его выражения, в том числе музыкальные произведения с текстом или без текста. Таким образом, для использования в учебном процессе основной школы (образовательная некоммерческая организация) подходят только музыкальные произведения, распространяемые по свободной лицензии (Creative Commons) или являющиеся общественным достоянием (Public Domain).

Для лицензий типа Creative Commons действует следующий принцип: каждая лицензия позволяет пользователям произведения, если они следуют условиям лицензии, по крайней мере в некоммерческих целях [10]:

- копировать произведение;
- распространять его;
- показывать или исполнять его публично;
- делать его цифровые публичные представления (например, вебкастинг);
- переводить произведение в другой формат как точную копию.

Понятие «общественное достояние» относится к музыкальным произведениям, авторские права на

которые прекратили действовать за истечением срока или никогда не существовали.

Вопрос с использованием музыкальных произведений встает при размещении своего аудио- или видеопроекта (ролика) на видеохостинге YouTube, на котором производится проверка всего загружаемого контента на соблюдение авторских прав.

Что касается различных звуков и шумов, то с ними также могут возникнуть проблемы с загрузкой на YouTube. Так, например, в некоторых случаях даже птичьи песни могут быть распознаны как авторский материал и заблокированы. Вероятно, алгоритмы YouTube считают птичьи песни в фильме очень похожими на звук, защищенный авторским правом [1]. Однако если мы используем звуки и шумы из интернет-коллекций, на сайтах которых декларируется, что все звуки получены из открытых источников, то с точки зрения законодательства Российской Федерации в этом вряд ли можно усмотреть нарушение авторского права. В качестве примера такой коллекции звуков можно привести шумотеку NoiseFX.ru [17]. Команда NoiseFX отмечает на своем сайте, что на нем можно бесплатно скачать звуки для видеомонтажа, рекламных аудио- и видеороликов, театральных представлений, видеоблогов, а также что находящиеся на сайте шумовые эффекты собраны из открытых источников, отсортированы, классифицированы и представлены для широкого использования.

Таким образом, для скачивания музыки, которая может быть легально использована в дальнейшем при создании учащимися различных аудиоклипов, необходимо использовать только сайты с произведениями, распространяемыми как общественное достояние или по лицензии Creative Commons. В качестве примера такого источника можно привести портал Free Music Archive (FMA), основанный в 2009 году радиостанцией WFMU, который предлагает бесплатный доступ к оригинальной музыке с открытой (свободной) лицензией [20].

Скачанные из открытых ресурсов сети Интернет звуки в дальнейшем импортируются в звуковой редактор Audacity с помощью команды *Файл, Импорт, Аудиоданных* и открываются в виде новой звуковой дорожки, как показано на рисунке 1. Перед началом создания нового проекта разумно открыть все звуки, которые планируется использовать в проекте, на отдельных дорожках.

Далее мы переходим к рассмотрению основных операций со звуковыми фрагментами, которые позволяют совершать функционал звукового редактора Audacity.

5.5. Этап 5. Параллельное использование нескольких звуковых дорожек, основные операции редактирования звуковых файлов (выделение, вырезание, копирование, вставка звукового фрагмента). Последующее сохранение готового проекта во внутреннем формате редактора Audacity — AUP

В редакторе Audacity со звуковыми фрагментами можно совершать следующие операции: *вырезать, удалить, копировать, вставить, дублировать, вырезать без соединения, удалить без соединения, заменить тишиной, обрезать звук*. Данные операции доступны пользователю через меню *Правка*. При использовании операций *вырезать без соединения* и *удалить без соеди-*

нения на месте удаленного фрагмента остается пустота. При использовании операции *обрезать звук* удаляется все, кроме выделенного звукового фрагмента.

При построении методики обучения теме «Создание и обработка звуковой информации» на уровне основного общего образования для отработки учащимися операций редактирования звуковых файлов с помощью стандартных операций редактирования, перечисленных выше, в качестве задания можно предложить **мини-проект**, целью которого будет создание аудиоклипа. Аудиоклип должен состоять из нескольких звуков и соответствующего музыкального оформления, причем предполагается наличие нескольких (минимум двух) звуковых дорожек для того, чтобы некоторые звуки были расположены последовательно, один за другим на одной из дорожек, а другие — параллельно им на других дорожках (дорожке).

На рисунке 2 показан возможный вариант реализации мини-проекта по созданию аудиоклипа с использованием двух звуковых дорожек. Цифрами 1–6 показаны различные звуки, расположенные последовательно на звуковой дорожке 1, а именно:

- 1) звук открывания двери автомобиля;
- 2) звук закрывания двери автомобиля;
- 3) звук включения двигателя автомобиля;
- 4) звук проезжающего по шоссе автомобиля;
- 5) звук резкого торможения;
- 6) реакция окружающих на происходящее.

Цифрами 7–8 на рисунке 2 показано музыкальное сопровождение на звуковой дорожке 2, которое имитирует включение музыки в салоне автомобиля, цифрой 9 — продублированный фрагмент звука 8, полученный с использованием стандартного встроенного эффекта *Повтор* звукового редактора.

Для промежуточного сохранения звукового проекта в редакторе Audacity используется собственный внутренний формат этого редактора — AUP. Данный формат не открывается медиапроигрывателями, и для сохранения готового проекта используется экспорт файлов в форматы MP3, WAV и OGG, которые были рассмотрены в разделе 5.1.

5.6. Этап 6. Самостоятельная запись звука (голоса) с помощью редактора Audacity и последующее наложение на него мелодии

После того как учащиеся усваивают основные навыки применения функционала звукового редактора Audacity, на этапе 6 обучения теме «Создание и обработка звуковой информации» можно переходить к разработке мини-проекта, заключающегося в записи собственного голоса и наложении на него какой-либо фоновой мелодии.

Процесс записи в Audacity очень прост: необходимо установить тот микрофон, который пользователь хочет использовать при записи, на панели звуковых устройств и нажать кнопку *Запись* на панели основных инструментов, а после окончания записи — кнопку *Смон.* В процессе записи пользователь будет видеть графически заполняющуюся звуковую дорожку.

Отметим, что правильной последовательностью действий на данном этапе будет следующая:

- 1) записать собственный голос;
- 2) прослушать получившуюся запись;
- 3) вырезать участки тишины или звуковых шумов;



Рис. 2. Пример реализации мини-проекта по созданию аудиоклипа в редакторе Audacity

- 4) импортировать выбранную мелодию;
- 5) подобрать тот участок мелодии, который больше всего соответствует голосовой записи, и удалить лишние части звуковой дорожки, содержащие фоновую мелодию, добившись одинаковой длительности голосовой и фоновой звуковых дорожек;
- 6) прослушать получившийся вариант наложения и скорректировать относительную громкость голоса и фоновой мелодии для обеспечения четкой слышимости голоса;
- 7) сохранить готовый проект во внутреннем формате Audacity — AUP для последующего применения к нему звуковых эффектов.

На данном этапе также важно не забывать, что фоновые мелодии необходимо брать только из интернет-ресурсов, на которых выложены музыкальные компо-

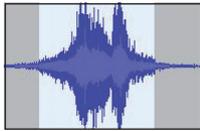
зиции, распространяемые как общественное достояние или по лицензии Creative Commons, иначе полученный звуковой клип будет создан с нарушением законодательства Российской Федерации в области защиты авторских прав на музыкальные произведения.

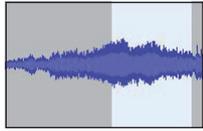
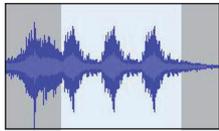
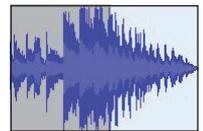
5.7. Этап 7. Применение различных эффектов к фрагментам звукозаписи

Звуковой редактор Audacity в своем арсенале имеет большое количество различных звуковых эффектов, использование которых делает аудиоклип, который разрабатывается учащимися в качестве конечного продукта изучения технологии обработки цифрового звука в рамках предлагаемой методики, значительно интереснее и разнообразнее. Далее в таблице 4 рассмотрим подробнее некоторые основные звуковые эффекты, ко-

Таблица 4

Основные звуковые эффекты меню *Эффекты* аудиоредактора Audacity

№ п/п	Наименование звукового эффекта	Описание звукового эффекта	Графическое отображение звукового эффекта
1	<i>Усиление</i>	Увеличивает или уменьшает громкость выбранного звука в децибелах. Audacity автоматически вычисляет максимальную величину, на которую можно усилить выбранный звук, не вызывая клиппинга (искажения, вызванного попыткой сделать звук слишком громким). На рисунке 3 показано применение эффекта <i>Усиление</i> с клиппингом (перегрузкой).	 <p>Рис. 3. Применение эффекта <i>Усиление</i> с клиппингом</p>
2	<i>НЧ и ВЧ (низкие частоты и высокие частоты)</i>	Увеличивает или уменьшает низкие и высокие частоты звука независимо друг от друга. Данный эффект ведет себя точно так же, как эквалайзер низких и высоких частот в домашней стереосистеме. На рисунке 4 показано применение эффекта НЧ и ВЧ, при котором высокие частоты усилены на 20 дБ, а низкие частоты уменьшены на 20 дБ.	 <p>Рис. 4. Применение эффекта <i>НЧ и ВЧ</i></p>

№ п/п	Наименование звукового эффекта	Описание звукового эффекта	Графическое отображение звукового эффекта
3	<i>Изменение скорости</i>	Изменяет (увеличивает или уменьшает) скорость воспроизведения выделенного звукового фрагмента и одновременно меняет частоту звука. Увеличение скорости приводит к увеличению частоты звука, и наоборот. На рисунке 5 к участку звукового сигнала на светлом фоне применен эффект <i>Изменение скорости</i> в сторону увеличения в два раза, и, следовательно, его длина уменьшилась в два раза.	 <i>Рис. 5. Применение эффекта Изменение скорости</i>
4	<i>Повтор</i>	Повторяет выделенный звуковой фрагмент указанное количество раз. На рисунке 6 показано применение эффекта <i>Повтор</i> к звуковому фрагменту два раза, в результате чего на светлом фоне видно три одинаковых звука, повторяющихся один за другим.	 <i>Рис. 6. Применение эффекта Повтор</i>
5	<i>Реверс</i>	Переворачивает выбранный звуковой фрагмент таким образом, чтобы он звучал так, как будто он воспроизводится с конца до начала (в обратную сторону). На рисунке 7 показано, что к левому звуковому фрагменту сначала был применен эффект <i>Повтор</i> один раз, а затем к повторенной части звука — эффект <i>Реверс</i> , в связи с чем полученный результат визуально воспринимается как отраженный относительно вертикали.	 <i>Рис. 7. Применение эффекта Реверс</i>
6	<i>Фейд-подъем</i>	Применяет нарастание к выбранному фрагменту звука, чтобы амплитуда постепенно возрастала от тишины в начале выделения до исходной амплитуды в конце выделения. Форма затухания линейная. Скорость нарастания полностью зависит от длины выделения, к которому оно применяется. На рисунке 8 показано применение эффекта <i>Фейд-подъем</i> к звуковому фрагменту на светлом фоне.	 <i>Рис. 8. Применение эффекта Фейд-подъем</i>
7	<i>Фейд-спад</i>	Применяет затухание к выбранному звуковому фрагменту, чтобы амплитуда постепенно уменьшалась от исходной амплитуды в начале выделения до тишины — в конце выделения. Форма затухания — линейная. Скорость затухания полностью зависит от длины выделения, к которому оно применяется. На рисунке 9 показано применение эффекта <i>Фейд-спад</i> к звуковому фрагменту на светлом фоне.	 <i>Рис. 9. Применение эффекта Фейд-спад</i>
8	<i>Эхо</i>	Повторяет выбранный звук снова и снова, обычно с каждым разом тише. Время задержки между повторами фиксировано, паузы между повторами нет. На рисунке 10 показано сначала применение эффекта <i>Повтор</i> к левому фрагменту звука, а затем к повторенному участку — применение эффекта <i>Эхо</i> . На рисунке видно, что область с большей амплитудой повторяется два раза при условии постепенного убывания амплитуды со временем.	 <i>Рис. 10. Применение эффекта Эхо</i>

торые доступны пользователю в меню *Эффекты* аудиоредактора Audacity и могут быть с успехом использованы учащимися при создании ими собственных аудиоклипов.

6. Заключение

В статье описан каркас методики обучения школьников на уровне основного общего образования теме «Создание и обработка звуковой информации» с использованием свободного кроссплатформенного офлайн-звукового редактора Audacity. Показано, что в качестве базового программного обеспечения для преподавания

данной темы можно использовать исключительно свободные кроссплатформенные офлайн-звуковые редакторы, имеющие скромные минимальные системные требования к аппаратному обеспечению персональных компьютеров, на которые планируется их установка.

Оптимальным вариантом такого программного обеспечения, с нашей точки зрения, является звуковой редактор Audacity, в некоторой степени обеспеченный учебно-методической литературой, посвященной технологии его использования в учебном процессе, а также учебными и справочными интернет-ресурсами. Кроме того, аудиоредактор Audacity сочетает в себе весьма вну-

шительный функционал, близкий к профессиональным аналогам, и достаточно простой, интуитивно понятный школьнику русифицированный интерфейс пользователя.

Предложенная в статье методика обучения теме «Создание и обработка звуковой информации» имеет следующие преимущества перед методиками, построенными на использовании проприетарного программного обеспечения, и методиками, в которых не конкретизируется тип применяемого программного обеспечения:

- может быть использована как для очной, так и для дистанционной форм обучения, так как программное обеспечение, используемое в ней в качестве базового, может быть установлено практически на любые конфигурации домашних персональных компьютеров, работающих под основными типами операционных систем (Windows, Linux, MacOS);
- соответствует законодательству Российской Федерации в области охраны здоровья учащихся (СП 2.4.3648-20 [12]), так как не предполагает использования мобильных устройств в учебном процессе;
- не требует дополнительных финансовых вложений в закупку лицензий на использование проприетарного программного обеспечения как для учебных компьютеров, так и для домашних компьютеров учащихся;
- обеспечивает высокую устойчивость учебно-воспитательного процесса к деструктивному влиянию на информационную структуру российских образовательных организаций, заключающемуся в том числе в немотивированном отказе в лицензиях на использование проприетарного программного обеспечения иностранного производства [11]. Отметим, что максимальный уровень устойчивости может быть достигнут в результате перевода учебного процесса на отечественные операционные системы, специально ориентированные на нужды образовательных организаций, такие как, например, «Альт Образование 10» [2]. Кроме того, устойчивость учебного процесса обусловливается тем, что применяемый офлайн-звукоредактор не предполагает наличия обязательного постоянного подключения к сети Интернет в течение всего урока, что необходимо в случае использования онлайн-аудиоредакторов (как это предлагается, например, в [9]);
- обеспечивает соблюдение авторских прав учащимися, причем в двояком смысле: как в отношении использования лицензионного свободного программного обеспечения в учебном процессе, так и в отношении использования только музыкальных произведений, распространяемых как общественное достояние или по лицензии Creative Commons;
- обеспечивает высокую мотивацию обучающихся к изучению материала темы, так как используемый звуковой редактор — достаточно мощный по функционалу и в то же время обладает интуитивно понятным русифицированным интерфейсом; кроме того, его можно использовать в домашних условиях для реализации учащимися своих собственных проектов;
- обеспечивает высокий интерес обучающихся к изучаемой теме в связи с предпрофессиональ-

ной подготовкой к возможному для некоторых школьников будущему овладению профессией звукорежиссера — ввиду использования программного обеспечения, эквивалентного профессиональному, и соответствующих алгоритмов обработки цифрового звука.

Список источников

1. Ализар А. Шум ветра тоже может быть «авторским материалом» // Хабр. 18 июня 2012 года. <https://habr.com/ru/post/145984/>
2. Альт Образование 10.0. Документация. Руководство пользователя. <https://docs.altlinux.org/ru-RU/alt-education/10.0/html/alt-education/ch39.html#idm45732522625936>
3. Бреснахэн К., Блум Р. Linux на практике. СПб.: Питер, 2017. 384 с.
4. Волков В. Б. Линукс Юниор: книга для учителя. М.: АIT Linux; ДМК Пресс, 2010. 362 с.
5. Глушаков С. В., Харьковский А. В. Цифровое видео и аудио. Секреты обработки на ПК. М.: АСТ Москва; Владимир: ВКТ, 2008. 320 с.
6. Горюнов В. А., Стась А. Н. Обработка и монтаж аудиозаписей с использованием Audacity (ПО для обработки и монтажа аудиозаписей): учебное пособие. М., 2008. 40 с. [https://pdf.zlibcdn.com/dtoken/52b94de7a8bac97239ccdf39a531dd73/Obrabotka_i_montazh_audiozapisei_s_ispolzovaniem_A_792223_\(z-lib.org\).pdf](https://pdf.zlibcdn.com/dtoken/52b94de7a8bac97239ccdf39a531dd73/Obrabotka_i_montazh_audiozapisei_s_ispolzovaniem_A_792223_(z-lib.org).pdf)
7. Грошев А. С., Захляков П. В. Информатика: учебник для вузов. 4-е изд., перераб. и доп. М.: ДМК Пресс, 2018. 672 с.
8. ЕГЭ по информатике. Задания КИМ № 7. Информационный объем звуковых данных // Сайт К. Ю. Полякова. <https://kpolyakov.spb.ru/school/ege/gen.php?action=viewAllEgeNo&egeId=7&cat39=on>
9. Королева Н. Ю. Формирование умений работы с цифровым звуком в школьном курсе информатики // Информатика в школе. 2021. № 10. С. 28–31.
10. Лицензии и инструменты Creative Commons // Википедия. https://ru.wikipedia.org/wiki/Лицензии_и_инструменты_Creative_Commons
11. Пешкова И. МГТУ Баумана остается без Windows: Microsoft отказалась поставлять ему свое ПО // CNews. 9 декабря 2020 года. https://www.cnews.ru/news/top/2020-12-09_mgtu_baumana_ostalsya_bez
12. Постановление Главного государственного санитарного врача Российской Федерации № 28 от 28 сентября 2020 года «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»». <http://publication.pravo.gov.ru/Document/View/0001202012210122>
13. Прокудин А., Эцманн Э., Мацонни Д. Audacity 1.2.1. Вводный курс. ALT LINUX TEAM, 2004. <https://glashkoff.com/wp-content/uploads/2017/02/audacity.pdf>
14. Системные требования Audacity. <https://легион.net/audacity-2-0-4-rus-portable/>
15. Федосов А. Ю., Маркушевич М. В. Актуальные вопросы применения отечественного программного обеспечения в учебно-воспитательном процессе общеобразовательной школы // Информатика и образование. 2018. № 6. С. 11–21.
16. Федосов А. Ю., Маркушевич М. В. Элементы методики преподавания векторной графики на основе свободного графического редактора LibreOffice Draw на уровне основного общего образования // Информатика в школе. 2021. № 9. С. 4–13.
17. Шумотека. Онлайн коллекция звуков NoiseFX.ru. <https://noisefx.ru>
18. Audacity // Википедия. <https://ru.wikipedia.org/wiki/Audacity>
19. Audacity Tour Guide — Audacity Manual // Audacity Manual. https://manual.audacityteam.org/man/audacity_tour_guide.html#other
20. Free Music Archive. <https://freemusicarchive.org/>
21. MP3 // Википедия. <https://ru.wikipedia.org/wiki/MP3>
22. Ogg // Википедия. <https://ru.wikipedia.org/wiki/Ogg>
23. WAV // Википедия. <https://ru.wikipedia.org/wiki/WAV>

DOI: 10.32517/2221-1993-2022-21-1-67-80

В. О. Дженжер, Л. В. Денисова*Оренбургский государственный педагогический университет, Оренбург, Россия*

ДИНАМИЧЕСКИЕ МАССИВЫ И СПИСКИ В PASCALABC.NET

Аннотация

В статье описывается методика преподавания школьникам и студентам темы «Одномерные динамические массивы» на языке PascalABC.NET. Показывается, что переход от изучения статических массивов к динамическим хотя и потребует от преподавателя времени и некоторых усилий, но все затраты окупаются приобретением нового мощного инструмента. Помимо динамических массивов излагаются основы работы со списками List<T>, которые во многих случаях оказываются удобнее массивов. Списки являются одной из основных структур данных во многих современных языках программирования. В статье описываются область применимости списков и отличие их от массивов. Приводятся примеры написания функций, входными и/или выходными параметрами которых являются массивы или списки. Рассматривается техника использования срезов для работы с массивами и списками. Срезы предоставляют лаконичный синтаксис для обращения к части массива. Это позволяет существенно сократить код, который в традиционном стиле содержит много циклических действий. В заключение делается вывод о том, что изучение динамических массивов и списков позволяет легче перейти к программированию на других современных языках.

Ключевые слова: программирование, динамический массив, список, PascalABC.NET.

1. Введение

Изучение линии «Алгоритмизация и программирование» в средней школе предполагает освоение темы «Массивы». В разных языках программирования, изучаемых в школе, используются разные виды массивов. Так, в традиционном школьном Pascal и в алгоритмическом языке рассматриваются только статические массивы. В C++ (STL), C# все с самого начала знакомятся с динамическими массивами (см., например, [2]). В PascalABC.NET принят такой же подход, правда, статические массивы в языке остаются для желающих. В Python привычных всем массивов нет, но их можно с некоторой степенью удобства заменить списками, ближайшими «родственниками» динамических массивов [7].

В школьных учебниках, опирающихся при изучении программирования на Pascal (например, [1, 3, 6, 8]), для решения задач применяются статические массивы. По всей видимости, это связано с устоявшейся традицией, а также с тем, что в ЕГЭ по информатике и ИКТ используются примеры кода в таком же стиле. По нашему мнению, сегодня при изучении Pascal в школе следует переходить к использованию динамических массивов, и тому есть несколько причин.

Во-первых, школьники познакомятся с понятием динамической памяти и ее отличием от статической памяти, что позволит им впоследствии легко осваивать современные структуры данных, являющиеся по преимуществу динамическими. Кроме того, изучение динамических переменных подготовит ученика к изуче-

Контактная информация

Дженжер Вадим Олегович, канд. физ.-мат. наук, доцент, зав. кафедрой информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Оренбург, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* vdjenjer@yandex.ru

Денисова Людмила Викторовна, канд. пед. наук, доцент кафедры информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Оренбург, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* lv-denisova@yandex.ru

V. O. Dzhenzher, L. V. Denisova

Orenburg State Pedagogical University, Orenburg, Russia

DYNAMIC ARRAYS AND LISTS IN PASCALABC.NET

Abstract

The article describes the methodology for teaching to schoolchildren and students the topic "One-dimensional dynamic arrays" in the PascalABC.NET programming language. It is shown that the transition from studying static arrays to dynamic arrays, although it will require time and some effort from the teacher, but all costs are paid off by the acquisition of a new powerful tool. In addition to dynamic arrays, the basics of working with List<T> lists are outlined, which in many cases turn out to be more convenient than arrays. Lists are one of the basic data structures in many modern programming languages. The article describes the scope of lists and their difference from arrays. Examples of writing functions whose input and/or output parameters are arrays or lists are given. The technique of using slices for working with arrays and lists is considered. Slices provide a concise syntax for accessing a portion of an array. This allows you to significantly reduce the code, which in the traditional style contains a lot of cyclic actions. In closing, conclusions are drawn that the study of dynamic arrays and lists makes it easier to switch to programming in other modern languages.

Keywords: programming, dynamic array, list, PascalABC.NET.

нию объектно-ориентированного программирования, в котором экземпляры классов (объекты) также являются динамическими переменными.

Во-вторых, динамические массивы имеют огромное количество полезных методов, которые избавляют программиста от рутинных и позволяют записывать краткий и понятный код. Заметим, что сами эти методы не могут и не должны быть предметом изучения на уроке. Но во внеурочной работе более продвинутые ученики, как показывает наша практика, с удовольствием их используют.

В-третьих, динамические массивы много лет назад введены в язык PascalABC.NET [4, 5, 9], который является одним из самых распространенных в школах нашей страны. Для того чтобы включить их изучение в школьную программу, не нужны сверхусилия, а требуется лишь добрая воля учителя.

Ну и, наконец, в-четвертых, переходя к изучению других языков, таких как C#, C++, Python и др., учащийся встречается именно с динамическими структурами данных. Поэтому знакомство с такими структурами сделает переход к изучению промышленных языков программирования более легким.

Далее мы попытаемся развернуть и обосновать некоторые из прозвучавших тезисов, а также изложить сведения, необходимые для получения первичных навыков работы с динамическими массивами. Поскольку «нельзя объять необъятное», в этой работе мы рассмотрим только одномерные динамические массивы и списки.

2. Статическая и динамическая память

Чтобы объяснить разницу между статическими и динамическими переменными, следует немного рассказать о логическом устройстве памяти компьютера. При этом не будем углубляться в детали организации памяти в Microsoft.NET, в тонкости управления памятью и сборки мусора. Приведем сведения, достаточные для понимания предмета нашего разговора.

С точки зрения программы (процесса), которую мы запускаем, доступная память делится на две части: на статическую и динамическую. Схематически это представлено на рисунке 1. Здесь *виртуальная память* — это оперативная память плюс часть дисковой памяти, которую операционная система использует в случае нехватки оперативной памяти. Для простоты будем считать, что *процесс* представляет собой выполняемую в данный момент программу, за которой закреплены некоторые ресурсы компьютерной системы.

Статическая память выделяется программе при ее запуске и служит для хранения кода самой программы, значений статических переменных и для оперативных целей. Все переменные, которые мы описываем в программе при помощи служебного слова *var*, являются статическими, следовательно, хранятся в статической памяти. Объем такой памяти ограничен, так что мы не можем разместить в ней больше определенного количества данных. Еще одна особенность статических переменных состоит в том, что их невозможно удалить из памяти, если они больше не нужны. Также нельзя изменить размер статической переменной (например, количество элементов статического массива), если она однажды была создана. Очень неудобно разрабатывать

Виртуальная память



Рис. 1. Три программы, размещенные в памяти компьютера

подпрограммы, использующие статические массивы, потому что размер такого массива придется жестко прописать в заголовке подпрограммы.

Динамическая память (ее еще называют *кучей*, англ. heap) — это часть виртуальной памяти, не занятая процессами. Она может быть использована для размещения переменных, принадлежащих процессам. Если процессу нужна дополнительная память, он просит операционную систему выделить ему некоторый участок кучи. Операционная система выделяет память и закрепляет ее за соответствующим процессом. Чтобы процесс мог обращаться к этой памяти, ему передается ссылка на начало выделенного участка (рис. 2).

Виртуальная память

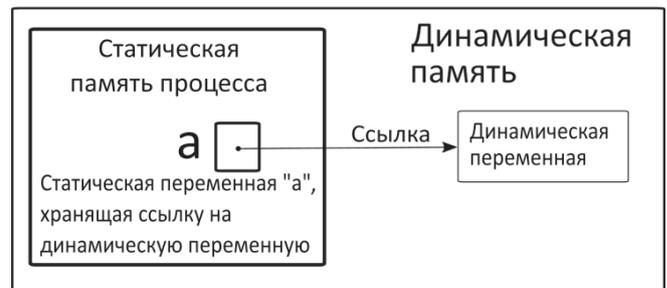


Рис. 2. Программа, в которой определена статическая переменная *a*, ссылающаяся на динамическую переменную. Динамическая переменная не имеет своего имени, доступ к ней может быть осуществлен только по ссылке

Ссылка похожа на указатель, т. е. переменную, хранящую адрес памяти. Но ссылка приспособлена к жизни в среде Microsoft.NET, где часто происходит перемещение динамических переменных с места на место при

очистке и уплотнении данных. При этом указатель не изменится, даже если переменная, на которую он указывал, будет перенесена на другое место. А вот ссылка будет обновлена, так что, где бы ни оказалась динамическая переменная, мы по-прежнему будем иметь доступ к ней.

Далее мы в основном будем говорить про динамические массивы, и может сложиться неверное впечатление, что только массивы могут быть размещены в динамической памяти. На самом деле это не так и существует множество других типов данных, которые размещаются в динамической памяти: списки, множества, стеки, очереди, деревья и др.

3. Статические и динамические массивы

Как мы уже выяснили, статические массивы размещаются в статической памяти, а динамические — в динамической. Рассмотрим код для создания обычного **статического массива**. Желательно описывать именованный тип массива, поскольку это позволяет использовать такой тип для создания подпрограмм с соответствующими параметрами (рис. 3).

Как мы видим, при работе со статическими массивами динамическая память не используется.

Динамические массивы, напротив, располагаются в динамической памяти. Рассмотрим программный код и фрагмент памяти для него (рис. 4).

Обратим внимание на несколько моментов.

1. *Описание динамического массива (в отличие от статического) не приводит к выделению памяти для него.* Статическая переменная, содержащая ссылку, создается, но содержит значение *nil*, т. е. ведет «в никуда»:

```
var a: array of integer;
```

Если проводить аналогию с приобретением земельного участка, то мы уже завели папку для документов (которую назвали *a*), но она пока пуста. Создание массива, т. е. выделение памяти для него и получение ссылки, производится при помощи команды **new**:

```
a := new integer[5];
```

Эта команда используется для создания любых динамических переменных и выполняет две функции, описанные выше: 1) выделяет в динамической памяти

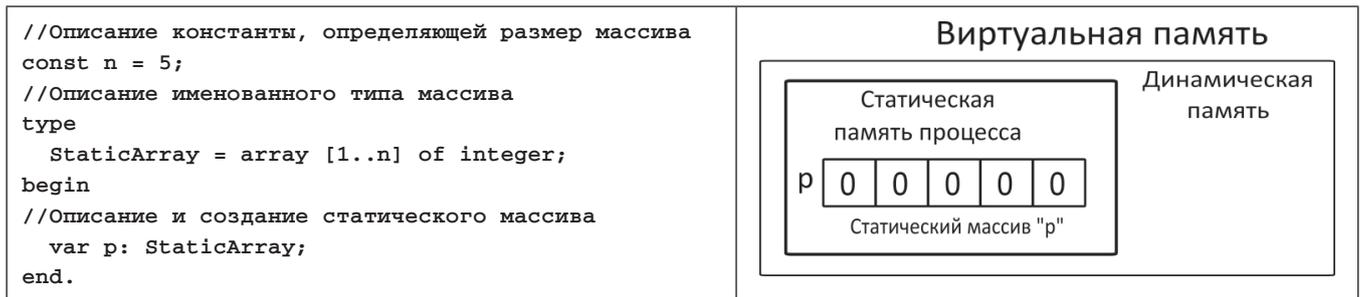


Рис. 3. Описание статического массива. Массив создается в памяти немедленно после описания переменной *p*. Каждая ячейка массива занимает 4 байта (размер переменной типа *integer*). Таким образом, полный размер массива = 4 байта × 5 = 20 байт

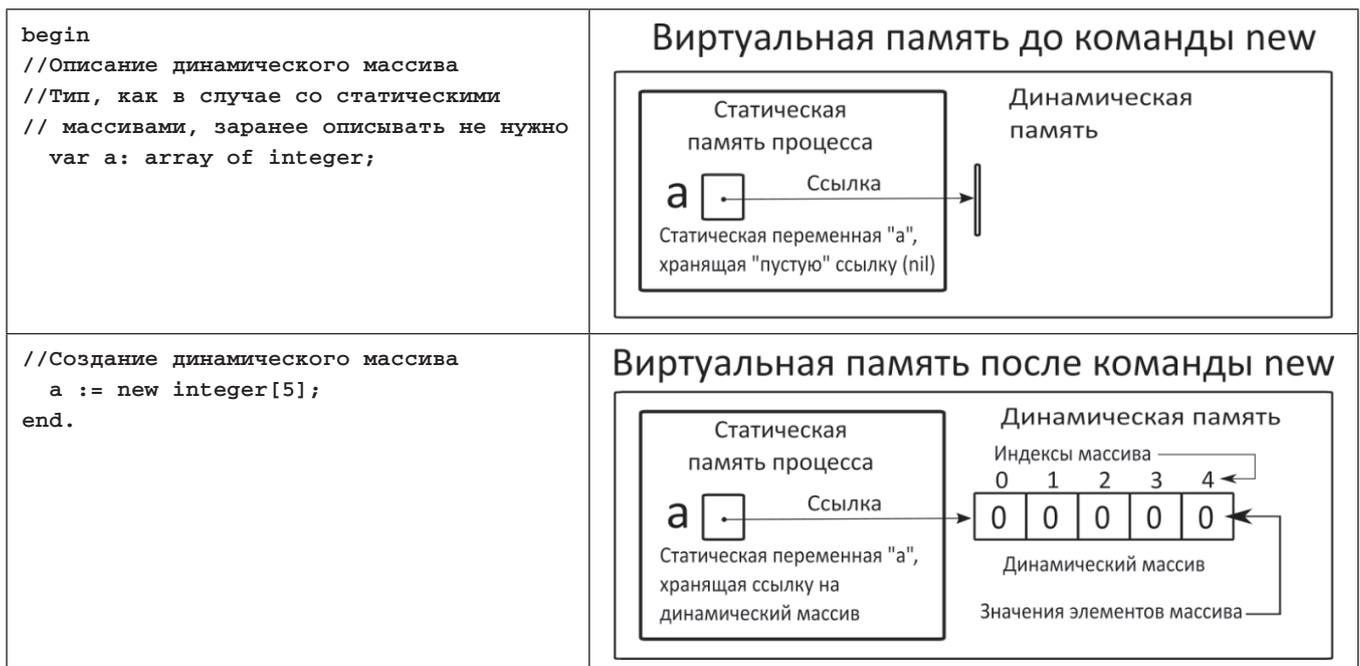


Рис. 4. Описание и создание динамического массива

участок необходимого размера и 2) возвращает ссылку на него. Ссылка посредством оператора присваивания записывается в статическую переменную *a*. Иногда в разговорной речи переменную *a* называют «динамической», что, в общем, неверно: динамическая переменная расположена в другом месте и не имеет имени. Но мы тоже будем говорить «динамический массив *a*», заменяя этой фразой более длинную: «статическая переменная *a*, ссылающаяся на динамический массив». Продолжая нашу аналогию, заметим, что после приведенной выше строки земельный участок нам выделен. Мы также получили географические координаты своего участка (ссылку на него), чтобы легко его отыскать, а сами координаты распечатали, и сложили листы с координатами в ранее заведенную папку *a*.

2. В команде **new** имеются указания на размер памяти, которую необходимо выделить. Действительно, запись «`new integer[5]`» означает, что нам нужен новый (*new*) динамический массив (на что указывают квадратные скобки), содержащий 5 (число в квадратных скобках) элементов целого (*integer*) типа. Выше мы уже отмечали, что на каждый такой элемент отводится по 4 байта, значит, общий размер массива составит 20 байт. Разумеется, вместо целого типа может быть указан любой другой тип. Где же именно расположен массив? А вот этого нам знать не нужно. Важно, что мы легко можем добраться до него при помощи ссылки. Более того, в процессе выполнения программы массив может быть перемещен в другую область памяти, о чем мы даже не узнаем. Но при этом ссылка в переменной *a* будет автоматически обновлена, и все это произойдет незаметно для нас.

3. При создании динамический массив инициализируется значениями по умолчанию. Статические массивы тоже как-то инициализируются, но как именно — может зависеть от языка и компилятора. Динамические массивы в *PascalABC.NET* обязательно будут проинициализированы значениями по умолчанию для

данного типа: числовые — нулями, строковые — пустыми строками и т. д. Поэтому после создания не обязательно производить инициализацию динамического массива, если нас устраивают значения по умолчанию.

4. Индексация элементов динамического массива всегда ведется целыми числами от нуля. И это невозможно изменить. Статические массивы могут иметь и символьные индексы, а нумерация элементов может идти, например, с -10 до $+10$. В некоторых задачах это удобно, но при работе с динамическими массивами такая возможность отсутствует.

5. Доступ к элементам динамического массива осуществляется точно так же, как доступ к элементам статического массива. Мы привыкли, что обращаться к элементу массива можно, указав имя массива и номер элемента в квадратных скобках. Так же мы обращаемся и к элементам динамического массива: `a[3]`. Надо только помнить, что если массив имеет пять элементов, то элемента с номером 5 у него нет (см. рис. 4).

6. Не требуется предварительное описание типа динамического массива. Мы можем описывать параметры-массивы процедур и функций без этого (см. ниже).

7. Все, что сказано о принципах работы с динамическими массивами, справедливо не только для *PascalABC.NET*, но и для других современных языков. Например, в *C#*, *Python* и многих других языках все происходит практически так же, с точностью до синтаксиса.

4. Инициализация, ввод и вывод динамического массива

Выше был приведен код, создающий динамический массив из пяти элементов и инициализирующий его нулями. Есть более удобные и короткие способы выполнить такое действие. Рассмотрим ряд примеров (табл. 1).

Ввод массива с клавиатуры выполняется при помощи функций (табл. 2).

Таблица 1

Код	Пояснение
<code>var a := new integer[5];</code>	// Описание динамического массива можно совмещать с его созданием
<code>var a: array of integer;</code>	// Описание динамического массива
<code>SetLength(a, 10);</code>	// Создание динамического массива длины 10
<code>var b := Arr(5, 7, 9);</code>	// Динамический массив из 3 элементов, заполненный значениями 5, 7 и 9. Размер массива определяется автоматически по количеству параметров функции <i>Arr</i>
<code>var c := Arr(-10..10);</code>	// Динамический массив из 21 элемента, заполненный целыми значениями от -10 до 10
<code>var c1 := Arr('a'..'z');</code>	// Динамический массив из символов (<i>char</i>) от 'a' до 'z'
<code>var d := 5, 7, 9 ;</code>	// Аналогично <i>Arr(5, 7, 9)</i>
<code>var d1 := 7 * 12;</code>	// Массив из 12 семерок. «Питоновский» стиль
<code>var e := ArrFill(7, 3);</code>	// Динамический массив из 7 элементов, каждый элемент равен 3
<code>var f := ArrRandom;</code>	// Динамический массив из 10 случайных целых элементов в диапазоне $[0; 100]$. Значения параметров берутся по умолчанию
<code>f := ArrRandom(5);</code>	// Динамический массив из 5 случайных целых элементов в диапазоне $[0; 100]$. Можно повторно использовать ссылку; при этом старый массив уничтожится

Код	Пояснение
<code>f := ArrRandom(5, -2, 2);</code>	// Динамический массив из 5 случайных целых элементов в диапазоне [-2; 2]
<code>var g := ArrRandomReal;</code>	// Динамический массив из 10 случайных действительных элементов в диапазоне [0; 10]
<code>g := ArrRandomReal(5);</code>	// Динамический массив из 5 случайных действительных элементов в диапазоне [0; 100]
<code>g := ArrRandomReal(5, -2, 2);</code>	// Динамический массив из 5 случайных действительных элементов в диапазоне [-2; 2]

Таблица 2

Код	Пояснение
<code>var a := ReadArrInteger(10);</code>	// Вводим с клавиатуры в массив 10 целых чисел
<code>var b := ReadArrReal(5);</code>	// Вводим с клавиатуры в массив 5 действительных чисел
<code>var c := ReadArrString(12);</code>	// Вводим с клавиатуры в массив из 12 строк
<code>var d := ReadArrInt64(7);</code>	// Вводим с клавиатуры в массив из 7 целых чисел типа Int64
<code>var e := ReadArrInteger('Введите десять целых элементов массива:', 10);</code>	// Перед размером массива можно указывать приглашение

Печать массива можно выполнять процедурами **Write (Writeln)** и **Print (Println)**, а также методом **Print (Println)**. Первые две процедуры для динамических массивов работают одинаково. Рассмотрим код и пример вывода (табл. 3).

Таблица 3

Код	Результат вывода
<code>var a := ArrRandom(7); Writeln(a);</code>	[11, 24, 81, 41, 72, 2, 92]
<code>Println(a);</code>	[11, 24, 81, 41, 72, 2, 92]
<code>a.Println;</code>	11 24 81 41 72 2 92

Последний вариант наиболее удобен, так как не выводит лишних символов (скобок и запятых). Так следует выводить массивы при сдаче решений задач в системы автоматического тестирования.

5. Особенности ссылочной модели

Работая только в статической памяти, мы привыкаем, что переменная — это именованный программный объект, хранящий некоторое значение. При использовании ссылочной модели, как уже говорилось, хранимое в переменной значение на самом деле является не просто значением, а *ссылкой* на участок памяти, в котором, в свою очередь, размещены наши данные. Рассмотрим пример кода, результат работы и данные в памяти (рис. 5).

В этом примере мы два раза печатаем массив *a*, не изменяя его явно после создания. Тем не менее второй элемент этого массива становится равен 7. Все дело в том, что в строке `var b := a;` мы создаем вторую ссылочную переменную (*b*), которая равна *a*. Это значит, что ссылка, хранящаяся в *a*, скопирована в *b*, и теперь обе переменные ссылаются на одну и ту же динамическую переменную (массив). Другими словами, у созданного динамического массива теперь *два* равноправных имени:

```
begin
  var a := new integer[5];
  Println('Результат:');
  a.Println;
  var b := a;
  b[2] := 7;
  a.Println;
end.
```

Результат:
0 0 0 0 0
0 0 7 0 0

Рис. 5. Две ссылочные переменные указывают на одну и ту же область памяти

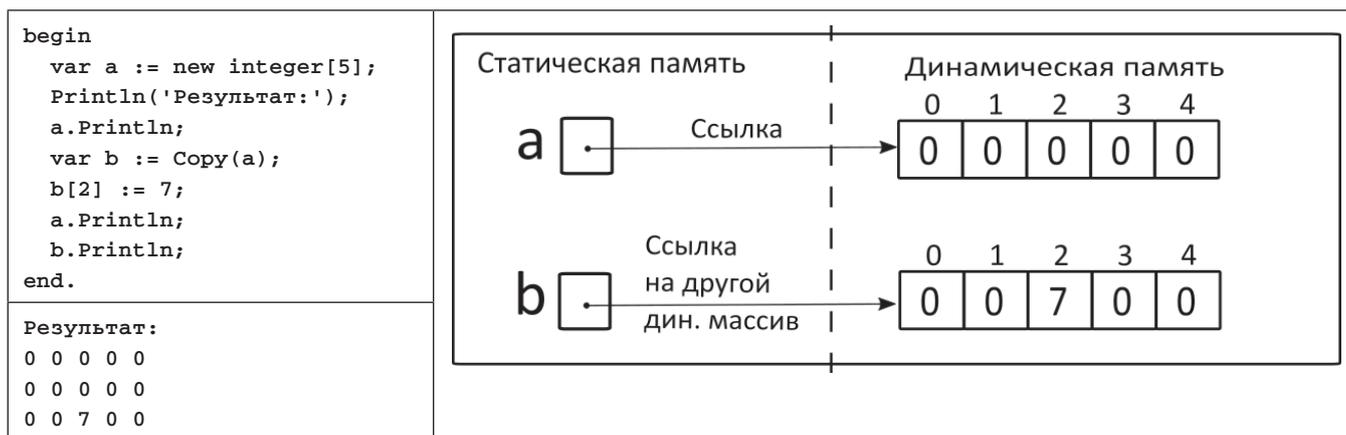


Рис. 6. Две копии одного и того же массива

a и *b*. Именно поэтому, когда меняется значение элемента массива *b*, эти изменения происходят и с массивом *a*. Разумеется, мы можем сделать и больше ссылок на один и тот же массив, как можем создать в папке на диске несколько ярлычков, ссылающихся на один и тот же файл.

Если нам нужны не две ссылки на один и тот же массив, а именно две копии одного и того же массива, то следует применить функцию `Copy` (рис. 6).

Оператор `var b := Copy(a)`; создает копию массива *a* и сохраняет ссылку на эту копию в переменной *b*.

6. Удаление динамических массивов. Сборщик мусора

В Microsoft.NET динамическая память находится под управлением **сборщика мусора** (garbage collector). Эта системная программа становится активной в некоторые, заранее нам неизвестные, моменты времени и выполняет работу по очистке и дефрагментации памяти. **Дефрагментация памяти** — это группировка динамических переменных в одной непрерывной области памяти. Она производится, если в памяти разбросано множество динамических переменных, так что не остается больших участков свободной памяти для новых переменных. Таким образом, сборщик мусора может перемещать динамические переменные по своему желанию, но при этом каждый раз будет автоматически обновлена ссылка на эту переменную.

Прибегать к очистке памяти следует, когда ее не хватает и при этом есть множество ненужных динамических переменных, лишь занимающих место. Сборщик мусора узнает, что переменная не нужна, если на нее нет ни одной ссылки (см. рис. 4, сверху). Таким образом, программист не должен вручную освобождать память, как в языках типа C++. Можно уничтожить динамическую переменную, просто уничтожив ссылку на нее:

```
var a := ArrFill(10, 3); // Новый массив
                          // из 10 троек
a := nil;                // Уничтожаем ссылку
                          // на массив a
```

Заметим, что такая распространенная ошибка, как утечка памяти, невозможна на платформе NET, именно благодаря наличию сборщика мусора.

7. Обработка массивов в цикле

Удобство массивов состоит в том, что мы можем работать с большим количеством значений как с единым целым. Одним из классических способов работы с массивами является обработка их элементов в цикле.

Рассмотрим несколько примеров с комментариями.

7.1. Удвоение каждого элемента в цикле For.

```
// Вместо Begin..End. можно использовать
// последовательность ## в начале программы
## var a := ArrRandom;
a.Println;
// a.High - индекс последнего элемента
// массива
// a.Length - длина (количество элементов)
// массива
// Т.е. a.High = a.Length - 1
for var i := 0 to a.High do
  a[i] *= 2; // То же, что a[i] := a[i] * 2
a.Println;
```

Это классический цикл For, который мы привыкли использовать для работы с массивами. Следует заметить, что динамический массив сам «знает» свой размер. Поэтому нет нужды в заведении дополнительных констант, хранящих эти сведения.

7.2. Нахождение суммы элементов массива в цикле Foreach.

Будем считать, что мы продолжаем писать предыдущую программу.

```
var sum := 0;
foreach var x in a do
  sum += x;
println(sum);
```

Цикл Foreach меньше знаком программистам на Pascal, чем цикл For. Между тем он бывает удобнее в тех случаях, когда не требуется изменения элементов массива, а нужно только перебрать их. В нашем случае мы перебираем все элементы и добавляем их к переменной *sum*. Здесь переменная *x* пробегает не по индексам, а по значениям элементов массива.

7.3. Удвоение каждого элемента в цикле Foreach с использованием свойства Indices.

Как упоминалось выше, в цикле Foreach нельзя изменять элементы массива. Однако у массива есть полезное свойство **Indices** (индексы), которое формально позволяет обойти указанное ограничение.

```
foreach var i in a.Indices do
  a[i] *= 2;
```

Здесь переменная *i* пробегает по всем индексам массива. То есть это просто другой способ записи примера 7.1. На самом деле свойство **Indices** гораздо интереснее, поскольку позволяет указывать в качестве своего параметра лямбда-функцию для отбора нужных элементов. Эта тема выходит за рамки статьи. Для интересующихся читателей приведем следующие примеры (подробнее см. [11]).

7.4. Обработка некоторых элементов с использованием свойства Indices.

```
foreach var i in a.Indices(p -> p in 3..7) do
  a[i] += 1;
```

В этом примере переменная *i* пробегает не по всем индексам массива, а только по индексам 3, 4, 5, 6 и 7. К каждому элементу на этих позициях прибавляется единица. Имя переменной в лямбда-функции (*p*) может быть любым. Читается это так: для всех индексов массива *a*, которые лежат на промежутке 3..7, увеличить соответствующий элемент на единицу.

```
foreach var i in a.Indices((x, p)
-> p.IsEven and (x > 0)) do
  a[i] += 1;
```

Здесь в условии отбора в лямбда-функции мы обрабатываемся как к позиции (*p*) элемента, так и к его значению (*x*). (По соглашению первый параметр лямбды здесь элемент, а второй — его индекс.) Прочитаем код: для всех *положительных* элементов массива *a* ($x > 0$), стоящих на *четных* (*IsEven*) позициях, увеличить такой элемент на единицу.

8. Динамические массивы как параметры и результат работы подпрограмм

Работая со статическими массивами, мы привыкаем писать подпрограммы для массивов определенного и неизменного размера. Динамические массивы более гибкие в этом плане. Поскольку они знают свой размер, отпадает необходимость в заведении констант, хранящих эти данные. А значит, имеется возможность написания функций и процедур, работающих с массивами произвольного размера.

8.1. Передача динамического массива в функцию.

Рассмотрим пример, в котором разработаем функцию для нахождения суммы элементов целочисленного массива.

Пример.

```
function ArrSum(a: array of integer): integer;
begin
  result := 0;
  foreach var x in a do
    result += x;
end;
begin
  var t := ArrRandom;
  t.Println;
  ArrSum(t).Print;
end.
```

В функцию передается динамический массив из целых чисел. Размер массива произволен, значение имеет только тип элементов. Далее обнуляем переменную *result*, через которую происходит возврат значений из функции. В принципе, этого можно и не делать, так как компилятор сам присвоит этой переменной значение по умолчанию (в данном случае ноль) в начале выполнения кода функции, но правила хорошего кода требуют от нас этого действия. Затем в цикле Foreach суммируются все элементы массива, а результат суммирования накапливается в переменной *result*.

В главной программе создается массив *t* из 10 случайных целых чисел. Массив выводится на экран. После этого вызывается функция, которой передается созданный массив, и сразу печатается результат (сумма элементов).

Обратите внимание на отсутствие раздела описания констант, типов и переменных, привычного всем по старому стилю программирования.

8.2. Возвращение динамического массива из функции.

Пример.

Рассмотрим две функции. В первой возьмем натуральное число *n* и создадим массив из чисел от 1 до *n*. Во второй прибавим к каждому элементу массива целое *k*.

```
function ArrMake(n: integer): array of integer;
begin
  result := new integer[n];
  // Или: SetLength(result, n)
  for var i := 0 to result.High do
    result[i] := i + 1;
end;
function ArrPlusK(a: array of integer;
  k: integer): array of integer;
begin
  result := Copy(a);
  for var i := 0 to a.High do
    result[i] += k;
end;
begin
  ArrPlusK(ArrMake(12), 3).Print;
end.
```

Функция *ArrMake* принимает целое *n* и возвращает массив из целых. Вначале создается новый динамический массив из целых и связывается с переменной *result*. Размер массива устанавливается равным параметру *n* функции. Затем в цикле For каждому *i*-му элементу при-

сваивается значение $(i + 1)$, так как нумерация элементов начинается с нуля, а в массиве по условию должны быть натуральные числа.

Функция *ArrPlusK* принимает два параметра: массив из целых и целое число k . Возвращает эта функция массив из целых. Поскольку нужно прибавлять k к каждому элементу массива a , в первой строке тела функции создается копия данного массива, которая связывается с переменной *result*. Затем в цикле For к каждому элементу массива *result* добавляется k .

В главной программе вызывается функция *ArrPlusK*, которой в качестве параметров передается то, что возвращает *ArrMake(12)* (т. е. массив из натуральных чисел 1..12), и число 3. Результат выводится на экран. В данном случае это натуральные числа 4..15.

Обратим особое внимание на первые строки тела функций. Если функция возвращает массив, то первым делом мы должны *создать* его.

Рассмотрим распространенные ошибки новичков. Первая ошибка легко находится компилятором, но для тех, кто не усвоил принципов работы с динамическими переменными, она может оказаться не вполне понятной.

```
function ArrMakeError(n: integer):
  array of integer;
begin
  for var i := 0 to n - 1 do
    result[i] := i + 1;
end;
```

Здесь не создан массив *result*. Он объявлен, но не создан. Поэтому при выполнении тела цикла получаем сообщение об ошибке:

Ошибка времени выполнения: Ссылка на объект не указывает на экземпляр объекта.

Аналогичная ошибка случится, если убрать первую строку в теле функции *ArrPlusK*:

```
function ArrPlusKError(a: array of integer;
  k: integer): array of integer;
begin
  //result := Copy(a);
  for var i := 0 to a.High do
    result[i] += k;
end;
```

Вторую ошибку можно продемонстрировать следующим кодом:

```
function ArrPlusKNotPure(a: array of integer;
  k: integer): array of integer;
begin
  for var i := 0 to a.High do
    a[i] += k;
  result := a;
end;
```

Эта функция будет работать практически так, как ожидалось. По крайней мере, она будет выдавать верный результат. Ошибка, однако, состоит в том, что в процессе своего выполнения эта функция изменит массив, который ей передали в качестве параметра, т. е. массив a . Если вы привыкли к тому, что параметр, переданный по значению (как в функции *ArrPlusKNotPure*), не может измениться вне функции, даже если поменять его

внутри, то с динамическими переменными это не так. Когда в подпрограмму передается динамический массив как параметр-значение, на самом деле передается ссылка на этот массив, что следует из самого устройства динамических массивов. Так что, если в подпрограмме размер динамического массива не меняется, ссылка останется прежней и при выходе из подпрограммы мы заметим, что массив поменялся, если его элементы менялись внутри подпрограммы. Если же нужно изменить размер массива в подпрограмме, то его передача должна происходить по ссылке, т. е. с указанием служебного слова *var*. При этом изменится и сама ссылка на массив.

Таким образом, в последнем случае описана функция, которая не является *чистой*, так как она меняет глобальные переменные. Это плохая практика программирования, и ее следует избегать.

8.3. Динамические массивы в процедурах.

Работа с массивами в процедурах мало отличается от таковой в функциях. Мы требуем от своих учеников выполнения стандартного правила: если параметр является входным, то передаем его по значению, если выходным или входным и выходным — то по ссылке. Это сокращает количество ошибок, о которых мы говорили выше.

Пример.

Написать процедуру, которая добавляет в конец массива сумму его элементов.

Из условия ясно, что размер массива изменится, поэтому передадим его по ссылке. У процедуры один параметр, который будет одновременно и входным, и выходным. Напишем два варианта процедуры: более традиционный и использующий конкатенацию массивов.

```
// Имена в PascalABC.NET могут быть русскими
procedure ДобавитьСумму1(var a: array of real);
begin
  SetLength(a, a.Length + 1);
  a[a.High] := a.Sum;
end;
procedure ДобавитьСумму2(var a: array of real);
begin
  a := a + |a.Sum|;
end;
begin
  var a := |1.0, 2, 3|;
  ДобавитьСумму1(a);
  a.Println;
  ДобавитьСумму2(a);
  a.Print;
end.
```

Процедура *ДобавьСумму1* принимает на вход массив из действительных чисел. Вначале мы увеличиваем длину массива на 1 при помощи процедуры *SetLength*. А затем в последнюю (только что добавленную) ячейку записываем сумму элементов массива. Заметим, что использовать вместо *SetLength* команду *new* неверно, так как она создаст массив со значениями по умолчанию (нулями), и мы потеряем содержимое массива.

Вторая процедура использует возможность конкатенации (склеивания) массивов. Эта операция знакома

многим по работе со строками. В единственной строке к массиву *a* приклеивается массив из одного элемента — суммы элементов данного массива. В контексте обсуждаемой темы полезно рассмотреть, как работает оператор присваивания:

```
a := a + |a.Sum|;
```

Сначала при помощи метода *a.Sum* вычисляется сумма элементов массива и из этого числа создается динамический массив. (Здесь мы забегаем вперед, в раздел 10, чтобы не загромождать код поиском суммы элементов.) Затем массив *a* и только что созданный массив из одного элемента склеиваются (операция конкатенации), так что в динамической памяти появляется новый динамический массив, а именно тот, который нам нужен. Ссылка на этот массив и присваивается переменной *a*. Какое-то время в памяти будут находиться все три массива: исходный *a*, новый *a* и массив из одного элемента. Но, поскольку ссылка есть только на увеличенный массив, он и останется в памяти после работы сборщика мусора. Остальные два массива будут уничтожены.

Код главной программы очевиден. Заметим только, что при создании массива командой `var a := |1.0, 2, 3|` мы были вынуждены хотя бы для одного элемента явно указать действительный тип: 1.0. Если этого не сделать, то при автовыводе типа массива будет установлено, что массив целочисленный (все его элементы — целые). А такой массив нельзя передавать в процедуру, принимающую на вход массив из действительных чисел.

В нашем случае программа выведет:

```
1 2 3 6
1 2 3 6 12
```

Кроме операции конкатенации (склеивания) для массивов определены операции повторения (*) и проверки вхождения элементов (in):

```
Var b := a * 3; // массив b есть массив a,
               // повторенный трижды
print(2 in a); // если число 2 есть в массиве a,
               // возвращается true, иначе false
```

Заметим, что операция in, которая раньше могла использоваться только при работе с множествами, теперь может применяться к любым контейнерам: массивам, спискам, строкам и т. п.

9. Списки List<T>

Говоря про динамические массивы, многие объясняют прилагательное «динамический» тем, что размер такого массива можно изменять в процессе выполнения программы. Последний разобранный нами пример демонстрирует, что изменение размера массива не является хорошей практикой, поскольку в процессе такого изменения создается несколько новых динамических массивов. Конечно, мы не замечаем этого, поскольку сборщик мусора работает очень эффективно. Но если размер массива значителен, да еще его изменение заключено в цикл, то мы можем столкнуться с неприятным замедлением работы программы и системы в целом.

В случае, если необходимо многократно менять размер контейнера данных, лучше использовать не массив,

а какой-то другой тип. В этой статье мы остановимся на самом ближайшем родственнике динамических массивов — на списках (List). Эти списки не следует путать с одно- или двунаправленными списками, построенными из записей или объектов и соединенными при помощи указателей или ссылок.

Если говорить совсем просто, то список — это по способу использования тот же массив, но в его конец можно легко добавлять элементы, не теряя при этом эффективности кода.

Рассмотрим основы работы со списками.

9.1. Описание и создание списка.

```
var a: List<integer>; // Описание списка
                       // из целых
a := new List<integer>; // Создание пустого
                       // списка
a := new List<integer>(12); // Создание списка
                           // из 12 элементов
```

Как обычно, описание и создание можно объединить:

```
var a := new List<integer>;
```

При описании мы указываем вид контейнера данных (List) и тип его элементов в угловых скобках (<integer>). То есть List<integer> семантически эквивалентно List of integer, но второй вариант использовать нельзя. Отличие в синтаксисе обусловлено тем, что массивы являются типами данных Pascal, а списки (а также стеки, очереди, множества на хэш-таблицах и др.) — типами платформы Microsoft.NET, где принята именно такая форма записи. Также можно указать размер списка при его создании.

9.2. Добавление элементов в список.

Пополнение списка элементами осуществляется двумя методами.

```
a.Add(1); // Добавление одного
          // элемента в конец списка
a.Add(2); // Добавление еще одного
          // элемента в конец списка
a.AddRange(5..8); // Добавление чисел из
                 // диапазона в конец списка
a.AddRange(|10, 40, 70|); // Добавление содержимого
                          // жимого массива
```

Результатом работы этих команд будет список: 1 2 5 6 7 8 10 40 70. Обычно операция добавления используется в цикле. Кроме того, для быстрого создания списка можно (по аналогии с командой Arr у массивов) воспользоваться командой Lst:

```
var a := Lst(3, 7, 5)
```

Пример.

Написать функцию, которая принимает на вход массив целых и возвращает список положительных элементов массива.

Сначала необходимо понять, почему возвращать надо не массив, а именно список. Заранее неизвестно, сколько положительных элементов будет в массиве. Какой размер выходного массива можно было бы указать? Неизвестно.

Пришлось бы многократно увеличивать размер выходного массива при обнаружении очередного положительного числа. Как было показано ранее, это не очень эффективно. Посмотрим, как в этой ситуации поможет список.

```
function Положительные(a: array of integer):
  List<integer>;
begin
  result := new List<integer>;
  foreach var x in a do
    if (x > 0) then
      result.Add(x);
end;
begin
  var t := ArrRandom(10, -20, 20);
  t.Println;
  var p := Положительные(t);
  p.Println;
end.
```

Как следует из заголовка, функция принимает на вход массив целых и возвращает список из целых. В первой строке тела функции создается динамическая переменная-список, связанная с *result*. Пока этот список пустой. Далее в цикле *ForEach* совершается проход по массиву, и каждый элемент *x* проверяется на соответствие условию ($x > 0$). Подходящие элементы добавляются в *result*.

В главной программе строится массив *t* из 10 случайных целых, лежащих на отрезке $[-20; 20]$, который передается в функцию. Результат работы функции записывается в переменную *p*. Тип этой переменной — *List<integer>*.

Как изменилась бы программа, если бы в функцию передавался не массив, а список? В заголовке функции надо было бы заменить *a: array of integer* на *a: List<integer>*. В главной программе нужно было бы сделать *t* не массивом, а списком. В языке наибольшее количество вспомогательных подпрограмм разработано для массивов, для списков их гораздо меньше. Например, нет функции создания случайного списка. В этом случае можно сделать массив, а затем преобразовать его в список. Взаимные преобразования из

массива в список и обратно осуществляются методами *ToList* и *ToArray* соответственно. То есть при создании списка надо написать так:

```
var t := ArrRandom(10, -20, 20).ToList;
```

Еще одним отличием списков от массивов является общепринятый для Microsoft.NET способ определения длины контейнера. Это не *Length*, как принято в Pascal, а *Count*. Таким образом, если нужен цикл *For* по списку *p*, пишут так:

```
for var i := 0 to p.Count - 1 do ...
```

В остальном все, что говорилось выше о массивах, справедливо и для списков.

10. Некоторые методы массивов и списков

Методом называют подпрограмму, описанную внутри класса и реализующую функциональность класса. Здесь, чтобы не уходить глубоко в теорию объектно-ориентированного программирования, мы будем называть методами подпрограммы, которые можно вызывать «через точку»: **ИмяПеременной.ИмяМетода**. Не будем делать различий между методами класса и методами расширения. Кроме того, рассмотрим некоторые методы последовательностей, которые широко используются при работе с массивами и списками. В этом изложении нам также придется опустить те методы, которые имеют параметры-лямбды, так как последние не рассматривались в статье. Далее запись: **a.IndexMax(start := 0)** означает, что метод *IndexMax* может быть вызван с параметром, либо параметр можно опустить. В последнем случае (т. е. по умолчанию) значение параметра *start* будет принято равным нулю.

В таблице 4 представлено несколько примеров. Пусть *a* — динамический массив или список из чисел.

Нам нередко приходилось вступать в дискуссию с преподавателями, утверждавшими, что наличие такого огромного арсенала методов отучает учащихся думать,

Таблица 4

Код	Пояснение
a.Sum	Возвращает сумму элементов
a.Product	Возвращает произведение элементов
a.CountOf(x)	Возвращает количество элементов <i>x</i> в последовательности
a.PrintLines	Печатает все элементы, причем каждый с новой строки. Удобно, когда элемент имеет структурный тип (строки, записи и т. п.)
a.High	Возвращает индекс последнего элемента. Только для массивов
a.Max	Возвращает максимальный элемент
a.Min	Возвращает минимальный элемент
a.IndexMax(start := 0)	Возвращает индекс первого максимального элемента. Поиск ведется с позиции <i>start</i> . Заметим, что максимум может встречаться в массиве несколько раз
a.IndexMin(start := 0)	Возвращает индекс первого минимального элемента. Поиск ведется с позиции <i>start</i>
a.LastIndexMax()	Возвращает индекс последнего максимального элемента.

Окончание табл. 4

Код	Пояснение
<code>a.LastIndexMin()</code>	Возвращает индекс последнего минимального элемента
<code>a.LastIndexMax(stop)</code>	Возвращает индекс последнего максимального элемента в диапазоне $[0; stop-1]$
<code>a.LastIndexMin(stop)</code>	Возвращает индекс последнего минимального элемента в диапазоне $[0; stop-1]$
<code>a.IndexOf(x)</code>	Возвращает индекс первого вхождения элемента x в массив, или (-1) , если элемент не найден
<code>a.IndexOf(x, start)</code>	Возвращает индекс первого вхождения элемента x в массив, или (-1) , если элемент не найден. Поиск ведется с позиции <i>start</i>
<code>a.LastIndexOf(x)</code>	Возвращает индекс последнего вхождения элемента x в массив, или (-1) , если элемент не найден
<code>a.LastIndexOf(x, stop)</code>	Возвращает индекс последнего вхождения элемента x в массив, или (-1) , если элемент не найден. Поиск ведется до позиции <i>stop</i>
<code>a.RandomElement</code>	Возвращает один случайно выбранный элемент
<code>a.Shuffle</code>	Возвращает массив, в котором все элементы перемешаны случайным образом
<code>a.Sort</code>	Сортирует массив по возрастанию значения элементов. Сортировка по ключу возможна с использованием лямбда-функций
<code>a.SortDescending</code>	Сортирует массив по убыванию значения элементов
<code>a.Indices</code>	Возвращает последовательность индексов массива (см. раздел 7)
<code>a.Insert(p, x)</code>	<i>Для списков:</i> вставляет элемент x в список на позицию p . Позиция должна лежать в диапазоне $[0; a.Length]$
<code>a.Remove(x)</code>	<i>Для списков:</i> удаляет первое вхождение элемента x в список. Возвращает True, если удаление произошло успешно, и False в противном случае
<code>a.RemoveAt(pos)</code>	<i>Для списков:</i> удаляет элемент на позиции pos . Если позиция выходит за границы списка, возникает ошибка

готовит программистов, которые хотят иметь отдельную кнопку для решения любой задачи. Мы считаем, что это глубоко ошибочное утверждение. Если ученик напишет 10 раз алгоритм нахождения максимума в массиве, он не станет лучше думать, он просто выучит алгоритм. А когда он выучил, не нужно писать этот алгоритм в 11-й раз, можно воспользоваться готовыми методами. Мы опираемся в обучении на следующий принцип: вначале изучение алгоритма и его реализация, потом — использование готовой подпрограммы. Это позволяет работать на более высоком уровне абстракции, думать более «крупными блоками» и не загромождать код тривиальными алгоритмами, что особенно ценно при решении масштабных задач, например, в проектной деятельности.

11. Срезы

Срезом (slice) называется некоторая (не обязательно непрерывная) подпоследовательность исходной последовательности. Использование срезов в программировании не является чем-то новомодным. Впервые срезы появились в языке Fortran 66 (1966 год) [10]. С тех пор срезы являются традиционным способом работы с массивами, строками и другими последовательностями. Синтаксис срезов в различных языках может существенно отличаться, но семантика в целом одинакова: возьми элементы с такого-то по такой-то с таким-то шагом. Синтаксис

срезов PascalABC.NET больше всего похож на таковой в Python. Общий формат следующий:

```
a[from:to:step]
```

что означает: выбрать из массива a все элементы с позиции *from* (включая) до позиции *to* (не включая) с шагом *step*. Обратим еще раз внимание на то, что *правая граница не включается*. Заметим также, что шаг среза можно не указывать: по умолчанию он равен единице. После имени переменной (перед открывающей квадратной скобкой) может стоять символ вопроса:

```
a?[from:to:step]
```

Это будет означать *безопасный срез*, при котором выход за границы диапазона индексов не вызывает ошибку, а в срезе останутся только допустимые значения индексов. Рассмотрим ряд примеров.

11.1. Срезы в направлении увеличения индексов (табл. 5).

11.2. Срезы в направлении уменьшения индексов (табл. 6).

Обратите внимание, что теперь правая граница среза меньше левой. Поэтому именно меньшая (она теперь стала правой) граница исключается из диапазона. Последний пример очень показателен: так можно развернуть массив от конца к началу.

Таблица 5

Код	Результат работы и пояснение
<code>## var a := 10, 1, 2, 3, 4, 5, 6, 7, 8, 9 ;</code>	Создаем массив из 10 элементов. Номер элемента совпадает с его значением
<code>a[2:7].Println;</code>	2 3 4 5 6
<code>a[5:].Println;</code>	5 6 7 8 9 Можно не указывать правую границу. При этом будет считаться, что мы берем элементы до последнего <i>включительно</i>
<code>a[:5].Println;</code>	0 1 2 3 4 Можно не указывать левую границу. При этом будет считаться, что мы берем все элементы с нулевого
<code>a[2:70].Println;</code>	Ошибка времени выполнения: Параметр <i>to</i> за пределами диапазона
<code>a[20:7].Println;</code>	Ошибка времени выполнения: Параметр <i>from</i> за пределами диапазона
<code>Println(a?[2:70]);</code>	[2,3,4,5,6,7,8,9] Ошибки нет. Справа ограничиваемся последним индексом
<code>Println(a?[20:7]);</code>	[] Ошибки нет, но возвращен пустой массив
<code>a[2:8:2].Println;</code>	2 4 6
<code>a[:].Println;</code>	0 1 2 3 4 5 6 7 8 9 Весь массив. То же, что просто <i>a</i>

Таблица 6

Код	Результат работы
<code>Println(a[7:2]);</code>	[]
<code>a[7:2:-1].Println;</code>	7 6 5 4 3
<code>a[:2:-2].Println;</code>	9 7 5 3
<code>a[7::-2].Println;</code>	7 5 3 1
<code>a[:: -1].Println;</code>	9 8 7 6 5 4 3 2 1 0

11.3. Срезы с нумерацией от конца массива.

Мы привыкли нумеровать элементы массива слева направо, от нуля и далее. В PascalABC.NET можно нумеровать элементы справа налево. В этом случае *нумерация будет вестись с единицы, а не с нуля*. Для этого перед индексом нужно поставить знак «^» (крышечка).

Примеры приведены в таблице 7.

11.4. Использование срезов при решении задач.

В [12] предлагается интересный способ объяснения того, почему при определении среза мы берем интервалы, открытые справа. Нужно представить себе, что индексы в срезе нумеруют не элементы массива, а «перегородки» между элементами (позиции среза). Двоеточия тут обозначают крайние положения. На примере строки

Таблица 7

Код	Результат работы и пояснение
<code>a[^2].Println;</code>	8 Обращаемся ко второму элементу с конца
<code>a[a.Length - 2].Println;</code>	8 То же, но более длинная запись
<code>a[^7:^3].Println;</code>	3 4 5 6
<code>a[:^3].Println;</code>	0 1 2 3 4 5 6
<code>a[^4:].Println;</code>	6 7 8 9
<code>a[^1:1:-2].Println;</code>	9 7 5 3

Индексы:	0	1	2	3	4	5	6	7	8										
Позиция среза:	:	1	2	3	4	5	6	7	8	:									
	+-----+																		
		P		a		s		c		a		l		A		B		C	
	+-----+																		
Индексы с конца:		^9		^8		^7		^6		^5		^4		^3		^2		^1	
Позиция среза:	:		^8		^7		^6		^5		^4		^3		^2		^1	:	

Рис. 7. Срезы

`s := "PascalABC"` это выглядит так, как показано на рисунке 7.

Таким образом, срез `s[2:7]` выдаст подстроку «scalA», т. е. с позиции среза 2 до позиции среза 7.

Использование срезов может значительно сократить код программы, но при этом потребуются дополнительная память для хранения среза и время для его создания в памяти. Рассмотрим пример.

Пример.

Написать функцию, которая принимает на вход массив, число и позицию внутри массива и возвращает новый массив, в котором данное число вставлено на данную позицию.

То есть нам нужно вставить элемент в массив на заданную позицию. Конечно, решить эту задачу можно и без помощи срезов. Придется увеличить размер массива на единицу, сдвинуть часть элементов вправо, чтобы освободить место для данного элемента и записать его на освободившееся место. Но мы напишем функцию с использованием срезов:

```
function Вставка(t: array of integer; x, p: integer): array of integer;
```

```
begin
  result := t[:p] + |x| + t[p:];
end;
begin
  var a := |1, 2, 3, 4, 5|;
  a := Вставка(a, 100, 2);
  a.Print; // 1 2 100 3 4 5
end.
```

К срезу от начала массива до позиции p (не включая) прибавили (подклеили) массив из данного элемента, а справа еще подклеили массив с позиции p (включая) до конца. Это же можно записать и в виде короткой функции:

```
function Вставка1(t: array of integer;
  x, p: integer) := t[:p] + |x| + t[p:];
```

Здесь важно не только заметное сокращение кода, но и то, что выполняемые действия ясны и доступны даже начинающему. Получить тот же результат с использованием цикла сложнее.

Правда, эта функция работает неверно для одного из граничных случаев, а именно, когда мы пытаемся вставить элемент в самый конец массива:

```
a := Вставка1(a, 100, 5); // Ошибка времени
// выполнения: Параметр from
// за пределами диапазона
```

Действительно, посмотрим, какие массивы будут склеиваться в этом случае:

```
t[:5] + |100| + t[5:]
```

Первый срез вполне нормальный: он задает весь массив. А вот второй срез пытается взять элементы с 5-го и далее. Но элемента с индексом 5 в массиве нет, отсюда и ошибка. Чтобы ее исправить, нужно воспользоваться безопасным срезом (добавляем знак вопроса после имени массива). Окончательно функция примет такой вид:

```
function Вставка2(t: array of integer;
  x, p: integer) := t[:p] + |x| + t?[p:];
```

На этом же примере становится ясно, почему при срезах используют полуоткрытые интервалы: не нужно задумываться, где надо писать $p + 1$ или $p - 1$. Мы пишем только p , и в одном случае соответствующий элемент будет взят, а в другом — нет.

Надо, однако, понимать, что в некоторый момент времени при работе этой функции в памяти будут существовать одновременно пять массивов: исходный массив, два среза, массив из одного вставляемого элемента и склеенный массив-результат. То, что на них будет потрачена память, — это полбеды. Самое неприятное, что эти массивы надо *создавать* в памяти, а это требует времени, особенно если исходный массив велик. Об этом следует помнить, применяя срезы, и стараться не выполнять подобные операции над большими массивами в цикле.

12. Заключение

История использования обычных статических массивов в школе уходит корнями в тридцатилетнее прошлое. Наблюдая, как быстро меняются информационные технологии в последние десятилетия, мы с удивлением видим в школьных программах нечто действительно незбылемое, поистине соответствующее своему названию, — статические массивы. В этой статье мы попытались показать, что переход к изучению динамических массивов не сильно усложнит задачу учителя. Зато ученики получают в руки гибкий и удобный инструмент со множеством встроенных методов, а также смогут приобрести навыки современного программирования. Тем, кто впоследствии решит освоить Python, C#, C++ (STL), навыки работы с динамическими массивами будут очень полезны, а изучение второго языка пройдет гораздо легче. Потому что действительно важно не то, какой язык выучил школьник, а какие концепции программирования он усвоил и насколько быстро сможет перейти на другой язык, если ему это понадобится.

Список источников

1. *Босова Л. И., Босова А. Ю.* Информатика. 9 класс: учебник. 6-е изд. М.: БИНОМ. Лаборатория знаний, 2016. 208 с.
2. *Галовиц Я.* C++17. Стандартная библиотека шаблонов. СПб.: Питер, 2018. 432 с.
3. *Калинин И. А., Самылкина Н. Н.* Информатика. 10 класс. Углубленный уровень: учебник. М.: БИНОМ. Лаборатория знаний, 2019. 256 с.
4. *Михалкович С. С., Дженджер В. О.* Новые примитивы в школьном программировании // Преподавание информационных технологий в Российской Федерации: сборник научных трудов. Материалы Восемнадцатой открытой всероссийской конференции (Москва, онлайн, 14–15 мая 2020 г.) / отв. ред. А. В. Альминдеров. М., 2020. С. 333–334. https://it-education.ru/conf2020/thesis/IT-education_in_Russia_Thesis_2020_preview.pdf
5. *Осипов А.* PascalABC.NET. Введение в современное программирование. Ростов-на-Дону, 2019. 572 с. <http://pascalabc.net/downloads/OsipovBook/ModernProgr.pdf>
6. *Поляков К. Ю., Еремин Е. А.* Информатика. 10 класс. Базовый и углубленный уровни: учебник: в 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2020. 352 с.
7. *Седжвик Р., Уэйн К., Дондеро Р.* Программирование на языке Python: учебный курс / пер. с англ. СПб.: Альфа-книга, 2017. 736 с.
8. *Семакин И. Г., Шеина Т. Ю., Шестакова Л. В.* Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2014. 184 с.
9. Система программирования PascalABC.NET. <http://pascalabc.net/>
10. Array slicing // Wikipedia. https://en.wikipedia.org/wiki/Array_slicing
11. PascalABC.NET: статьи, примеры. Программы и алгоритмы с использованием массивов. https://pascalabcnet.github.io/progr_arrays.html
12. The Python Tutorial. An Informal Introduction to Python. <https://docs.python.org/3/tutorial/introduction.html#strings>



DOI: 10.32517/2221-1993-2022-21-1-81-84

А. М. Магомедов, Н. Ш. Раджабова

Дагестанский государственный университет, г. Махачкала, Республика Дагестан, Россия

ЗАМОЩЕНИЕ КЛЕТЧАТОЙ ПОЛОСЫ ШИРИНОЙ 4*

Аннотация

Рассматривается проблема подсчета числа всевозможных покрытий клетчатого прямоугольника $n \times m$ (n строк, m столбцов) плитками размером 1×2 без пропусков и наложений. Поскольку при нечетных n и m покрытий не существует, предполагается, что хотя бы один из этих двух параметров имеет четное значение. Предлагается решение задачи о подсчете количества всевозможных покрытий плитками размером 1×2 полосы клетчатой бумаги с шириной 4 и заданной высотой.

Приведен краткий обзор известных подходов к решению задачи определения числа всевозможных покрытий a_n при $m = 2$ и $m = 3$. В данной статье получена рекуррентная формула для вычисления a_n в случае $m = 4$.

Для этого предлагается метод начальной клетки для укладки плиток, на основе которого построена система из трех рекуррентных формул для вычисления a_n при $n \geq 3$ и $m = 4$. Затем из системы выведена одна рекуррентная формула для a_n .

Ключевые слова: задача «Паркет», замощение костями домино, количество покрытий, рекуррентное соотношение.

1. Задача подсчета покрытий прямоугольника

Рассмотрим задачу подсчета числа всевозможных покрытий клетчатого прямоугольника $n \times m$ (n строк, m столбцов) плитками размером 1×2 без пропусков и наложений. Когда значение m однозначно определяется из контекста, число всевозможных покрытий будем обозначать через a_n . Поскольку при нечетных n и m покрытий не существует, предполагается, что хотя бы один из этих двух параметров имеет четное значение.

Здесь термин *покрытие* сходен по значению со словами *паркетаж*, *разбиение* и *замощение* [9, с. 16] (см. также [3], задача 20 на с. 53). Впервые задача исследована в [2, 8], где показано, что для вычисления a_n в случае $m = 2$ достаточно одной рекуррентной формулы, в случае

$m = 3$ удается построить систему из двух рекуррентных формул. Для небольших n и m задача была предложена на VI Всероссийской олимпиаде школьников по информатике [1, 5].

При малых n и m число всевозможных покрытий может быть найдено методом динамического программирования [1, 4]. В [10] найдена формула для a_n как функции от n при $0 \leq n \leq 8$. В [6] представлен способ вычисления коэффициентов рекуррентной формулы, использующей для нахождения a_n (количества разбиений при фиксированном значении параметра m) лишь операции сложения и умножения целых чисел. Исследование задачи замощения продолжает вызывать интерес (см. [4, 6, 7, 11] и др.).

В случае $m = 2$ (см., например, [1, с. 52–54]) укладка плиток у верхней границы прямоугольника соответству-

* Работа выполнена при поддержке Отдела математики и информатики ДФИЦ РАН.

Контактная информация

Магомедов Абдулкарим Магомедович, доктор физ.-мат. наук, профессор, зав. кафедрой дискретной математики и информатики, Дагестанский государственный университет, г. Махачкала, Республика Дагестан, Россия; *адрес:* 367000, Россия, Республика Дагестан, г. Махачкала, ул. Гаджиева, д. 43-а; *e-mail:* magomedtagir1@yandex.ru

Раджабова Наима Шамильевна, канд. физ.-мат. наук, доцент кафедры дискретной математики и информатики, Дагестанский государственный университет, г. Махачкала, Республика Дагестан, Россия; *адрес:* 367000, Россия, Республика Дагестан, г. Махачкала, ул. Гаджиева, д. 43-а; *e-mail:* naimasha@gmail.com

A. M. Magomedov, N. Sh. Radzhabova

Dagestan State University, Makhachkala, The Republic of Dagestan, Russia

TILING A CHECKERED STRIP WITH A WIDTH OF 4

Abstract

The problem of calculating the number of all possible coverings of a checkered rectangle $n \times m$ (n rows, m columns) by tiles 1×2 without gaps and overlaps is considered. Since there are no coverings for odd n and m , it is assumed that at least one of these two parameters has an even value. A solution to the problem of calculating the number of various coverings with tiles of sizes 1×2 strips of checkered paper with a width of 4 and a given height is proposed.

A brief overview of known approaches to solving the problem of determining the number of all possible coverings a_n for $m = 2$ and $m = 3$ is given. In this article, a recursive formula is obtained for calculating a_n in the case $m = 4$.

For this, a method of the initial cell for laying tiles is proposed, on the basis of which a system of three recurrent formulas for calculating a_n for $n \geq 3$ and $m = 4$ is built. Then one recurrent formula for a_n is deduced from the system.

Keywords: task "Parquet", paving with dominoes, number of coverings, recurrent ratio.

ет рисункам 1, a или b . Число способов продолжения каждого из этих двух вариантов равно соответственно a_{n-1} и a_{n-2} , поэтому:

$$a_n = a_{n-1} + a_{n-2}. \quad (1)$$

С учетом очевидных равенств $a_1 = 1, a_2 = 2$ получим числа Фибоначчи: 1, 2, 3, 5, 8,

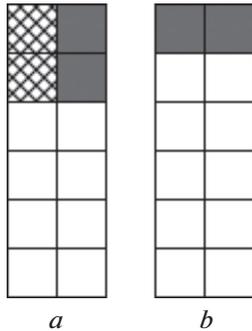


Рис. 1. При $m = 2$ покрытие верхней левой клетки горизонтальной или вертикальной плиткой соответствует слагаемым рекуррентной формулы (1)

Для случая $m = 3$ задача рассмотрена, например, в [7], где, в частности, получена рекуррентная формула:

$$a_n = a_{n-2} + a_{n-4},$$

где n — четное число.

В данной статье будет получена рекуррентная формула для вычисления a_n в случае $m = 4$.

2. Система рекуррентных формул

В этом разделе мы построим систему из трех формул для вычисления a_n в случае $m = 4$.

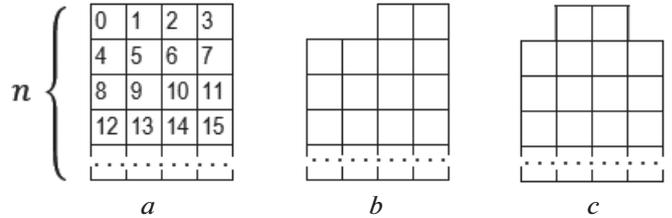


Рис. 2. Высоты фигур равны n

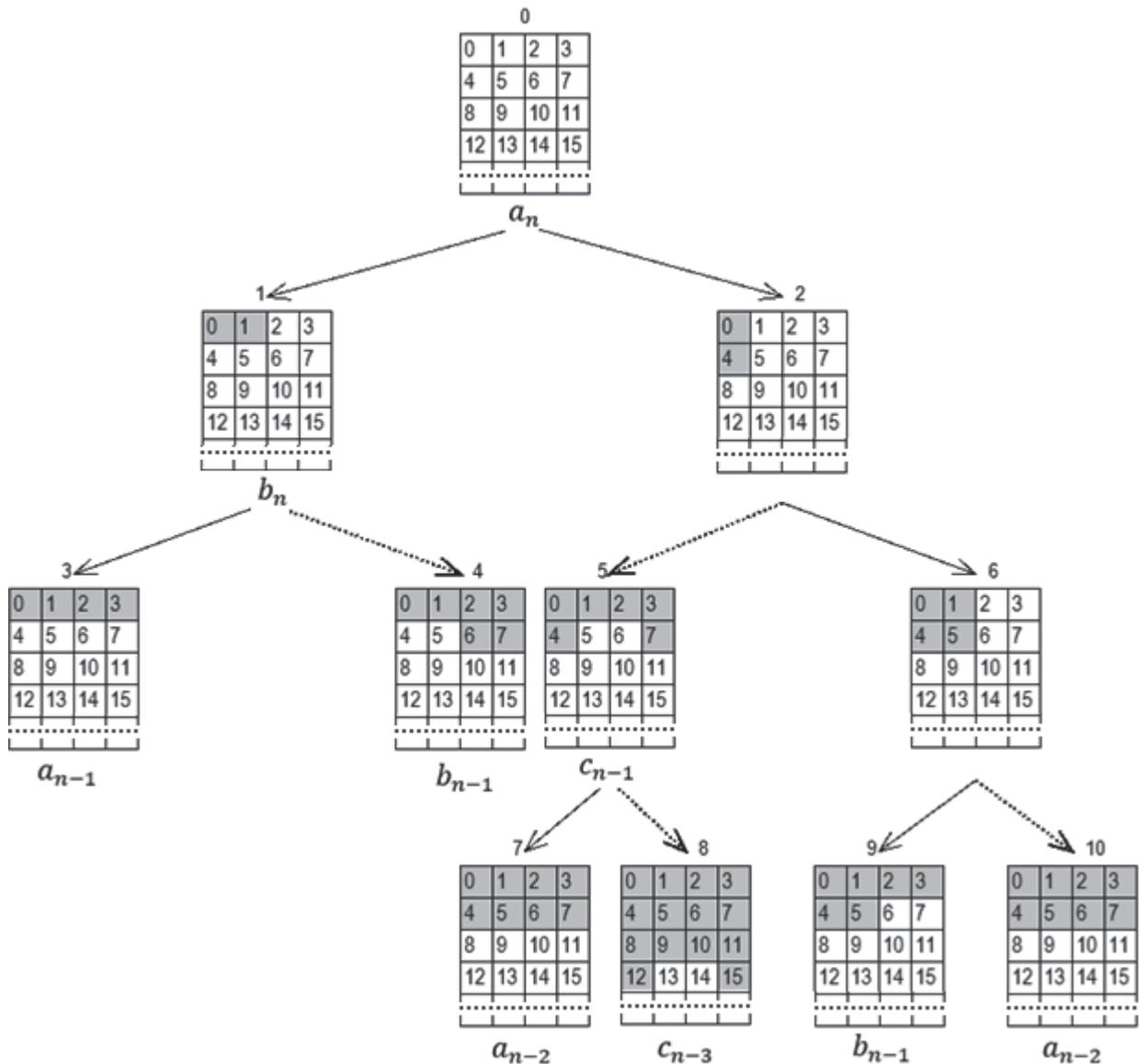


Рис. 3. Для каждой фигуры $ч$. с.п. равно сумме $ч$. с.п. двух фигур, к которым от данной фигуры проведены стрелки

Упорядочим клетки прямоугольника так, как показано на рисунке 2, а. Множество непокрытых пока клеток будем называть *фигурой*, из них клетку с наименьшим номером — *начальной клеткой*.

Обозначим число способов покрытия (далее кратко: ч. с.п.) фигур с высотой n , приведенных на рисунках 2, а, 2, b и 2, c , через a_n , b_n и c_n соответственно. Будем укладывать плитки на клетчатый прямоугольник таким образом, чтобы очередная плитка покрыла начальную клетку. Такой способ укладки плиток будем называть *методом начальной клетки*.

Начальную клетку исходного прямоугольника, отмеченного на рисунке 3 номером 0, можно покрыть как «горизонтальной» (0–1), так и «вертикальной» (0–4) плиткой. В результате непокрытыми останутся фигуры с номерами 1 и 2 соответственно. Если начальную клетку 2 фигуры 1 покрыть горизонтальной плиткой 2–3, получим фигуру с номером 3; в фигуре, полученной покрытием этой же клетки вертикальной плиткой 2–6, начальной клеткой будет клетка 3, а единственно допустимым продолжением — укладка вертикальной плитки 3–7, что приводит к фигуре 4. Таким образом, ч. с.п. фигуры 1 равно сумме ч. с.п. фигур 3 и 4:

$$b_n = a_{n-1} + b_{n-1}$$

или, в более удобном для дальнейшего использования виде:

$$a_{n-1} = b_n - b_{n-1}. \quad (2)$$

Рассмотрим на рисунке 3 фигуру с номером 5 и ч. с.п., равным c_{n-1} . Ее начальная клетка 5 допускает покрытие как горизонтальной плиткой 5–6 (результат — фигура с номером 7 и числом способов покрытия a_{n-2}), так и вертикальной плиткой 5–9. Начальная клетка 6 фигуры F_1 , полученной во втором случае, допускает только покрытие вертикальной плиткой 6–10. У полученной в результате фигуры F_2 начальной клеткой будет клетка с номером 8, которая также допускает единственное покрытие (плиткой 8–12), а вслед за ней однозначно будет положена вертикальная плитка 11–15. В результате мы получили фигуру, размещенную на рисунке 3 под номером 8, число способов покрытия которой равно c_{n-3} . Таким образом, ч. с.п. фигуры с номером 5 равно сумме ч. с.п. фигуры с номером 7 и фигуры с номером 8:

$$c_{n-1} = a_{n-2} + c_{n-3}. \quad (3)$$

Приступим к выводу третьей формулы. Искомое число a_n для фигуры 0 равно сумме ч. с.п. фигур с номерами 1 (число b_n) и 2. Ч.с. п. фигуры с номером 2 равно сумме ч. с.п. фигур с номерами 5 (число c_{n-1}) и 6. Наконец, ч. с.п. фигуры 6, как легко убедиться, равно сумме чисел способов покрытий фигур 9 и 10, то есть $b_{n-1} + a_{n-2}$.

Получили формулу:

$$a_n = a_{n-2} + b_n + b_{n-1} + c_{n-1}. \quad (4)$$

Запишем полученные формулы (2), (3) и (4) в виде системы:

$$\begin{cases} a_{n-1} = b_n - b_{n-1}, \\ c_{n-1} = a_{n-2} + c_{n-3}, \\ a_n = a_{n-2} + b_n + b_{n-1} + c_{n-1}. \end{cases} \quad (5)$$

Начальные значения a_n , b_n , c_n при $n = 1$ и $n = 2$ можно легко вычислить:

$$a_1 = 1, a_2 = 5, b_1 = 1, b_2 = 2, c_1 = 1, c_2 = 1.$$

Используя эти значения и соотношения (5), можно вычислить a_n для $n \geq 3$.

Следует, однако, признать, что необходимость вычислять элементы трех последовательностей — $\{a_n\}$, $\{b_n\}$ и $\{c_n\}$ — несколько снижает ценность полученных соотношений. В следующем разделе мы покажем, как найти одну рекуррентную формулу для a_n .

3. Рекуррентная формула для вычисления a_n

Из формулы (4) выразим c_{n-1} :

$$c_{n-1} = a_n - a_{n-2} - b_n - b_{n-1}.$$

Заменив в этой формуле индекс n на $n - 2$, получим равенство:

$$c_{n-3} = a_{n-2} - a_{n-4} - b_{n-2} - b_{n-3}.$$

Подставив полученные выражения для c_{n-1} и c_{n-3} в формулу (3), получим соотношение:

$$a_n - a_{n-2} - b_n - b_{n-1} = a_{n-2} + a_{n-2} - a_{n-4} - b_{n-2} - b_{n-3}.$$

Или, после очевидных упрощений:

$$a_n - 3a_{n-2} + a_{n-4} = b_n + b_{n-1} - b_{n-2} - b_{n-3}. \quad (6)$$

Если преобразовать правую часть (6) к виду

$$(b_n - b_{n-1}) + 2(b_{n-1} - b_{n-2}) + (b_{n-2} - b_{n-3})$$

и заменить заключенные в скобки разности в соответствии с формулой $a_{n-1} = b_n - b_{n-1}$ (см. формулу (2)), то получим равенство:

$$a_n - 3a_{n-2} + a_{n-4} = a_{n-1} + 2a_{n-2} + a_{n-3}.$$

Или окончательно:

$$a_n = a_{n-1} + 5a_{n-2} + a_{n-3} - a_{n-4}. \quad (7)$$

Надо заметить, что для использования соотношения (7) необходимо будет найти четыре первых значения a_1, a_2, a_3 и a_4 . Но сделать это совсем не сложно (для этой цели удобнее всего воспользоваться системой (5)).

4. Заключение

Одним из интересных продолжений представляется решение линейных рекуррентных соотношений, подобных (7), т. е. поиск формулы для a_n как функции от n . Но это уже совсем другая история...

Список источников

1. Волченков С. Г. Задача «Паркет» // Информатика и образование. 1994. № 3. С. 52–54.
2. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. М.: Мир, 1998. 703 с.
3. Канель-Белов А. Я., Ковальджи А. К. Как решают нестандартные задачи / под ред. В. О. Бугаенко. 4-е изд., стереотип. М.: МЦНМО, 2008. 96 с.

4. *Караваев А. М.* Метод динамического программирования для подсчета замощений домино на прямоугольной решетке и цилиндре // Актуальные проблемы гуманитарных и естественных наук. 2013. № 6 (53). С. 13–18.
5. *Кирюхин В. М.* VI Всероссийская олимпиада школьников по информатике // Информатика и образование. 1994. № 3. С. 47–50.
6. *Магомедов А. М., Магомедов Т. А., Лавренченко С. А.* Взаимно-рекуррентные формулы для перечисления разбиений прямоугольника // Прикладная дискретная математика. 2019. № 46. С. 108–121.
7. *Нурлигареев Х. Д.* Полоски из домино // Наука и общество. Новости науки и техники. <https://ru.sciencenetnews.com/poloski-iz-domino>
8. *Graham R. L., Knuth D. E., Patashnik O.* Concrete Mathematics. Massachusetts: Addison-Wesley, 1994. 657 p.
9. *Grünbaum B., Shephard G. C.* Tilings and Patterns. New York: W. H. Freeman & Co., 1987.
10. *Klarner D., Pollack J.* Domino tilings of rectangles with fixed width // Discrete Mathematics, 1980. Vol. 32. P. 45–52.
11. *Rao M.* Exhaustive search of convex pentagons which tile the plane. <https://perso.ens-lyon.fr/michael.rao/publi/penta.pdf>



ИТОГИ XXII ВСЕРОССИЙСКОГО КОНКУРСА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ И ФОТОГРАФИЙ ФОТО 1-2022

Уважаемые коллеги!

В октябре 2021 года издательство «Образование и Информатика» объявило очередной конкурс цифровых изображений и фотографий для размещения их на обложке журнала «Информатика в школе» первого полугодия 2022 года.

В конкурсе приняли участие школьники, студенты и педагоги из Беларуси, Казахстана, России, Украины.

На конкурс было прислано более тысячи работ, среди них было много интересных, хороших рисунков, однако проверка на плагиат показала, что, к сожалению, почти половина присланных рисунков — это либо явный плагиат, либо перерисовка. Надеемся, что в будущих конкурсах авторы работ и педагоги-руководители более ответственно отнесутся к подготовке конкурсных работ и соблюдению норм публикационной этики.

Благодарим всех конкурсантов за участие и представляем авторов победивших работ.

В номинации «Цифровые изображения» победителями стали:

Авагян Арман Артакович,

*ученик VII класса средней общеобразовательной школы № 26 г. Сочи имени Героя Советского Союза Диброва Кирилла Селиверстовича, Россия
(руководитель работы — Тыщенко Наталья Юрьевна)*

Барышева Дарья Павловна,

*ученица IX класса гимназии № 1, г. Новосибирск, Россия
(руководитель работы — Челак Антонина Александровна)*

Белякова Карина Антоновна,

*учащаяся студии анимации «Рыжая Ворона», ученица III класса лицея № 21, г. Иваново, Россия
(руководитель работы — Гурская Наталья Викторовна)*

Белякова Каролина Антоновна,

*учащаяся студии анимации «Рыжая Ворона», ученица III класса лицея № 21, г. Иваново, Россия
(руководитель работы — Гурская Наталья Викторовна)*

Брошкова Юлия Алексеевна,

*ученица V класса средней общеобразовательной школы № 5 с углубленным изучением отдельных предметов, г. Солнечногорск, Московская область, Россия
(руководитель работы — Пушкиова Ирина Александровна)*

Булатов Степан Матвеевич,

*ученик V класса гимназии № 4 им. А. С. Пушкина, г. Йошкар-Ола, Республика Марий Эл, Россия
(руководитель работы — Мусинская Мария Альбертовна)*

Булатова Анастасия Матвеевна,

*ученица V класса гимназии № 4 им. А. С. Пушкина, г. Йошкар-Ола, Республика Марий Эл, Россия
(руководитель работы — Мусинская Мария Альбертовна)*

Гавло Анастасия Павловна,

*студентка Барнаульского государственного педагогического колледжа, г. Барнаул, Россия
(руководитель работы — Лукьянова Наталия Владимировна)*

Замесина Анна Андреевна,

*ученица VII класса средней общеобразовательной школы № 26 г. Сочи имени Героя Советского Союза Диброва Кирилла Селиверстовича, Россия
(руководитель работы — Тыщенко Наталья Юрьевна)*

Захарова Мария Игоревна,

*учащаяся Дворца детского (юношеского) творчества, ученица VII класса средней общеобразовательной школы № 7, г. Чебоксары, Чувашская Республика, Россия
(руководитель работы — Чопик Марина Степановна)*

Кадырова Рамзия Камилевна,

студентка Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Легоцкая Алина Алексеевна,

ученица VII класса средней общеобразовательной школы № 26 г. Сочи имени Героя Советского Союза Диброва Кирилла Селиверстовича, Россия
(руководитель работы — Тыщенко Наталья Юрьевна)

Максимов Вадим Владиславович,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Немиро Татьяна Викторовна,

ученица VII класса средней общеобразовательной школы № 26 г. Сочи имени Героя Советского Союза Диброва Кирилла Селиверстовича, Россия
(руководитель работы — Тыщенко Наталья Юрьевна)

Победённая Ксения Андреевна,

ученица X класса средней общеобразовательной школы № 2, г. Зеленокумск, Ставропольский край, Россия
(руководитель работы — Букина Елена Юрьевна)

Шафигуллин Фирдус Фидаилевич,

студент Мамадышского политехнического колледжа, г. Мамадыш, Республика Татарстан, Россия
(руководитель работы — Порываева Нафиса Сабитовна)

Швецова Анастасия Александровна,

учащаяся Центра технического творчества, ученица III класса Вятской гуманитарной гимназии с углубленным изучением английского языка, г. Киров, Россия
(руководитель работы — Ренжина Анна Анатольевна)

В номинации «Цифровые фотографии» победителями стали:

Гаевская Вера Алексеевна,

учитель информатики средней школы № 206, г. Минск, Беларусь

Климина Наталья Владимировна,

учитель информатики лицея имени Героя Советского Союза П. И. Викулова, г. Сызрань, Самарская область, Россия

Крекин Михаил Евгеньевич,

учитель информатики школы № 1228 «Лефортово», г. Москва, Россия

Саварцов Михаил Сергеевич,

учитель информатики и математики средней общеобразовательной школы № 50, г. Ставрополь, Ставропольский край, Россия

Скрылев Андрей Алексеевич,

учитель информатики средней общеобразовательной школы № 4, г. Покачи, Ханты-Мансийский автономный округ — Югра, Россия

Рябчикова Ольга Дмитриевна,

педагог дополнительного образования Потьминской средней общеобразовательной школы, п. Потьма, Zubovo-Полянский район, Республика Мордовия, Россия

Все победители конкурса получают (бесплатно):

- диплом от издательства «Образование и Информатика»;
- печатный экземпляр журнала «Информатика в школе» № 1-2022 с итогами конкурса;
- авторский печатный экземпляр журнала, на обложке которого будет опубликована работа.

Уважаемые коллеги! Обращаем ваше внимание, что с 2022 года электронные версии журналов «Информатика и образование» и «Информатика в школе» будут размещаться на сайте издательства в режиме открытого доступа (Open Access), поэтому электронная подписка победителям не оформляется (как это было в предыдущих конкурсах).

Остальные конкурсанты могут получить сертификат об участии, который будет подготовлен по индивидуальному запросу.

Спасибо за внимание и участие в конкурсе!

Следите за информацией о новых конкурсах в журналах «Информатика в школе» и «Информатика и образование», а также на сайте ИНФО: <http://www.infojournal.ru/>

В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов:



Юлия Брошкова



Рамзия Кадырова



В. А. Гаевская



Н. В. Климина



Рамзия Кадырова



Карина Белякова, Каролина Белякова



М. С. Саварцов



Мария Захарова



А. А. Скрылев

Уважаемые авторы и читатели!

Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» во втором полугодии 2022 года.

Подробности — на сайте издательства «Образование и Информатика»: <http://www.infojournal.ru/>

ПОДПИСКА

Журнал «Информатика в школе»

Индекс подписки
на 1-е полугодие 2022 года
(«Урал-Пресс», «АРЗИ» и другие агентства подписки)

81407

Периодичность выхода: 3 номера в полугодие (февраль, апрель, июнь)
Объем — не менее 88 полос

Редакционная стоимость — 500 руб.

ПРАВИЛА ДЛЯ АВТОРОВ

Уважаемые коллеги!

Статьи для публикации в журналах «Информатика и образование» и «Информатика в школе» должны отправляться в редакцию **только через электронную форму на сайте ИНФО (раздел «Авторам → Отправка статьи»):**

<http://infojournal.ru/authors/send-article/>

Обращаем ваше внимание, что для отправки статьи необходимо предварительно зарегистрироваться на сайте ИНФО (или авторизоваться — для зарегистрированных пользователей).

С требованиями к оформлению представляемых для публикации материалов можно ознакомиться на сайте ИНФО в разделе «Авторам»:

<http://infojournal.ru/authors/>

Обратите внимание: требования к оформлению файла рукописи — **разные** для журналов «Информатика и образование» и «Информатика в школе». При подготовке файла рукописи ориентируйтесь на требования для того журнала, в который вы представляете статью. Если вы представляете рукопись в оба журнала (для публикации в одном из изданий — на усмотрение редакции), при ее оформлении следует руководствоваться требованиями к оформлению рукописи в журнал «Информатика и образование».

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

E-mail: readinfo@infojournal.ru

Телефон: (495) 140-19-86

ВСЕРОССИЙСКИЙ КОНКУРС «ШКОЛЬНАЯ ИНФОРМАТИКА: ВЧЕРА, СЕГОДНЯ, ЗАВТРА»

НОМИНАЦИИ КОНКУРСА

1. Номинация для педагогов: Школьная информатика: вчера, сегодня, завтра.
2. Номинация для учащихся: Школьная информатика: заглянем в будущее.

СРОКИ И ЭТАПЫ ПРОВЕДЕНИЯ КОНКУРСА

1. Работы на конкурс принимаются с 5 апреля по 25 мая 2022 года включительно.
2. Итоги конкурса будут подведены в № 3-2022 (июнь) журнала «Информатика в школе», а также опубликованы на сайте издательства «Образование и Информатика».
3. Лучшие работы будут опубликованы в журнале «Информатика в школе».

Заявки на участие в конкурсе принимаются только через заполнение формы на сайте издательства «Образование и Информатика».

К участию могут быть представлены работы как от одного автора, так и от группы авторов. Представленные на конкурс материалы должны быть оригинальными – не опубликованными ранее в печатных или электронных изданиях, в том числе в сети Интернет.

ПОДРОБНАЯ ИНФОРМАЦИЯ

<http://infojournal.ru/competition/informatics-2022/>



24 сентября - 1 октября
международный конгресс
Суперкомпьютерные дни в России 2022

<https://Congress.RussianSCDays.org>

Научные школы:
24.09 - 01.10.2022

Научная конференция:
26.09 - 27.09.2022

Семинары

Выставка

Экскурсии

Новый расширенный формат объединяет научную конференцию, научные школы Суперкомпьютерной академии, серию специализированных научных семинаров, экскурсии в ведущие суперкомпьютерные центры и множество других событий, проводимых на различных площадках Москвы и России.

ТЕМАТИКА мероприятий конгресса — суперкомпьютерные технологии во всем многообразии: параллельные и распределенные вычисления, высокопроизводительные программные и аппаратные решения, масштабируемые алгоритмы, индустриальные суперкомпьютерные решения, большие данные, машинное обучение, суперкомпьютерное образование и многое другое.

АУДИТОРИЯ — российские и зарубежные представители науки, промышленности, бизнеса, образования, государственных органов.

НАУЧНЫЕ ШКОЛЫ СУПЕРКОМПЬЮТЕРНОЙ АКАДЕМИИ — это специализированные мероприятия по актуальным направлениям развития науки и технологий, организуемые и проходящие под руководством известных российских специалистов.

Рабочие дни академии: 24.09 - 01.10.2022

<https://academy.hpc-russia.ru/>

ОДНА НЕДЕЛЯ — МНОЖЕСТВО ПАРАЛЛЕЛЬНЫХ СОБЫТИЙ!

КЛЮЧЕВЫЕ ДАТЫ НАУЧНОЙ КОНФЕРЕНЦИИ

До 15 апреля — представление полных версий работ

15 мая — уведомление о включении работы в программу конференции

30 мая — представление окончательного варианта работы

НАУЧНАЯ КОНФЕРЕНЦИЯ — это множество параллельно идущих секций: выступления мировых лидеров HPC-сообщества, научные и индустриальные секции, постерная секция, конференция молодых ученых. Совещания, круглые столы, живые дискуссии, обмен опытом и инновациями.

Рабочие дни конференции: 26.09 - 27.09.2022

<https://RussianSCDays.org>

РЕГИСТРАЦИЯ
участников
конференции:
с 15 апреля

<https://RussianSCDays.org>

СТАНЬТЕ ЧАСТЬЮ «СУПЕРКОМПЬЮТЕРНЫХ ДНЕЙ В РОССИИ»!

Посетите конференцию и научные школы, узнайте о работе ведущих российских и мировых суперкомпьютерных центров, организуйте свое мероприятие в рамках конгресса!

Приглашаем к организации семинаров и мастер-классов суперкомпьютерного конгресса! Семинары могут проводиться удаленно на различных площадках в пределах России.

Приглашаем принять участие в выставке!