

ИНФОРМАТИКА В ШКОЛЕ

№ 6'2022

ISSN 2221-1993

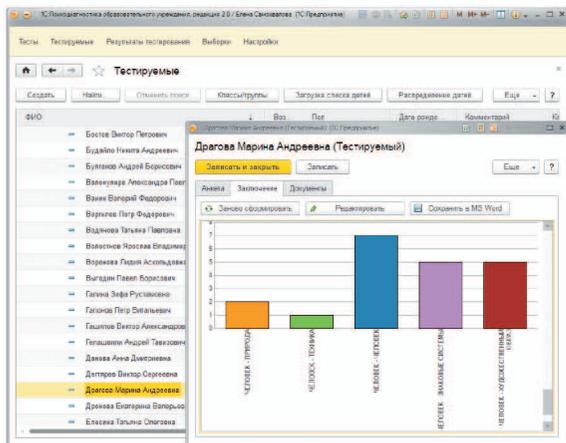
www.infojournal.ru



1С:ПСИХОДИАГНОСТИКА

Программно-методические комплексы линейки «1С:Психодиагностика» представляют собой инструментарий для проведения компьютерной психодиагностики детей и подростков, для сбора и консолидации результатов тестирования. Программы разработаны при поддержке группы ведущих психологов МГУ им. М.В. Ломоносова под общим руководством доктора психологических наук, профессора А.Н. Гусева. Программы линейки «1С:Психодиагностика»

- одобрены ФГАУ «Федеральный институт развития образования» в качестве программного обеспечения для использования психологами образовательных учреждений;
- включены в Единый реестр российских программ для электронных вычислительных машин и баз данных.



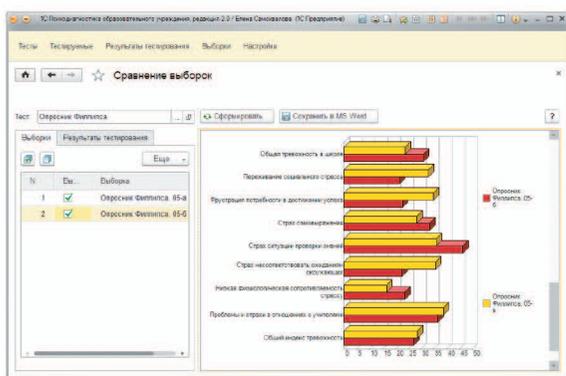
Функциональные возможности

- Хранение информации о тестируемых, их родителях, учителях в единой базе данных.
- Хранение результатов тестирования.
- Ведение истории работы психолога с тестируемым.
- Удаленное и массовое тестирование при помощи проекторов. Поддерживаются батареи тестов.
- Ввод и обработка данных с бумажных бланков, сформированных в программе.
- Сравнение результатов тестирования отдельных тестируемых.
- Автоматический расчет результатов тестирования.
- Формирование выборок результатов тестирования: по классам (группам), полу, возрасту и т.д.

Наименование	Блок	Возраст от	Возраст до	Время тестирования	Для учащихся групп
Опросник метод работы	Адаптация в коллективе	5	14	15	
Опросник Кеттелла - Подструк...	Общие	12	16	40	
Опросник креативности Дилон...	Креативность	7	10	15	
Опросник Селлербюргера - Хан...	Общие	15	99	15	
Опросник темперамента Тома...	Общие	3	7	15	✓
Опросник толерантности	Толерантность	15	99	10	
Опросник Томаса	Общие	14	99	15	
Опросник Фоллинкса	Адаптация в коллективе	7	17	10	
Опросник Штайна	Мотивация	11	99	15	
Отношение к сверстникам	Адаптация в коллективе	3	7	5	✓
Отношение к чужому взросло...	Адаптация в коллективе	3	7	5	✓
Оценка черт личности	Аддитивное поведение	13	99	20	
ГЕО	Общие	14	16	40	
Попытка	Имплайт	4	7	20	
Попытки	Мотивация	11	99	15	
Проба на произвольность	Имплайт	9	7	15	
Проведение агрессии	Адаптация в коллективе	9	7	5	✓
Повторно-оценочная ха...	Общие	3	10	5	✓
Распознавание кружков	Общие	5	9	20	
Распознавание	Полнота и широта	8	7	15	
САН	Общие	7	10	5	

Блоки методик

- Профорентация.
- Индивидуально-психологические особенности:
 - Оценка уровня тревожности,
 - Оценка уровня агрессии,
 - Исследование самооценки,
 - Исследование темперамента,
 - Исследование креативности,
 - Оценка познавательной сферы
 - Оценка ценностных ориентаций.
- Адаптация в коллективе.
- Детско-родительские отношения.
- Готовность к школе.



Преимущества использования

- Улучшение качества психологического сопровождения воспитательного процесса.
- Повышение производительности труда психологов.
- Соблюдение конфиденциальности психологической информации.
- Оценивание динамики психического развития детей.
- Формирование отчетов о проделанной работе.
- Снижение вероятности ошибок в результатах расчета психодиагностического исследования.
- Автоматизация процесса написания заключений.

ИНФОРМАТИКА В ШКОЛЕ

ИЗДАЕТСЯ С СЕНТЯБРЯ 2002 ГОДА

№ 6 (179) декабрь 2022

Содержание

От редакции 4

МЕТОДИЧЕСКАЯ КОПИЛКА

Босова Л. Л., Аквилянов Н. А. Обработка символьных данных: от простого к сложному 5

Маркушевич М. В. Элементы методики преподавания темы «Растровая компьютерная графика» с помощью свободного графического редактора GIMP на уровне основного общего образования 12

Гербут С. С. Пропедевтика программирования: как подготовить учащихся к восприятию объектно-ориентированного программирования 30

Розов К. В., Шрайнер Б. А. Дистанционное обучение школьников технологиям искусственного интеллекта 37

Дронова Е. Н. Реализация межпредметных связей информатики и математики при изучении темы «Обработка числовой информации в электронных таблицах» 44

Паравина А. С. Чек-лист по теме урока как средство обратной связи для учителя 50

ЗАДАЧИ

Долинский М. С. Ускорение рекурсивных решений при помощи мемоизации 55

СРЕДСТВА ИКТ В УЧЕБНОМ ПРОЦЕССЕ

Маркелов В. К., Завьялова О. А. Инструментарий облачной среды Google Colaboratory для обучения программированию и подготовки к ЕГЭ по информатике 68

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Баракина Т. В. Система обучения детей дошкольного и младшего школьного возраста основам начального конструирования и робототехники 81

НАПЕЧАТАНО В 2022 ГОДУ 89

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

Издатель ООО «Образование и Информатика»
119261, Россия, г. Москва, Ленинский пр-т, д. 82/2, комн. 6
Телефон: +7 (495) 140-19-86
E-mail: readinfo@infojournal.ru
Сайт издательства: <http://infojournal.ru/>
Почтовый адрес: 119270, Россия, г. Москва, а/я 15

Подписано в печать 12.12.22.
Формат 60×90^{1/8}. Усл. печ. л. 11,5
Тираж 2000 экз. Заказ № 1833.
Отпечатано в типографии ООО «Принт сервис групп»,
105187, Россия, г. Москва, Борисовская ул., д. 14, стр. 6,
тел./факс: +7 (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) 140-19-86

Почтовый адрес
119270, Россия, г. 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

Редактор отдела
БАСЫРОВА Зифа Аббясовна

Корректоры
СИРОТКИН Никита Сергеевич
ШАРАПКОВА Людмила Михайловна

Верстка
ФЕДОТОВ Дмитрий Викторович

Дизайн
ГЛАВНИЦКИЙ Евгений Николаевич

Отдел распространения и рекламы
КУЗНЕЦОВА Елена Александровна

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ «ИНФОРМАТИКА В ШКОЛЕ»

УЧРЕДИТЕЛЬ:

ИЗДАТЕЛЬСТВО
«ОБРАЗОВАНИЕ И ИНФОРМАТИКА»

ГЛАВНЫЙ РЕДАКТОР

БОСОВА Людмила Леонидовна

чл.-корр. РАО, 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, Академия социального управления, Московская область, профессор кафедры общеобразовательных дисциплин

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»

ЗАВОДЧИКОВА Надежда Ивановна

канд. пед. наук, доцент, Ярославский государственный педагогический университет им. К. Д. Ушинского, доцент кафедры теории и методики обучения информатике

ЗАСЛАВСКАЯ Ольга Юрьевна

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

ЗУБРИЛИН Андрей Анатольевич

канд. филос. наук, доцент, Мордовский государственный педагогический университет имени М. Е. Евсевьева, г. Саранск, доцент кафедры физики, информационных технологий и методик обучения

КИРИЧЕНКО Ирина Борисовна

заместитель главного редактора журнала «Информатика в школе»

ЛЕВЧЕНКО Ирина Витальевна

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

ПИЛОСЯН Элина Анатольевна

канд. тех. наук, Сочинский государственный университет, декан факультета информационных технологий и математики

ПОЛЯКОВ Константин Юрьевич

доктор тех. наук, Санкт-Петербургский государственный морской технический университет, профессор кафедры судовой автоматики и измерений; средняя общеобразовательная школа № 163, Санкт-Петербург, учитель информатики

РЫБАКОВ Даниил Сергеевич

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

СЛИНКИНА Ирина Николаевна

канд. пед. наук, доцент, Шадринский государственный педагогический университет, доцент кафедры физико-математического и информационно-технологического образования, руководитель Центра по работе с одаренными детьми и молодежью «Малый университет»

ТРОИЦКАЯ Ольга Николаевна

канд. пед. наук, доцент, Высшая школа информационных технологий и автоматизированных систем Северного (Арктического) федерального университета имени М. В. Ломоносова, г. Архангельск, зав. кафедрой экспериментальной математики и информатизации образования

ФЕДОРОВА Галина Аркадьевна

доктор пед. наук, доцент, Омский государственный педагогический университет, профессор кафедры информатики и методики обучения информатике

ФЕДОСОВ Александр Юрьевич

доктор пед. наук, доцент, Российский государственный социальный университет, Москва, профессор кафедры информационных технологий, искусственного интеллекта и общественно-социальных технологий цифрового общества

ЧЕРНЕЦКАЯ Татьяна Александровна

канд. пед. наук, фирма «ТС», Москва, ведущий методист отдела образовательных программ

ХІХ ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ПО МЕТОДИКЕ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ ИНФО-2022

**Издательство «Образование и Информатика»
объявляет о проведении в 2022 году
конкурса по следующим номинациям:**

- 1. Инновации в методике обучения информатике.**
- 2. Технологии искусственного интеллекта в сфере образования.**
- 3. Цифровизация образовательной организации и учебного процесса.**
- 4. Развитие профессиональных компетенций педагогов в контексте цифровизации системы образования.**

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

Цели и задачи конкурса

1. Поддержка и распространение опыта педагогов и образовательных организаций по внедрению в образовательную практику современных методов и средств обучения и управления образованием.
2. Выявление и поддержка талантливых педагогов, методистов, руководителей образовательных организаций и органов управления образованием, заинтересованных в развитии инновационных образовательных технологий.
3. Включение педагогов, методистов, руководителей образовательных организаций и органов управления образованием в деятельность по разработке нового содержания образования, новых образовательных технологий, методик обучения и управления образованием.
4. Создание информационно-образовательного пространства на сайте издательства «Образование и Информатика», а также на страницах журналов «Информатика и образование» и «Информатика в школе» по обмену и распространению опыта внедрения инновационных образовательных технологий.
5. Повышение информационной культуры и информационно-коммуникационной компетентности всех участников образовательного процесса.

Сроки и этапы проведения конкурса

- 1. Работы на конкурс принимаются** с 1 октября по 15 декабря 2022 года включительно. Работы, присланные позже 15 декабря 2022 года, к участию в конкурсе допускаться не будут.
- 2. Итоги конкурса** будут подведены до 1 февраля 2023 года и опубликованы на сайте издательства «Образование и Информатика», а также в журналах «Информатика и образование» и «Информатика в школе» № 1-2023.
- 3. Лучшие работы** будут опубликованы в журналах «Информатика и образование» и «Информатика в школе».

Победители конкурса получат (бесплатно):

1. Диплом от издательства «Образование и Информатика».
2. Подписку в печатном виде на журналы «Информатика и образование» и «Информатика в школе» на 2023 год.

Подробную информацию о конкурсе вы можете найти на сайте ИНФО:

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

Контакты Оргкомитета:

Телефон: +7 (495) 140-19-86

E-mail: readinfo@infojournal.ru

<http://www.infojournal.ru/>



Уважаемые коллеги! Дорогие друзья!

Стремительно приближающийся к концу 2022 год принес много изменений в систему образования, в нашу профессиональную деятельность, в школьный курс информатики.

Мы получили обновленные ФГОС начального общего, основного общего и среднего общего образования, в которых детализированы требования к личностным, метапредметным и предметным результатам и к условиям реализации основных образовательных программ.

Во исполнение поручения Президента РФ «обеспечить совершенствование преподавания учебных предметов “Математика” и “Информатика” в общеобразовательных организациях, установив их приоритет в учебном плане и скорректировав содержание примерных основных образовательных программ общего образования» на основе требований обновленных ФГОС ООО и СОО разработаны примерные рабочие программы по учебному предмету «Информатика»:

- базового уровня для V—VI классов;
- базового уровня для VII—IX классов;
- углубленного уровня для VII—IX классов;
- базового уровня для X—XI классов;
- углубленного уровня для X—XI классов.

Каждая образовательная организация, опираясь на новую нормативную базу, получает возможность в рамках единого образовательного пространства выстраивать собственную, в том числе непрерывную, траекторию освоения школьного курса информатики. А значит, особую актуальность приобретают вопросы поиска, разработки, апробации новых методических решений, современного учебно-методического обеспечения, отвечающего задачам развития образования в области информатики и информационных технологий.

Редакционная коллегия журнала «Информатика в школе» ждет такие материалы от вас, наши уважаемые авторы, готова публиковать их на своих страницах для вас, дорогие наши читатели.

Желаем вам встретить наступающий 2023 год, объявленный указом Президента России Годом педагога и наставника, с прекрасным настроением в кругу родных и друзей, полными творческих замыслов и сил для их воплощения.

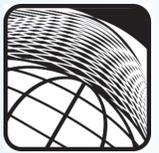
И пусть создаст вам праздничное настроение отрывок из стихотворения Александра Твардовского «Утро»:

Кружась легко и неумело,
Снежинка села на стекло.
Шел ночью снег густой и белый —
От снега в комнате светло.

Чуть порошит пушок летучий,
И солнце зимнее встает.
Как каждый день — полней и лучше,
Да будет лучше новый год.

С Новым годом, дорогие друзья!

*Искренне ваша,
редакция журнала
«Информатика в школе»*



DOI: 10.32517/2221-1993-2022-21-6-5-11

Л. Л. Босова

Московский педагогический государственный университет, г. Москва, Россия

Н. А. Аквиланов

Фирма «1С», г. Москва, Россия

ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ: ОТ ПРОСТОГО К СЛОЖНОМУ

Аннотация

В статье предложены разноуровневые материалы для освоения учащимися VII–VIII классов темы «Обработка символьных данных» на языке программирования Python.

Первый уровень сложности предполагает рассмотрение минимальных теоретических сведений и базовых алгоритмов обработки символьных данных, в том числе алгоритмов редактирования текстов (замена символа, удаление и вставка символа, поиск вхождения заданного образца); материал ориентирован на учеников, осваивающих информатику на базовом уровне.

Второй (повышенный) уровень сложности предполагает не только фактическое знание изучаемого материала, включая методы обработки строк, но и способность применить его в новых ситуациях; материал может быть предложен учащимся основной школы, осваивающим информатику на углубленном уровне.

Третий (высокий) уровень сложности включает знакомство обучающихся с олимпиадными задачами; материал ориентирован на наиболее подготовленных и мотивированных школьников.

Работа со строковыми данными, реализованная на разных уровнях сложности, может стать как альтернативой, так и дополнением для задач с математическим контекстом, традиционно включаемых в содержание курса информатики основной школы.

Ключевые слова: информатика, обучение программированию, язык программирования Python, символьные данные.

1. Введение

В последние годы в условиях цифровой трансформации многих сфер нашей жизни во всем мире наблюдается большой интерес к обучению школьников программированию [3].

Отмечаются развивающие и социальные аспекты программирования [1, 2]:

- занятия программированием развивают мышление человека;
- изучение программирования способствует формированию у людей новых ценностей цифрового

Контактная информация

Босова Людмила Леонидовна, доктор пед. наук, доцент, член-корреспондент РАО, зав. кафедрой теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* akulll@mail.ru

Аквиланов Никита Александрович, руководитель проекта, фирма «1С», г. Москва, Россия; *адрес:* 127473, Россия, г. Москва, ул. Селезневская, д. 34, стр. 4; *e-mail:* akvilianov@gmail.com

L. L. Bosova

Moscow Pedagogical State University, Moscow, Russia

N. A. Akvilianov

1C Company, Moscow, Russia

CHARACTER DATA PROCESSING: FROM SIMPLE TO COMPLEX

Abstract

The article proposes multi-level materials for students of grades 7–8 to master the theme "Character data processing" in the Python programming language.

The first level of complexity involves consideration of the minimum theoretical information and basic algorithms for processing character data, including algorithms for texts editing (replacing a character, deleting and inserting a character, searching for an occurrence of a given pattern); the material is aimed at students who master informatics at a basic level.

The second (advanced) level of complexity implies not only actual knowledge of the material being studied, including string processing methods, but also the ability to apply it in new situations; the material can be offered to the basic school students who master informatics at an advanced level.

The third (high) level of complexity involves the acquaintance of students with the Olympiad tasks; the material is aimed at the most prepared and motivated students.

Working with string data, implemented at different levels of complexity, can become both an alternative and a supplement for tasks with a mathematical context, traditionally included in the content of the informatics course of the basic school.

Keywords: informatics, programming training, Python programming language, character data.

общества (сообщества, совместная работа, обмен знаниями);

- благодаря навыкам программирования человек лучше понимает правила поведения в цифровой среде, чувствует себя в ней более уверенно и комфортно.

Тематический раздел «Алгоритмы и программирование» является традиционным для школьного курса информатики, традиционными являются и трудности, возникающие у школьников при его освоении. Одна из основных причин трудностей, возникающих у обучающихся при знакомстве с языком программирования, — преобладание задач с математическим контекстом (проверка делимости одного целого числа на другое; решение квадратного уравнения, имеющего вещественные корни; нахождение наибольшего общего делителя двух натуральных чисел; разложение натурального числа на простые сомножители; разбиение записи натурального числа в позиционной системе с основанием, меньшим или равным 10, на отдельные цифры и др.) [9, 10]. Устранить такого рода причину можно, если наряду с программированием алгоритмов обработки чисел *познакомить школьников с алгоритмами обработки символьных данных*. Заметим, что соответствующая тематика достаточно полно представлена в примерных рабочих программах по учебному предмету «Информатика» для основной школы базового и углубленного уровней: «Обработка символьных данных. Символьные (строковые) переменные. Посимвольная обработка строк. Подсчет частоты появления символа в строке. Встроенные функции для обработки строк» [9, 10].

Покажем, как может быть организовано освоение темы «Обработка символьных данных» на языке программирования Python учениками VII—VIII классов.

2. Краткие теоретические сведения

Приведем краткие теоретические сведения о символьных данных (данных строкового типа), с которыми необходимо познакомить обучающихся [4].

Значением строковой величины (тип *str*) в языке Python является произвольная последовательность символов, заключенная в одинарные или двойные кавычки. Символьная строка рассматривается как единый объект.

В тексте программы переменную строкового типа можно задать, заключив цепочку символов в одинарные или двойные кавычки.

Новое значение может быть записано в строку с помощью оператора ввода с клавиатуры: $s = \text{input}()$. Если значение символьной переменной считывается с клавиатуры, то его следует набирать без апострофов.

Можно проверить равенство (совпадение) строк или выявить, какая из двух строк меньше (при этом используется поочередное сравнение кодов пар символов, образующих слова; меньшим будет то слово, у которого код символа окажется меньше).

В Python строки можно сцеплять: $a + b$ (к концу строки a прикрепляется, или «приписывается», строка b).

В результате операции $a * k$, где k — целое число, строка a повторяется k раз.

Строка состоит из последовательности символов. Узнать длину строки (количество символов в строке) можно при помощи функции $\text{len}()$.

Из строки можно выделить срез — любое количество последовательно идущих символов:

- $s[i]$ — извлечение из строки одного символа, имеющего номер i ; при этом считается, что нумерация начинается с 0; чтобы извлекать элементы строки справа налево, указывают отрицательные значения i , считая, что последний символ строки имеет номер -1 , предпоследний — -2 и т. д.;
- $s[m:n]$ — извлечение из строки последовательности символов, начиная с символа, имеющего номер m , до символа с номером n ; при этом символ с номером n в срез не входит. Если не указывать параметр m , то срез будет взят от начала строки; если не указывать параметр n , то срез будет взят до конца строки.

С помощью оператора *in* можно проверить, входит ли значение некоторой строковой переменной в ту или иную строку.

Функция $\text{str}()$ переводит число или любой другой объект к строке.

Функция $\text{int}()$ переводит строку в число в десятичной системе счисления. Если вызвать функцию $\text{int}()$ без аргументов, она вернет 0. Чаще всего используется функция $\text{int}()$ с одним аргументом — символьным представлением целого числа. В случае, когда указывается второй аргумент для функции $\text{int}()$, он обозначает систему счисления, в которой находится число, указанное в строке первого аргумента; функция $\text{int}()$ возвращает его значение в десятичной системе счисления.

Другие теоретические сведения на начальном этапе работы со строковыми величинами не требуются.

3. Программирование базовых алгоритмов обработки строк

Рассмотрим программирование базовых алгоритмов обработки строк на примере последовательности задач, представленных в пособии [5].

Задание 1. Две строки.

Составьте программу, которая предлагает ввести с клавиатуры две строки a и b , а затем:

- соединяет их в новую строку c и выводит результат;
- определяет количество символов в новой строке c ;
- выводит на экран более длинную из строк a и b ;
- выводит на экран большую из строк a и b .

Пример ввода	Пример вывода
тепло ход	а) теплоход
	б) 8
	в) тепло
	г) ход

Решение.

```
a=input()
b=input()
c=a+b
print(c)
k=len(a+b)
```

```
print(k)
print(a) if len(a)>=len(b) else print(b)
print(a) if a>=b else print(b)
```

Задание 2. Составление слов.

Составьте программу, которая из слова ИНФОРМАТИКА получает слова:

- ФОРМА;
- ФИРМА;
- МАК.

Решение.

```
a="ИНФОРМАТИКА"
print(a[2:7])
print(a[2]+a[0]+a[4:7])
print(a[5:7]+a[-2])
```

Задание 3. Работа со словами строки.

Вводится строка из двух слов, разделенных пробелом. Составьте программу, которая:

- находит позицию пробела во введенной строке и выводит номер этой позиции на экран;
- выводит на экран первое слово введенной строки;
- выводит на экран второе слово введенной строки;
- меняет местами слова в исходной строке и выводит новую строку на экран.

Пример ввода	Пример вывода
обработка строк	а) 10
	б) обработка
	в) строк
	г) строк обработка

Решение.

```
a=input()
for i in range(len(a)):
    if a[i]==" ":
        n=i
print(n+1)
print(a[0:n])
print(a[n+1:len(a)])
b=a[n+1:len(a)]+" "+a[0:n]
print(b)
```

Задание 4. Подсчет количества разных букв в строке.

Составьте программу, которая предлагает ввести строку, состоящую из строчных английских букв, и находит:

- количество букв «s» в этой строке;
- общее количество букв «s» и «t» в этой строке;
- каких букв — «s» или «t» — больше в этой строке.

Пример ввода	Пример вывода
rststsrssr	а) 5
	б) 7
	в) Больше s

```
a=input()
ks=kt=0
for i in range(len(a)):
    if a[i]=="s":
        ks=ks+1
    if a[i]=="t":
        kt=kt+1
print(ks)
print(ks+kt)
print("Больше s") if ks>kt else print("Больше t")
```

Задание 5. Подсчет слов в предложении.

Составьте программу, которая осуществляет подсчет слов в предложении, если слова в предложении разделены одним пробелом.

Решение.

```
a=input()
k=1
for i in range(len(a)):
    if a[i]==" ":
        k=k+1
print(k)
```

Задание 6. Замена букв.

Составьте программу, которая заменяет в строке из строчных английских букв все буквы «a» на букву «o».

Пример ввода	Пример вывода
abcaaf	obcoof

Решение.

```
a=input()
n=len(a)
c=""
for i in range(n):
    if a[i]=="a":
        c=c+"o"
    else:
        c=c+a[i]
print(c)
```

Задание 7. Удаление символов.

Составьте программу, которая удаляет в строке из строчных английских букв все вхождения буквы, введенной с клавиатуры.

Пример ввода	Пример вывода
rsassarssra r	sassassa

Решение.

```
a=input()
b=input()
c=""
for i in range(len(a)):
    if a[i]!=b:
        c=c+a[i]
print(c)
```

Задание 8. Вставка символов.

Составьте программу, позволяющую:

- вставить после каждого символа исходной строки пробел и вывести полученную строку на экран;
- удвоить каждый символ исходной строки и вывести полученную строку на экран.

Пример ввода	Пример вывода
ПРИВЕТ	а) П Р И В Е Т
	б) ППРРИИВВЕЕТТ

Решение.

```
a=input()
c=""
d=""
for i in range(len(a)):
    c=c+a[i]+" "
    d=d+2*a[i]
print(c)
print(d)
```

Обратите внимание! Программный код для части б рассматриваемого задания можно записать короче:

```
a=input()
for i in a:
    print(2*i, end="")
```

Задание 9. Строка-палиндром.

Составьте программу, позволяющую определить, является ли введенная строка:

- палиндромом, т. е. верно ли, что она читается одинаково слева направо и справа налево;
- палиндромом после удаления из нее всех пробелов.

	Пример ввода	Пример вывода
а)	КАЗАК	Палиндром
	БАНАН	Нет
б)	КОТУ ТАЩАТ УТОК	Палиндром
	КОТУ ТАЩАТ УТКУ	Нет

Решение.

Программный код для части а:

```
a=input()
c=""
for i in range(len(a)):
    c=a[i]+c
print("Палиндром") if a==c else print("Нет")
```

Обратите внимание! Этот программный код можно записать короче:

```
a=input()
print("Палиндром") if a==a[::-1] else
print("Нет")
```

Программный код для части б:

```
a=input()
d=""
c=""
```

```
for i in range(len(a)):
    if a[i]==" ":
        d=d+" "
for i in range(len(d)):
    c=d[i]+d
print("Палиндром") if d==c else print("Нет")
```

Выполнив представленную выше последовательность заданий, учащиеся:

получат представление:

- о том, как устроены базовые алгоритмы обработки символьных данных;

научатся:

- определять длину строки;
- применять операцию конкатенации к строкам;
- определять номер заданного символа в строке;
- подсчитывать количество слов в строке;
- менять местами слова в строке;
- осуществлять поиск и подсчет количества заданных элементов строки;
- осуществлять поиск и замену заданных элементов строки;
- удалять заданные элементы из строки;
- вставлять заданные элементы в строку;
- разрабатывать несложные программы обработки строк с использованием ветвлений и циклов.

4. Задачи повышенного уровня сложности**Задание 10. Строка-число.**

Составьте программу, которая проверяет, может ли введенная с клавиатуры строка быть записью двоичного числа.

Решение.

```
a=input()
binary="01"
flagB=1
for i in range(len(a)):
    if a[i] not in binary:
        flagB=0
if flagB:
    print("Перед вами двоичный код")
else:
    print("Нет")
```

Задание 11. Проверка правописания.

Составьте программу, которая проверяет во введенной строке соблюдение правила правописания, выводит на экран количество допущенных ошибок и исправленную строку:

- правописание «ча» и «ща»;
- правописание «жи» и «ши».

	Пример ввода	Пример вывода
а)	рощя, чашя, чяща, чяша	4, роцца, чаща, чаша, чаша
б)	шыло, лужы, ошыбка, дружыть	4, шило, лужи, ошибка, дружить

Решение.

```
a=input()
k=0
for i in range (len(a)-1):
    if (a[i]+a[i+1]=="чя") or (a[i]+a[i+1]=="щя"):
        a=(a[: (i+1)]+"a"+a[(i+2):])
        k+=1
    if (a[i]+a[i+1]=="жы") or (a[i]+a[i+1]=="шы"):
        a=(a[: (i+1)]+"и"+a[(i+2):])
        k+=1
print(k, " ", a, sep="")
```

5. Методы обработки строк

Во всех программах, рассмотренных выше, данные и функции использовались отдельно. Но это не единственный способ организации кода: объектно-ориентированный код строится на объединении данных и функций в одну сущность — **объект**. Данные в таком случае называются **атрибутами**, а функции — **методами**. Вызов метода происходит через точку, которая идет сразу за именем переменной. В остальных методах работают как обычные функции.

После того как школьники освоят представленные выше базовые алгоритмы обработки строк (задания 1–9) и попробуют применять их для решения задач (задания 10–11), можно познакомить их с некоторыми методами обработки строк (см. табл.) [8].

Таблица

Некоторые методы обработки строк

Метод	Описание
<code>upper()</code>	Возвращает строку, символы которой приведены к верхнему регистру. Исходная строка не изменяется
<code>lower()</code>	Возвращает строку, символы которой приведены к нижнему регистру. Исходная строка не изменяется
<code>capitalize()</code>	Возвращает строку, в которой первая буква прописная, остальные — строчные
<code>replace(old, new)</code>	Возвращает строку, в которой вхождения строки <i>old</i> заменены строкой <i>new</i>
<code>find(str)</code>	Поиск подстроки в строке. Возвращает номер первого вхождения или -1.
<code>rfind(str)</code>	Поиск подстроки в строке. Возвращает номер последнего вхождения или -1
<code>split(символ)</code>	Разбиение строки по разделителю
<code>count(str)</code>	Возвращает количество непересекающихся вхождений подстроки
<code>join(список)</code> <code>join(str)</code>	Сборка строки из списка с разделителем. Если передать в качестве аргумента функции строку, то она будет разбита по символам с определенным разделителем

Покажем, как с использованием приведенных в таблице методов обработки строк можно оптимизировать программный код для рассмотренных выше заданий.

Программный код к заданию 3 с использованием метода `find(str)` будет выглядеть так:

```
a=input()
print(a.find(" ") + 1)
print(a[:a.find(" ")])
print(a[a.find(" ") + 1:])
print(a[a.find(" ") + 1:] + " " + a[a.find(" ")])
```

Программный код к заданию 4 с использованием метода `count(str)` будет выглядеть так:

```
a=input()
print(a.count("s"))
print(a.count("s")+a.count("t"))
print("Вольше s") if (a.count("s")>a.count("t"))
else print("Вольше t")
```

Программный код к заданию 5 с использованием метода `count(str)` будет выглядеть так:

```
print(input().count(" ") + 1)
```

Программный код к заданию 6 с использованием метода `replace(old, new)` будет выглядеть так:

```
a=input()
print(a.replace("a", "o"))
```

Программный код к заданию 7 с использованием метода `replace(old, new)` будет выглядеть так:

```
a=input()
b=input()
print(a.replace(b, ""))
```

Этот код можно «свернуть» в одну строку:

```
print(input().replace(input(), ""))
```

Программный код к заданию 8а с использованием метода `join(str)` будет выглядеть так:

```
print(" ".join(input()))
```

Программный код к заданию 8б с использованием метода `join(str)` и генератора списка будет выглядеть так:

```
print("".join([x*2 for x in (input())]))
```

Программный код к заданию 9б с использованием метода `replace(old, new)` будет выглядеть так:

```
a=input()
a=a.replace(" ", "", a.count(" "))
print("Палиндром") if a==a[::-1] else
print("Нет")
```

Программный код к заданию 10 с использованием метода `count(str)` будет выглядеть так:

```
x=input()
print("Перед вами двоичный код")
if (x.count("1")+x.count("0") == len(x))
else print("Нет")
```

Программный код к заданию 11 с использованием методов `replace(old, new)` и `count(str)` будет выглядеть так:

```

a=input()
k=a.count("чя")+a.count("щя")+a.count("жы")+
  a.count("шы")
a=a.replace("щя","ща")
a=a.replace("чя","ча")
a=a.replace("жы","жи")
a=a.replace("шы","ши")
print(k," ",a,sep=" ")

```

Мы рассмотрели лишь некоторые методы обработки строк. Сдругими из них мотивированные ученики могут познакомиться самостоятельно.

6. Задания высокого уровня сложности

Задание 12. Системы счисления (задание 14 демонстрационного варианта ЕГЭ–2023) [6].

Операнды арифметического выражения записаны в системе счисления с основанием 15:

$$123x5_{15} + 1x233_{15}.$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 15-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения кратно 14. Для найденного значения x вычислите частное от деления значения арифметического выражения на 14 и укажите его в ответе в десятичной системе счисления. Основание системы счисления в ответе указывать не надо.

Решение.

```

a="0123456789ABCDE"
for i in a:
    x1="123"+i+"5"
    x2="1"+i+"233"
    x=int(x1,15)+int(x2,15)
    if x%14==0:
        print(x//14)
        break

```

Задание 13. Маска числа (задание 25 демонстрационного варианта ЕГЭ–2023) [6].

Назовем маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «?» означает ровно одну произвольную цифру;
- символ «*» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.

Например, маске $123*4?5$ соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2139*4$, делящиеся на 2023 без остатка.

Решение.

Сначала исследуем числа, имеющие вид: 1?21394. Другими словами, в записи чисел по заданной маске знаку «*» соответствует пустая последовательность. Вместо знака «?» может стоять любая из цифр от 0 до 9. Построим такие числа и проверим их делимость на число 2023:

```

for i in range(0,10):
    x=int("1"+str(i)+"21394")
    if x%2023==0:
        print(x)

```

Добавим к рассмотрению числа, в которых вместо знака «*» может быть любая последовательность из одной, двух или трех десятичных цифр. Такие числа можно получить, перебрав в цикле и подставив на место знака «*» все однозначные, двузначные или трехзначные числа (при попытке подставить вместо знака «*» большее количество цифр мы выйдем за границы числа 10^{10}):

```

for j in range(0,1000):
    y=x//10
    y=int(str(y)+str(j)+"4")
    if y%2023==0:
        print(y)

```

Важно не упустить из рассмотрения и такие числа, в которых вместо знака «*» стоят строки, начинающиеся с одного или двух нулей. Дополним программный код:

```

if j<100:
    y=int(str(y)+"0"+str(j)+"4")
    if y%2023==0:
        print(y)
if j<10:
    y=int(str(y)+"00"+str(j)+"4")
    if y%2023==0:
        print(y)

```

В заключение рассмотрим олимпиадную задачу, сложность решения которой связана с разработкой алгоритма, а не с его записью на языке программирования.

Задание 14. Счастливые числа (задача муниципального этапа Всероссийской олимпиады школьников по информатике, IX—XI классы, 2021 год, Московская область) [7].

Федя совсем недавно поступил в лучший вуз страны. Особенно ему стала интересна кафедра изучения счастливых чисел, т. е. тех чисел, которые состоят только из цифр 2 и 5. Научные сотрудники этой кафедры исследуют их распределение. Они поняли, что существует последовательность всех счастливых чисел в порядке возрастания (2 — первое число, 5 — второе, 22 — третье и т. д.). Они хотят найти порядковый номер счастливого числа N в данной последовательности. Федю очень заинтересовала эта задача. Он думал над ней целый день, но так ни к чему и не пришел. Можете ли вы помочь Феде и кафедре счастливых чисел найти ответ?

Решение.

Выпишем начало последовательности счастливых чисел:

2, 5, 22, 25, 52, 55, 222, 225, 252, 255, 522, 525, 552, 555, ...

Все числа записаны в алфавите из двух цифр — 2 и 5. Перейдем к более привычному двоичному алфавиту — 0 и 1. Получим:

0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, ...

Полученные двоичные цепочки различны; цепочки одной длины расположены в алфавитном порядке.

Если бы не нули, стоящие в начале некоторых цепочек, то эти цепочки можно было бы трактовать как разные двоичные числа, а их десятичный эквивалент можно было бы связать с номером, занимаемым числом в последовательности.

Есть прием, позволяющий превратить полученные двоичные цепочки в двоичные числа: надо дописать в начале каждой цепочки единицу:

10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, ...

Мы получили двоичное представление последовательности натуральных чисел, начиная с 2:

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...

Номер, занимаемый числом в последовательности, на 1 меньше значения этого числа.

Итак, чтобы выяснить номер произвольного счастливого числа, следует:

- 1) в записи исходного числа произвести замену цифры 2 на 0, цифры 5 — на 1;
- 2) в начало полученной двоичной строки приписать цифру 1;
- 3) считать полученную двоичную цепочку двоичным числом, перевести это двоичное число в десятичную систему счисления;
- 4) вычесть из полученного десятичного числа 1.

Запишем описанный выше алгоритм на языке программирования:

```
s=input()
s=s.replace("2","0").replace("5","1")
b="1"+s
d=int(b,2)
print(d-1)
```

7. Заключение

Содержание данной статьи условно можно разбить на три части.

В первой части мы рассмотрели минимальные теоретические сведения (понятие символьной величины, операции над строками, некоторые встроенные функции языка программирования для обработки строк) и базовые алгоритмы обработки символьных данных, в том числе алгоритмы редактирования текстов (замена символа, удаление и вставка символа, поиск вхождения заданного образца). К этой части относятся задания 1–9. Их выполнение по силам ученикам, осваивающим информатику на базовом уровне.

Во второй части (задания 10–11) мы рассмотрели задания повышенного уровня сложности, требующие не только фактического знания изучаемого материала, но и умения его применять в новых ситуациях. Здесь же мы рассмотрели ряд методов обработки строк, перейдя, по сути, от записи программного кода базовых алгоритмов к использованию соответствующих функций. Такой

переход целесообразно делать именно после того, как обучающиеся познакомились с «внутренним» устройством функции, поняли, как она работает. Материал второй части может быть предложен учащимся основной школы, осваивающим информатику на углубленном уровне, а также мотивированным школьникам, изучающим информатику на базовом уровне.

Третья часть представлена заданиями 12–14, ориентированными на наиболее подготовленных и мотивированных учащихся, осваивающих информатику на углубленном уровне.

Таким образом, работа со строковыми данными, реализованная на разных уровнях сложности, может стать как альтернативой, так и дополнением для задач с математическим контекстом, традиционно включаемых в содержание курса информатики основной школы.

Список источников

1. Босова Л. Л. Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе // Информатика в школе. 2018. № 6. С. 3–11. EDN: XZOOJV.
2. Босова Л. Л. Программирование как инструмент формирования вычислительного мышления обучающихся // Информатика в школе. 2020. № 10. С. 4–10. DOI 10.32517/2221-1993-2020-19-10-4-10. EDN: GURIPH.
3. Босова Л. Л. Школьная информатика в России и в мире // Информатизация образования и науки. 2018. № 3. С. 134–145. EDN: XSMSVV.
4. Босова Л. Л., Аквилянов Н. А., Кочергин И. О., Штепа Ю. Л., Бурцева Т. А. Информатика. 8–9 классы. Начала программирования на языке Python: Дополнительные главы к учебнику. М.: БИНОМ. Лаборатория знаний, 2020. 96 с. EDN: LPMXCV.
5. Босова Л. Л., Босова А. Ю., Аквилянов Н. А. Информатика: 7–9 классы: компьютерный практикум. М.: Просвещение, 2022. 192 с.
6. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2023 года по информатике. Подготовлен федеральным государственным бюджетным научным учреждением «Федеральный институт педагогических измерений». <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!/tab/151883967-5>
7. Задания муниципального этапа всероссийской олимпиады школьников по информатике 2021–2022 учебный год. 9–11 классы. Московская область // Образовательный центр «Взлёт». https://olympmo.ru/news_img/training/iikt/olympmun_2021-2022/tasks-iikt-9-11-mun-2020-2021.pdf
8. Кольцов Д. В. Python: создаем программы и игры. 2-е изд. СПб.: Наука и Техника, 2019. 400 с.
9. Примерная рабочая программа основного общего образования. Информатика. Базовый уровень (для 7–9 классов образовательных организаций). Одобрена решением федерального учебно-методического объединения по общему образованию, протокол 3/21 от 27.09.2021 г. <https://fgosreestr.ru/oop/237>
10. Примерная рабочая программа основного общего образования. Информатика. Углубленный уровень (для 7–9 классов образовательных организаций). Одобрена решением федерального учебно-методического объединения по общему образованию, протокол 2/22 от 29.04.2022 г. <https://fgosreestr.ru/oop/313>

DOI: 10.32517/2221-1993-2022-21-6-12-29

М. В. Маркушевич

Школа № 1352, г. Москва, Россия

ЭЛЕМЕНТЫ МЕТОДИКИ ПРЕПОДАВАНИЯ ТЕМЫ «РАСТРОВАЯ КОМПЬЮТЕРНАЯ ГРАФИКА» С ПОМОЩЬЮ СВОБОДНОГО ГРАФИЧЕСКОГО РЕДАКТОРА GIMP НА УРОВНЕ ОСНОВНОГО ОБЩЕГО ОБРАЗОВАНИЯ

Аннотация

Статья содержит описание каркаса методики обучения учащихся основной школы технологии как обработки цифровой фотографии, так и создания растровых графических изображений с нуля. В качестве программного инструмента предлагается использовать свободный растровый редактор GIMP.

Растровая компьютерная графика является более сложной по отношению к векторной графике, так как в растровой графике используется большое количество инструментов, а также активно задействованы кистевые графические инструменты (*Кисть, Карандаш, Аэрограф*), эффективное применение которых предполагает наличие у пользователя базовых художественных навыков.

Выбор именно свободного графического редактора GIMP обусловлен рядом существенных преимуществ данного программного инструмента перед аналогами. К таковым можно отнести лицензию GNU GPL на его использование, кроссплатформенность, относительно низкие системные требования, функциональность, близкую к профессиональным растровым редакторам, а также существенный стек учебно-методической литературы, посвященной технологии применения GIMP в учебном процессе, что вообще не является характерным для свободного программного обеспечения.

Непростой пользовательский интерфейс редактора GIMP обусловлен большим количеством инструментов и полной функциональностью данного программного обеспечения и может быть освоен учащимися основной школы при условии применения в учебном процессе эффективной методики обучения растровой графике.

При построении каркаса методики обучения школьников растровой графике технология применения всех групп инструментов редактора GIMP демонстрируется в процессе проведения практических работ, направленных на создание различных

Контактная информация

Маркушевич Михаил Владимирович, учитель информатики, школа № 1352, г. Москва, Россия; *адрес:* 107207, Россия, г. Москва, Щелковское ш., д. 77А; *e-mail:* mihael11@yandex.ru

M. V. Markushevich

School 1352, Moscow, Russia

ELEMENTS OF THE METHODOLOGY OF TEACHING THE THEME "RASTER COMPUTER GRAPHICS" WITH THE HELP OF THE GIMP FREE GRAPHIC EDITOR AT THE LEVEL OF BASIC GENERAL EDUCATION

Abstract

The article describes the framework of the methodology for teaching basic school students the technology of both processing digital photography and creating raster graphic images from scratch. As a software tool, it is proposed to use the free GIMP raster editor.

Raster computer graphics is more complex in relation to vector graphics, since a large number of tools are used in raster graphics, and brush graphic tools (*Brush, Pencil, Airbrush*) are also actively used, the effective use of which requires user of basic artistic skills.

The choice of the free graphic editor GIMP is due to a number of significant advantages of this software tool over analogues. These include the GNU GPL license for its use, cross-platform, relatively low system requirements, functionality close to professional raster editors, as well as a significant stack of educational and methodological literature on the technology of using GIMP in the educational process, that is generally not typical for free software.

The complex user interface of the GIMP editor is due to the large number of tools and the full functionality of this software and can be mastered by basic school students, provided that an effective methodology for teaching raster graphics is used in the educational process.

When creating a framework for teaching schoolchildren about raster graphics, the technology for using all groups of GIMP editor tools is demonstrated in the process of conducting practical work aimed at creating various raster graphics, the complexity of which gradually increases in proportion to the number of tools mastered by students.

Also, the proposed teaching methodology takes into account all the main structural elements of professional raster graphics, namely: the technology for the effective use of the group of operations for selecting image fragments, the technology for the effective use of the group for the transformation operation, the technology for the effective use of the group of drawing tools, image layers, color channels, contours, built-in filters, creation of text elements, creation of GIF-animation.

Keywords: raster graphics, GIMP, freeware.

растровых графических изображений, сложность которых постепенно возрастает пропорционально количеству освоенных учащимися инструментов.

Также предлагаемая методика обучения учитывает все основные структурные элементы профессиональной растровой графики, а именно: технологию эффективного применения группы операции выделения фрагментов изображения, технологию эффективного применения группы операций преобразования, технологию эффективного использования группы инструментов рисования, слои изображения, цветовые каналы, контуры, встроенные фильтры, создание текстовых элементов, создание GIF-анимации.

Ключевые слова: растровая графика, GIMP, свободное программное обеспечение.

1. Введение

Настоящая статья является продолжением серии работ автора, написанных в соавторстве с А. Ю. Федосовым, направленных на разработку каркаса методики обучения учащихся основной школы различным информационным технологиям с использованием исключительно свободного программного обеспечения [8, 22].

В данной статье речь пойдет о построении методики обучения технологии создания и обработки растровой графики. Отметим, что эта информационная технология является более сложной по сравнению с ранее рассмотренными автором технологиями создания и обработки векторной графической информации, текстовой информации, звуковой информации.

Причины указанной сложности заключаются в непростой структуре объектов растровой графики (слоев изображения, контуров, масок изображений, цветовых каналов и т. д.), а также в том, что создание изображений в растровой графике аналогично рисованию кистями на холсте, что требует определенной художественной подготовки в отличие от технологии создания векторных графических изображений, в которой изображения собираются из графических примитивов как из кирпичиков [21].

Еще одним доказательством сложности построения методики обучения учащихся основной школы технологии создания и обработки растровой графики является отсутствие таковой методики в настоящий момент, несмотря на наличие достаточно большого количества литературы, посвященной применению свободного растрового графического редактора GIMP. Когда мы говорим об одной из информационных технологий, *под понятием «методика обучения» можно подразумевать эффективную организацию процесса передачи от преподавателя обучающимся некоего необходимого и достаточного набора алгоритмов создания тех или иных информационных объектов, соответствующих изучаемой информационной технологии, включающих в себя знание и умение использовать стандартный инструментарий той программной среды (сред), которая является оптимальной для данных целей на основе сравнительного анализа доступного программного обеспечения.*

Хотелось бы еще раз подчеркнуть, что речь идет о законченной методике обучения теме «Растровая компьютерная графика» для уровня основного общего образования, а не о некоторых фрагментах данной методики, которые очевидно присутствуют в работах Г. Ю. Пожариной [13], И. А. Хахаева [23], Т. А. Панюковой [12]. В то же время некоторые авторы под методикой подразумевают отдельные практические задания, выполняемые в программной среде, в нашем случае — в растровом редакторе GIMP [11], с чем сложно согласиться.

Отметим, что сама по себе проблема преподавания растровой компьютерной графики в школе с использованием свободных графических редакторов встала только в 2007 году, когда бывший директор Сепычевской средней школы Пермского края А. М. Поносов был приговорен судом к штрафу за использование нелегальных копий ОС Windows и Microsoft Office на школьных компьютерах. Для предотвращения повторения подобных ситуаций в 2008 году в российские школы был направлен для тестирования и использования пакет свободного программного обеспечения (ПСПО) на базе отечественной операционной системы Alt Linux, содержащий в списке устанавливаемых по умолчанию программ также и GIMP, что, к сожалению, на тот момент не дало существенных положительных результатов ввиду практически полного отсутствия учебно-методической литературы по данному программному обеспечению.

До 2007 года проблема выбора оптимального программного обеспечения для поддержки учебного процесса по информатике в общеобразовательной российской школе явно не относилась к основным проблемам, и авторы учебной литературы в качестве базового нередко использовали проприетарное программное обеспечение, обычно применяемое для профессиональных целей (для дизайна рекламы, дизайна промышленной упаковки и т. п.), такое как графические редакторы Adobe Photoshop и CorelDRAW, для которых и разрабатывались фрагменты методик обучения компьютерной графике [3].

С одной стороны, данный подход понятен, так как корпорация Adobe фактически монополизировала профессиональный рынок обработки цифровой фотографии, растровый редактор Photoshop стал программным инструментом в данной области «по умолчанию», но крайне высокая стоимость на лицензию для его использования и отсутствие в рамках основного общего образования целей формирования специалиста в области дизайна делают данный вид программного обеспечения совершенно неподходящим для применения в учебном процессе на уровне основного общего образования. С другой стороны, редактор GIMP не поддерживает цветовой режим CMYK, который используется для подготовки цветных изображений к печати, но в рамках изучения темы «Растровая компьютерная графика» в основной школе данная опция не является принципиальной, в отличие, например, от среднего специального образования по специальности «Дизайн».

В настоящее время ситуация изменилась в лучшую сторону, и, например, в учебнике по информатике для VII класса основной школы авторским коллективом под руководством И. Г. Семакина в разделе «Работа с графическим редактором растрового типа» упоминается свободный редактор GIMP, а в разделе «Работа

с графическим редактором векторного типа» — опять же свободный векторный редактор Inkscape [15].

Аналогично, другой авторский коллектив в лице К. Ю. Полякова и Е. А. Еремина в учебнике по информатике для VII класса в разделе «Обработка фотографий» в качестве программного инструмента предлагает GIMP, отмечая, что этот редактор относится к свободному программному обеспечению и является кроссплатформенным [14]. Действительно, в настоящее время необдуманное использование проприетарного программного обеспечения иностранного производства в учебном процессе может привести к очень печальным последствиям, таким, как отказ в предоставлении (продлении) лицензии на его использование и последующая аварийная перестройка всего учебного процесса с учетом перехода на отечественное свободное программное обеспечение. В частности, такая ситуация сложилась с лицензией на использование программного обеспечения производства корпорации Microsoft в МГТУ им. Н. Э. Баумана в конце 2021 года [9].

2. Выбор оптимального варианта свободного растрового графического редактора для поддержки преподавания темы «Растровая компьютерная графика» на уровне основного общего образования

Как мы уже отмечали ранее в [21], для поддержки преподавания какой-либо информационной технологии в основной школе оптимальным программным обеспечением является следующее:

- отечественное;
- свободное;
- кроссплатформенное;
- обладающее низкими минимальными системными требованиями к аппаратному обеспечению персональных компьютеров;
- не являющееся онлайн-сервисом (т. е. офлайн-ное, не зависящее от подключения к сети Интернет);
- обладающее простым, не перегруженным, интуитивно понятным русифицированным интерфейсом пользователя;
- обладающее достаточно богатым, но не избыточным функционалом;
- обладающее хорошей методической поддержкой в плане его использования в учебном процессе (наличие учебно-методической литературы, интернет-источников).

В статье Википедии, предлагающей сравнительную характеристику растровых графических редакторов [19], представлены 23 свободных графических редактора, но большинство из них являются совершенно не известными в отечественном педагогическом сообществе и не упоминаются в учебно-методической литературе, что является достаточно серьезным барьером для их использования.

В свою очередь, авторы статьи «Анализ бесплатного программного обеспечения для создания растровых изображений» [10] В. П. Назарова, С. А. Таричко

и Д. В. Лучанинов в своей работе сравнивают растровые графические редакторы GIMP, Paint.NET, Krita, Photo Pos Pro по следующим критериям:

- возможность инсталляции в различных операционных системах;
- арсенал инструментов для работы с изображениями;
- поддержка различных цветовых моделей (RGB, CMYK);
- поддержка распространенных форматов графических файлов;
- удобство работы с интерфейсом пользователя программного обеспечения.

В результате проведенного анализа высшую оценку авторов статьи [10] получил редактор GIMP. Авторы отмечают, что GIMP, по сравнению с другими редакторами, имеет множество инструментов для работы с изображениями, поддерживает все операционные системы и большинство распространенных форматов графических файлов. Этот графический редактор обладает всеми необходимыми функциями для обработки изображений и для создания собственных изображений с нуля.

А. В. Кайсина и С. В. Николаев провели аналогичный сравнительный анализ растровых графических редакторов GIMP, Krita и Photo Pos Pro и пришли к выводу в пользу использования редактора GIMP, подчеркивая следующие его **преимущества перед другими растровыми редакторами** [4]:

- является кроссплатформенным, свободно распространяемым графическим редактором;
- в нем есть возможность работы с векторной графикой;
- каждое открытое окно программы является независимым; интерфейс программы легко настраивается под нужды конкретного пользователя;
- существует возможность работать с большим количеством графических форматов;
- можно создавать анимированные изображения (GIF-анимация);
- существует большое количество подключаемых модулей и плагинов, значительно расширяющих функционал GIMP.

Проанализируем **минимальные системные требования графического редактора GIMP** [16]:

- разрядность операционной системы — x86 (32-bit) или x64 (64-bit);
- видеоадаптер — любой с поддержкой 3D;
- видеопамять — от 64 мегабайт;
- жесткий диск — 100 мегабайт;
- оперативная память — 512 мегабайт.

Данные минимальные системные требования означают, что графический редактор GIMP может быть использован также на персональных компьютерах с очень скромными характеристиками аппаратного обеспечения под управлением таких минималистичных отечественных операционных систем, как, например, Simply Linux [25], что существенно расширяет парк компьютеров, которые могут быть использованы для осуществления учебного процесса по теме «Растровая компьютерная графика» в основной школе. Данный момент является особенно актуальным для школ в отдаленных районах,

учащиеся которых могут быть не обеспечены современной мощной компьютерной техникой.

Таким образом, для того чтобы сделать вывод о возможности рекомендовать свободный графический редактор GIMP в качестве оптимального программного средства для поддержки учебного процесса по теме «Растровая компьютерная графика» на уровне основного общего образования, надо проанализировать доступные учебно-методическую литературу и интернет-ресурсы, посвященные вопросу эффективного использования данного программного обеспечения в учебном процессе, что будет сделано в разделе 3 данной статьи.

3. Обзор учебной и учебно-методической литературы, а также интернет-источников, посвященных применению графического редактора GIMP в учебном процессе

GIMP — достаточно давний проект в области свободного программного обеспечения, он был основан в 1995 году и начинался как дипломный проект студентов Университета Калифорнии в Беркли Спенсера Кимбелла (Spencer Kimball) и Питера Маттиса (Peter Mattis). Аббревиатура GIMP расшифровывается как GNU Image Manipulation Program, что подчеркивает, что программа стала частью проекта GNU [6]. За прошедшее время проект GIMP успел заинтересовать довольно большое количество педагогов-исследователей, результатом работы которых стала учебно-методическая литература, посвященная технологии использования графического редактора GIMP в учебном процессе. Здесь хотелось бы особо выделить несколько работ.

Учебное пособие И. А. Хахаева «Свободный графический редактор GIMP. Первые шаги» [23] содержит описание некоторых аспектов технологии обработки и создания изображений, использования контуров и встроенных эффектов преобразования изображений, а также достаточно подробное описание инструментария редактора GIMP.

В самоучителе достаточно известного автора **Д. Н. Колисниченко «GIMP 2 — бесплатный аналог Photoshop для Windows/Linux/Mac OS»** [5] приведено подробное описание элементов интерфейса GIMP, а также его инструментов и их настройки. Даны сведения об элементах растровых изображений — слоях, масках, каналах, контурах, а также о цветовых моделях и форматах файлов изображений. Рассмотрена работа с фотографиями: масштабирование, вращение, кадрирование, устранение «эффекта красных глаз». Представлены основные приемы и инструменты рисования: применение кистей, текстур, градиентов, палитр и фильтров. Показаны методы создания анимации и логотипов, карт изображений.

GIMP обычно по умолчанию используется в качестве растрового графического редактора в большинстве дистрибутивов операционных систем семейства Linux, в том числе в дистрибутиве Ubuntu. В связи с этим в самоучителе по операционной системе Ubuntu 10 «**Ubuntu 10. Библия пользователя**» [6] его автор **Д. Н. Колисниченко** также вкратце рассмотрел основной базовый

функционал редактора GIMP, а именно: изменение размера изображения, изменение графического формата изображения, кадрирование изображения, поворот на произвольный угол.

Особое внимание хотелось бы обратить на учебно-методическое пособие по применению свободного программного обеспечения на уроках информатики **Г. Ю. Пожариной «Свободное программное обеспечение на уроке информатики»** [13], в котором автор рассматривает наиболее популярные на момент издания книги (2010 год) виды СПО (Tux Paint, OpenOffice.org), а также достаточно подробно описывает интерфейс пользователя, функционал редактора GIMP, основные группы инструментов, дает методические рекомендации по использованию GIMP в учебном процессе, приводит свой вариант поурочного планирования по теме «Растровая компьютерная графика». Важно отметить, что в этом пособии его автор приводит конкретные практические задания для отработки технологии использования того или иного функционала графического редактора GIMP, что делает книгу ценной именно в методическом отношении.

Интересна книга **Т. А. Панюковой «GIMP и Adobe Photoshop. Лекции по растровой графике»** [12], представляющая собой опубликованные лекции по курсу «Компьютерные издательские системы», прочитанные автором пособия в Южно-Уральском государственном университете. В книге приведены основные приемы для быстрого создания и редактирования растровых изображений и монтажа фотографий. Рассматриваются также основы создания коллажей, цветовой и тоновой коррекции изображений, применение различных встроенных фильтров.

Если говорить об учебных пособиях для основной школы, то свободный графический редактор GIMP упоминается в **учебнике Л. Ф. Соловьевой по информатике и ИКТ для IX класса** [18], но, к сожалению, описанию технологии применения GIMP автор уделила всего несколько страниц.

Учебное пособие А. Г. Жексенаева «Основы работы в растровом редакторе GIMP (ПО для обработки и редактирования растровой графики)» [2] включает в себя краткий теоретический и практический материал, позволяющий получить представление о работе в графическом редакторе GIMP. Этот материал позволяет на конкретных примерах изучить основные возможности применения различных инструментов GIMP. В пособии также дан материал для самостоятельного контроля.

В англоязычном фундаментальном труде американских авторов **Дж. ван Гумстера (J. van Gumster) и Р. Шимонски (R. Shimonski) «Библия GIMP» («GIMP Bible»)** [26] очень подробно описан весь функционал графического редактора GIMP. Основным недостатком для русскоязычного читателя — это то, что книга написана на английском языке, в процессе чтения приходится переводить специальные термины, и для работы с данной книгой лучше использовать англоязычную версию GIMP. Кроме того, книга довольно дорогая (в феврале 2022 года ее можно было приобрести на «Озоне» за 3510 рублей), но, если вы хотите самостоятельно разобраться в тонкостях инструментария и функционала

рассматриваемого нами программного обеспечения, тогда, с точки зрения автора, затраты оправданны.

В англоязычной книге **Ф. Витта (P. Whitt)** «Начальные приемы ретуши и восстановления фотографий с помощью GIMP» («*Beginning photo retouching and restoration using GIMP*») [27] автор учит читателей, как выполнять цифровую ретушь и восстанавливать поврежденные фотографии с помощью GIMP версии 2.8.

Англоязычный учебник **О. Лекарма (O. Lecarme)** и **К. Дельвар (K. Delvare)** «Книга о GIMP» («*The book of GIMP*») [24] в первой половине знакомит читателей с основными навыками работы с GIMP, такими как изменение размера и обрезка изображений, ретуширование пятен и царапин, а также настройка размера холста и изображения. Иллюстрированные пошаговые инструкции показывают, как:

- улучшить освещение и композицию изображений;
- удалить искажения и шум, чтобы старые и поврежденные фотографии выглядели как новые;
- создавать потрясающие панорамы и цифровые коллажи из серии фотографий;
- создавать, редактировать и экспортировать собственные текстуры, логотипы и анимированные GIF-файлы;
- работать с выделениями, каналами и масками, чтобы редактировать изображения на профессиональном уровне.

Вторая половина книги предлагает исчерпывающий справочник по многим функциям GIMP, включая балансировку цвета, маски, фильтры и плагины.

Когда мы говорим об информационных технологиях, интерес в плане учебных ресурсов представляют сайты, содержащие видеуроки (скринкасты) по данной технологии и соответствующему ей программному обеспечению. Действительно, часто видеурок является более информативным, так как показывает весь процесс создания какого-либо информационного объекта в деталях, тогда как текстовое описание, даже сопровождающееся иллюстрациями, проигрывает по детализации и удобству восприятия. В качестве примера такого обучающего русскоязычного интернет-ресурса для свободного растрового графического редактора GIMP можно привести блог **Антон Ляпина** «Уроки GIMP для начинающих» [20].

Электронный учебник доступен пользователю на **официальном сайте проекта GIMP** в сети Интернет: <https://www.gimp.org/tutorials/>

Бесплатный курс по отечественной операционной системе Альт Образование 9 [1], расположенный на портале дистанционной поддержки пользователей операционных систем семейства Альт, содержит в том числе модуль, посвященный растровому графическому редактору GIMP. Для доступа к материалам курса необходима регистрация на портале. В содержании модуля рассматриваются следующие темы:

- Назначение и запуск GIMP в операционной системе Альт Образование.
- Особенности интерфейса графического редактора GIMP.
- Формат изображений, экспорт изображений в форматы PNG, JPEG.
- Способы создания/загрузки изображений.

- Слои: технология создания GIF-анимации.
- Инструменты выделения: различные способы выделения фрагмента изображения.
- Инструменты рисования: создание простого орнамента.
- Инструменты преобразования.
- Создание снимка экрана в редакторе GIMP.
- Изменение размера изображения.
- Ретуширование изображения.
- Обводка по выделению.

4. Поурочное планирование по теме «Растровая компьютерная графика» с использованием свободного графического редактора GIMP на уровне основного общего образования

В таблице на с. 17 приведены темы уроков по растровой графике с использованием растрового графического редактора GIMP и их основное содержание, а также указано, в какой форме проводится урок.

5. Основное содержание обучения по теме «Растровая компьютерная графика» с использованием свободного графического редактора GIMP на уровне основного общего образования

5.1. Урок 1

Растровая компьютерная графика. Основные характеристики растрового изображения (размер, разрешение, цветовое пространство).

С изложением данного теоретического материала можно познакомиться, например, в учебном пособии Т. А. Панюковой «GIMP и Adobe Photoshop: Лекции по растровой графике» [12].

5.2. Урок 2

Растровые графические редакторы. Типы растровых графических редакторов (свободные, проприетарные). Принципы выбора оптимального типа графического редактора для целей использования в основной школе и для профессиональной деятельности по специальности «Дизайн».

Данный материал достаточно подробно разобран в разделе 2 настоящей статьи.

5.3. Урок 3

Свободный растровый графический редактор GIMP. Официальный сайт проекта: <https://www.gimp.org/>, скачивание дистрибутива GIMP, его установка под различными операционными системами (Windows, Linux, MacOS). Достоинства и недостатки. Основные элементы интерфейса пользователя.

Данный материал подробно рассмотрен, например, в главах 1 и 2 учебного пособия Д. Н. Колисниченко «GIMP 2 — бесплатный аналог Photoshop для Windows/Linux/Mac OS» [5], а также в разделе 2 настоящей статьи.

Поурочное планирование по теме «Растровая компьютерная графика»

№ урока	Тема и основное содержание урока	Тип урока
1	Растровая компьютерная графика. Основные характеристики растрового изображения (размер, разрешение, цветовое пространство).	Лекция
2	Растровые графические редакторы. Типы растровых графических редакторов (свободные, проприетарные). Принципы выбора оптимального типа графического редактора для целей использования в основной школе и для профессиональной деятельности по специальности «Дизайн».	Лекция
3	Свободный растровый графический редактор GIMP. Официальный сайт проекта: https://www.gimp.org/ , скачивание дистрибутива GIMP, его установка под различными операционными системами (Windows, Linux, MacOS). Достоинства и недостатки. Основные элементы интерфейса пользователя.	Лекция
4	Открытие исходных и сохранение измененных изображений. Изменение размера изображения. Инструмент перемещения и инструменты выделения фрагмента изображения (прямоугольное и эллиптическое выделение). Создание нового изображения из буфера обмена, содержащего фрагмент исходного изображения. Технология применения инструмента <i>Штамп</i> для коррекции изображения. <i>Практическая работа № 1 «Птицы».</i>	Семинар, практическая работа
5	Различные инструменты выделения областей изображений (<i>Прямоугольное выделение, Эллиптическое выделение, Свободное выделение, Выделение переднего плана, Ножницы, Выделение смежных областей, Выделение по цвету</i>) и их использование на практике. Создание новых изображений (коллажей) на основе выделенных фрагментов. Удаление монохромного и полихромного фона в изображении. <i>Практическая работа № 2 «Рекламный постер “Минеральная вода”.</i> Подготовка графических фрагментов для создания рекламного постера «Минеральная вода».	Семинар, практическая работа
6	Слои в изображении. Свойства слоев. Операции со слоями. Инструменты преобразования (<i>Искавление, Вращение, Перспектива, Зеркало, Масштаб, Перемещение</i>). <i>Практическая работа № 2 «Рекламный постер “Минеральная вода” (продолжение).</i> Создание рекламного постера «Минеральная вода» с использованием нескольких слоев, инструментов рисования (<i>Ластик</i>) и преобразования.	Семинар, практическая работа
7	Работа с текстом в GIMP. Редактор текста, параметры текста. Экспорт изображения в различные графические форматы (JPEG и т. д.). <i>Практическая работа № 2 «Рекламный постер “Минеральная вода” (окончание).</i>	Семинар, практическая работа
8	Кистевые инструменты рисования (<i>Карандаш, Кисть</i>). Рисование кистями, свойства кистей. Создание собственных кистей. Инструменты <i>Плоская заливка</i> и <i>Градиент</i> . Инструменты <i>Размазывание</i> (палец), <i>Размытие — Резкость, Осветление — Затемнение</i> . <i>Практическая работа № 3 «Создание изображений “Текст под снегом”.</i>	Семинар, практическая работа
9	<i>Практическая работа № 4 «Создание изображения “В лесу”.</i>	Практическая работа
10	Использование масок в изображениях. Изменение прозрачности пикселей определенного фрагмента изображения. <i>Практическая работа № 5 «Создание социальной рекламы».</i>	Семинар, практическая работа
11	Встроенные фильтры графического редактора GIMP. Основные группы фильтров. Примеры применения различных фильтров к изображениям.	Семинар, практическая работа
12	Вставка в растровое изображение векторных фрагментов. <i>Практическая работа № 4 «Создание изображения “В лесу” (окончание).</i>	Семинар, практическая работа
13	Контуры. Создание выделений из контуров. Создание рисунков с помощью контуров. <i>Практическая работа № 6 «Создание новогодней открытки».</i>	Семинар, практическая работа
14	Создание анимированных изображений средствами GIMP. <i>Практическая работа № 7 «Бегающий человек».</i>	Семинар, практическая работа

5.4. Урок 4

*Открытие исходных и сохранение измененных изображений. Изменение размера изображения. Инструмент перемещения и инструменты выделения фрагмента изображения (прямоугольное и эллиптическое выделение). Создание нового изображения из буфера обмена, содержащего фрагмент исходного изображения. Технология применения инструмента **Штамп** для коррекции изображения.*

Практическая работа № 1 «Птицы».

На данном этапе учащиеся выполняют практическую работу № 1, заключающуюся в открытии в редакторе GIMP изображения большого размера, содержащего несколько объектов (например, несколько птиц, сидящих на ветке), и в последующем уменьшении размера с помощью команды *Изображение, Изменить размер*. Полученное новое изображение с уменьшенным разрешением сохраняется во внутреннем формате редактора GIMP xcf. Далее учащиеся выделяют один из объектов с помощью инструментов *Прямоугольное выделение* и *Эллиптическое выделение* и создают на их основе новые изображения с помощью команды *Создать, Из буфера обмена*, затем сохраняют их во внутреннем формате редактора GIMP xcf.

Полученные таким образом изображения могут включать части, содержащие скопированные ненужные фрагменты, например части соседних птиц, как показано на рисунке 1.

Такие фрагменты желательно перекрасить в цвет фона для придания полученному изображению законченности. Для этих целей рационально использовать инструмент *Штамп*. Этот инструмент копирует цвета определенной группы пикселей (обычно фрагмента фона) в цвета выбранной области в зависимости от формы штампа. В графическом редакторе GIMP для запоминания цветов пикселей выбранной зоны необходимо нажать клавишу Ctrl, затем, отпустив Ctrl, можно переносить скопированные пиксели в выбранную зону изображения многократным нажатием левой кнопки



Рис. 1. Новое изображение, созданное на основании вырезанного фрагмента



Рис. 2. Изображение попугая после удаления ненужных фрагментов с помощью **Штампа**

мыши (в таком случае при перемещении мыши зона копирования не перемещается) либо с зажатой левой кнопкой мыши (в таком случае при перемещении мыши зона копирования перемещается синхронно с перемещением мыши). Результат применения инструмента *Штамп* к изображению попугая, первоначально содержащему ненужные фрагменты (рис. 1), показан на рисунке 2.

5.5. Урок 5

*Различные инструменты выделения областей изображений (**Прямоугольное выделение**, **Эллиптическое выделение**, **Свободное выделение**, **Выделение переднего плана**, **Ножницы**, **Выделение смежных областей**, **Выделение по цвету**) и их использование на практике. Создание новых изображений (коллажей) на основе выделенных фрагментов. Удаление монохромного и полихромного фона в изображении.*

Практическая работа № 2 «Рекламный постер «Минеральная вода»». Подготовка графических фрагментов для создания рекламного постера «Минеральная вода».

На данном этапе учащиеся приступают к выполнению практической работы № 2, целью которой является создание рекламного постера «Минеральная вода». Для этого учащимся необходимо подготовить изображения пустыни в заданном разрешении, а также отделить от монохромного (белого) и полихромного фона изображение бутылки минеральной воды и сохранить полученные фрагменты в отдельных файлах в формате xcf.

Для отделения изображения от монохромного (белого) фона в GIMP используется инструмент *Выделение смежных областей*, который выделяет соседние пиксели одного цвета.

Таким образом, для отделения изображения бутылки минеральной воды от белого фона можно реализовать следующий алгоритм:

1. Открываем первоначальное изображение бутылки с минеральной водой в GIMP с помощью команды *Правка, Вставить как, Новое изображение*.
2. Щелкаем инструментом *Выделение смежных областей* по белому фону, в результате чего все белые пиксели попадают в выделение.
3. Инвертируем выделение с помощью команды *Выделение, Инвертировать*, в результате чего выделяется сама бутылка.
4. После этого с помощью команды *Правка, Копировать* можем скопировать только изображение самой бутылки.

Отметим, что в редакторе GIMP выделенный в текущий момент фрагмент изображения отмечается движущейся пунктирной линией.

Отделение объекта от полихромного (пестрого) фона можно реализовать с помощью инструмента *Свободное выделение*, который позволяет выделить произвольную область изображения, обводя ее курсором мыши. Данный метод является более трудоемким по сравнению с отделением от белого фона и реализуется следующим алгоритмом:

1. Открываем первоначальное изображение объекта, которое планируется отделить от фона.

- С помощью инструмента *Свободное выделение* проходим по границе изображения объекта, при необходимости меняя масштаб изображения, замыкаем выделение.
- Теперь с помощью команды *Правка, Копировать* можем скопировать выделенный фрагмент изображения.

5.6. Урок 6

Слои в изображении. Свойства слоев, операции со слоями. Инструменты преобразования (Искавление, Вращение, Перспектива, Зеркало, Масштаб, Перемещение).

Практическая работа № 2 «Рекламный постер «Минеральная вода»» (продолжение). Создание рекламного постера «Минеральная вода» с использованием нескольких слоев, инструментов рисования (Ластик) и преобразования.

На данном этапе изучения методики создания и редактирования растровых графических изображений в свободном графическом редакторе GIMP учащиеся продолжают выполнять практическую работу № 2, в ходе которой компилируют из ранее созданных фрагментов (изображений бутылки минеральной воды) рекламный постер «Минеральная вода». В процессе создания коллажа учащиеся знакомятся с парадигмой разработки растровых графических изображений, состоящих из множества слоев, содержащих различные фрагменты изображения и обладающих разным уровнем прозрачности.

Данная парадигма предполагает следующие подходы к использованию слоев в растровом изображении [17]:

- В любом документе можно создавать произвольное количество слоев, их число ограничивается лишь объемом оперативной памяти компьютера. На каждом слое может располагаться любое количество объектов. Сквозь прозрачные участки слоя видны объекты, расположенные на нижних слоях.
- Можно отображать или скрывать отдельные слои изображения, можно защищать слои, а также изменять порядок взаимного расположения слоев.
- Существует возможность дублирования слоев. При этом создается новый слой, который помещается над исходным и принимает все его параметры. На дублированном слое размещаются копии всех объектов оригинала.
- Существует возможность объединения нескольких слоев в один. Объединение слоев заключается в переносе всех слоев на самый верхний из них.
- Существует возможность перемещения объектов с одного слоя на другой с помощью стандартных функций копирования/вставки.
- Существует возможность изменения порядка расположенных слоев и удаления ненужных.
- Существует возможность создания корректирующих слоев, что позволяет применять к отдельным слоям или группам слоев различные методы цветокоррекции. Благодаря этому можно свободно экспериментировать с комбинациями графиче-

ских элементов, спецэффектами и др., не изменяя при этом фактического содержания слоев.

В ходе создания коллажа учащиеся вставляют в открытое ранее изображение пустыни предварительно сохраненные в отдельных файлах фрагменты изображения бутылки с минеральной водой, а также изображение автомата для газированной воды в качестве новых слоев с использованием команды *Вставить как, новый слой*. Границы вставленного фрагмента могут быть подкорректированы простейшим из инструментов рисования — *Ластиком*. Далее графический фрагмент на вставленном слое может быть при необходимости отредактирован с помощью инструментов преобразования, а именно:

- его размер и пропорции могут быть изменены с помощью инструмента *Масштаб*;
- положение фрагмента относительно других слоев может быть изменено с помощью инструмента *Перемещение*;
- фрагмент можно повернуть на произвольный угол с помощью инструмента *Вращение*, и т. д.

Результат создания коллажа показан на рисунке 3 в окне программы GIMP. На этом рисунке в панели слоев справа показаны слои, на которых находятся изображения бутылки с минеральной водой (название слоя — «минеральная вода») и автомата для газированной воды (название слоя — «автомат»).

5.7. Урок 7

Работа с текстом в GIMP. Редактор текста, параметры текста. Экспорт изображения в различные графические форматы (JPEG и т. д.).

Практическая работа № 2 «Рекламный постер «Минеральная вода»» (окончание).

На этом уроке учащиеся заканчивают создание коллажа «Минеральная вода» в ходе выполнения практической работы № 2.

С помощью инструмента *Текст* они создают рекламный слоган на постере. Отметим, что при создании текста в GIMP он автоматически размещается на новом слое, что позволяет свободно перемещать его и поворачивать относительно группы слоев с изображениями.

Инструмент *Текст* позволяет менять кегль вводимого текста, начертание и цвет. На рисунке 3 показан результат вставки текста на рекламный постер «Минеральная вода», справа внизу в панели слоев виден слой, на котором расположен текст, имя слоя совпадает с текстом.

5.8. Урок 8

Кистевые инструменты рисования (Карандаш, Кисть). Рисование кистями, свойства кистей. Создание собственных кистей. Инструменты Плоская заливка и Градиент. Инструменты Размазывание (палец), Размытие — Резкость, Осветление — Затемнение.

Практическая работа № 3 «Создание изображений «Текст под снегом»».

Несмотря на то что свободный растровый редактор GIMP в первую очередь предназначен для редактирования цифровых фотографий, в нем также можно

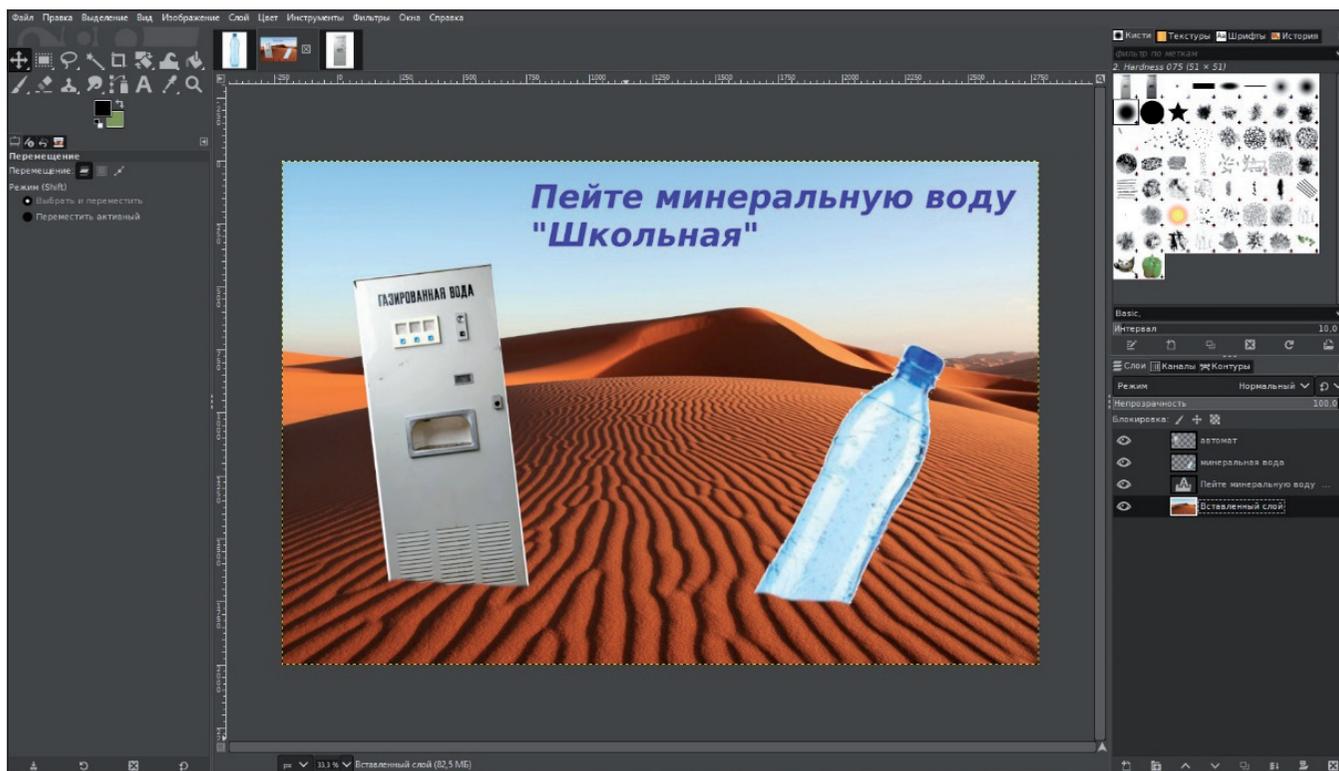


Рис. 3. Окно программы GIMP с созданным коллажем «Минеральная вода»

создавать собственные изображения полностью с нуля с использованием инструментов рисования. Очевидно, что для этого пользователь должен иметь хотя бы базовые навыки рисования кистями, которые приобретаются учащимися на уроках изобразительного искусства.

Рассмотрим простейшую технику использования кистей и градиентов при создании изображения «Текст под снегом», показанного на рисунке 4, в ходе выполнения практической работы № 3, алгоритм которой заключается в следующем:

1. Создаем изображение и заливаем фон градиентом (голубой цвет, переходящий в темно-синий).
2. Пишем произвольный крупный текст с помощью инструмента *Текст* (в нашем случае слово «Холод»).
3. Устанавливаем цвет переднего плана — белый, выбираем круглую кисть с мягкими краями и с ее помощью создаем имитацию накопившегося снега в соответствующих местах букв.
4. С помощью инструмента *Размазывание* имитируем эффект сползания снега вниз.

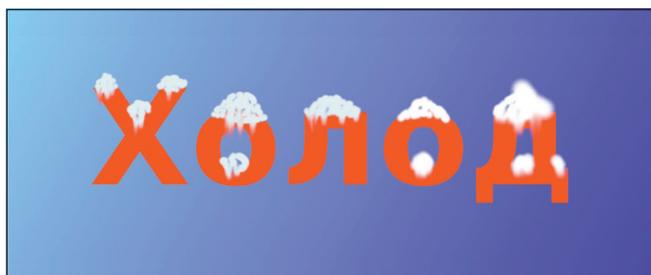


Рис. 4. Изображение «Холод», снег создан с помощью круглой кисти белого цвета

5.9. Урок 9

Практическая работа № 4 «Создание изображения «В лесу»».

Более сложная техника использования кистей, а также инструментов *Размазывание*, *Осветление* отрабатывается учащимися при создании изображения «В лесу» в ходе проведения практической работы № 4, идея которой взята автором в книге Т. А. Панюковой «GIMP и Adobe Photoshop. Лекции по растровой графике» [12].

Практическая работа № 4 проводится по следующему алгоритму:

1. Создаем новый файл в редакторе GIMP размером 1920 на 1080 пикселей с единственным прозрачным слоем (устанавливается в разделе *Расширенные параметры* диалогового окна *Создать новое изображение*) с помощью команды *Файл, Создать проект*.
2. Создаем новый прозрачный слой, рисуем круглой кистью размером примерно 25 пикселей контур облака, заливаем его тем же цветом переднего плана (рис. 5). Прозрачные пиксели в GIMP обозначаются, как фон на рисунке 5, в виде шахматных клеток.
3. Внутри облака рисуем такой же контур и заливаем его более темным оттенком того же цвета (рис. 5).
4. С помощью инструмента *Размазывание* размером примерно 50 пикселей круговыми движениями придаем облаку натуральный эффект размытости (рис. 6).
5. Дублируем данный слой с помощью команды *Создать копию слоя* в контекстном меню для слоя с первым облаком и сдвигаем его немного в сторону с помощью инструмента *Перемещение*.

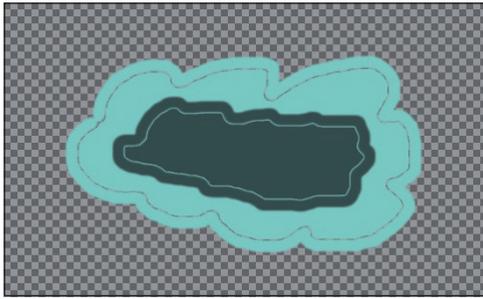


Рис. 5. Первая стадия рисования облака

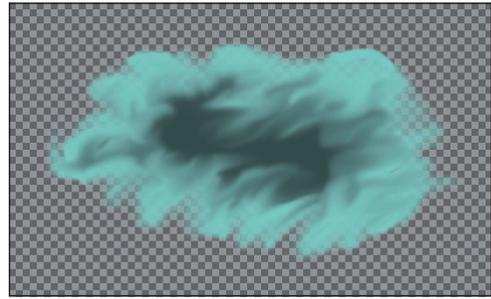


Рис. 6. Вторая стадия рисования облака

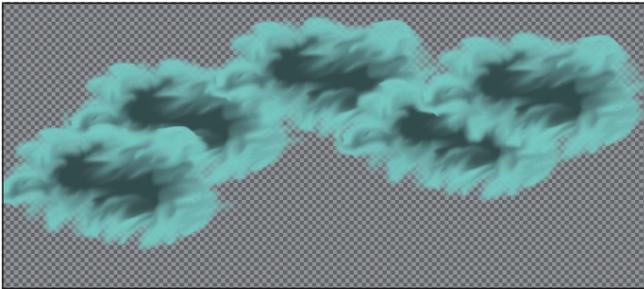


Рис. 7. Результат множественного копирования слоя с облаком

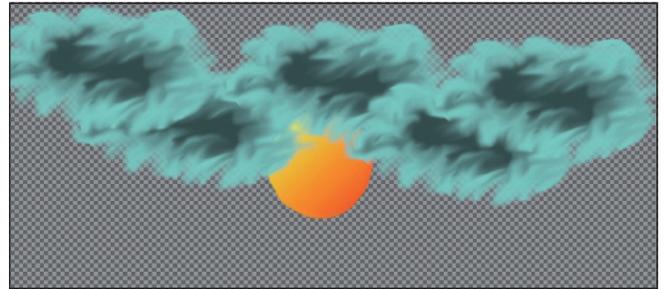


Рис. 8. Солнце расположено на отдельном слое

- Повторяем данную процедуру несколько раз. Таким образом, каждое облако оказывается на своем персональном отдельном слое. Примерный результат показан на рисунке 7.
6. Создаем новый прозрачный слой, с помощью инструмента *Эллиптическое выделение* создаем выделение в форме круга, заливаем его градиентом красно-оранжевых цветов.
 7. Снимаем выделение, проходим инструментом *Размазывание* по границе круга.
 8. Перемещаем слой с солнцем на задний план в панели слоев, с помощью инструмента *Перемещение* располагаем оптимальным образом относительно облаков (рис. 8).
 9. Переходим к рисованию ствола и ветвей дерева, для чего создаем новый прозрачный слой. Остальные слои можно временно скрыть с помощью кнопки с изображением глаза, расположенной слева от соответствующего слоя в панели слоев. Для рисования дерева будем использовать круглую кисть, размер которой можно менять в зависимости от толщины веток, которые мы изображаем (рис. 9).
 10. С помощью инструмента *Осветление — Затемнение* постараемся придать эффект объемности нарисованному на предыдущем шаге дереву. Интенсивность осветления зависит от количества проходов инструментов по заданному месту изображения. После осветления рационально использовать инструмент *Размазывание* для обеспечения плавности перехода оттенков (рис. 10).
 11. Для оформления кроны дерева используем технологию создания собственных кистей. Создадим кисть в форме кленового листа. Для этого откроем в редакторе GIMP изображение кленового листа на белом фоне с помощью команды *Правка, Вставить как, Новое изображение*, отделим лист от фона, скопируем его и сохраним как новую кисть с помощью команды *Правка, Вставить как, Новая кисть*. После чего новая кисть добавится в палитру кистей (рис. 11).
 12. Для рисования кроны создадим новый прозрачный слой, в панели слоев дадим ему имя «крона» и с помощью кисти в форме кленового листа нарисуем крону дерева. При использовании кисти в форме кленового листа имеет смысл периодически изменять размер кисти и угол поворота для придания естественной формы кроне дерева (рис. 12).
 13. Далее на новом прозрачном слое нарисуем с помощью кисти траву в форме кустика травы, действуя по алгоритму создания пользовательских кистей, описанному в п. 11. Дадим данному слою имя «травы», расположим слой «травы» перед слоем «дерево».
 14. Создадим копию слоя «травы» и поместим ее за слоем «дерево» так, чтобы дерево было окружено травой (рис. 13).
 15. Создадим слой «небо», зальем его светло-голубым цветом и расположим на самом заднем плане в панели слоев. Результат вместе с панелью слоев показан на рисунке 13.
 16. Добавим на созданный нами пейзаж какое-нибудь живое существо, например немецкую овчарку, которую отделим от полихромного фона с помощью инструмента *Свободное выделение*, а затем вставим в наш проект с помощью команды *Правка, Вставить как, Новый слой*. Границу изображения доработаем с помощью инструмента *Ластик* так, чтобы убрать все лишние элементы. При этом рационально менять масштаб обрабатываемого изображения для обработки мелких деталей.

17. Таким же образом вставим в наш проект в качестве нового слоя изображение летящей совы, предварительно отделив его от полихромного фона. Результат показан на рисунке 14.

На рисунках 15 и 16 приведены примеры работ учеников восьмого класса одной из московских школ, изучавших GIMP по предлагаемой в настоящей статье методике.

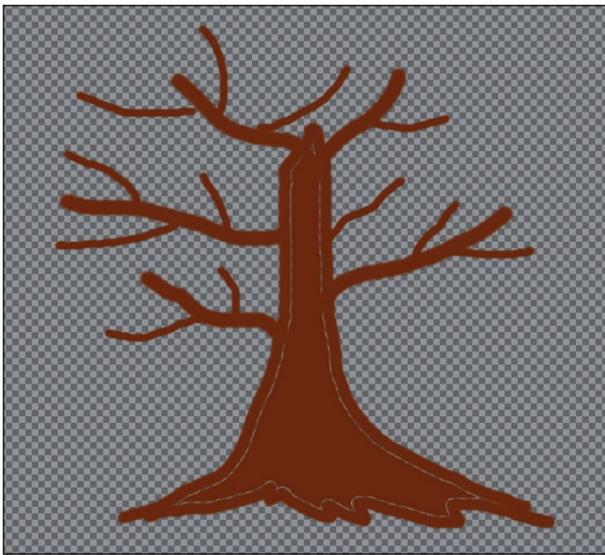


Рис. 9. Первая стадия рисования ствола и ветвей дерева

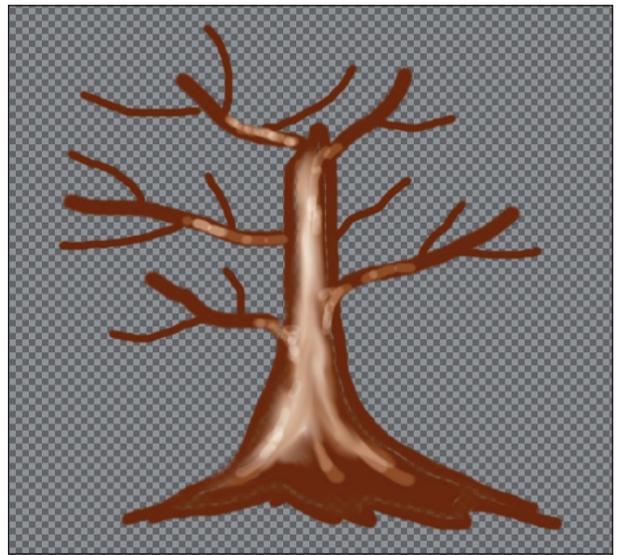


Рис. 10. Вторая стадия рисования ствола и ветвей дерева

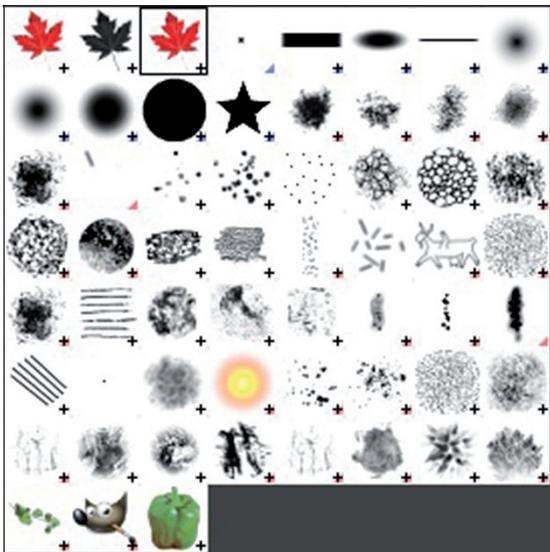


Рис. 11. Новая кисть в форме кленового листа добавилась в палитру кистей

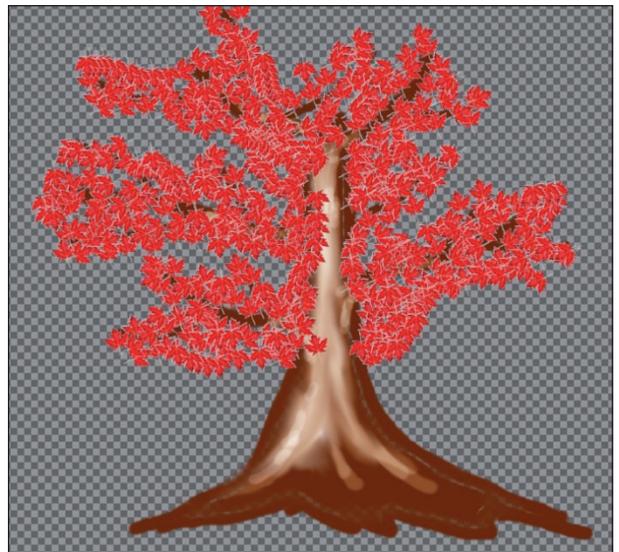


Рис. 12. Крона дерева сформирована кистью в форме кленового листа



Рис. 13. Пейзаж с деревом и палитра слоев справа



Рис. 14. Пейзаж с деревом с добавленными в него изображениями овчарки и совы



Рис. 15. Первый образец работы ученика VIII класса



Рис. 16. Второй образец работы ученика VIII класса

5.10. Урок 10

Использование масок в изображениях. Изменение прозрачности пикселей определенного фрагмента изображения.

Практическая работа № 5 «Создание социальной рекламы».

Маски являются эффективным инструментом выделения некоего фрагмента (фрагментов) в обрабатываемом изображении для последующего изменения прозрачности пикселей выделенного фрагмента.

Проиллюстрируем технологию применения масок в ходе практической работы № 5, целью которой будет обесцвечивание фона изображения, приведенного на рисунке 16.

Работа проводится по следующему алгоритму:

1. Откроем в графическом редакторе GIMP пейзаж (рис. 16).
2. В панели слоев создадим копию единственного слоя изображения с помощью пункта контекстного меню *Создать копию слоя* (вызывается правой кнопкой мыши).
3. Изменим цветность скопированного слоя с помощью команды *Цвет, Обесцвечивание, Обесцвечивание*.
4. К обесцвеченному слою добавим маску с помощью контекстного меню — команда *Добавить маску слоя*. При добавлении маски инициализируем ее белым цветом, который означает полную первоначальную непрозрачность всех пикселей слоя, к которому добавляется маска. После добавления маски в панели слоев и пиктограммы слоя появится белый прямоугольник, символизирующий добавленную маску.

5. Установим в качестве цвета переднего плана черный, выберем инструмент *Кисть* (желательно с мягкими краями), выделим в панели слоев маску слоя с помощью левой кнопки мыши и начнем кистью закрашивать дерево на изображении. Пиксели, выделенные черным цветом, становятся полностью прозрачными. В случае выделения лишних пикселей в маске их можно сделать непро-



Рис. 17. Социальная реклама, полученная на основе изображения дерева

зрачными, вновь закрасив белым цветом. На пиктограмме маски слоя в панели слоев появляется контур закрашиваемого дерева.

6. Так как нижний слой изображения остается полноцветным, то под ставшими прозрачными пикселями верхнего обесцвеченного слоя мы видим цветные пиксели, в результате чего дерево на изображении будет в цвете, а фон остается обесцвеченным.
7. Для придания смысловой законченности созданному нами изображению превратим его в социальную рекламу, добавив слоган «Берегите природу!» с помощью инструмента *Текст*. Результат показан на рисунке 17.

5.11. Урок 11

Встроенные фильтры графического редактора GIMP. Основные группы фильтров. Примеры применения различных фильтров к изображениям.

Свободный графический редактор GIMP содержит большое количество встроенных фильтров, которые используются для создания некоторого сложного эффекта, применяемого к создаваемому изображению или отдельному фрагменту изображения.

Фильтры в GIMP рассортированы по следующим группам:

- *Размытие* — группа фильтров, которая тем или иным способом размывает все изображение или его часть.
- *Улучшение* — группа фильтров используется для устранения дефектов растрового изображения, таких как пыль, шум, недостаточная четкость.
- *Искажения* — группа фильтров, которые изменяют геометрию изображения различными способами.
- *Свет и тень* — группа фильтров, обеспечивающих создание эффектов освещения и теневых эффектов.
- *Шум* — группа фильтров, добавляющих шумовые эффекты к изображению в целом или выделенному фрагменту.
- *Выделение края* — группа фильтров, которая обеспечивает поиск границ между различными

цветами, которые рассматриваются как границы объектов на изображении.

- *Общие* — к данной группе относятся фильтры, которые невозможно отнести ни к какой другой группе, такие как *Матрица свертки*, *Эрозия* и т. п.
- *Объединение* — к данной группе относятся фильтры, которые тем или иным способом объединяют два или более изображений в одно.
- *Имитация* — данная группа объединяет фильтры, которые обеспечивают имитацию данным изображением какого-либо стандартного стиля в живописи, например *Комикс*, *Кубизм*, *Ван Гог*, *Масляная краска*.
- *Декорация (оформление)* — группа фильтров, зависящих от конкретного изображения; они добавляют декоративные рамки и специальные эффекты к изображению.
- *Проекция* — группа фильтров, реализующих проектирование изображения или фрагмента изображения на какой-либо объект, например, на сферу или рельеф. Таким образом можно создавать в изображении трехмерные эффекты.
- *Визуализация* — очень интересная группа фильтров, с помощью которых можно создавать новые изображения, например, визуализировать различные виды фракталов или шумов. На рисунке 18 показан визуализированный фрактал.
- *Веб* — группа фильтров для интернета; чаще всего используются при работе с изображениями, предназначенными для оформления веб-сайтов. Фильтр *ImageMap* используется для добавления в изображение чувствительных к воздействию мышью «горячих точек». Фильтр *Полусведение* используется для симуляции полупрозрачности форматов изображений без добавления альфа-канала.
- *Анимация* — фильтры, вспомогательные для оптимизации GIF-анимации.

С нашей точки зрения, искусство художника заключается в выборе из имеющегося арсенала наиболее эффективного для данного изображения фильтра, подчеркивающего и акцентирующего сильные стороны рисунка. Действительно, практика показывает, что



Рис. 18. Пейзаж до обработки фильтром

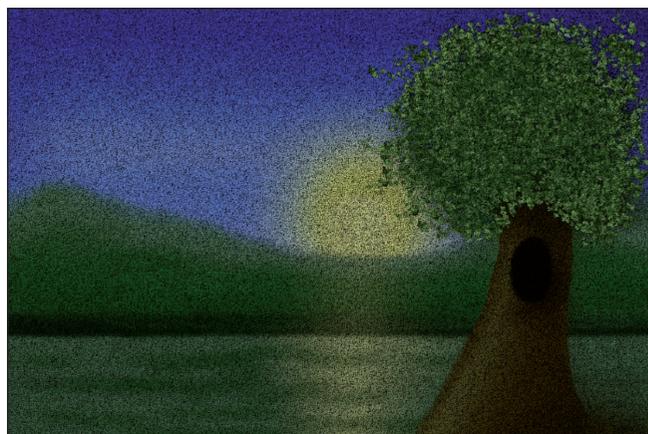


Рис. 19. Пейзаж, показанный на рисунке 18, с примененным к нему фильтром *Холст*

применение большей части фильтров ухудшает изображение, другие фильтры практически не меняют изображение и только малая часть фильтров могут увеличить его художественную ценность.

Для пейзажа, показанного на рисунке 18, эффективным будет применение фильтров из группы *Имитация*, например фильтра *Холст*, в результате чего мы получаем имитацию картины, написанной на холсте, как показано на рисунке 19.

5.12. Урок 12

Вставка в растровое изображение векторных фрагментов.

Практическая работа № 4 «Создание изображения “В лесу”» (окончание).

Интересной опцией в графическом редакторе GIMP является возможность вставки в растровое изображение векторных фрагментов. Источником векторных графических фрагментов может являться любой векторный редактор, например LibreOffice Draw, работа с которым подробно описана автором (в соавторстве с А. Ю. Федосовым) в статье [22].

Продемонстрируем данную возможность на уже созданном нами ранее лесном пейзаже:

1. Нарисуем домик в векторном редакторе LibreOffice Draw и скопируем его в буфер обмена.
2. Затем вставим его в лесной пейзаж с помощью команды *Правка, Вставить как, Новый слой*, в случае необходимости изменим масштаб с помощью инструмента *Масштаб* и переместим так, чтобы он был между слоями, содержащими траву (рис. 20).

5.13. Урок 13

Контур. Создание выделений из контуров. Создание рисунков с помощью контуров.

Практическая работа № 6 «Создание новогодней открытки».

Контур является одним из инструментов GIMP, с помощью которого можно создавать выделение.

Рассмотрим технологию использования контуров в процессе создания поздравительной новогодней открытки в ходе практической работы № 6, реализуемой по следующему алгоритму:

1. Создадим новый файл размером 1920 на 1080 пикселей с прозрачным заполнением с помощью команды *Файл, Создать проект*.
2. С помощью инструмента *Контур* создадим контур, показанный на рисунке 21, следующим способом:
 - щелкаем левой кнопкой мыши и создаем первую опорную точку контура;
 - сдвигаемся вниз относительно этой точки, еще раз щелкаем левой кнопкой мыши и ставим вторую опорную точку контура;
 - возвращаемся к первой точке, нажимаем клавишу Ctrl и замыкаем контур;
 - сдвигаем с помощью левой кнопки мыши вертикальный отрезок контура влево, а затем второй отрезок контура вправо; в процессе сдвигания с помощью нажатой левой кнопки мыши отрезок изгибается.
3. Создаем выделение из контура с помощью команды *Выделение, Из контура*.
4. Устанавливаем в качестве цвета переднего плана светло-серый цвет, а в качестве цвета фона — средне-серый цвет и заливаем выделение с помощью инструмента *Градиент* с формой *Очертание (сфера)*.
5. Снимаем выделение с помощью команды *Выделение, Снять выделение*. Результат показан на рисунке 22.
6. Аналогичным образом (п. 2–5) создаем среднюю и верхнюю части снеговика, а также ноги



Рис. 20. Лесной пейзаж со вставленным в него «векторным» домиком

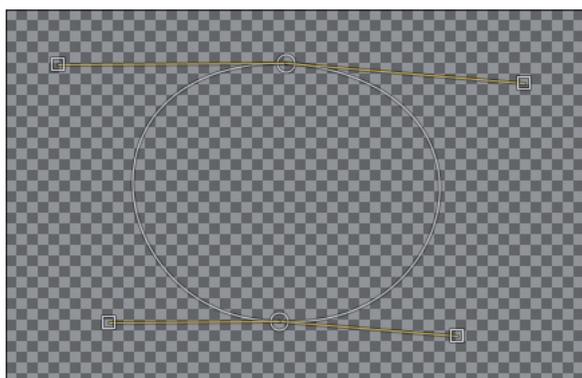


Рис. 21. Создание контура для нижней части снеговика

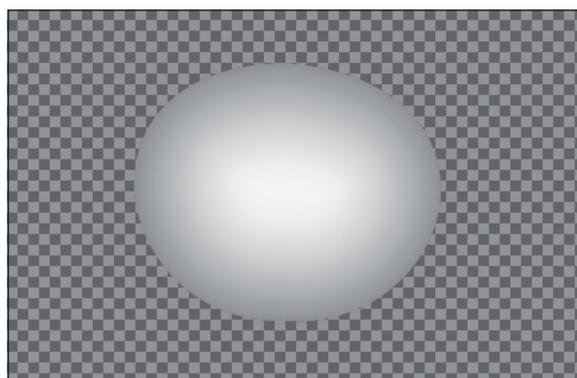


Рис. 22. Нижняя часть снеговика

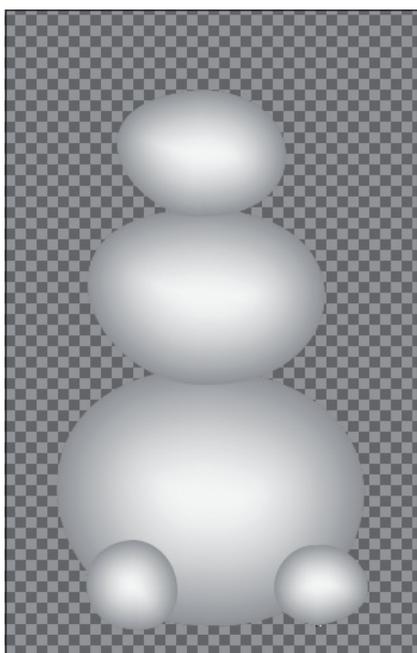


Рис. 23. Снеговик

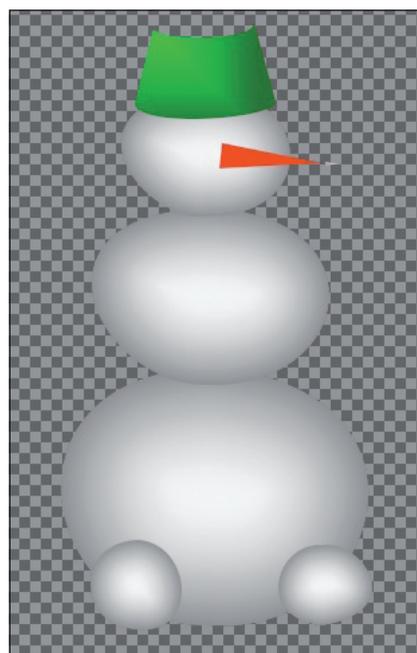


Рис. 24. Снеговик с ведром и носом



Рис. 25. Добавили глаза, рот, руки и градиент



Рис. 26. Добавили шарф



Рис. 27. Законченная новогодняя открытка

снеговика. При этом каждая из частей снеговика располагается на отдельном прозрачном слое (рис. 23).

7. Далее нарисуем ведро на голове снеговика с помощью создания выделения из контура, а также нос снеговика. Каждый объект создается на своем слое (рис. 24).
8. Создадим новый прозрачный слой, на котором с помощью кисти нарисуем глаза, рот и руки снеговика.
9. Создадим новый прозрачный слой, который зальем градиентом голубого цвета, переходящего в синий, и переместим его в панели слоев на задний план (рис. 25).
10. Нарисуем шарф снеговика на отдельном слое также с помощью технологии выделений, получаемых из контуров (рис. 26).
11. Создадим надпись «С Новым годом!», добавим на отдельном слое эффект имитации снега (аналогично изображенному на рисунке 4). Законченная работа показана на рисунке 27.

5.14. Урок 14

Создание анимированных изображений средствами GIMP.

Практическая работа № 7 «Бегущий человек».

Идея, положенная в основу создания GIF-анимации, очень проста: в изображении на разных прозрачных слоях располагают различные стадии состояния некоего движущегося объекта, которые воспроизводятся с заданным интервалом, выраженным в миллисекундах.

Продemonстрируем данную технологию в редакторе GIMP при создании GIF-анимации «Бегущий человек» в ходе практической работы № 7.

1. С помощью команды *Файл, Создать проект* создадим новый файл размером 1920 на 1080 пикселей с прозрачным заполнением.
2. С помощью инструмента *Кисть* схематично нарисуем бегущего человека. Для прочерчивания прямой линии в GIMP используется клавиша Shift (рис. 28).
3. Скопируем слой с данным изображением с помощью команды контекстного меню в панели слоев *Создать копию слоя*.
4. Выделим руку человека инструментом *Свободное выделение* и вырежем с помощью команды *Правка, Вырезать*, а потом вставим с помощью команды *Вставить как, Новый слой*.
5. Вставленный фрагмент изображения (например, рука бегущего человека) далее должен быть повернут и перемещен так, чтобы была получена следующая стадия (форма) бегущего человека (рис. 29, 30).
6. Для создания GIF-анимации надо выполнить экспорт файла в формат GIF с помощью команды *Файл, Экспортировать как, Изображение GIF*, в диалоговом окне поставить галочку *Сохранить как анимацию*, а в пункте *Расположение кадра — Один кадр на слой (замена)*. Задержку между кадрами рационально ставить примерно в 200 миллисекунд.
7. Созданный файл в формате GIF может быть просмотрен через любой интернет-браузер.



Рис. 28. Бегущий человек, стадия 1



Рис. 29. Бегущий человек, стадия 2



Рис. 30. Бегущий человек, стадия 3

6. Выводы

Свободный графический растровый редактор GIMP в настоящее время является, с точки зрения автора, оптимальным программным средством для преподавания растровой графики на уровне основного общего образования.

К его достоинствам можно отнести следующие:

- Графический растровый редактор GIMP относится к свободному программному обеспечению, распространяется согласно условиям Стандартной общественной лицензии GNU (GNU General Public License).
- GIMP является кроссплатформенным ПО, имеет сборки для всех основных типов операционных систем (Windows, Linux, Mac OS).
- GIMP обладает низкими системными требованиями к аппаратному обеспечению, на котором планируется его установка.
- Существует большое количество учебно-методической литературы, учебных онлайн-курсов, а также интернет-источников, посвященных технологии эффективного использования редактора GIMP в учебном процессе.
- Функционал графического редактора GIMP постоянно дорабатывается и развивается командой разработчиков, в настоящий момент на официальном сайте проекта доступна версия 2.10.30.
- Функционал GIMP позволяет использовать его как для обработки цифровой фотографии, так и для создания собственных изображений «с нуля».
- Графический редактор GIMP является полнофункциональным программным обеспечением, вполне сравнимым с проприетарными профессиональными аналогами типа Adobe Photoshop; в нем используются все основные инструменты растровой графики, такие как слои, контуры, каналы, встроенные фильтры и т. п.

Графический редактор GIMP обладает достаточно сложным интерфейсом пользователя, но тем не менее учащийся основной школы вполне может освоить его при условии использования эффективной методики обучения растровой графике, учитывающей особенности использования данного программного обеспечения.

Список источников

1. Графический редактор GIMP // ООО «Базальт СПО» — российский разработчик операционных систем «Альт». <https://kurs.basealt.ru/mod/lesson/view.php?id=2662>
2. Жексенаев А. Г. Основы работы в растровом редакторе GIMP (ПО для обработки и редактирования растровой графики): учебное пособие. М., 2008. 80 с.
3. Залогова Л. А. Компьютерная графика. Практикум. 2-е изд. М.: Лаборатория Базовых Знаний, 2005. 320 с.
4. Кайсина А. В., Николаев С. В. GIMP как средство обучения компьютерной графике в условиях внедрения свободного программного обеспечения // XXIII Царскоельские чтения. Материалы международной научной конференции. Т. II / отв. ред. А. А. Беляева. СПб.: Ленинградский государственный университет имени А. С. Пушкина, 2019. С. 349–352. EDN: KVKSUJ.
5. Колисниченко Д. Н. GIMP 2 — бесплатный аналог Photoshop для Windows/Linux/Mac OS: 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2010. 368 с.
6. Колисниченко Д. Н. Ubuntu 10. Библия пользователя. М.: Вильямс, 2010. 592 с.
7. Краснов А. Н., Маркушевич М. В., Низаметдинова И. А. Обучение компьютерной графике младших школьников на основе применения свободного растрового графического редактора GIMP // Герценовские чтения. Начальное образование. 2018. Т. 9. № 1. С. 131–144. EDN: YQCWTY.
8. Маркушевич М. В., Федосов А. Ю. Основные методические подходы к использованию свободного программного обеспечения в курсе информатики на уровне основного общего образования // Информатика и образование. 2022. Т. 37. № 1. С. 37–48. DOI: 10.32517/0234-0453-2022-37-1-37-48. EDN: JVMRGW.
9. МГТУ Баумана остается без Windows: Microsoft отказалась поставлять ему свое ПО // CNews. https://www.cnews.ru/news/top/2020-12-09_mgtu_baumana_ostalsya_bez
10. Назарова В. П., Таричко С. А., Лучанинов Д. В. Анализ бесплатного программного обеспечения для создания растровых изображений // Постулат. 2016. № 12. С. 20. EDN: XSBKWP.
11. Нуромская Н. Н., Попова И. К., Шестакова О. А. Методика работы с графическим редактором GIMP // Психолого-педагогический журнал Гаудеамус. 2010. Т. 2. № 16. С. 185–188. EDN: MUEUJJ.
12. Панюкова Т. А. GIMP и Adobe Photoshop. Лекции по растровой графике. М.: ЛИБРОКОМ, 2018. 280 с.
13. Пожарина Г. Ю. Свободное программное обеспечение на уроке информатики. СПб.: БХВ-Петербург, 2010. 336 с.
14. Поляков К. Ю., Еремин Е. А. Информатика. 7 класс: в 2 ч. Ч. 2. 2-е изд., стереотип. М.: БИНОМ. Лаборатория знаний, 2018. 160 с.
15. Семанин И. Г., Залогова Л. А., Русаков С. В., Шестакова Л. В. Информатика: учебник для 7 класса. 7-е изд., стереотип. М.: БИНОМ. Лаборатория знаний, 2018. 168 с.

16. Системные требования GIMP // Системные требования. <https://systemrequirements.ru/gimp/>
17. Слои. Каналы. Маски. <https://www.sites.google.com/site/grafgimp/home/sloi-kanaly-maski>
18. *Соловьева Л. Ф.* Информатика и ИКТ. Учебник для 9 класса. СПб.: БХВ-Петербург, 2011. 384 с.
19. Сравнение растровых графических редакторов // Википедия. https://ru.wikipedia.org/wiki/Сравнение_растровых_графических_редакторов
20. Уроки GIMP для начинающих // Блог Антона Лапшина. <https://www.gimpart.org/video-uroki-gimp/kollekziya-video-uroki-po-gimp>
21. *Федосов А. Ю., Маркушевич М. В.* Актуальные вопросы построения методики преподавания темы «Векторная и растровая графика» в основной школе на основе свободных графических редакторов OpenOffice.org Draw и GIMP // Информатика в школе. 2019. № 8. С. 9–18. DOI: 10.32517/2221-1993-2019-18-8-9-18. EDN: НАЕВЕФ.
22. *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. DOI: 10.32517/2221-1993-2021-20-9-4-13. EDN: SCENFM.
23. *Хахаев И. А.* Свободный графический редактор GIMP. Первые шаги. М.: Alt Linux; ДМК Пресс, 2010. 223 с.
24. *Lecarme O., Delvare K.* The book of GIMP: A complete guide to nearly everything. No Starch, 2013. 676 p.
25. Simply Linux // ООО «Базальт СПО» — российский разработчик операционных систем «Альт». <https://www.basealt.ru/simplylinux/description>
26. *van Gumster J., Shimonski R.* GIMP Bible. Wiley, 2010. 770 p.
27. *Whitt P.* Beginning photo retouching and restoration using GIMP. CA: Apress Berkeley, 2014. 308 p. DOI: 10.1007/978-1-4842-0403-0

DOI: 10.32517/2221-1993-2022-21-6-30-36

С. С. Гербут

Лицей № 2, г. Тула, Россия

ПРОПЕДЕВТИКА ПРОГРАММИРОВАНИЯ: КАК ПОДГОТОВИТЬ УЧАЩИХСЯ К ВОСПРИЯТИЮ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ*

Аннотация

В статье описаны методические основы подготовки к программированию в основной школе и, в частности, к восприятию объектно-ориентированного программирования в будущем. Данную методику можно реализовать в курсе внеурочной деятельности или воспользоваться ее фрагментами для реализации на уроке. Программирование — один из важнейших разделов в базовом школьном курсе информатики. Это интересный и одновременно довольно сложный для начального обучения раздел. С учетом того, что базовый курс информатики начинается с седьмого класса и составляет всего один час в неделю (углубленный — два часа в неделю), на программирование выделяется время только в восьмом классе и не так уж много. А познакомить детей нужно с большим числом понятий, научить писать программы — от линейных до обработки массивов. В связи с этим предлагается использовать пропедевтический курс внеурочной деятельности для учащихся шестых-седьмых классов с целью их знакомства с основами структурного программирования на примере работы с исполнителями, позволяющими реализовать знакомство со структурой программы, синтаксическими требованиями, основными алгоритмическими конструкциями через рисование и движение графических объектов (на выбор учителя — на алгоритмическом языке, PascalABC, Python, C++). Методические разработки, описанные в статье, можно использовать для тех исполнителей, которых не будет на уроках, или, наоборот, познакомить учеников с написанием программ на языке, который будет изучаться дальше на уроках, что позволит им легче воспринимать материал по программированию в основной школе и развивать свои способности в этом направлении по индивидуальной образовательной траектории — как на уроках, так и во внеурочной деятельности.

Ключевые слова: информатика, начала программирования, пропедевтика программирования, программирование графических объектов, структурное программирование, язык PascalABC, язык Python, язык C++ для начинающих, КуМир, Рисователь.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_06-2022/

Контактная информация

Гербут Светлана Сергеевна, учитель информатики, лицей № 2, г. Тула, Россия; адрес: 300002, Россия, г. Тула, ул. Галкина, д. 14; e-mail: gerbut-ss@yandex.ru

S. S. Gerbut

Lyceum 2, Tula, Russia

PROPAEDEUTICS OF PROGRAMMING: HOW TO PREPARE STUDENTS FOR THE PERCEPTION OF OBJECT-ORIENTED PROGRAMMING

Abstract

The article describes the methodological foundations for preparing for programming in basic school and, in particular, for the perception of object-oriented programming in the future. This methodics can be implemented in the course of extracurricular activities or use its fragments for implementation in the lesson. Programming is one of the most important sections in the basic school informatics course. This is an interesting and at the same time quite difficult section for initial learning. Given that the basic informatics course starts in the seventh grade and is only one hour per week (advanced — two hours per week), programming is allocated only in the eighth grade and not so much. And you need to introduce pupils to a large number of concepts, teach them how to write programs, from linear programs to array processing. In this regard, it is proposed to use a propaedeutic course of extracurricular activity for students of the sixth-seventh grades in order to familiarize them with the basics of structured programming using the example of working with performers, which allow them to realize familiarity with the structure of the program, syntactic requirements, basic algorithmic constructions through drawing and movement of graphic objects (according the teacher's choice — in algorithmic language, PascalABC, Python, C++). The methodological developments described in the article can be used for those performers who will not be in the lessons, or, conversely, to introduce students to writing programs in a language that will be studied further in the lessons, which will make it easier for them to perceive the material on programming in basic school and develop their abilities in this direction along an individual educational trajectory — both in the classroom and in extracurricular activities.

Keywords: informatics, basics of programming, propaedeutics of programming, programming of graphic objects, structural programming, PascalABC, Python, C++ for beginners, KuMir, Drawman.

1. Введение

В настоящее время все большей популярностью пользуется такое направление в информатике, и в частности в программировании, как раннее знакомство учащихся с объектно-ориентированным программированием (ООП). ООП можно рассматривать как модульное программирование нового уровня, когда вместо во многом случайного, механического объединения процедур и данных главным становится их смысловая связь [3].

Объектно-ориентированное программирование предполагает такой взгляд на моделируемые объекты, при котором они рассматриваются как единое целое со всем многообразием свойств и особенностей поведения [5].

Многие учителя являются сторонниками перехода к понятию объектов, методов и других понятий ООП только после изучения структурного программирования через консольные приложения. Чтобы все успеть рассказать и показать ученикам, конечно, надо ответить на острый вопрос: когда начать? Ведь обучение программированию в основной школе начинается, как правило, во второй половине восьмого класса, что довольно поздно в том смысле, что алгоритмическое мышление у них уже в основной массе сформировано. Понятие алгоритмического мышления тесно связано с понятием логического мышления. Психологи утверждают, что для развития логического мышления природой отведены определенные возрастные рамки, примерно соответствующие по срокам обучению в начальной школе.

Описанные ниже методические аспекты и структура пропедевтического курса нацелены на раннее знакомство с «настоящим» программированием в среде, в которой можно программировать серьезные проекты. Знакомство с программированием графических элементов, анимации в шестом-седьмом классах и с началами программирования в восьмом классе позволит ученикам восьмых-девярых классов уже выбрать направление программирования, которое им будет интересно для дальнейшего саморазвития (олимпиадное программирование, проектное программирование (ООП), работа с базами данных, веб-программирование и др.) и, возможно, для реализации индивидуального учебного проекта в старшей школе.

В данной статье подробно описано итоговое занятие по программированию, на котором учащиеся должны продемонстрировать полученные знания, проявить свои творческие и лидерские качества, потренироваться работать в команде. Такое занятие необходимо учителю для оценивания результатов своей работы по подготовке детей к восприятию ООП. Оценка — это всегда измерение определенных характеристик достигнутого образовательного результата. Объектом педагогической оценки в учебном процессе является деятельность обучающихся, а предметом — оценивание результатов этой деятельности, выраженное в различных качественных характеристиках и количественных показателях [7].

Для понимания значения этого занятия опишем сначала общую структуру пропедевтического курса.

2. Выбор среды программирования

Программирование графики (рисование объектов с помощью графических примитивов, анимация) —

это раздел программирования, с которого легче начать знакомство учеников с программированием и который наиболее интересен для учащихся.

Выбор конкретного языка программирования и среды программирования во многом зависит от используемой в конкретной школе программы обучения и от навыков учителя:

- для знакомства с программированием в шестом классе можно начать со среды «КуМир», выбрать исполнителя Рисователь;
- если в основной школе изучается язык программирования PascalABC, то можно и в пропедевтическом курсе сразу начать с него (с использованием модуля GraphABC);
- для знакомства с современным языком программирования Python можно воспользоваться авторской библиотекой К. Ю. Полякова graph.py;
- если цель учителя — в дальнейшем перейти к олимпиадному программированию, то можно выбрать язык C++ (в частности, начать работу с библиотеки TXLib).

Если требуется написать программу с графическими возможностями на языке Python, обычно используют библиотеки, которые скрывают от программиста сложности взаимодействия с операционной системой. Одна из библиотек, позволяющая работать с графикой на языке Python, — это tkinker (устанавливается вместе с интерпретатором Python). Библиотека graph, с которой намного проще начать знакомство с программированием графики, является «оберткой» библиотеки tkinker [8].

Писать простые графические программы на C++ не всегда просто. Для вывода графики требуется с помощью операционной системы создать специальное окно, обеспечить его взаимодействие с пользователем и т. п. Для того чтобы упростить вход в мир графического программирования, были разработаны специальные библиотеки. Наиболее удобна библиотека TX Library, разработанная И. Р. Дединским (см. [8]).

Модуль GraphABC в PascalABC представляет собой простую графическую библиотеку и предназначен для создания несобытийных графических и анимационных программ в процедурном и, частично, в объектном стиле. Рисование осуществляется в специальном графическом окне, возможность рисования в нескольких окнах отсутствует. Кроме того, в модуле GraphABC определены простейшие события мыши и клавиатуры, позволяющие создавать элементарные событийные приложения [9].

PascalABC.NET — мультипарадигменный язык. Он позволяет программировать в структурном, объектно-программированном и функциональном стилях, а также смешивать эти стили в одной программе. Большое количество «синтаксического сахара» делает программирование на PascalABC.NET простым и комфортным как для начинающих, так и для имеющих опыт в создании программ [6].

Что общего у перечисленных выше сред программирования? В них используется компьютерная система координат, имеется набор похожих графических примитивов, возможность реализации процедурного программирования, в PascalABC и C++ можно продолжить изучение в направлении разработки собственных



Рис. 1. Компьютерная система координат

классов с объектами и методами (осуществить плавный переход к понятиям ООП).

Компьютерная система координат (рис. 1) характеризуется следующими особенностями: ось x — горизонтальная, направлена вправо, ось y — вертикальная, направлена вниз, точка $(0, 0)$ — начало координат находится в верхнем левом углу, все координаты — неотрицательные целые числа.

3. Темы курса внеурочной деятельности по программированию графики

Предполагается, что до итогового занятия учащиеся уже прошли вводное обучение программированию графики в выбранной учителем среде по следующим темам:

- Знакомство с компьютерной системой координат.
- Рабочая область исполнителя, программирование рисования графических примитивов.
- Управление цветом фона, заливки, линий фигур (встроенные цвета).
- Добавление в программу цветовых констант (в системе RGB).
- Знакомство с переменными. Целый и вещественный тип. Тип для цвета.
- Пересчет координат рисования объектов через переменные.
- Рисование одинаковых фрагментов из разных точек и разными цветами.
- Знакомство с процедурами и функциями.
- Функции с параметрами (координаты, цвет).
- Вывод текста на область рисования. Анимация через циклы.
- Программирование фрагментов с повторениями.
- Проект «Многоэтажный дом» с разными по цвету окнами (выключен и включен свет, цвет занавесок).

4. Сценарий итогового занятия (командная работа)

Тема занятия: Разработка проекта «Программируем созвездия»

Цель занятия: создание условий для развития алгоритмического мышления и творческого самовыражения школьников через программирование проектов, способствующих их будущим развитию и самосовершенствованию в области объектно-ориентированного программирования.

Задачи занятия:

- закрепить понятие алгоритма, структурного программирования путем деления алгоритма на под-

задачи и реализации в виде функций; подвести обучающихся к понятиям объектно-ориентированного программирования: объект, свойство, метод;

- сформировать у учащихся навыки работы в команде: распределить обязанности в команде по выполнению фрагментов задания, выбрать ответственного за объединение работ в единый проект; развивать творческие способности каждого школьника при реализации им выделенной части проекта;
- воспитывать у учащихся умения: работать в команде, высказывать свою точку зрения, прислушиваться к чужому мнению, находить оптимальное решение, принимать мнение команды, оказывать помощь напарнику и, наоборот, просить о помощи; способствовать передаче опыта от старшеклассников к более младшим ученикам.

Целевая аудитория: ученики шестого-седьмого классов.

Характеристика класса (группы).

Учащиеся:

- хорошо владеют навыками работы в выбранной среде программирования;
- понимают основы работы с компьютерной системой координат;
- знают структуру программы;
- владеют навыками создания линейных программ;
- умеют добавлять собственные цвета в программу;
- умеют писать программы с использованием переменных;
- понимают назначение подпрограмм;
- могут реализовать подпрограмму с параметрами (как минимум с тремя: координаты, цвет).

На некоторых занятиях уже использовалась работа в парах, но опыта работы в команде более двух человек у учеников не было.

Подготовительный этап.

В зависимости от уровня освоения материала учащиеся к текущему занятию учителю необходимо спланировать задание (внести какие-то изменения — упростить или усложнить отдельные фрагменты).

Вопросы и ответы на них для этапа «Актуализация опорных знаний» следует оформить в слайдах презентации, чтобы можно было подкрепить ответы учеников фактическими командами, напомнить изученный материал. Наличие презентации на занятии обязательно, так как в активном восприятии и осмыслении изучаемого материала большое значение имеет умение учителя придавать изложению увлекательный характер, делать его живым и интересным [1].

Учитель должен продумать, как (в команды по сколько человек) можно разделить группу, посещающую внеурочные занятия, чтобы получилось несколько команд-участников. При небольшой посещаемости это может быть и одна команда. Если команд несколько, стоит пригласить старшеклассников для курирования работы каждой команды, обсудить с этими ребятами их роль в проведении занятия.

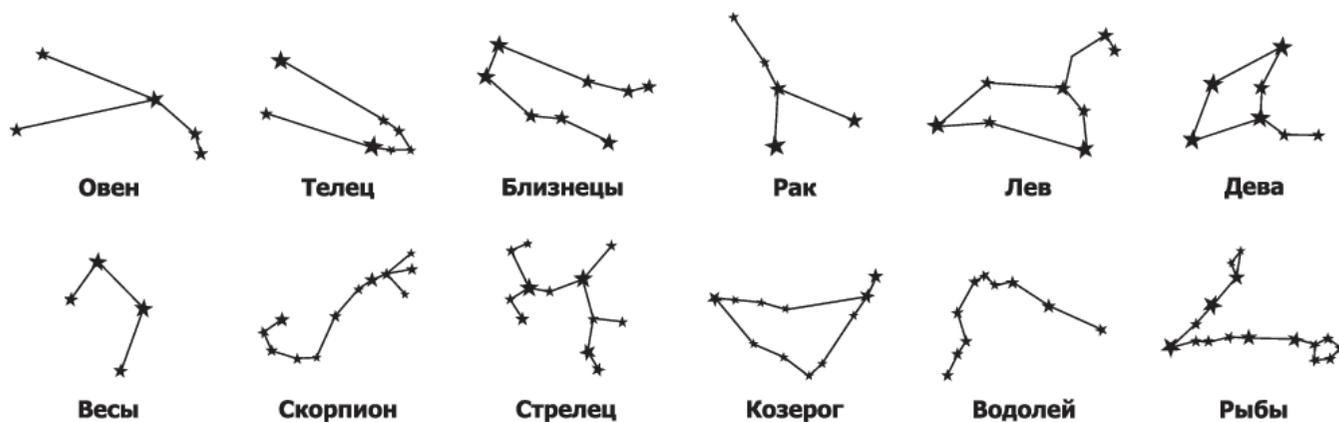


Рис. 2. Изображения созвездий (<https://mychatte.ru/catalog/raznoe/nabor-zodiakalnykh-sozvezdiy-/>)

Для выполнения задания надо подготовить изображения созвездий (один из примеров приведен на рисунке 2). Можно подготовить и любые другие изображения, главное — чтобы их можно было распечатать на белом фоне. Ученики с помощью линейки и карандаша должны будут разлиновать рисунок, чтобы просчитать координаты для рисования звезд выбранного созвездия в компьютерной системе координат.

Надо заранее распечатать карточки, каждая из которых содержит:

- формулировку задания;
- рисунки созвездий;
- таблицу «Командная работа» (табл. 1) в помощь ученикам для организации командной работы.

Также следует подготовить распечатки-подсказки по пройденным темам, которые могут понадобиться слабым ученикам во время выполнения задания.

Необходимо обратить особое внимание на подготовку техники: у каждого ученика в команде должно быть рабочее место (компьютер или ноутбук с установленным ПО); места должны быть распределены так, чтобы участники одной команды могли общаться друг с другом и видеть работу друг друга.

Учитель должен заранее подготовить поощрения за участие, это могут быть грамоты победителям, сертификаты участникам, памятные призы и т. п. — на усмотрение учителя.

Занятие рассчитано на 2,5 часа (три академических урока).

Таблица 1

Командная работа

№ п/п	Работа в команде	Отметка о выполнении
1	Выбрать ученика, который будет готовить диалог программы и собирать проект в единое целое	
2	Обсудить, на какие блоки разбивается алгоритм	
3	Обсудить реализацию диалога с пользователем	
4	Обсудить шаблон подпрограмм, названия, параметры	
5	Распределить созвездия между участниками	
6	Договориться о размере окна рисования	
7	Договориться о размерах и формах звезд	
8	Обсудить цветовую гамму проекта	
9	Обсудить наличие или отсутствие анимации	
10	На распечатках созвездий просчитать координаты каждой звезды	

Таблица 2

Ход занятия

№ этапа	Этап занятия	Время, мин	Деятельность учителя	Деятельность учащихся
1	Организационный момент	3	Приветственное слово. Озвучивает цель занятия: реализовать командный проект «Программируем созвездия» с применением полученных знаний в программировании. Поясняет, что основной формой деятельности на занятии будет командная работа. Представляет старшеклассников, поясняет, что их роль не реализовать проект, а помочь, подсказать в организационных вопросах, ответить на вопросы по программе.	Приветствуют учителя, старшеклассников. Задают организационные вопросы.

Продолжение табл. 2

№ этапа	Этап занятия	Время, мин	Деятельность учителя	Деятельность учащихся
2	Актуализация опорных знаний	15	Предлагает ученикам вспомнить основные особенности программирования в изучаемой среде: 1) Что такое «компьютерная система координат»? 2) Какова структура программы на выбранном языке? 3) Какие вам известны команды исполнителя для рисования графических примитивов? 4) Какие вам известны команды для работы с цветом? 5) Что такое переменные? Зачем они нужны? К какому типу данных они относятся? 6) Каким образом можно организовать рисование одинаковых фрагментов из разных точек и разными цветами? 7) Что такое анимация и какими средствами программирования ее можно реализовать?	Ответы учеников будут зависеть от используемой среды программирования и исполнителя. Но общая идея одинакова: 1) Учащиеся должны вспомнить, что начало координат находится в верхнем левом углу рабочей области. 2) Структура программы зависит от конкретного языка программирования. 3) Линия, прямоугольник, окружность, эллипс и др. (конкретные названия команд зависят от используемого языка программирования). 4) Цвет линии, цвет заливки, задание цветов в модели RGB. 5) Нужны для упрощения рисования похожих объектов из разных точек. Есть целые и вещественные переменные. 6) С помощью подпрограмм, подпрограмм с параметрами. 7) Учащиеся должны вспомнить известные циклы (количество итераций, с параметром, с предусловием — ответы зависят от того, что учитель рассказывал ранее).
3	Деление на команды	3	Руководит распределением учащихся по командам.	Учащиеся делятся на команды — команду набирает капитан или распределение по командам определяется жеребьевкой.
4	Формулировка задания	5	Объясняет задание : Разработать проект, в котором будет реализована демонстрация некоторых созвездий (из распечаток). В программе должны быть реализованы диалог между пользователем и программой, позволяющий выбрать созвездие для демонстрации, а также возможность очистить область рисования. Поскольку это командная работа, следует проговорить необходимость распределения обязанностей между участниками в команде.	Слушают задание, задают уточняющие вопросы.
5	Распределение учащихся по рабочим местам	2	Размещает учеников так, чтобы они могли общаться с остальными членами команды.	Сдвигают парты, чтобы каждая команда работала за одним большим столом, или просто занимают заранее подготовленные учителем места.
6	Работа в команде	17	Координирует командную работу, при этом старшеклассники помогают ему следить за распределением членов каждой команды по пунктам, указанным в карточке с заданием.	Обсуждают распределение обязанностей по программированию отдельных блоков программы. Выбирают члена команды, который будет собирать проект в единое целое. Обсуждают, на какие подпрограммы будет разбиваться алгоритм, наличие параметров в функциях. Распределяют между участниками фрагменты задания. Договариваются о размерах окна рисования, о цветовой гамме визуализации проекта, о размерах звезд, наличии или отсутствии анимации. На распечатках созвездий просчитывают координаты каждой звезды.

Окончание табл. 2

№ этапа	Этап занятия	Время, мин	Деятельность учителя	Деятельность учащихся
	Перерыв	10		
7	Индивидуальная работа каждого участника	45	<p>Вместе со старшеклассниками следит за работой команд, отвечает на вопросы. Каждые 15 минут напоминает учащимся о необходимости проведения гимнастики для глаз.</p> <p>У учителя заранее подготовлены фрагменты программ с предыдущих уроков на случай, если кто-то из учеников ошибается в синтаксисе или структуре программы. Но использование таких подсказок учитывается при подведении итогов. У учителя должны быть подготовлены распечатки с созвездиями для раздачи учащимся в случае дополнительного запроса.</p>	<p>Ученики за компьютерами выполняют свою часть проекта, они могут обращаться за помощью или советом к участникам своей команды, к старшекласснику.</p> <p>Капитан каждой команды создает подготовку проекта, в которую будут добавлены все остальные части. В случае быстрого выполнения какими-то участниками своей части работы команда может принять решение добавить в программу анимацию или дополнительные созвездия.</p>
	Перерыв	10		
8	Работа в команде	20	Вместе со старшеклассниками следит за правильностью сохранения программы на флеш-накопитель или в сетевую папку, передачи кода капитану команды, помогает при возникновении трудностей.	<p>Ответственный член команды (обычно капитан) объединяет полученные от участников команды решения в общий проект. Команда обсуждает выступление на защите.</p>
9	Демонстрация проектов командами, защита	20	<p>Учитель и старшеклассники:</p> <ul style="list-style-type: none"> • слушают выступления команд; • задают вопросы; • оценивают проекты по критериям (табл. 3); • награждают участников грамотами и памятными призами (если это предусмотрено). 	<p>Демонстрируют проекты. Выступают перед другими командами: рассказывают о том, что сделано в работе, что не получилось реализовать, какие варианты развития проекта они видят, отвечают на вопросы.</p>
10	Подведение итогов, рефлексия	5	<p>Благодарит всех участников за их работу. Отмечает, что сегодня они программировали собственные объекты, у которых есть свойства: цвет, местоположение и др., в каких-то проектах появились методы: находиться на месте, двигаться. Это задел на будущее объектно-ориентированное программирование.</p> <p>Подводит всех к мысли, что программирование имеет огромные масштабы для реализации творческих замыслов, что нужно его учить более углубленно и создавать свои собственные разработки. Отмечает помощь старшеклассников.</p> <p>Подчеркивает, что работа в команде — это непростое, но продуктивное дело. Все молодцы, что смогли распределить задания, помочь друг другу справиться с трудностями и в итоге представить отличный совместный проект.</p>	<p>Высказываются о проделанной работе, обсуждают, в чем разница между индивидуальной и командной работой, с какими трудностями они столкнулись, какие проекты им бы хотелось еще реализовать.</p>

Примеры проектов, выполненных учащимися, а также задание и презентацию к занятию можно скачать на сайте ИНФО в дополнительных материалах к статье:

http://infojournal.ru/journals/school/school_06-2022/

Оценивание.

Каждая работа оценивается по определенным критериям (табл. 3), оценка по каждому критерию — до 3 баллов, при этом:

- 0 — полное отсутствие (критерий не выполняется);
- 1 — частичная реализация;
- 2 — реализация хорошая, но есть неточности;
- 3 — отличная реализация.

5. Заключение

В рамках описанного выше занятия у учащихся в разной степени формируются все четыре вида универсальных учебных действий: личностные, регулятивные, познавательные, коммуникативные. Они обеспечивают возможность учащегося самостоятельно осуществлять деятельность учения, ставить учебные цели, искать и использовать необходимые средства и способы достижения этих целей, контролировать и оценивать процесс и результаты своей и коллективной деятельности [2].

Подготовленный сценарий занятия можно использовать как:

- конкурсное (состязательное) мероприятие;

Критерии оценки работ

№ п/п	Критерии	Баллы
1	Объединение программ в единый проект	
2	Реализация диалога программы с пользователем	
3	Структура программы в одном шаблоне (название подпрограмм и порядок параметров при наличии)	
4	Написание подпрограмм с параметрами	
5	Единый стиль оформления результата (цвет, форма и размер звезд)	
6	Наличие анимации	
7	Просчет координат на распечатках	
	Штрафной балл за использование подсказок учителя (вычитается из общей суммы)	
	ИТОГО:	

- зачетную работу;
- занятие для подготовки команды к участию в сто-роннем конкурсе.

Описанный выше сценарий можно доработать до самостоятельного мероприятия, целью которого будет знакомство со структурным программированием (для новичков) для привлечения внимания школьников к программированию и, возможно, с целью дальнейшего посещения курса внеурочной деятельности. Данное мероприятие будет рассчитано уже на целый день (т. е. полностью меняется формат его проведения).

Подготовка к объектно-ориентированному программированию начинается со структурного программирования, а именно с умения написать процедуры и функции — разного назначения, но имеющие что-то общее. Тогда ученики смогут разобраться с понятиями «объект», «свойство объекта», «метод». На примере вышеописанного занятия объектами являются звезды, у которых есть свойства: цвет, местоположение (координаты), размер и др., а также методы: рисование звезды, движение звезды. Это самое наглядное, но практика показывает, что кто-то из учеников обязательно заметит, что также объектом являются область рисования, диалог с программой и т. д. Это большой задел на будущее изучение объектно-ориентированного программирования.

Список источников

1. Босова Л. Л., Босова А. Ю. Информатика: методическое пособие для 5–6 классов. М.: БИНОМ. Лаборатория знаний, 2014. 384 с.
2. Босова Л. Л., Босова А. Ю. Информатика: методическое пособие для 7–9 классов. М.: БИНОМ. Лаборатория знаний, 2015. 472 с.
3. Гвоздева В. А. Введение в специальность программиста: учебник. 2-е изд., испр. и доп. М.: ИД «ФОРУМ»; ИНФРА-М, 2007. 208 с.
4. Голицына О. Л., Попов И. И. Программирование на языках высокого уровня: учебное пособие. М.: ФОРУМ, 2008. 496 с.
5. Куперман В. Г., Торина Е. Г. Основы информатики и начала программирования: учебное пособие. Тула: Изд-во Тульского государственного педагогического университета им. Л. Н. Толстого, 1997. 264 с.
6. Осипов А. В. PascalABC.NET: Введение в современное программирование. Ростов-на-Дону, 2019. 572 с.
7. Основы общей теории и методики обучения информатике: учебное пособие / под ред. А. А. Кузнецова. М.: БИНОМ. Лаборатория знаний, 2010. 207 с.
8. Поляков К. Ю. Программирование. Python. C++. Ч. 1: учебное пособие. 2-е изд., испр. М.: БИНОМ. Лаборатория знаний, 2020. 144 с.
9. Система программирования PascalABC.NET. <http://pascalabc.net/>

DOI: 10.32517/2221-1993-2022-21-6-37-43

К. В. Розов*Новосибирский государственный педагогический университет;**Гимназия № 4, г. Новосибирск, Россия***Б. А. Шрайнер***Новосибирский государственный педагогический университет, г. Новосибирск, Россия*

ДИСТАНЦИОННОЕ ОБУЧЕНИЕ ШКОЛЬНИКОВ ТЕХНОЛОГИЯМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА*

Аннотация

В настоящее время технологии искусственного интеллекта являются одними из перспективных и стремительно развивающихся цифровых технологий, оказывающих существенное влияние на экономику страны и многие сферы деятельности человека. Это обуславливает необходимость в информировании населения еще со школы о возможностях технологий искусственного интеллекта и в осуществлении соответствующей профессиональной ориентации школьников для подготовки востребованных специалистов. На уровне общего образования тема искусственного интеллекта пока представлена слабо, но исследования и разработки учебных материалов в этом направлении активно ведутся. В статье представлены опыт и результаты дистанционного обучения школьников технологиям искусственного интеллекта в асинхронном формате. Приведены описание и элементы содержания авторского учебного курса по искусственному интеллекту, являющегося одним из разделов большого курса «Сквозные цифровые технологии для школьников», размещенного в системе управления обучением Moodle Новосибирского государственного педагогического университета. Выявлены элементы содержания, к которым обучающиеся проявляли наибольший интерес.

* Исследование выполнено при финансовой поддержке Министерства просвещения РФ в рамках исполнения государственного задания № 073-03-2022-037 от 13.01.2022 г. по проекту «Обучение сквозным технологиям в условиях персонализации образовательных траекторий школьников».

Контактная информация

Розов Константин Владимирович,

ст. преподаватель кафедры информационных систем и цифрового образования, Институт физико-математического, информационного и технологического образования, Новосибирский государственный педагогический университет, г. Новосибирск, Россия; *адрес:* 630126, Россия, г. Новосибирск, ул. Вилюйская, д. 28;

учитель информатики, гимназия № 4, г. Новосибирск, Россия; *адрес:* 630004, г. Новосибирск, ул. Сибирская, д. 35;

e-mail: konstantin_dubrava@mail.ru

Шрайнер Борис Александрович, канд. психол. наук, зав. лабораторией «Технологии искусственного интеллекта», доцент

кафедры информационных систем и цифрового образования, Институт физико-математического, информационного и технологического образования, Новосибирский государственный педагогический университет, г. Новосибирск, Россия; *адрес:* 630126, Россия, г. Новосибирск, ул. Вилюйская, д. 28; *e-mail:* boris.shrayner@gmail.com

K. V. Rozov

Novosibirsk State Pedagogical University;

Gymnasium 4, Novosibirsk, Russia

B. A. Shrayner

Novosibirsk State Pedagogical University, Novosibirsk, Russia

DISTANCE LEARNING OF SCHOOLCHILDREN IN ARTIFICIAL INTELLIGENCE TECHNOLOGIES

Abstract

Currently, artificial intelligence technologies are one of the promising and rapidly developing digital technologies that have a significant impact on the country's economy and many areas of human activity. This necessitates informing the population from school about the possibilities of artificial intelligence technologies and the implementation of appropriate vocational guidance for schoolchildren to train in-demand specialists. At the level of general education, the theme of artificial intelligence is still poorly represented, but research and development of educational materials in this direction are actively underway. The article presents the experience and results of distance learning of schoolchildren in artificial intelligence technologies in an asynchronous format. The description and elements of the content of the author's training course on artificial intelligence, which is one of the sections of the large course "End-to-end digital technologies for schoolchildren", hosted in the Moodle learning management system of Novosibirsk State Pedagogical University, are given. The elements of the content to which the students showed the greatest interest were identified. The main problems that students encountered when performing practical work are listed. The problems of distance learning of schoolchildren in artificial intelligence technologies, in particular in an asynchronous format, are outlined, and ways to solve them are proposed.

Keywords: artificial intelligence, artificial intelligence technologies, distance learning, e-learning, data analysis, machine learning, computer vision, natural language processing.

Перечислены основные проблемы, с которыми обучающиеся столкнулись при выполнении практических работ. Обозначены проблемы дистанционного обучения школьников технологиям искусственного интеллекта, в частности в асинхронном формате, и предложены пути их решения.

Ключевые слова: искусственный интеллект, технологии искусственного интеллекта, дистанционное обучение, электронное обучение, анализ данных, машинное обучение, компьютерное зрение, обработка естественного языка.

1. Введение

В последнее время технологии искусственного интеллекта стали все чаще применяться в самых разнообразных сферах жизни человека и общества. На данный момент можно констатировать, что имеется высокая потребность государства и бизнеса в специалистах профессий, связанных с искусственным интеллектом. Так как системы на основе искусственного интеллекта могут использоваться практически в любой сфере жизни человека, то требуется массово информировать людей о принципах, возможностях и ограничениях таких технологий. Соответственно, имеет смысл затрагивать тему искусственного интеллекта при работе со школьниками.

Проблемам обучения школьников элементам искусственного интеллекта посвящены работы О. М. Корчажиной [1], И. В. Левченко, А. Р. Садыковой [4], Н. Н. Самылкиной, А. А. Салаховой [8], Л. Н. Ясницкого, Ф. М. Черепанова [12] и др. Исследователями пермской школы искусственного интеллекта Ф. М. Черепановым и Л. Н. Ясницким в 2011 году был представлен лабораторный практикум элективного курса для учащихся старших классов с применением авторского тренажера для конструирования и обучения искусственных нейронных сетей [2]. Содержащиеся в нем сведения и лабораторно-практические работы по-прежнему актуальны и позволяют закрепить теоретические представления об основах функционирования искусственных нейронных сетей, продемонстрировать задачи, решаемые с помощью них. Н. Н. Самылкина, А. А. Салахова в своих работах представляют учебные материалы для школьников, ориентированные преимущественно на изучение анализа данных с помощью машинного обучения, реализованного с использованием специализированных Python-библиотек и сред разработки, таких как Jupyter Notebook, входящий в состав Anaconda [8, 9]. Коллектив авторов под руководством И. В. Левченко разработал и представил в 2021 году ряд образовательных модулей для общеобразовательного курса информатики, посвященных основам искусственных нейронных сетей, компьютерному зрению, обработке естественного языка, разработке интеллектуальных компьютерных игр [3], учебные пособия и методические рекомендации по обучению искусственному интеллекту в основной школе, в частности в V—VI классах [5]. Однако данные учебные материалы не находятся в публичном доступе.

Знакомство школьников с технологиями искусственного интеллекта также осуществляют ведущие отечественные ИТ-компании совместно с вузами. Примерами крупных проектов являются «AI-ACADEMY Академия искусственного интеллекта для школьников» [13] и «Урок Цифры», предлагающий уроки по темам «Искусственный интеллект в образовании», «Нейросети и коммуникации», «Искусственный интеллект и машинное обучение» [10]. Учебные материалы, представленные на этих ресурсах, находятся в открытом доступе.

Нами были разработаны учебное пособие и практикум, направленные на знакомство школьников и студентов педагогического вуза физико-математических и информационно-технологических профилей подготовки с технологиями искусственного интеллекта, реализующими их современными Python-библиотеками и популярными в этой области инструментами и сервисами [6, 11].

На основе разработанных нами материалов был создан бесплатный дистанционный курс для школьников. Основной целью курса является популяризация тем искусственного интеллекта среди учащихся VII—X классов. Пройдя этот курс, обучающиеся смогут составить представление об основных принципах работы систем искусственного интеллекта, изучив теоретические разделы, а также попробовать свои силы в практических работах, позволяющих самостоятельно разработать свой мини-проект на основе искусственного интеллекта.

2. Содержание обучения школьников технологиям искусственного интеллекта

Наш практический опыт позволил выработать некоторую концепцию обучения школьников технологиям искусственного интеллекта, дающую им, с одной стороны, достаточные теоретические представления о сути современных технологий искусственного интеллекта, а с другой стороны, возможность получить реальный практический опыт, позволяющий в итоге разработать свой проект, основанный на применении искусственного интеллекта для решения конкретной практической задачи.

Ставилась цель познакомить учащихся средней школы с технологиями искусственного интеллекта, причем так, чтобы вне зависимости от их опыта в этом направлении они могли почерпнуть для себя что-то новое и сделать некий шаг вперед. Для новичков это возможность изучить язык программирования Python и среду разработки Jupyter Notebook, для продолжающих — опробовать функционал библиотек анализа и визуализации данных, а для более опытных — попробовать свои силы в машинном обучении, компьютерном зрении и обработке естественного языка, выполняя практические работы. В рамках этих практических работ обучающимся предлагается:

- разработать программы, решающие конкретные прикладные задачи из области применения искусственного интеллекта с использованием пошаговых инструкций;
- выполнить задания для самостоятельной работы по доработке программы;
- ответить на вопросы, требующие проведения экспериментов.

Авторский курс по искусственному интеллекту базируется на системе управления обучением Moodle Новосибирского государственного педагогического университета и размещен на сайте: <https://cifra.nspu.ru/>. Он

является одним из разделов большого курса «Сквозные цифровые технологии для школьников», включающего в себя различные темы из области цифровых технологий, такие как: большие данные, робототехника, беспроводная связь, виртуальная и дополненная реальность, искусственный интеллект, блокчейн, информационная безопасность, цифровая грамотность. Школьники, изу-

чая курс, выбирают наиболее интересные для них темы, проходят квесты и решают задачи тестов.

Курс «Искусственный интеллект» состоит из шести разделов, элементы содержания которых представлены в таблице 1.

Далее подробно рассмотрим содержание всех разделов курса.

Таблица 1

Элементы содержания разделов дистанционного курса «Искусственный интеллект»

№ раздела	Раздел	Элементы содержания раздела
1	Введение	<ul style="list-style-type: none"> • Сущность понятия «искусственный интеллект». • История искусственного интеллекта. • Разделы и классификация искусственного интеллекта. • Интересные онлайн-ресурсы с искусственным интеллектом
2	Язык программирования Python	<ul style="list-style-type: none"> • Специфика Python для изучения тем искусственного интеллекта. • Синтаксис Python. • Первая программа на Python в Anaconda. • Задачник по программированию на Python. • Задания по Python
3	Анализ данных	<ul style="list-style-type: none"> • Введение в анализ данных. • Установка библиотек. • Список библиотек. • Введение в Pandas. • Получение статистических данных в Pandas. • Отбор по условию в Pandas. • Удаление строк и столбцов. • Сортировка в Pandas. • Документация по Pandas. • Библиотека NumPy. • Визуализация данных. • Задачник по анализу данных
4	Машинное обучение	<ul style="list-style-type: none"> • Введение в машинное обучение. • Обучение с учителем. • Обучение без учителя. • Обучение с подкреплением. • Этапы разработки модели. • Проблема недообучения и переобучения. • Искусственные нейронные сети. • <i>Практическая работа 1</i> «Искусственный нейрон. Классификация». • <i>Практическая работа 2</i> «Кластеризация данных методом k-средних». • <i>Практическая работа 3</i> «Решение задачи регрессии с помощью искусственной нейронной сети»
5	Компьютерное зрение	<ul style="list-style-type: none"> • Введение в компьютерное зрение. • История компьютерного зрения. • Генеративно-состязательная сеть. • Сверточная нейронная сеть. • Популярные датасеты компьютерного зрения. • Библиотека OpenCV. • <i>Практическая работа 4</i> «Поиск объектов по цвету и контуру». • <i>Практическая работа 5</i> «Распознавание образов на фотографиях и видео с помощью каскадов Хаара». • <i>Практическая работа 6</i> «Обучение искусственной нейронной сети с помощью Google Teachable Machine»
6	Обработка естественного языка	<ul style="list-style-type: none"> • Введение в обработку естественного языка. • Мешок слов. • TF-IDF метрики. • Косинусное сходство. • <i>Практическая работа 7</i> «Предобработка текста и простая поисковая система». • <i>Практическая работа 8</i> «Распознавание и синтез речи»

Раздел 1. Введение

В разделе рассматриваются базовые теоретические сведения об искусственном интеллекте как отрасли информационных технологий. В конце раздела представлена подборка ссылок на онлайн-сервисы, функционирующие на основе технологий искусственного интеллекта, чтобы обучающиеся могли на практике, на конкретных примерах познакомиться с областями применения искусственного интеллекта, не прибегая к программированию.

Раздел 2. Язык программирования Python

Раздел направлен на знакомство обучающихся с Python как с языком, популярным и эффективным для разработок в области искусственного интеллекта. Рассматриваются среда разработки Jupyter Notebook, входящая в пакет приложений и библиотек для анализа данных и машинного обучения Anaconda, и облачный сервис Google Colab; изучаются основы программирования на языке Python; представлены задачи для формирования базовых навыков программирования на этом языке. Обучающиеся решают задачи и прикрепляют скрипты на Python (файлы с расширением .py) либо ноутбуки (файлы с расширением .ipynb) к заданию для проверки преподавателем. Планируется подключение модуля редактирования, запуска, тестирования и автоматической проверки программного кода.

Раздел 3. Анализ данных

В данном разделе рассматриваются ключевые понятия науки о данных (Data Science) — одной из ключевых областей применения технологий искусственного интеллекта. Прежде чем приступить к выявлению закономерностей в данных средствами машинного обучения, данные необходимо предобработать (привести к единому формату, удалить некорректные или неважные для текущей задачи объекты или их признаки и т. д.). Для повышения эффективности анализа полученных результатов важно научиться визуализировать данные, т. е. представлять их в наглядном виде с помощью графиков и диаграмм. Предобработке и визуализации данных посвящена практическая часть раздела, знакомящая обучающихся с возможностями Python-библиотек Pandas, NumPy, Matplotlib.

Более подробно остановимся на **содержании подраздела «Визуализация данных»**. В нем описываются особенности визуализации данных на Python с помощью библиотеки Matplotlib. Отмечается важность представления данных в визуальном формате, приведены кейсы использования визуализаций для выявления корреляции между признаками, поиска выбросов и выявления на глаз особенностей распределения данных. Приведены примеры кода для визуализации результатов (построения диаграмм и графиков разных типов) с подробным описанием принципов его работы. Предлагается воспользоваться разделом примеров на сайте документации Matplotlib для выбора нужного типа визуализации и ознакомиться с примером кода для ее создания на Python.

Раздел 4. Машинное обучение

В этом разделе рассматриваются понятия «машинное обучение» (Machine Learning, ML) и «модель машинного

обучения»; кратко охарактеризованы виды машинного обучения, этапы разработки модели, а также проблемы, возникающие при обучении. Отдельным элементом содержания раздела являются искусственные нейронные сети, в рамках этого элемента раскрываются понятия «искусственный нейрон», «искусственная нейронная сеть» и «глубокое обучение» (Deep Learning, DL). Даны ссылки на онлайн-сервисы A Neural Network Playground (<https://playground.tensorflow.org/>) и Teachable Machine (<https://teachablemachine.withgoogle.com/>), в которых можно попробовать построить и обучить искусственные нейронные сети.

Практическая часть раздела включает в себя три практические работы, посвященные основным задачам, решаемым с помощью машинного обучения (с учителем/без учителя): классификации, кластеризации, регрессии. Обучающимся предлагается написать программы на языке Python с использованием библиотек NumPy, Scikit-learn, TensorFlow/Keras для решения поставленных задач, провести эксперименты и представить отчет в виде текстового файла, который должен включать в себя листинги программ, изображения, демонстрирующие результаты, и ответы на вопросы.

Более подробно рассмотрим **содержание подраздела «Проблема недообучения и переобучения»**. В нем обсуждаются проблемы недостаточно точной работы модели на тестовой выборке. Для ситуации переобучения, когда модель вместо «понимания» общих принципов решения задачи в данных их «выучила» и из-за этого дает плохой результат, предлагаются различные варианты решения. Рассматриваются варианты, связанные с увеличением количества тренировочных данных, уменьшением степеней свободы модели (например, ограничение глубины у дерева решений), а также в ситуации с обучением нейронной сети в процессе обучения предлагается отключать случайную часть нейронов на каждом шаге обучения — применять метод Dropout.

Раздел 5. Компьютерное зрение

Раздел направлен на знакомство обучающихся с основными понятиями компьютерного зрения, сферами его применения и историей.

Уделено внимание популярным датасетам изображений, в частности набору рукописных цифр MNIST, применение которого используется как материал для изучения работы сверточных нейронных сетей в углубленном курсе информатики в школе [7], и библиотеке OpenCV как одной из самых популярных, наиболее востребованных современных библиотек компьютерного зрения.

Практическая часть раздела состоит из трех практических работ, направленных на использование библиотеки OpenCV для решения задач поиска объектов на изображениях по цвету и контуру (количеству углов), обнаружения лиц, глаз, улыбок и автомобильных номеров на фотографиях и в видеороликах (или используя веб-камеру) с помощью методов «классического компьютерного зрения» (без использования искусственных нейронных сетей); создание мини-проекта «Умная корзина» с применением обученной сверточной нейронной сети (модели TensorFlow) с помощью онлайн-сервиса Teachable Machine.

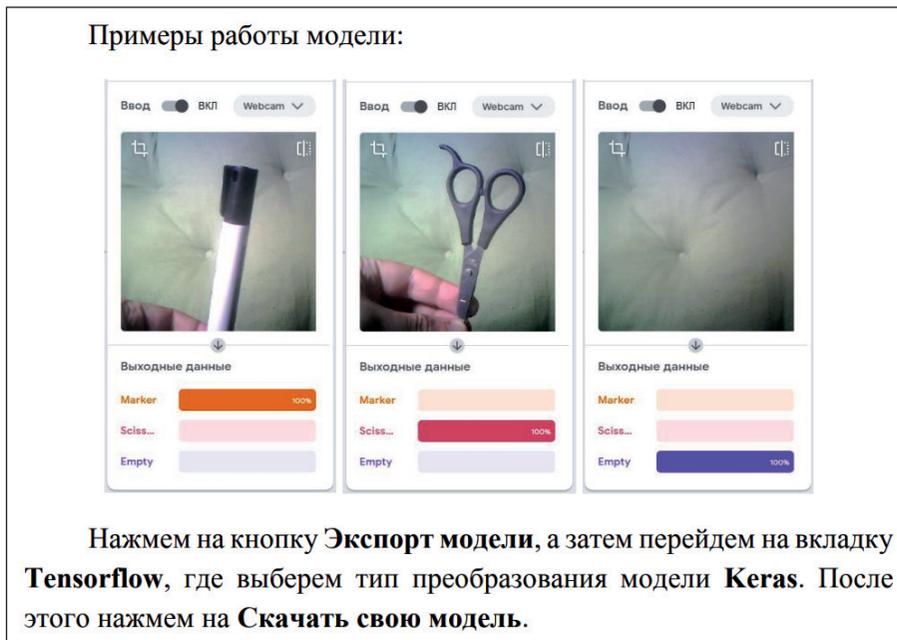


Рис. 1. Скриншот инструкции из практической работы с использованием Teachable Machine

Подробно рассмотрим **содержание практической работы 6 «Обучение искусственной нейронной сети с помощью Google Teachable Machine»**. В этой работе предлагается воспользоваться популярным онлайн-сервисом Teachable Machine, который позволяет достаточно просто создать классификатор компьютерного зрения. При этом образцы изображений для каждого класса можно как загрузить с компьютера, так и получить, сделав снимки объекта на веб-камеру в разных ракурсах. Дается подробная пошаговая инструкция по использованию данного сервиса (рис. 1). Результат обучения классификатора (модель TensorFlow) предлагается скачать и воспользоваться моделью в своем проекте на основе компьютерного зрения, применив код на Python

в среде разработки для целей распознавания объектов с видеопотока веб-камеры.

Раздел 6. Обработка естественного языка

В теоретической части раздела рассматривается понятие обработки естественного языка (Natural Language Processing, NLP) как направления искусственного интеллекта, приведены примеры применения систем, основанных на NLP. Особое внимание уделено этапам предобработки текстов: сегментации, токенизации, удалению стоп-слов, лемматизации/стеммингу и др. как универсальным действиям перед преобразованием текстов в наборы чисел (векторы), которые можно будет использовать для алгоритмов искусственного интеллекта.

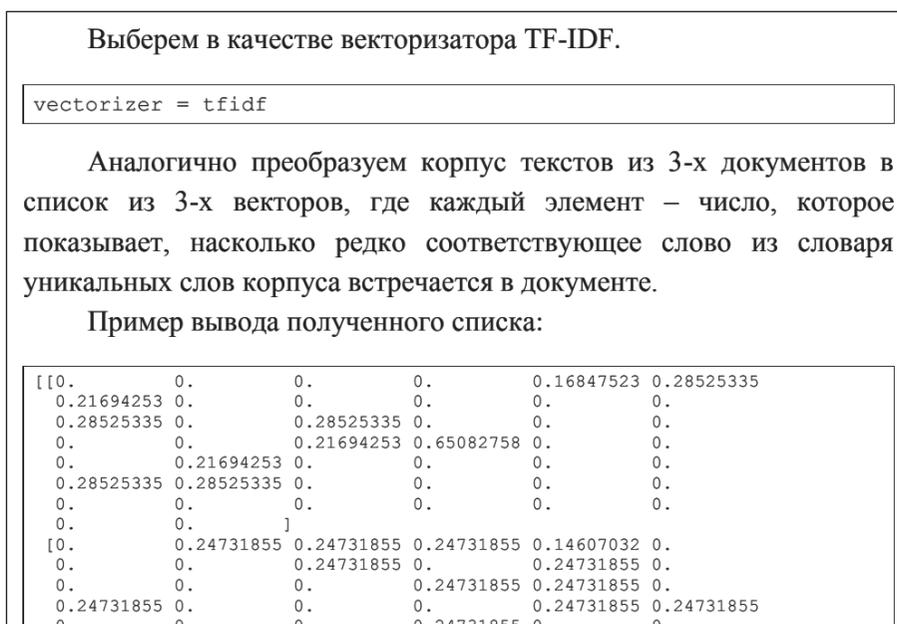


Рис. 2. Скриншот фрагмента практической работы по предобработке текста и созданию простой поисковой системы

Далее рассматриваются непосредственно методы таких преобразований, а также метод определения сходства двух векторов.

Первая практическая работа в данном разделе (практическая работа 7 «Предобработка текста и простая поисковая система») позволяет обучающимся выполнить все описанные этапы преобразования текста на практике с использованием библиотек NLTK, Scikit-learn и Rymorphy2 для создания простой поисковой системы (поиска предложения в заданном тексте, похожего на предложение, введенное с клавиатуры). Вторая практическая работа (практическая работа 8 «Распознавание и синтез речи») посвящена технологиям распознавания и синтеза речи. Обучающимся предлагается создать простого голосового помощника с помощью библиотек SpeechRecognition и Pyttsx3.

Более подробно рассмотрим **практическую работу 7 «Предобработка текста и простая поисковая система»**. В этой практической работе предлагается, применяя библиотеки NLTK, на основе подготовленного текста создать поисковую систему, которая будет отвечать на простые запросы. В процессе выполнения работы обучающиеся проводят все основные манипуляции по предобработке текста, включающие в себя токенизацию, удаление стоп-слов, лемматизацию, два варианта векторизации текста («Мешок слов» и «TF-IDF») и получение результата на основе косинусного сходства (рис. 2). Далее предлагается продолжить разработку подобной системы, добавив русскоязычный корпус текста и сделав циклический ввод — от получения ответа на запрос до ввода стоп-слова.

В конце курса школьники проходят **тестирование**, где проверяются их знания по искусственному интеллекту. В случае успешного прохождения курса обучающимся выдается сертификат.

3. Опыт и результаты обучения школьников

Всего на курс записались (создали учетные записи на образовательной платформе) 252 школьника (учащиеся VII—X классов), итоговый тест завершили 196 человек, на положительную оценку — 63 человека. Обучающиеся приглашались на курс путем рассылки приглашений в образовательные организации общего образования г. Новосибирска и Новосибирской области.

В процессе обучения, которое длилось с 1 октября по 15 декабря 2021 года, школьники самостоятельно обучались на платформе курса, проходя через все его разделы, изучая теоретический материал и выполняя практические работы. В конце обучения по курсу предлагалось пройти тестирование и дать обратную связь через форму, содержащую вопросы о процессе прохождения курса, интересности отдельных разделов, а также о сложностях, с которыми пришлось столкнуться обучающимся в процессе прохождения курса. По итогам тестирования (при наличии 50 % правильных ответов и более) выдавались электронные сертификаты о прохождении курса. Консультации преподавателей не проводились, учебные группы не организовывались, также не было и чата поддержки. Таким образом, *курс был полностью асинхронным*.

Статистика посещаемости по разделам представлена в таблице 2. Значения получены путем анализа стати-

стических данных образовательной платформы курса, построенного на базе LMS Moodle.

Таблица 2

Посещаемость основных разделов курса

№ п/п	Раздел	Завершили раздел, % от всех обучающихся на курсе
1	Введение	69,05
2	Язык программирования Python	64,29
3	Анализ данных	59,52
4	Машинное обучение	59,52
5	Компьютерное зрение	59,52
6	Обработка естественного языка	57,14

Видно, что обучающиеся в большинстве (57–69 %) посетили все разделы курса, ознакомились с их содержанием.

После завершения обучения с помощью формы обратной связи, позволяющей «проголосовать» за подразделы (из общего списка), которые обучающимся были наиболее интересны, было определено, какие направления искусственного интеллекта и **практические работы в рамках каждой из тем** (которые были представлены выше в содержании обучения) **вызвали у обучающихся наибольший интерес** (в списке ниже расположены сверху вниз от наиболее интересных к наименее интересным):

1. Компьютерное зрение.
 - 1.1. Практическая работа 5 «Распознавание образов на фотографиях и видео с помощью каскадов Хаара».
 - 1.2. Практическая работа 6 «Обучение искусственной нейронной сети с помощью Google Teachable Machine».
 - 1.3. Практическая работа 4 «Поиск объектов по цвету и контуру».
2. Обработка естественного языка.
 - 2.1. Практическая работа 8 «Распознавание и синтез речи».
 - 2.2. Практическая работа 7 «Предобработка текста и простая поисковая система».
3. Машинное обучение.
 - 3.1. Практическая работа 2 «Кластеризация данных методом k-средних».
 - 3.2. Практическая работа 1 «Искусственный нейрон. Классификация».
 - 3.3. Практическая работа 3 «Решение задачи регрессии с помощью искусственной нейронной сети».

Обучающиеся выбирали, прежде всего, такие практические работы, в которых осуществлялась работа с изображениями и звуком, а не с числами и символами.

Основная сложность, с которой обучающиеся столкнулись при переходе к практике, заключалась в установке необходимых библиотек, несмотря на наличие в курсе подробной инструкции. Здесь можно отметить такие **основные проблемы**:

- не учитывались особенности конфигурации компьютера и установленной операционной системы, права доступа у отдельно взятого пользователя;
- невнимательность обучающихся при написании путей к директории с установленным Python или скачанным на компьютер библиотекам в формате .whl;
- попытки использования Python Shell вместо командной строки операционной системы для установки библиотек через менеджер pip.

При выполнении практических работ основной проблемой, указанной обучающимися, являлся недостаток опыта программирования на языке Python.

По результатам проведения курса можно выделить следующие **недостатки асинхронного формата**:

- потеря интереса и мотивации к изучению темы;
- отсутствие поддержки наставника;
- отсутствие персонализации — все получают один и тот же образовательный контент;
- сложность оценки качества обучения — на основе ответов на тестовые задания, прикрепленных файлов с кодом и отчетов о выполнении практических работ невозможно оценить реальные знания конкретного обучающегося и его практические навыки.

Для компенсации этих недостатков целесообразно:

- проводить консультации в дистанционном формате;
- оказывать поддержку в чате курса в мессенджере;
- организовывать дистанционные устные защиты практических работ.

4. Заключение

Можно констатировать, что обучение школьников технологиям искусственного интеллекта возможно в дистанционном формате при реализации массового онлайн-курса. Однако для повышения результативности обучения, на наш взгляд, оно должно осуществляться *при интеграции синхронного и асинхронного форматов*.

В рамках синхронного формата обучения в учебный курс могут быть включены вебинары по установке необходимого программного обеспечения и введению в программирование на языке Python, онлайн-конференции по защите практических работ или защите итоговых проектов (при небольшом числе участников; разделении обучающихся на группы/потоки; достаточном количестве преподавателей).

Организовывать освоение основного материала по искусственному интеллекту целесообразно *в асинхронном формате*, который позволит учитывать индивидуальный темп каждого обучающегося по изучению достаточно сложных компьютерных технологий.

Список источников

1. Корчажкина О. М. Искусственный интеллект в программе средней школы: чему учить? // Вестник МГПУ. Серия: Информатика и информатизация образования. 2019. № 4. С. 29–42. DOI: 10.25688/2072-9014.2019.50.4.04. EDN: VHWAQW.
2. Лабораторный практикум по ИИ. Приложение к учебно-методическому комплексу. <https://www.lbai.ru/>
3. Левченко И. В., Меренкова П. А. Формирование содержательных модулей для обучения искусственному интеллекту в основной школе // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2021. Т. 18. № 3. С. 227–237. DOI: 10.22363/2312-8631-2021-18-3-227-237. EDN: UOMFOA.
4. Левченко И. В., Садыкова А. Р. Системно-деятельностный подход к обучению искусственному интеллекту в основной школе // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2021. Т. 18. № 2. С. 162–171. DOI: 10.22363/2312-8631-2021-18-2-162-171. EDN: QRADUP.
5. Левченко И. В., Садыкова А. Р., Абушкин Д. Б., Карташова Л. И., Кондратьева В. А., Меренкова П. А. Методические рекомендации по обучению искусственному интеллекту в основной школе: учебно-методическое пособие. М.: Образование и Информатика, 2021. 48 с. EDN: MMKTJL.
6. Розов К. В. Технологии искусственного интеллекта на языке Python 3. 2-е изд., доп. и перераб. Новосибирск: Новосибирский государственный педагогический университет, 2021. 211 с.
7. Самылкина Н. Н., Калинин И. А. Изучение работы нейронной сети на примере распознавания рукописных цифр с использованием языка программирования Python 3.8 в углубленном курсе информатики // Информатика в школе. 2021. № 10. С. 21–29. DOI: 10.32517/2221-1993-2021-20-10-19-27. EDN: WYTCYS.
8. Самылкина Н. Н., Салахова А. А. Обучение основам искусственного интеллекта и анализа данных в курсе информатики на уровне среднего общего образования: монография. М.: МПГУ, 2022. 242 с. DOI: 10.31862/9785426310643. EDN: BACMCW.
9. Самылкина Н. Н., Салахова А. А. Основы искусственного интеллекта и анализа данных в курсе информатики на уровне среднего общего образования // Информационные технологии в образовательном процессе вуза и школы. Материалы XVI Всероссийской научно-практической конференции. Воронеж: Воронежский государственный педагогический университет, 2022. С. 225–233. EDN: WFUYRC.
10. Урок Цифры — всероссийский образовательный проект в сфере информационных технологий. <https://урокцифры.рф>
11. Шрайнер Б. А., Розов К. В. Введение в искусственный интеллект. Новосибирск: Новосибирский государственный педагогический университет, 2021. 101 с.
12. Ясницкий Л. Н., Черепанов Ф. М. Возможности преподавания курса «Искусственный интеллект» в школе // Информатика в школе: прошлое, настоящее и будущее. Материалы Всероссийской научно-методической конференции по вопросам применения ИКТ в образовании. Пермь: Пермский государственный национальный исследовательский университет, 2014. С. 271–273. EDN: SEGORP.
13. AI-ACADEMY Академия искусственного интеллекта для школьников. <https://ai-academy.ru/>

DOI: 10.32517/2221-1993-2022-21-6-44-49

Е. Н. Дронова

Алтайский государственный педагогический университет, г. Барнаул, Россия

РЕАЛИЗАЦИЯ МЕЖПРЕДМЕТНЫХ СВЯЗЕЙ ИНФОРМАТИКИ И МАТЕМАТИКИ ПРИ ИЗУЧЕНИИ ТЕМЫ «ОБРАБОТКА ЧИСЛОВОЙ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ»

Аннотация

В статье рассмотрены возможности реализации межпредметных связей информатики и математики в школе при изучении темы «Обработка числовой информации в электронных таблицах». Обосновано, что межпредметные связи в учебных дисциплинах устанавливаются через межпредметные понятия. Выделены ключевые понятия в учебной теме «Обработка числовой информации в электронных таблицах» школьного курса информатики. С опорой на эти понятия представлены примеры решения задач из школьного курса математики в электронных таблицах: построение таблицы квадратов двухзначных чисел, построение графиков функций, решение системы уравнений графическим способом. Приведенные примеры демонстрируют возможность реализации межпредметных связей информатики и математики в контексте изучения обработки числовой информации с помощью электронных таблиц: с одной стороны, применение электронных таблиц позволяет автоматизировать рутинные табличные вычисления в математике, с другой стороны, решение таких задач позволяет учащимся глубже осознать организацию вычислений в электронных таблицах.

Ключевые слова: информатика, математика, межпредметные связи, электронные таблицы, обработка числовой информации.

1. Введение

Информатика как наука об обработке информации имеет широкое прикладное значение. Этой особенностью отличается и содержание школьного учебного предмета «Информатика». Методы, средства, технологии обработки информации, которые изучаются на уроках информатики, могут использоваться во всех без преувеличения школьных учебных предметах.

Особо тесная связь присутствует между школьными учебными предметами «Информатика» и «Математика». Содержание этих предметов в отдельных учебных темах очень сильно переплетается, но, к сожалению,

и зачастую несогласованно, что отрицательно влияет на качество успеваемости учащихся по информатике [10]. О необходимой согласованности в обучении информатике и математике, о важности реализации межпредметных связей информатики и математики говорится в работах многих педагогов-исследователей, особо отметим статью М. М. Абдуразакова, О. Ю. Лягиновой, О. Н. Цветковой [1].

В нашей статье мы рассмотрим такую тему школьного курса информатики, как «Обработка числовой информации с помощью электронных таблиц», и на основе содержания этой темы продемонстрируем реализацию межпредметных связей информатики и математики.

Контактная информация

Дронова Екатерина Николаевна, канд. пед. наук, доцент, доцент кафедры теоретических основ информатики, Институт информационных технологий и физико-математического образования, Алтайский государственный педагогический университет, г. Барнаул, Россия; адрес: 656031, Россия, Алтайский край, г. Барнаул, ул. Молодежная, д. 55; e-mail: dronova_ekn@altspu.ru

E. N. Dronova

Altai State Pedagogical University, Barnaul, Russia

IMPLEMENTATION OF INTERDISCIPLINARY CONNECTIONS OF INFORMATICS AND MATHEMATICS IN THE STUDY OF THE THEME "PROCESSING OF NUMERICAL INFORMATION IN SPREADSHEETS"

Abstract

The article describes the possibilities of implementing interdisciplinary connections of informatics and mathematics at school when studying the theme "Processing numerical information in spreadsheets". It is substantiated that interdisciplinary connections in academic disciplines are established through interdisciplinary concepts. The key concepts in the educational theme "Processing numerical information in spreadsheets" of the school informatics course are highlighted. Based on these concepts, examples of solving problems from a school mathematics course in spreadsheets are presented: building a table of squares of two-digit numbers, plotting of functions, solving a system of equations graphically. The examples demonstrate the possibility of implementing interdisciplinary connections of informatics and mathematics in the context of studying the processing of numerical information using spreadsheets: on the one hand, the use of spreadsheets allows you to automate routine tabular calculations in mathematics, on the other hand, solving such problems allows students to better understand the organization of calculations in spreadsheets.

Keywords: informatics, mathematics, interdisciplinary connections, spreadsheets, processing of numerical information.

2. Ключевые понятия учебной темы «Обработка числовой информации в электронных таблицах» как средство реализации межпредметных связей

Межпредметные связи — это связи между учебными предметами, а точнее между структурными элементами их содержания, выраженными в понятиях, научных фактах, законах, теориях. А так как научные факты, законы и теории формируются через понятия или выражают связь между ними, то в итоге межпредметные связи — это связи между понятиями в различных дисциплинах. Этот факт подчеркивается и в статье И. А. Васильевой, М. Ю. Солощенко «Осуществление межпредметных связей математики и информатики»: межпредметные связи «дают наилучшее формирование межпредметных понятий, полное понимание о которых не может быть передано учащимся на уроках какой-нибудь одной дисциплины» [5, с. 164].

Проанализируем содержание учебной темы «Обработка числовой информации в электронных таблицах» и выявим ключевые понятия в ней. За основу возьмем УМК по информатике Л. Л. Босовой и А. Ю. Босовой. Тема «Обработка числовой информации в электронных таблицах» изучается в IX классе и включает следующие параграфы: «Электронные таблицы», «Организация вычислений в электронных таблицах», «Средства анализа и визуализации данных» [2]. В таблице приведен перечень ключевых понятий, изучаемых в теме «Обработка числовой информации в электронных таблицах», он разбит в соответствии с выделенными в учебнике параграфами.

Отметим, что в XI классе учащиеся вновь сталкиваются с электронными таблицами на уроках информатики [3]. Осуществляются повторение и систематизация

Таблица

Ключевые понятия темы «Обработка числовой информации в электронных таблицах»

Параграфы темы	Ключевые понятия
Электронные таблицы	<ul style="list-style-type: none"> • электронные таблицы • табличный процессор • столбец • строка • ячейка • диапазон ячеек • лист • книга
Организация вычислений в электронных таблицах	<ul style="list-style-type: none"> • относительная ссылка • абсолютная ссылка • смешанная ссылка • встроенная функция • логическая функция • условная функция
Средства анализа и визуализации данных	<ul style="list-style-type: none"> • сортировка • поиск (фильтрация) • диаграмма • график • круговая диаграмма • гистограмма (столбчатая диаграмма) • ярусная диаграмма • ряды данных • категории

учебного материала, пройденного в IX классе, происходит его обобщение до уровня обработки разных видов информации в электронных таблицах. В XI классе соответствующая тема называется «Обработка информации в электронных таблицах» [2].

Однако, несмотря на то что в электронных таблицах можно обрабатывать разные виды информации, важнейшей областью их применения является обработка именно числовой информации. Этот факт имеет отражение и в определении понятия «электронные таблицы», которое дается в учебнике информатики IX класса: «Электронные таблицы (табличный процессор) — это прикладная программа, предназначенная для организации табличных вычислений на компьютере. Электронными называют и таблицы, созданные с помощью одноименных программ в памяти компьютера» [2, с. 118].

Закономерны вопросы: «А встречаются ли табличные вычисления в школьном курсе математики? Если да, то где именно?» Приведем соответствующие примеры и на их основе продемонстрируем возможность реализации межпредметных связей информатики и математики.

3. Примеры реализации межпредметных связей информатики и математики при изучении темы «Обработка числовой информации в электронных таблицах»

Пример 1.

Построим в электронных таблицах таблицу квадратов двухзначных чисел.

Межпредметные связи с математикой.

Эта таблица широко используется в школьном курсе алгебры при решении квадратных уравнений, а именно при вычислении квадратного корня из дискриминанта. С этой задачей учащиеся впервые сталкиваются на уроках алгебры в VIII классе [7], далее обращаются к этой таблице на протяжении всего школьного курса математики вплоть до XI класса.

Решение.

1. Создаем структуру таблицы квадратов двухзначных чисел в электронных таблицах (рис. 1). Рекомендуем придерживаться «классической» структуры, используемой в математике: по строкам располагаем десятки, по столбцам — единицы.

2. Продумываем формулу для вычисления квадрата двухзначного числа в произвольной ячейке таблицы. Например, в ячейке F7 формула будет следующей:

$$=(B7*10+F2)^2$$

В этой формуле:

- 1) из ячеек B7 и F2 мы формируем двухзначное число, в данном случае 43;
- 2) сформированное двухзначное число возводим во вторую степень.

Вычисление квадратов остальных двухзначных чисел в таблице будет аналогичное, поэтому сформированная формула будет копироваться в другие ячейки таблицы, а значит, необходимо продумать тип ссылок (абсолют-

	A	B	C	D	E	F	G	H	I	J	K	L
1			Единицы									
2			0	1	2	3	4	5	6	7	8	9
3	Десятки	0										
4		1										
5		2										
6		3										
7		4										
8		5										
9		6										
10		7										
11		8										
12		9										

Рис. 1. Структура таблицы квадратов двухзначных чисел в электронных таблицах

ные/относительные/смешанные) на используемые в формуле ячейки B7 и F2.

В условиях нашей задачи необходимы: смешанная ссылка \$B7 (столбец B фиксируется, а номер строки меняется при копировании формулы) и смешанная ссылка F\$2 (строка 2 фиксируется, а столбцы будут меняться при копировании формулы).

Итоговая формула в ячейке F7 будет следующей (рис. 2):

$$=($B7*10+F$2)^2$$

3. Копируем формулу из ячейки F7 на всю таблицу (рис. 3).

Замечание к примеру 1.

Таблица квадратов двухзначных чисел является самой известной в школьном курсе математики для уча-

щихся средних и старших классов, она часто печатается на форзацах школьных учебников математики, в справочных материалах ОГЭ по математике и в справочных материалах ЕГЭ по математике базового уровня.

Однако нужно обратить внимание девятиклассников на то, что в математике, в том числе в школьной математике, используются и другие виды таблиц:

- в начальной школе учащиеся изучают таблицу Пифагора, более известную как таблица умножения;
- в VII классе на уроках алгебры при изучении темы «Степень с натуральным показателем» учащиеся знакомятся с таблицей квадратов и кубов однозначных чисел;
- в старших классах при углубленном изучении математики учащиеся могут познакомиться с пособием В. М. Бродиса «Четырехзначные математические таблицы» [4], в котором собраны

	A	B	C	D	E	F	G	H	I	J	K	L
1			Единицы									
2			0	1	2	3	4	5	6	7	8	9
3	Десятки	0										
4		1										
5		2										
6		3										
7		4					=(\$B7*10+F\$2)^2					
8		5										
9		6										
10		7										
11		8										
12		9										

Рис. 2. Формула для вычисления квадрата двухзначного числа в электронных таблицах

	A	B	C	D	E	F	G	H	I	J	K	L
1			Единицы									
2			0	1	2	3	4	5	6	7	8	9
3	Десятки	0	0	1	4	9	16	25	36	49	64	81
4		1	100	121	144	169	196	225	256	289	324	361
5		2	400	441	484	529	576	625	676	729	784	841
6		3	900	961	1024	1089	1156	1225	1296	1369	1444	1521
7		4	1600	1681	1764	1849	1936	2025	2116	2209	2304	2401
8		5	2500	2601	2704	2809	2916	3025	3136	3249	3364	3481
9		6	3600	3721	3844	3969	4096	4225	4356	4489	4624	4761
10		7	4900	5041	5184	5329	5476	5625	5776	5929	6084	6241
11		8	6400	6561	6724	6889	7056	7225	7396	7569	7744	7921
12		9	8100	8281	8464	8649	8836	9025	9216	9409	9604	9801

Рис. 3. Таблица квадратов двухзначных чисел в электронных таблицах

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	x	-3	-2,5	-2	-1,5	-1	-0,5	0	0,5	1	1,5	2	2,5	3
2	y1	9	6,25	4	2,25	1	0,25	0	0,25	1	2,25	4	6,25	9
3	y2	7	4,25	2	0,25	-1	-1,75	-2	-1,75	-1	0,25	2	4,25	7
4	y3	11	8,25	6	4,25	3	2,25	2	2,25	3	4,25	6	8,25	11

Рис. 4. Таблица значений функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$ в электронных таблицах

математические таблицы для практических вычислений, не требующих большой точности (точность таблиц в пособии — 4 знака после запятой, поэтому они и называются четырехзначными). В этом пособии приводятся, например, такие таблицы, как: «Синусы и косинусы», «Тангенсы и котангенсы», «Радианная мера», «Тригонометрические функции от аргумента в радианах» и др.

Пример 2.

Построим в электронных таблицах графики функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$ и выявим их взаимосвязь.

Межпредметные связи с математикой.

В IX классе на уроках алгебры учащиеся изучают преобразования графиков функций, которые выводятся на основе рассмотрения частных случаев [8]. Частные случаи построения графиков функций реализуются по схеме, известной учащимся еще с VII класса [6]:

- 1) построение таблицы значений рассматриваемой функции;
- 2) построение графика рассматриваемой функции на основе координат точек, вычисленных в п. 1.

Точность данной схемы построения графика функции увеличивается при увеличении количества вычисленных координат точек, что требует значительной технической работы по получению таблицы значений функции, — эту работу могут взять на себя электронные таблицы.

Решение.

1. Построим в электронных таблицах таблицу значений для функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$. Значения независимой переменной x возьмем от -3 до 3 с шагом $0,5$ (рис. 4).

В ячейку B2 запишем формулу:

$$=B1^2$$

В ячейку B3 запишем формулу:

$$=(B1^2)-2$$

В ячейку B4 запишем формулу:

$$=(B1^2)+2$$

В условиях нашей задачи во всех трех формулах необходима относительная ссылка B1 (при копировании формулы вправо столбцы будут меняться).

2. Используя в электронных таблицах инструмент «Диаграммы», построим графики функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$ на основе координат точек, вычисленных в п. 1. Для этой цели в электронных таблицах можно использовать два типа диаграмм: «График» и «Точечная». Эти типы диаграмм очень похожи друг на друга, особенно если точечная диаграмма отображается

с соединительными линиями, однако они отличаются способом печати данных по горизонтальной оси и вертикальной оси.

Мы использовали для построения графиков рассматриваемых функций тип диаграмм «Точечная». Точечный тип диаграмм разрешает редактировать цену деления и по горизонтальной оси, и по вертикальной оси, что позволяет в итоге получить картинку графиков функций, соответствующую привычному представлению учащихся о них (рис. 5).

3. Рисунок 5 демонстрирует взаимосвязь графиков функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$: график функции $y_2 = x^2 - 2$ получен в результате параллельного переноса

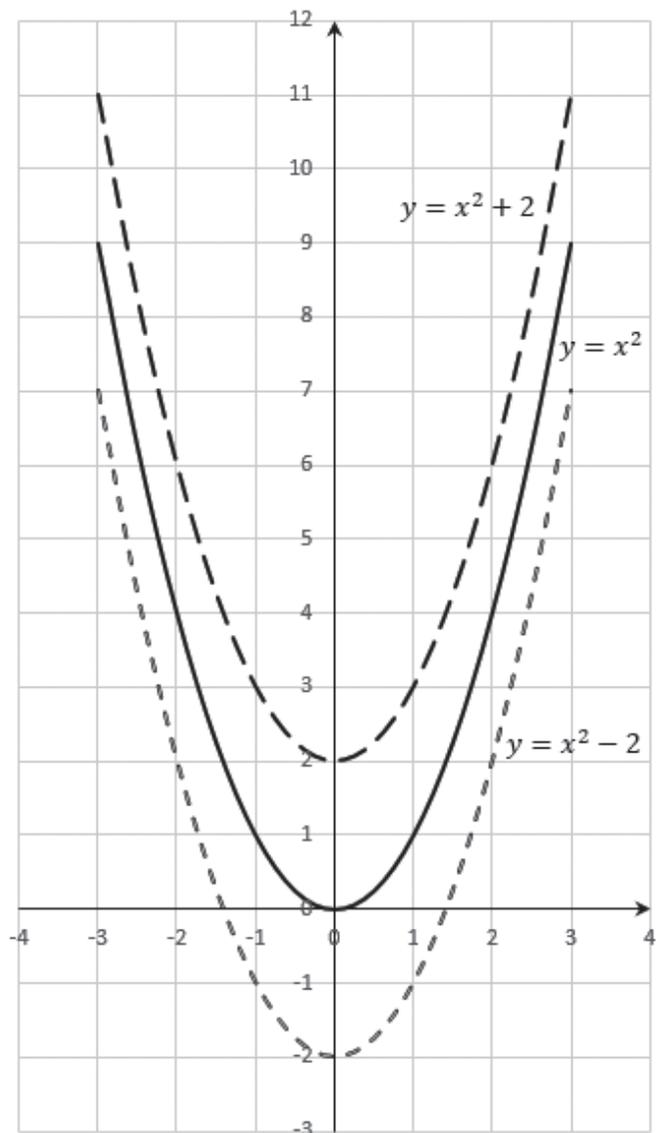


Рис. 5. Графики функций $y_1 = x^2$, $y_2 = x^2 - 2$, $y_3 = x^2 + 2$ в электронных таблицах

	A	B	C	D	E	F	G	H	I	J	K	L
1	x	-5	-4	-3	-2	-1	0	1	2	3	4	5
2	y1	5	4	3	2	1	0	1	2	3	4	5
3	y2	-23	-14	-7	-2	1	2	1	-2	-7	-14	-23
4												
5	a=	2										

Рис. 6. Таблица значений функций $y_1 = |x|$, $y_2 = a - x^2$ в электронных таблицах

графика функции $y_1 = x^2$ на 2 единицы вниз, а график функции $y_3 = x^2 + 2$ получен в результате параллельного переноса графика функции $y_1 = x^2$ на 2 единицы вверх.

Замечание к примеру 2.

Рассмотренный пример подсказывает правило построения графика функции $y = f(x) + b$ из графика функции $y = f(x)$: график функции $y = f(x) + b$ можно получить в результате параллельного переноса графика функции $y = f(x)$ на b единиц вверх, если $b > 0$, и на b единиц вниз, если $b < 0$.

Аналогичные задания можно предложить учащимся для получения двух других правил преобразования графиков функций:

- график функции $y = f(x + a)$ можно получить в результате параллельного переноса графика функции $y = f(x)$ на a единиц влево, если $a > 0$, и на a единиц вправо, если $a < 0$;
- график функции $y = kf(x)$ можно получить из графика функции $y = f(x)$ в результате растяжения в k раз от оси абсцисс, если $k > 1$, или в результате сжатия в k раз к оси абсцисс, если $0 < k < 1$.

Пример 3.

В электронных таблицах исследуем количество решений системы уравнений

$$\begin{cases} y = |x|; \\ y = a - x^2. \end{cases}$$

в зависимости от значения a .

Межпредметные связи с математикой.

В IX классе на уроках алгебры учащиеся изучают методы решения систем уравнений (графический метод, метод подстановки, метод сложения) [8]. Для исследования количества решений данной системы уравнений в зависимости от значения a нужно использовать графический метод. Существенно сократить техническую работу при использовании графического метода решения приведенной системы уравнений помогут электронные таблицы.

Решение.

1. Построим таблицу значений для функций $y_1 = |x|$ и $y_2 = a - x^2$ в электронных таблицах. Значения независимой переменной x возьмем от -5 до 5 с шагом 1.

В ячейку B2 запишем формулу:

$$=ABS(B1)$$

(встроенная функция ABS возвращает модуль числа).

В ячейку B3 запишем формулу:

$$=B5-B1^2$$

(в ячейке B5 будем попеременно вводить разные значения параметра a).

В условиях нашей задачи: в обеих формулах необходима относительная ссылка B1 (при копировании формулы вправо столбцы будут меняться), во второй формуле — абсолютная ссылка \$B\$5 (при копировании формулы вправо ссылка не изменится). На рисунке 6 представлена таблица значений для функций $y_1 = |x|$, $y_2 = a - x^2$ при $a = 2$.

2. Используя в электронных таблицах тип диаграмм «Точечная», построим графики функций $y_1 = |x|$, $y_2 = a - x^2$ при $a = 2$ (рис. 7).

Из рисунка 7 видим, что при $a = 2$ построенные графики функций пересекаются в двух точках, а значит, исходная система уравнений имеет два решения.

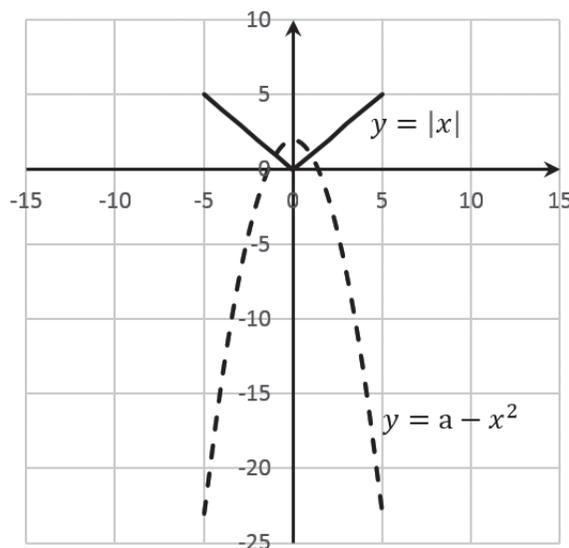


Рис. 7. Графики функций $y_1 = |x|$, $y_2 = a - x^2$ при $a = 2$ в электронных таблицах

3. Далее исследуем количество решений заданной системы уравнений в зависимости от значения a : в ячейку B5 попеременно вводим разные значения параметра a ($a = 1$, $a = 0$, $a = -1$, $a = -2$, $a = -3$, ... и далее $a = 3$, $a = 4$, $a = 5$, ...) (рис. 8, 9) и наблюдаем за взаимным расположением графиков функций на диаграмме.

В ходе исследования находим, что:

- при $a > 0$ система имеет два решения (две точки пересечения графиков функций y_1 и y_2);
- при $a = 0$ система имеет одно решение (одну точку пересечения графиков функций y_1 и y_2);
- при $a < 0$ система не имеет решений (графики функций y_1 и y_2 не пересекаются).

Ответ. При $a > 0$ система имеет два решения, при $a = 0$ — одно решение, при $a < 0$ нет решений.

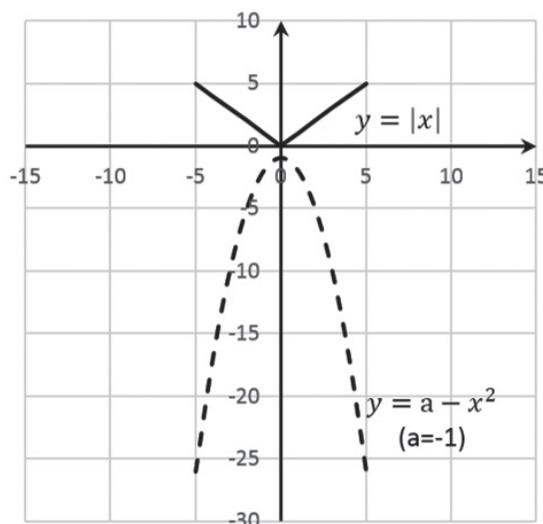
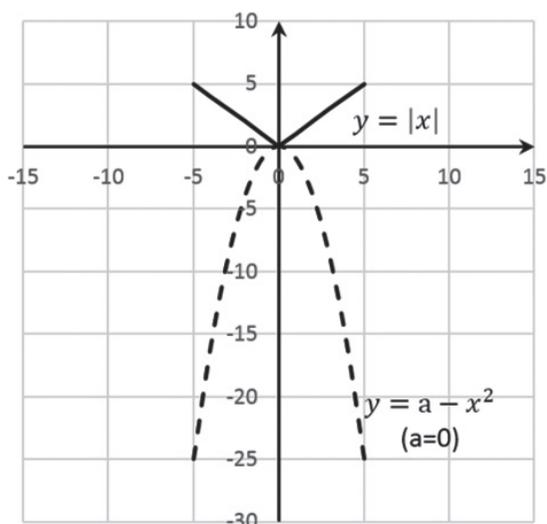


Рис. 8. Графики функций $y_1 = |x|$, $y_2 = a - x^2$ при $a = 0$, $a = -1$ в электронных таблицах

4. Заключение

Представленные примеры решения математических задач в электронных таблицах демонстрируют межпредметные связи информатики и математики. С одной стороны, применение электронных таблиц позволяет автоматизировать рутинные табличные вычисления в математике. С другой стороны, решение таких задач позволяет учащимся глубже осознать организацию вычислений в электронных таблицах через использование разных типов ссылок (относительных, абсолютных, смешанных). Использование электронных таблиц как инструмента для построения графиков функций способствует более глубокому освоению учащимися различных настроек по форматированию диаграмм, а также за счет использования встроенных математических функций позволяет расширить список функций, графики которых учащиеся строят на уроках математики.

Отметим, что на уроках математики учащиеся работают не только с таким средством визуализации данных, как график. В VI классе на уроках математики [9] учащиеся знакомятся со столбчатой диаграммой и круговой диаграммой. Это знакомство ограничивается «чтением» готовых диаграмм учащимися; построением столбчатых и круговых диаграмм «вручную» на уроках математики не занимаются (это требует достаточно громоздких математических вычислений). Изучение на уроках информатики в IX классе электронных таблиц как инструмента построения разных типов диаграмм является логическим продолжением темы «Диаграммы» школьного курса математики VI класса. Здесь учащиеся вновь видят, что электронные таблицы освобождают человека от рутинных громоздких вычислений и автоматизируют процесс построения диаграмм разных типов.

В завершение сделаем несколько выводов:

- анализ ключевых понятий по теме «Обработка числовой информации в электронных таблицах» (см. табл.) показывает, что только два понятия, используемые в электронных таблицах, — «сор-

тировка» и «поиск (фильтрация)» — не находят отражения в школьных математических задачах;

- учебная тема по информатике «Обработка числовой информации в электронных таблицах» имеет большие возможности для реализации межпредметных связей информатики и математики как связей между понятиями в этих учебных дисциплинах;
- реализация межпредметных связей информатики и математики при изучении темы «Обработка числовой информации в электронных таблицах» обогащает учебное содержание и школьного курса информатики, и школьного курса математики.

Список источников

1. Абдуразаков М. М., Лягинова О. Ю., Цветкова О. Н. Информатика, математика и логика в аспекте межпредметной и метапредметной образовательной связи // Чебышевский сборник. 2021. Т. 22. № 2 (78). С. 373–388. DOI: 10.22405/2226-8383-2021-22-2-373-388. EDN: RUTLFH.
2. Босова Л. Л., Босова А. Ю. Информатика. 9 класс. М.: БИНОМ. Лаборатория знаний, 2017. 208 с.
3. Босова Л. Л., Босова А. Ю. Информатика. 11 класс. Базовый уровень. М.: БИНОМ. Лаборатория знаний, 2017. 256 с.
4. Бродис В. М. Четырехзначные математические таблицы. М.: Дрофа, 2010. 93 с.
5. Васильева И. А., Солощенко М. Ю. Осуществление межпредметных связей математики и информатики // Modern Science. 2021. № 11-4. С. 164–168. EDN: ZRVIJX.
6. Мерзляк А. Г., Полонский В. Б., Якир М. С. Алгебра: 7 класс: учебник для учащихся общеобразовательных учреждений. М.: Вентана-Граф, 2015. 272 с.
7. Мерзляк А. Г., Полонский В. Б., Якир М. С. Алгебра: 8 класс: учебник для учащихся общеобразовательных учреждений. М.: Вентана-Граф, 2013. 256 с.
8. Мерзляк А. Г., Полонский В. Б., Якир М. С. Алгебра: 9 класс: учебник для учащихся общеобразовательных учреждений. М.: Вентана-Граф, 2014. 304 с.
9. Мерзляк А. Г., Полонский В. Б., Якир М. С. Математика: 6 класс: учебник для учащихся общеобразовательных организаций. М.: Вентана-Граф, 2014. 304 с.
10. Нефедов Д. Е. Об интеграции содержания школьных предметов математики и информатики // Ярославский педагогический вестник. 2015. Т. 2. № 2. С. 34–38. EDN: TQJVOZ.

DOI: 10.32517/2221-1993-2022-21-6-50-54

А. С. Паравина*Средняя общеобразовательная школа № 5 имени Героя Советского Союза В. Ф. Кравченко,
г. Сызрань, Самарская область, Россия*

ЧЕК-ЛИСТ ПО ТЕМЕ УРОКА КАК СРЕДСТВО ОБРАТНОЙ СВЯЗИ ДЛЯ УЧИТЕЛЯ*

Аннотация

Задачей современного учителя является создание условий для личностного, эмоционального и умственного развития учеников, включение процессов саморефлексии и самооценивания в урочную деятельность для формирования внутренней мотивации школьников. Анализ общей картины понимания учащимися темы урока, доли учеников, усвоивших тему и имеющих проблему усвоения, может дать применение такого средства обратной связи, как чек-лист. В статье рассматривается применение чек-листов на уроке информатики, особенности создания чек-листов по темам урока, приведены образцы чек-листов, используемых на уроках информатики в V–IX классах. Предложенные образцы чек-листов, разработанные с учетом требований педагогического дизайна, являются эффективным средством обратной связи для учителя и способствуют развитию у обучающихся навыков самоконтроля, самооценивания, самоанализа и самокоррекции.

Ключевые слова: чек-лист, технология формирующего оценивания, контрольно-оценочная деятельность, самооценка.

1. Введение

Современные школьники все чаще интересуются: для чего вообще им нужна та информация, которую они изучают на уроке? Как она им может пригодиться в жизни? Сниженная мотивация к обучению приводит к лавинообразному нарастанию проблем с пониманием учебного материала, с заучиванием необходимого объема информации и, как следствие, к привыканию к ситуации неуспешности в школе. Задача учителя — создать условия для личностного, эмоционального и умственного развития учащихся, включить процессы саморефлексии и самооценивания в урочную деятельность для формирования внутренней мотивации школьников. Необходимо создавать на уроке такие учебные ситуации, в которых каждый школьник сможет наблюдать собственный рост и чувствовать себя успешным.

Анализ общей картины понимания учащимися темы урока, доли учеников, усвоивших тему и имеющих проблему усвоения, может дать применение такого средства обратной связи, как **чек-лист (проверочный лист)** [1, 3, 9]. Понятие «чек-лист» заимствовано из технологии формирующего оценивания [4].

Использование чек-листов включает школьников в контрольно-оценочную деятельность и формирует навыки самооценки. Чек-лист по теме урока позволит ученику понять, насколько успешно идет усвоение темы, и наблюдать даже небольшой свой прогресс. То есть в данном случае процесс оценивания осуществляется не учителем, а самим учеником, причем ученик может оценить себя достаточно объективно. Что касается учителя, то ему чек-лист поможет скорректировать учебный процесс в зависимости от проблем, выявленных после анализа того, как учащиеся заполнили чек-листы.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_06-2022/

Контактная информация

Паравина Алина Сергеевна, учитель информатики, средняя общеобразовательная школа № 5 имени Героя Советского Союза В. Ф. Кравченко, г. Сызрань, Самарская область, Россия; *адрес:* 446022, Россия, Самарская область, г. Сызрань, ул. Новостроящаяся, д. 18-А; *e-mail:* semakina.alina@mail.ru

A. S. Paravina

School 5 named after Hero of the Soviet Union V. F. Kravchenko, Syzran, Samara Region, Russia

CHECKLIST ON THE THEME OF THE LESSON AS A MEANS OF FEEDBACK FOR THE TEACHER

Abstract

The aim of a modern teacher is to create conditions for the personal, emotional and mental development of students, to include the processes of self-reflection and self-assessment in class activities to form the internal motivation of students. An analysis of the overall picture of students' understanding of the theme of the lesson, the proportion of students who have mastered the theme and have a problem with mastering, can provide the use of such a feedback tool as a checklist. The article discusses the use of checklists in an informatics lesson, the features of creating checklists on the theme of the lesson, and provides samples of checklists used in informatics lessons in grades 5–9. The proposed samples of checklists, developed taking into account the requirements of pedagogical design, are an effective means of feedback for the teacher and contribute to the development of students' skills of self-control, self-assessment, self-analysis and self-correction.

Keywords: checklist, technology of formative assessment, control and assessment activity, self-assessment.

2. Педагогический дизайн чек-листов

Чек-лист содержит задания различного типа, которые подходят для учащихся как с высокими, так и с низкими образовательными результатами. Задания подбираются так, чтобы любой ученик смог решить хотя бы одно задание. А для сильных учеников нужно включать задания посложнее. Важно помнить, что чек-лист по итогам урока — это не задачник, а единичные задания для понимания того, усвоены ли навыки, отрабатываемые на уроке. Чек-лист по теме урока может содержать задания, выполняемые в течение всего урока. Такой чек-лист можно разделить на блоки, отвечающие каждый за свой навык.

Чтобы пробудить интерес учащегося к выполнению заданий, желательно подобрать удачное оформление чек-листа: картинки, фоновые изображения, стильное оформление заданий. Текст должен быть читаемым, графики и диаграммы — с хорошо читаемыми данными.

Весь объем материала желательно разместить на одной странице, максимум на двух, так как большое количество заданий может отпугнуть учеников.

Для самоанализа учащимся своих проблем и достижений рекомендуется включить в чек-лист блок с воз-

можностью оценить свой уровень по параметрам типа: знаю/умею/затрудняюсь, знаю/сомневаюсь/требуется помощь, знаю/могу выполнить сам(а)/нужна помощь и т. п.

Чек-листы позволяют обучающемуся получить результаты в виде выводов, а не отметок, тем самым сохраняется его мотивация к обучению.

По итогам оценивания учениками самих себя у учителя будет представление о том, скольким обучающимся удалось хорошо усвоить материал, а скольким нужно помочь дополнительно, и скорректировать свою педагогическую деятельность в дальнейшем.

Чек-листы также можно использовать для проверки домашнего задания.

Для осуществления системной работы с учащимися чек-листы желательно готовить к каждому уроку [3, 7, 9].

3. Примеры чек-листов для основной школы

Чек-лист для V класса «Информация вокруг нас» (рис. 1) и чек-лист для VI класса «Графики и диаграммы» (рис. 2) являются чек-листами по итогам урока и при-

Чек-лист по теме «Информация вокруг нас»

И
Н
Ф
О
Р
М
А
Ц
И
Я

По способу восприятия

Знаю

Сомневаюсь

Требуется помощь

По способу представления

Знаю

Сомневаюсь

Требуется помощь

Было интересно

Я все понял(а)

Рис. 1. Чек-лист для V класса

Рис. 2. Чек-лист для VI класса

Чек-лист по теме «Графики и диаграммы»

Г
Р
А
Ф
И
К
И
И
Д
И
А
Г
Р
А
М
М

Виды диаграмм?

График холдинга Винни-Пуха по гостям

Динамика количества уч. в классе

Знаю

Сомневаюсь

Требуется помощь

Поедание плюшек Карлсоном

3. Сколько всего плюшек съел Карлсон за выходные дни?

На графике, изображенном на рисунке 11, представлено изменение биржевой стоимости акций газодобывающей компании в первые две недели апреля. В первую неделю апреля бизнесмен купил 14 акций, а потом продал их на второй неделе. Какую наибольшую прибыль он мог получить?

Было интересно

Я всё понял(а)

Чек-лист по теме «Измерение информации»

И
З
М
Е
Р
Е
Н
И
Е
И
Н
Ф
О
Р
М
А
Ц
И
И

$N = 2^i$

$I = K * i$

Дано: $N = 1024$

Решение: $K = 1024$

$I = ?$

$1,5 \text{ Гб} = \text{байт}$

$3,5 \text{ Мб} = \text{Кб}$

$3072 \text{ Гб} = \text{Тб}$

$1024 \text{ Кб} = 2 \text{ бит}$

Минимальная единица количества информации называется . Это такая информация, которую мы получаем при выборе одного из вариантов. Например, ответ на вопрос "Вы будете чай или кофе?" даёт нам один информации.

Знаю

Сомневаюсь

Требуется помощь

Было познавательно

Я всё понял(а)

Рис. 3. Чек-лист для VII класса

меняются в конце занятия для подведения итогов урока и систематизации знаний по теме урока [2, 5–8].

Чек-лист для VII класса «Измерение информации» (рис. 3) и чек-лист для VIII класса «Двоичная система счисления» (рис. 4) являются чек-листами по теме урока и применяются в течение всего урока для систематизации знаний по теме урока [2, 5–8].

Чек-лист для IX класса «Графы» (рис. 5) является чек-листом по итогам урока и применяется в конце урока для подведения итогов урока и систематизации знаний по теме урока [2, 5–8].

4. Заключение

Чек-листы выступают эффективным средством обратной связи для учителя по итогам урока или изучения раздела (темы) курса. Обратная связь необходима и учителю, и ученику в процессе обучения. При помощи обратной связи можно скорректировать процесс обучения для движения к поставленной цели обучения. Обратная связь может способствовать формированию

у обучающихся уверенности в своих силах и повышению успеваемости.

Список источников

1. *Гаванде А.* Чек-лист (система предотвращения ошибок): пер. с англ. М.: Альпина Паблишер, 2022. 200 с. (*Gawande A.* The checklist manifesto: How to get things right.)
2. ИнтернетУрок. Библиотека видеоуроков по школьной программе. <https://interneturok.ru/>
3. *Лазарева С. В.* Методическое пособие «Педагогика творчества: от теории к практике». https://toipkro.ru/content/files/documents/metodicheskoe_posobie.pdf
4. *Пинская М. А.* Формирующее оценивание: оценивание в классе: учебное пособие. М.: Логос, 2010. 264 с.
5. Российская электронная школа. <https://resh.edu.ru/>
6. УМК «Информатика». Авторы Босова Л. Л., Босова А. Ю. <https://bosova.ru/>
7. Чек-лист // Учитель. Club. <https://uchitel.club/parents-checklists/>
8. ЯКласс. <https://www.yaklass.ru/>
9. *Ярыгина З. А., Ярыгин А. Н.* Чек-лист — новая технология в школьном образовании? // Центр научного сотрудничества «Интерактив плюс». <https://interactive-plus.ru/e-articles/607/Action607-496353.pdf>



DOI: 10.32517/2221-1993-2022-21-6-55-67

М. С. Долинский

Гомельский государственный университет имени Франциска Скорины, г. Гомель, Беларусь

УСКОРЕНИЕ РЕКУРСИВНЫХ РЕШЕНИЙ ПРИ ПОМОЩИ МЕМОИЗАЦИИ*

Аннотация

В статье на примере решения двух задач проиллюстрирована методика изучения темы «Ускорение рекурсивных решений при помощи мемоизации» при подготовке школьников к олимпиадам по информатике. Изучение основано на последовательном решении усложняющихся задач. Для каждой задачи приводятся следующие материалы: условие задачи, идея решения с предложением придумать самостоятельно реализацию, решение на языке программирования Pascal. Серьезной технической основой является разработанная под управлением автора инструментальная система дистанционного обучения (<http://dl.gsu.by>), которая позволяет: предложить ученику условие задачи; отправить решение на проверку; получить от системы вердикт — правильное или неправильное решение; для неправильных решений указывается номер теста, на котором решение не прошло. Ученик может взять тест (входные и выходные данные), на котором не прошло его решение, разобраться, в чем ошибка в его программе, исправить и послать решение повторно. Кроме того, для каждой задачи есть ссылка по ней на тему в форуме, где можно задать вопрос по решению этой задачи и/или почитать ответ, если вопросы уже задавались ранее.

Ключевые слова: рекурсия, мемоизация, олимпиады по информатике, инструментальная система дистанционного обучения.

1. Введение

Подготовка школьников к олимпиадам по информатике и программированию требует от учащихся значительного объема теоретических знаний и практических навыков решения задач. Поэтому в мире компьютерной книжной и журнальной литературы регулярно появляются книги и статьи как по общей теории алгоритмов (см., например, [5, 10, 11, 13, 14, 16, 18–20, 23]), так и на отдельные актуальные темы: динамическое программирование [12, 27, 28], графы [15, 22, 23, 25, 26, 29], хеши [24, 30], рекурсия [1, 2–4, 17]. Автор вносит свою лепту в разработку темы «Рекурсия»

[5–10] специальным подходом, связанным с ранним началом обучения. Как следствие, уже в пятом–шестом классах появляются ученики, которым приходится объяснять такие темы, как «Рекурсия». Поэтому приходится модифицировать изложение в сторону более простого и наглядного объяснения и более медленного продвижения по учебному материалу, явно выделяя и обозначая все этапы этого продвижения. Введение в решение задач с помощью рекурсии изложено в работе [5]. Решение задач с помощью рекурсивной генерации таких комбинаторных объектов, как: множество всех подмножеств, сочетания, перестановки, перестановки с повторениями, размещения, описано в работе [6].

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_06-2022/

Контактная информация

Долинский Михаил Семенович, канд. тех. наук, доцент, доцент кафедры математических проблем управления и информатики, Гомельский государственный университет имени Франциска Скорины, г. Гомель, Беларусь; *адрес:* 246000, Республика Беларусь, г. Гомель, ул. Советская, д. 104; *e-mail:* dolinsky@gsu.by

M. S. Dolinsky,

Francisk Skorina Gomel State University, Gomel, Belarus

SPEEDING UP RECURSIVE SOLUTIONS USING MEMOIZATION

Abstract

In the article, using the example of solving two problems, the methodology for studying the theme "Speeding up recursive solutions using memoization" is illustrated in preparing schoolchildren for Olympiads in informatics. The study is based on the sequential solution of increasingly complex problems. For each problem the following materials are given: the formulation of the problem, the idea of a solution with a proposal to come up with an implementation on their own, the solution in the Pascal programming language. Distance learning system (<http://dl.gsu.by>) is the effective technical base for teaching. The system allows to offer for a student a formulation of the problem; to submit the solution for review; to get a verdict from the system — a correct or incorrect solution; for incorrect solution, the number of the test on which the solution did not pass is indicated. A student can take a test (input and output data), on which his solution did not pass, figure out what the error is in his program, correct and send the solution again. In addition, for each problem there is a link on it to the topic in the forum at site, where you can ask a question on solving this problem and / or read the answer if the questions have already been asked before.

Keywords: recursion, memoization, programming training, Olympiads in informatics, distance learning tools.

В работе [10] описано решение рекурсивных задач «по определению». В работе [8] описано решение задач рекурсивной генерацией чисел. Решение игровых задач с помощью рекурсии описывается в работе [9]. Работа [7] предлагает материал по решению задач с помощью основанного на рекурсии метода «разделяй и властвуй». В данной статье описывается использование мемоизации для ускорения рекурсивных решений.

Проверка решений осуществляется автоматически на сайте DL.GSU.BY. Вдумчивым читателям предлагается после чтения условия задачи/подзадачи попытаться решить ее самостоятельно. Все задачи на указанном сайте находятся в учебном курсе «Олимпиады по информатике».

2. Понятие мемоизации

Мемоизация (англ. memoization от memo — память и optimization — оптимизация) — в программировании сохранение результатов выполнения функций для предотвращения повторных вычислений. Это один из способов оптимизации, применяемый для увеличения скорости выполнения компьютерных программ. Перед вызовом функции проверяется, вызывалась ли функция ранее; если не вызывалась, то функция вызывается и результат ее выполнения сохраняется; если вызывалась, то используется сохраненный результат.

3. Задача «Клеточки» (Гомельская городская олимпиада для I—IX классов, 2012 год)

Задано клетчатое поле размером $1 \times N$ клеток. Необходимо определить количество способов раскрасить это поле в два цвета так, чтобы никакие из K рядом

находящихся клеток не были закрашены одним цветом (каждая клетка может быть окрашена только в один из двух цветов).

Формат ввода.

Два целых числа — N и K .

Ограничения.

$2 \leq N, K \leq 50$.

Формат вывода.

Одно число — количество способов раскраски.

Пример ввода	Пример вывода
5 3	16

Идея решения задачи.

Идея решения — рекурсия с мемоизацией.

Запоминать (мемоизировать) будем количество способов раскрасить поле из $Current$ клеточек, заполнив одним цветом $Fill$ клеточек (в массиве $F[1..50, 0..50]$).

Если элемент массива уже вычислен, рекурсия не вызывается, а ответ берется из массива F . Иначе рекурсия вызывается, а вычисленный рекурсией ответ сохраняется в массиве F .

Рекурсивные вычисления таковы:

$$Rec(Current, Fill) = \begin{cases} 0, & \text{если } Fill \geq K \\ 1, & \text{если } Current > N \\ Rec(Current + 1, Fill + 1) + \\ + Rec(Current + 1, 1) \end{cases}$$

То есть значение вычисляется как количество способов, если мы раскрасим тем же цветом, плюс количество способов, если начнем красить другим цветом.

Полный текст решения приведен в листинге 1.

```
var
  F      : array [1..50,0..50] of int64;
  n,k,i,j: longint;

function Rec(Current,Fill: longint): int64;
var
  Res: int64;
begin
  if Fill>=k      then begin Res:=0; exit; end;
  if Current>N   then begin Res:=1; exit; end;
  if F[Current,Fill]<>0 then begin Res:=f[Current,Fill]; exit; end;
  Res:=Rec(Current+1,Fill+1) + Rec(Current+1,1);
  F[Current,Fill]:=Res;
  Rec      :=Res;
end;

begin
  readln(n,k);
  for i:=1 to n do
    for j:=0 to k do f[i,j]:=0;
  writeln(Rec(1,0));
end.
```

4. Задача «Трапеции» (Хорватская национальная олимпиада, 2017 год)

Мы можем создать гексагональный пазл размера n , поделив правильный шестиугольник на равные треугольники, проведя на одинаковых расстояниях $2n - 1$ параллельных линий между каждой из трех пар противоположных сторон шестиугольника. Некоторые из треугольников этого пазла заштрихованы и должны быть накрыты кусками пазла. Каждый кусок — это трапеция, которая состоит из трех равносторонних треугольников, приставленных сторона к стороне. Эти куски бывают шести различных цветов, обозначенных числами от 1 до 6. В нашем распоряжении есть неограниченное количество кусков каждого цвета.

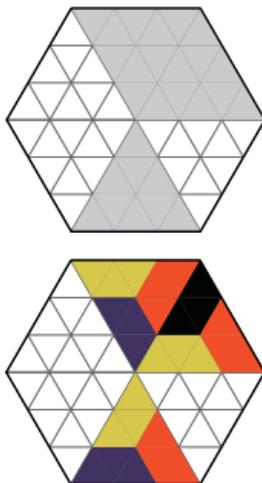
Цель пазла — положить куски на шестиугольник так, чтобы выполнялось следующее:

- 1) каждый кусок покрывает ровно три заштрихованных треугольника;
- 2) каждый треугольник накрывается ровно одним куском;
- 3) два куска одного цвета не касаются по стороне (но могут касаться по углу).

Определите, существует ли решение, если существует, выведите одно.

Время на прохождение каждого теста — одна секунда.

Ниже для примера приведены пазл размера $n = 3$ и решение задачи:



Формат ввода.

Первая строка ввода содержит положительное целое число n — размер пазла.

Следующие $2n$ строк описывают строки пазла сверху вниз. Каждая из этих строк содержит строку символов, которая описывает строку пазла слева направо. Цифра 0 означает заштрихованный треугольник, а символ точки «.» — незаштрихованный треугольник. Вы можете полагать, что хотя бы один треугольник заштрихован.

Формат вывода.

Если пазл невозможно решить, выведите в первой строке слово «nemoguće» («невозможно» по-хорватски). Заштрихованные треугольники должны быть обозначены цифрами от 1 до 6 вместо цифры 0. Эти цифры обозначают цвет соответствующего треугольника.

<i>Пример ввода</i>	3 .000000 ...0000000000000..... ...000... .00000.	1 .0. 0.0	2 0000. 0000000 ..00.0. .0000
<i>Пример вывода</i>	.111224 ...3324423111221..... ...112... .33322.	nemoguće	1222. 1133111 ..31.2. .1122

Решение.

Вернемся к рисунку из условия задачи.

Здесь $n = 3$, т. е. сторона шестиугольника равна 3, и, как видно из рисунка, шестиугольник составлен из шести рядов треугольников, в каждом из которых находится соответственно 7, 9, 11, 11, 9 и 7 треугольников.

Для случаев $n = 1$ и $n = 2$ соответствующий рисунок предлагается самостоятельно нарисовать заинтересованному читателю.

Дополним таблицу из формулировки задачи:

Размер стороны, n	3	1	2
Количество треугольников в рядах (сверху вниз)	7 9 11 11 9 7	3 3	5 7 7 5
<i>Пример ввода</i>	3 .000000 ...0000000000000..... ...000... .00000.	1 .0. 0.0	2 0000. 0000000 ..00.0. .0000
<i>Пример вывода</i>	.111224 ...3324423111221..... ...112... .33322.	nemoguće	1222. 1133111 ..31.2. .1122

Простейший подход к решению рассматриваемой задачи — рекурсивный перебор всех возможных представлений трапеций (листинг 2).

```
begin
  InputData;
  Row:=1;
  Found0 (Row, Column);
  DFS (Row, Column);
  writeln('nemoguće');
  close(input); close(output);
end.
```

Листинг 2

Функция *Found0* (листинг 3) ищет первую ячейку с символом «0», начиная со строки *Row*, позиции 1.

Если ячеек с символом «0» больше нет, значит, все поле заполнили, выводим построенный ответ и останавливаемся. Если все варианты перебрали и ответ не нашли — выводим слово «nemoguće».

```

function Found0(var Row,Column: longint):
boolean;
var
  i,j: longint;
begin
  for i:=Row to 2*N do
  for j:=1 to L[i] do
    if s[i,j]='0'
    then begin
      Row:=i;
      Column:=j;
      Found0:=true;
      exit;
    end;
  Found0:=false;
end;

```

Листинг 3

Процедура *DFS* (листинг 4) пытается приложить все варианты расположения трапеций.

На вводе в каждую строку добавляем по два пробела, чтобы не проверять отдельно границы и не менять четность столбца, от которой много зависит в этой задаче. А именно существуют описанные далее варианты прикладывания трапеции из трех треугольников в зависимости от номера строки и четности столбца (i — номер строки, j — номер столбца (номер элемента в строке)).

В трапецию включаются текущая позиция (i, j), а также еще две позиции — ($i1, j1$), ($i2, j2$), которые определяются смещением относительно (i, j) (+/-1 — вниз/вверх по строкам или влево/вправо по столбцам).

Введем обычные обозначения: R — вправо, L — влево, D — вниз, U — вверх.

Тогда вне зависимости от строки и столбца можно брать вариант RR: взяли два треугольника вправо, первый из них характеризуется смещением (0, 1), второй — (0, 2).

В зависимости от строк и столбцов имеем такую таблицу:

	Нечетный столбец (<i>Odd(j)</i>)	Четный столбец (not <i>Odd(j)</i>)
$i < n$	R&D : (0, 1) (1, 1) DL : (1, 1) (1, 0) DR : (1, 1) (1, 2)	RD : (0, 1) (1, 2)
$i = n$	R&D : (0, 1) (1, 0) DL : (1, 0) (1, -1) DR : (1, 0) (1, 1)	RD : (0, 1) (1, 1)
$i = n + 1$	RD : (0, 1) (1, 0)	R&D : (0, 1) (1, -1) DL : (1, -1) (1, -2) DR : (1, -1) (1, 0)
$i > n + 1$	RD : (0, 1) (1, 0)	R&D : (0, 1) (1, -1) DL : (1, -1) (1, -2) DR : (1, -1) (1, 0)

Процедура *Check*($i0, j0, i1, j1, i2, j2$) (листинг 5):

- проверяет, стоят ли символы «0» на позициях ($i1, j1$), ($i2, j2$) (на позиции ($i0, j0$) ноль стоит по построению);
- перебирает цвет C от 1 до 6 (до первого подходящего).

Функция *Bad* определяет, можно ли цветом C окрасить позиции ($i1, j1$) и ($i2, j2$) так, чтобы не получилось

```

procedure DFS(i,j: longint);
begin
  {1RR} Check(i,j,i,j+1,i,j+2);

  if (i<N) and      Odd(j) then begin
  {1R&D}              Check(i,j,i,  j+1,i+1,j+1);
  {1DR}              Check(i,j,i+1,j+1,i+1,j+2);
  {1DL}              Check(i,j,i+1,j+1,i+1,j );
  end;
  {1RD} if (i<N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+2);

  if (i=N) and      Odd(j) then begin
  {1R&D}              Check(i,j,i,  j+1,i+1,j );
  {1DL}              Check(i,j,i+1,j,  i+1,j-1);
  {1DR}              Check(i,j,i+1,j,  i+1,j+1);
  end;
  {1RD} if (i=N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+1);

  {2RD} if (i> N) and      Odd(j) then Check(i,j,i,j+1,i+1,j);
  if (i> N) and (not Odd(j)) then begin
  {2R&D}              Check(i,j,i,  j+1,i+1,j-1);
  {2DR}              Check(i,j,i+1,j-1,i+1,j);
  {2DL}              Check(i,j,i+1,j-1,i+1,j-2);
  end;
end;

```

Листинг 4

```

procedure Check(i0,j0,i1,j1,i2,j2: longint);
var
  Row,c: longint;
begin
  if (s[i1,j1]<>'0') or (s[i2,j2]<>'0') then exit;
  for c:=1 to 6 do
    begin
      if Bad(C,i0,j0,i1,j1,i2,j2) then continue;
      s[i0,j0]:=d[C]; s[i1,j1]:=d[C]; s[i2,j2]:=d[C];
      Row:=i0;
      if Found0(Row,Column)
        then DFS(Row,Column)
        else PrintHalt;
      s[i0,j0]:='0'; s[i1,j1]:='0'; s[i2,j2]:='0';
      break;
    end;
  end;
end;

```

Листинг 5

соприкосновения по стороне с ранее поставленной трапецией такого же цвета.

Если подходящий цвет найден, то рисуем им трапецию $(i0, j0)$, $(i1, j1)$, $(i2, j2)$ и, если еще есть ячейки с «0», запускаем DFS от нее, иначе выводим ответ и останавливаемся.

После выхода из рекурсии возвращаем символы «0» на позициях $(i0, j0)$, $(i1, j1)$, $(i2, j2)$ и выходим из цикла по цвету.

Функция *Bad* (листинг 6) проверяет с помощью *Bad3* каждую из ячеек $(i0, j0)$, $(i1, j1)$, $(i2, j2)$ по отдельности — можно ли ее покрасить цветом *C*.

Функция *Bad3* (листинг 7) рассматривает соседние треугольники по таким правилам:

- левый и правый треугольники — со смещениями $(0, -1)$ и $(0, 1)$ от текущей позиции;

- верхний и нижний треугольники по следующим правилам в зависимости от строки и четности столбца:

	Нечетный столбец (<i>Odd(j)</i>)	Четный столбец (not <i>Odd(j)</i>)
$i < n$	D: (1, 1)	U: (-1, -1)
$i = n$	D: (1, 0)	U: (-1, -1)
$i = n + 1$	U: (-1, 0)	D: (1, -1)
$i > n + 1$	U: (-1, +1)	D: (1, -1)

Полный текст решения приведен в листинге 8.

```

function Bad(C,i0,j0,i1,j1,i2,j2: longint): boolean;
begin
  Bad:=True;
  if Bad3(C,i0,j0) then exit;
  if Bad3(C,i1,j1) then exit;
  if Bad3(C,i2,j2) then exit;
  Bad:= false;
end;

```

Листинг 6

```

function Bad3(C,i,j: longint): boolean;
begin
  Bad3:=(s[i,j-1]=d[C]) or (s[i,j+1]=d[c]) or
    ((i<N) and Odd(j) and ((s[i+1,j+1]=d[c])) or
    (i<N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N) and Odd(j) and ((s[i+1,j ]=d[c])) or
    (i=N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N+1) and Odd(j) and ((s[i-1,j ]=d[c])) or
    (i=N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])) or
    (i>N+1) and Odd(j) and ((s[i-1,j+1]=d[c])) or
    (i>N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])));
end;

```

Листинг 7

```
const
  MaxN=5;
  d : string = '123456';
var
  s: array [0..2*MaxN+1] of string;
  L: array [1..2*MaxN+1] of longint;
  N,Row,Column,Color : longint;

procedure InputData;
var
  i,j: longint;
  t : string;
begin
  readln(N);
  for i:=1 to 2*N do
    begin
      readln(t);
      s[i]:= ' '+t+' ';
    end;
  s[0]:= ' ';
  for i:=1 to N+2 do s[0]:=s[0]+' ';
  s[2*N+1]:= ' ';
  for i:=1 to N+2 do s[2*N+1]:=s[2*N+1]+' ';
  for i:=1 to 2*N do L[i]:=length(s[i]);
end;

function Found0(var Row, Column: longint): boolean;
var
  i,j: longint;
begin
  for i:=Row to 2*N do
    for j:=1 to L[i] do
      if s[i,j]='0'
        then begin
          Row:=i;
          Column:=j;
          Found0:=true;
          exit;
        end;
  Found0:=false;
end;

procedure PrintHalt;
var
  k: longint;
begin
  for k:=1 to 2*N do
    begin
      delete(s[k],1,2);
      writeln(s[k]);
    end;
  close(input); close(output);
  halt;
end;

procedure DFS(i,j: longint); forward;
```

```

function Bad3(C,i,j: longint): boolean;
begin
  Bad3:=(s[i,j-1]=d[C]) or (s[i,j+1]=d[c]) or
    ((i<N) and Odd(j) and ((s[i+1,j+1]=d[c])) or
    (i<N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N) and Odd(j) and ((s[i+1,j ]=d[c])) or
    (i=N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N+1) and Odd(j) and ((s[i-1,j ]=d[c])) or
    (i=N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])) or
    (i>N+1) and Odd(j) and ((s[i-1,j+1]=d[c])) or
    (i>N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])));
end;

function Bad(C,i0,j0,i1,j1,i2,j2: longint): boolean;
begin
  Bad:=True;
  if Bad3(C,i0,j0) then exit;
  if Bad3(C,i1,j1) then exit;
  if Bad3(C,i2,j2) then exit;
  Bad:= false;
end;

procedure Check(i0,j0,i1,j1,i2,j2: longint);
var
  Row,c: longint;
begin
  if (s[i1,j1]<>'0') or (s[i2,j2]<>'0') then exit;
  for c:=1 to 6 do
    begin
      if Bad(C,i0,j0,i1,j1,i2,j2) then continue;
      s[i0,j0]:=d[C]; s[i1,j1]:=d[C]; s[i2,j2]:=d[C];
      Row:=i0;
      if Found0(Row,Column)
        then DFS(Row,Column)
        else PrintHalt;
      s[i0,j0]:='0'; s[i1,j1]:='0'; s[i2,j2]:='0';
      break;
    end;
  end;

procedure DFS(i,j: longint);
begin
  {1RR} Check(i,j,i,j+1,i,j+2);
  if (i<N) and Odd(j) then begin
  {1R&D} Check(i,j,i, j+1,i+1,j+1);
  {1DR} Check(i,j,i+1,j+1,i+1,j+2);
  {1DL} Check(i,j,i+1,j+1,i+1,j );
  end;
  {1RD} if (i<N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+2);
  if (i=N) and Odd(j) then begin
  {1R&D} Check(i,j,i, j+1,i+1,j );
  {1DL} Check(i,j,i+1,j ,i+1,j-1);
  {1DR} Check(i,j,i+1,j ,i+1,j+1);
  end;
  {1RD} if (i=N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+1);
  {2RD} if (i> N) and Odd(j) then Check(i,j,i,j+1,i+1,j);
  if (i> N) and (not Odd(j)) then begin
  {2R&D} Check(i,j,i, j+1,i+1,j-1);
  {2DR} Check(i,j,i+1,j-1,i+1,j);
  {2DL} Check(i,j,i+1,j-1,i+1,j-2);
  end;
end;
end;

```

```

begin
  InputData;
  Row:=1;
  Found0(Row,Column);
  DFS(Row,Column);
  writeln('nemoguće');
  close(input); close(output);
end.

```

Листинг 8 (окончание)

Такое решение проходит все предлагаемые (организаторами олимпиады) для данной задачи тесты, кроме двух:

<i>Ввод</i>	<pre> 5 0000000000 00.0000000000 0000.0000000000 000000.0000000000 00000000.0000000000 000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0 </pre>	<pre> 5 0000000000 00.0000000000 000.0000000000 000000.0000000000 00000000.0000000000 000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0000000000 0000000000.0 </pre>
<i>Вывод</i>	<pre> 12221112221 11.3332223311 1222.1114443222 211333.2221113331 12233111.3332224411 114431222.111333422 224311333.2221112 233222111.33322 111333222.112 222111333.1 </pre>	<pre> 11222111222 21.1112221112 122..3331113322 211333..222443111 12233111..111422233 114431222.222111443 224311333.3332224 233222111.11133 111333222.223 222111333.2 </pre>

Причем первый тест проходит за 5 секунд (а по ограничениям задачи надо уложиться в 1 секунду), а вот второму тесту не хватает и 30 секунд.

Идея решения с мемоизацией.

Для ускорения используем мемоизацию по профилю.

Заведем массив $Was[Row, P]$, все элементы которого изначально равны *false*, и по мере рекурсивного обхода будем заносить туда *true* после того, как убедимся рекурсивным перебором, что для строки Row из профиля P невозможно заполнить шестиугольник трапециями.

Пусть мы нашли очередной ноль в позиции (i, j) . Зафиксируем с помощью функции $Profile(i, j)$ состояние заполнения шестиугольника трапециями. Тогда при входе в рекурсию пишем проверку:

```
if Was[i,Profile(i,j)] then exit;
```

То есть, если мы уже пытались заполнять с такого профиля и не получилось, не будем повторять эту работу.

А перед выходом из рекурсии будем помечать тот факт, что заполнить не получилось:

```
Was[i,Profile(i,j)]:=true;
```

То есть рекурсивный перебор станет выглядеть так, как представлено в листинге 9.

Представленная в листинге 10 функция вычисления профиля $Profile(i, j)$ анализирует как профиль:

- текущую строку Row с позиции j — это будут младшие биты профиля;
- следующую строку $Row+1$ (если она есть) с позиции 1 до позиции j включительно — это будут старшие биты профиля;

Применение данного подхода ускоряет решение, все тесты проходятся за 0,1 секунды.

Однако в двух тестах, которые представлены в таблице ниже, получается неверный ответ «nemoguće»,

```

procedure DFS(i,j: longint);
begin
  if Was[i,Profile(i,j)] then exit;

{1RR} Check(i,j,i,j+1,i,j+2);

  ...

  if (i> N) and (not Odd(j)) then begin
{2R&D}          Check(i,j,i, j+1,i+1,j-1);
{2DL}          Check(i,j,i+1,j-1,i+1,j-2);
{2DR}          Check(i,j,i+1,j-1,i+1,j);
                end;

  Was[i,Profile(i,j)]:=true;
end;

```

Листинг 9

```

function Profile(i,j: longint): longint;
var
  t,p: longint;
begin
  p:=0;
  for t:=j to L[i] do
    if (s[i,t]>='1') and (s[i,t]<='6')
      then p:=p or (1 shl (4*N+2-(t-3)));
  if i<2*N
    then begin
      for t:=3 to j do
        if (s[i+1,t]>='1') and (s[i+1,t]<='6')
          then p:=p or (1 shl (4*N+1-(t-3)));
      end;
  Profile := p;
end;

```

Листинг 10

хотя на самом деле решение существует — см. вторую строку таблицы:

<i>Ввод</i>	5 0000000000 000000000000 00000000000000 0000000000000000 00.0.0.0.0.0.0.0.0.0. 000000000000000000 000000000000000000 0000000000000000 0000000000000000 000000000000	5 0000000000 000000000000 00000000000000 0000000000000000 00.....0.0.0.0.0.0. 000000000000000000 00.00000000000000 0000000000000000 0000000000000000 000000000000
<i>Вывод</i>	11122211122 2223331222123 111444113331133 12233112211223311 11.2.3.1.2.1.2.3.1. 2221112221112221122 333222333222333142 111333111444344 2221112211122 33344423332	11122211122 2223331222123 111444113331133 22333112211223311 12.....1.2.1.2.3.1. 1122211122211122211 22.22233322233321 211144411144422 2221112233311 33344424441

Можно предположить, что имеется ошибка в вычислении профиля с помощью функции *Profile(i, j)*.

На самом деле, для корректной мемоизации нужно запоминать профиль для позиции (i, j) , а не для строки i . Такое решение берет все тесты, но за время 1,3 секунды на половине тестов. В задаче же установлен лимит в 1 секунду.

Переход от строк к числам не принес принципиального изменения по времени выполнения.

Анализ приводит к пониманию того, что для ускорения надо чуть подправить функцию вычисления профиля для строки: вместо формирования профиля по заполненным позициям ($\geq '1'$ и $\leq '6'$) формировать профиль по незаполненным позициям текущей и следующей строк ($s[i, t]='0'$, $s[i+1, t]='0'$) (листинг 11).

Получаем *полное решение*, которое проходит все тесты за 0,1 секунды (листинг 12).

5. Заключение

В данной статье на примере решения двух задач рассмотрена методика изучения темы «Ускорение рекурсивных решений при помощи мемоизации», предлагающая, по мнению автора, наиболее простой способ постепенного осознания учащимися механизма рекурсии и способ решения задач с ее помощью. Методика включает

```

function Profile(i,j: longint): longint;
var
  t,p: longint;
begin
  p:=0;
  if i<2*N
    then begin
      for t:=3 to j do
        if s[i+1,t]='0'
          then p:=p or (1 shl (4*N+2-(t-3)));
      end;
  for t:=j to L[i] do
    if s[i,t]='0'
      then p:=p or (1 shl (4*N+1-(t-3)));
  Profile := p;
end;

```

Листинг 11

```

const
  MaxN=5;
  d: string = '123456';
  MaxW=8*1024*1024;
var
  s: array [0..2*MaxN+1] of string;
  L: array [1..2*MaxN+1] of longint;
  N,Row,Column: longint;
  Was: array [1..2*MaxN,0..MaxW] of boolean;

procedure InputData;
var
  i,j: longint;
  t : string;
begin
  readln(N);
  for i:=1 to 2*N do
    begin
      readln(t);
      s[i]:=' '+t+' ';
    end;
  s[0]:=' ';
  for i:=1 to N+2 do s[0]:=s[0]+' ';
  s[2*N+1]:=' ';
  for i:=1 to N+2 do s[2*N+1]:=s[2*N+1]+' ';
  for i:=1 to 2*N do L[i]:=length(s[i]);
  for i:=1 to 2*N do
    for j:=0 to MaxW do Was[i,j]:=false;
  end;

function Found0(var Row, Column: longint): boolean;
var
  i,j: longint;
begin
  for i:=Row to 2*N do
    for j:=1 to L[i] do
      if s[i,j]='0'
        then begin
          Row:=i; Column:=j; Found0:=true; exit;
        end;
  Found0:=false;
end;

procedure PrintHalt;
var
  k: longint;
begin
  for k:=1 to 2*N do
    begin
      delete(s[k],1,2); writeln(s[k]);
    end;
  close(input); close(output); halt;
end;

procedure DFS(i,j: longint); forward;

```

```

function Bad3(C,i,j: longint): boolean;
begin
  Bad3:=(s[i,j-1]=d[C]) or (s[i,j+1]=d[c]) or
    ((i<N) and Odd(j) and ((s[i+1,j+1]=d[c])) or
    (i<N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N) and Odd(j) and ((s[i+1,j ]=d[c])) or
    (i=N) and not Odd(j) and ((s[i-1,j-1]=d[c])) or
    (i=N+1) and Odd(j) and ((s[i-1,j ]=d[c])) or
    (i=N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])) or
    (i>N+1) and Odd(j) and ((s[i-1,j+1]=d[c])) or
    (i>N+1) and not Odd(j) and ((s[i+1,j-1]=d[c])));
end;

function Bad(C,i0,j0,i1,j1,i2,j2: longint): boolean;
begin
  Bad:=True;
  if Bad3(C,i0,j0) then exit;
  if Bad3(C,i1,j1) then exit;
  if Bad3(C,i2,j2) then exit;
  Bad:= false;
end;

procedure Check(i0,j0,i1,j1,i2,j2: longint);
var
  Row,c: longint;
begin
  if (s[i1,j1]<>'0') or (s[i2,j2]<>'0') then exit;
  for c:=1 to 6 do
    begin
      if Bad(C,i0,j0,i1,j1,i2,j2) then continue;
      s[i0,j0]:=d[C]; s[i1,j1]:=d[C]; s[i2,j2]:=d[C];
      Row:=i0;
      if Found0(Row,Column)
        then DFS(Row,Column)
        else PrintHalt;
      s[i0,j0]:='0'; s[i1,j1]:='0'; s[i2,j2]:='0';
      break;
    end;
  end;

function Profile(i,j: longint): longint;
var
  t,p: longint;
begin
  p:=0;
  if i<2*N
    then begin
      for t:=3 to j do
        if s[i+1,t]='0'
          then p:=p or (1 shl (4*N+2-(t-3)));
      end;
    for t:=j to L[i] do
      if s[i,t]='0'
        then p:=p or (1 shl (4*N+1-(t-3)));
    Profile := p;
  end;

```

```

procedure DFS(i,j: longint);
begin
  if Was[i,Profile(i,j)] then exit;
{1RR} Check(i,j,i,j+1,i,j+2);
  if (i<N) and      Odd(j)  then begin
{1R&D}                      Check(i,j,i,  j+1,i+1,j+1);
{1DL}                      Check(i,j,i+1,j+1,i+1,j  );
{1DR}                      Check(i,j,i+1,j+1,i+1,j+2);
                          end;
{1RD} if (i<N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+2);
  if (i=N) and      Odd(j)  then begin
{1R&D}                      Check(i,j,i,  j+1,i+1,j  );
{1DL}                      Check(i,j,i+1,j  ,i+1,j-1);
{1DR}                      Check(i,j,i+1,j  ,i+1,j+1);
                          end;
{1RD} if (i=N) and (not Odd(j)) then Check(i,j,i,j+1,i+1,j+1);

{2RD} if (i> N) and      Odd(j)  then Check(i,j,i,j+1,i+1,j);
  if (i> N) and (not Odd(j)) then begin
{2R&D}                      Check(i,j,i,  j+1,i+1,j-1);
{2DL}                      Check(i,j,i+1,j-1,i+1,j-2);
{2DR}                      Check(i,j,i+1,j-1,i+1,j);
                          end;

  Was[i,Profile(i,j)]:=true;
end;

begin
  InputData;
  Row:=1;
  Found0(Row,Column);
  DFS(Row,Column);
  writeln('nemoguće');
  close(input);  close(output);
end.

```

Листинг 12 (окончание)

в себя последовательность задач в порядке возрастания сложности, снабженных, где необходимо, предварительными общими пояснениями и последующими полными решениями предлагаемых задач. В статье предложены две таких задачи на применение мемоизации. Серьезной технической основой является разработанная под управлением автора инструментальная система дистанционного обучения (<http://dl.gsu.by>), которая позволяет максимально автоматизировать процесс предъявления условий задач и проверки их решений.

Список источников

1. Баррон Д. Рекурсивные методы в программировании. М.: Мир, 1974. 79 с.
2. Бердж В. Методы рекурсивного программирования. М.: Машиностроение, 1983. 256 с.
3. Головешкин В. А., Ульянов В. В. Теория рекурсии для программистов. М.: Физматлит, 2006. 296 с.
4. Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы. М.: МЦНМО, 2019. 320 с.
5. Долинский М. С. Введение в решение задач с помощью рекурсивных процедур и функций // Информатика в школе. 2016. № 9. С. 49–56. EDN: XEQEZR.
6. Долинский М. С. Генерация комбинаторных объектов с помощью рекурсивных процедур и функций // Информатика в школе. 2019. № 4. С. 59–63. DOI: 10.32517/2221-1993-2019-18-4-59-63. EDN: TCJKQY.
7. Долинский М. С. Рекурсивное решение задач с помощью метода «разделяй и властвуй» // Информатика в школе. 2022. № 2. С. 58–64. DOI: 10.32517/2221-1993-2022-21-2-58-64. EDN: JOEVGG.
8. Долинский М. С. Решение задач рекурсивной генерацией чисел // Информатика в школе. 2021. № 1. С. 46–51. DOI: 10.32517/2221-1993-2021-20-1-46-51. EDN: OIHIZJ.
9. Долинский М. С. Решение игровых задач с помощью рекурсии // Информатика в школе. 2021. № 9. С. 43–50. DOI: 10.32517/2221-1993-2021-20-9-43-50. EDN: XQIAMP.
10. Долинский М. С. Решение рекурсивных задач по определению // Информатика в школе. 2020. № 2. С. 60–66. DOI: 10.32517/2221-1993-2020-19-2-60-66. EDN: EAYQZS.
11. Златопольский Д. М. 1400 задач по программированию. М.: ДМК Пресс, 2019. 192 с.
12. Луридаш П. Алгоритмы для начинающих. Теория и практика для разработчика. М.: ЭКСМО, 2018. 608 с.
13. Паронджанов В. Д. Дружелюбные алгоритмы, понятные каждому. М.: ДМК Пресс, 2014. 464 с.
14. Потопахин В. В. Искусство алгоритмизации. М.: ДМК Пресс, 2013. 320 с.

15. Рафгарден Т. Совершенный алгоритм. Графовые алгоритмы и структуры данных. СПб.: Питер, 2020. 256 с.
16. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. СПб.: Питер, 2020. 256 с.
17. Рафгарден Т. Совершенный алгоритм. Основы. СПб.: Питер, 2019. 256 с.
18. Рубио-Санчес М. Введение в рекурсивное программирование. М.: ДМК Пресс, 2019. 436 с.
19. Скиена С. Алгоритмы. Руководство по разработке. СПб.: ВНУ, 2011. 720 с.
20. Солтис М. Введение в анализ алгоритмов. М.: ДМК Пресс, 2019. 278 с.
21. Стивенс Р. Алгоритмы. Теория и практическое применение. М.: ЭКСМО, 2016. 544 с.
22. Шень А. Программирование: теоремы и задачи. М.: МЦНМО, 2017. 320 с.
23. Castro R., Lehmann N., Pérez J., Subercaseaux B. Wavelet trees for competitive programming // Olympiad in Informatics. 2016. Vol. 10. P. 19–37.
24. Do P. T., Pham B. T., Than V. C. Latest algorithms on particular graph classes // Olympiad in Informatics. 2020. Vol. 14. P. 21–35.
25. Erdősné Németh A. Teaching graphs for contestants in lower-secondary-school-age // Olympiad in Informatics. 2017. Vol. 11. P. 41–53.
26. Erdősné Németh A., Zsakó L. The place of the dynamic programming concept in the progression of contestants' thinking // Olympiad in Informatics. 2016. Vol. 10. P. 61–72.
27. Forisek M. Towards a better way to teach dynamic programming // Olympiad in Informatics. 2015. Vol. 9. P. 45–55.
28. Manev K. Tasks on graphs // Olympiad in Informatics. 2008. Vol. 2. P. 90–104.
29. Manev K., Nikolov N., Markov M. Reconstruction of trees using metric properties // Olympiad in Informatics. 2011. Vol. 5. P. 82–91.
30. Pachocki J., Radoszewskij J. Where to use and how not to use polynomial string hashing // Olympiad in Informatics. 2013. Vol. 7. P. 90–100.



DOI: 10.32517/2221-1993-2022-21-6-68-80

В. К. Маркелов, О. А. Завьялова

Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия

ИНСТРУМЕНТАРИЙ ОБЛАЧНОЙ СРЕДЫ GOOGLE COLABORATORY ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ И ПОДГОТОВКИ К ЕГЭ ПО ИНФОРМАТИКЕ

Аннотация

Одной из облачных технологий, которую можно использовать для обучения программированию на базе языка программирования Python, является облачный сервис Google Colaboratory. Данный сервис может использоваться учителями как при изучении отдельных тем раздела «Алгоритмы и программирование» в рамках школьного курса информатики, так и при подготовке обучающихся к заданиям по программированию ЕГЭ по информатике.

В статье рассматриваются возможности облачного сервиса Google Colaboratory для создания интерактивных блокнотов на примере интерактивного блокнота и практической работы по теме «Цикл с предусловием», а также интерактивных блокнотов «Задание № 24 ЕГЭ по информатике» для обучающихся и для учителя. Обосновывается применимость данных блокнотов для обучения программированию на базе языка Python в рамках школьного курса информатики и для подготовки обучающихся к заданиям по программированию ЕГЭ по информатике.

Ключевые слова: программирование, интерактивный блокнот, методика обучения программированию, Google Colaboratory, Python.

1. Введение

Методика обучения программированию в рамках школьного курса информатики нуждается в постоянном обновлении и совершенствовании используемых методов, технологий и средств. В качестве одного из таких средств могут применяться различные облачные технологии. При этом в рамках образовательного процесса облачные технологии обычно используются для

предоставления доступа учащимся к учебно-методическим материалам, для файлообмена между обучающимся и учителем, а также между самими обучающимися с целью обмена опытом и совместной работы над учебными проектами.

Под облачными технологиями понимаются системы обработки данных, в которых компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис [9].

Контактная информация

Маркелов Валерий Константинович, студент 2-го курса магистратуры по направлению подготовки 44.04.01 «Педагогическое образование» (профиль «Информационные технологии в профессиональной деятельности педагога»), Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* v. a.l. e.m. a.r.k@yandex.ru

Завьялова Ольга Алексеевна, канд. пед. наук, доцент кафедры математики, информатики и методики обучения, Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* ooolga30@yandex.ru

V. K. Markelov, O. A. Zavyalova

Shuya branch of Ivanovo State University, Shuya, Ivanovo Region, Russia

TOOLS OF THE GOOGLE COLABORATORY CLOUD ENVIRONMENT FOR PROGRAMMING TRAINING AND THE PREPARATION FOR THE UNIFIED STATE EXAM IN INFORMATICS

Abstract

One of the cloud technologies that can be used to teach programming based on the Python programming language is the Google Colaboratory cloud service. This service can be used by teachers both when teaching certain themes of the section "Algorithms and programming" as part of a school informatics course, and when preparing students for the tasks in programming of the Unified State Exam in informatics.

The article discusses the possibilities of the Google Colaboratory cloud service for creating interactive notebooks using the example of an interactive notebook and practical work on the theme "A precondition loop", as well as interactive notebooks "Task No. 24 of the Unified State Exam in informatics" for students and teachers. The applicability of these notebooks for programming training based on the Python language as part of a school informatics course and for preparing students for tasks in programming of the Unified State Exam in informatics is substantiated.

Keywords: programming, interactive notebook, methodology of programming training, Google Colaboratory, Python.

Современный урок информатики допускает параллельную работу обучающихся с различными языками программирования, ставшую возможной благодаря наличию облачных сервисов, позволяющих изучать программирование без установки программного обеспечения на свой компьютер [4]. Одним из основных языков программирования в рамках школьного курса информатики является Python. Python — это язык программирования высокого уровня, простой синтаксис которого позволяет учащимся быстро освоить базовые алгоритмические структуры (следование, ветвление, выбор, повторение) и акцентировать основное внимание на оттачивании навыков программирования, при этом улучшенные отчеты об ошибках способствуют развитию у обучающихся умения находить ошибки в программе и самостоятельно их исправлять [7].

Использование облачных сервисов в образовательном процессе создает условия для естественного развития познавательных и творческих способностей обучающихся, развивает коммуникативные и личностные универсальные учебные навыки, а также способности к планированию и управлению своей и совместной деятельностью [5]. Кроме того, применение данных сервисов позволяет снизить затраты на процесс обучения, повысить качество подготовки учащихся за счет оперативного изменения учебных планов, формирования и предоставления совместного доступа к обучающим материалам [6].

Одним из облачных сервисов, который можно использовать для обучения программированию в школе на базе языка Python, является облачный сервис Google Colaboratory, который был разработан корпорацией Google.

Google Colaboratory, или Google Colab (<https://colab.research.google.com/>) — это бесплатная интерактивная облачная среда, которая используется в профессиональной разработке для анализа данных и машинного обучения на базе языка Python 3.8.16. Облачная среда Google Colaboratory стимулирует улучшение навыков программирования на Python, а также позволяет реализовывать алгоритмы углубленного обучения с использованием популярных библиотек, таких как Keras, TensorFlow, PyTorch и OpenCV [10].

Данный сервис позволяет специалистам в области цифровых технологий [8]:

- совместно анализировать структурированные и неструктурированные данные огромных объемов (Big Data);
- проектировать и тестировать новые модели машинного обучения;
- разрабатывать нейронные сети;
- осуществлять эксперименты в области искусственного интеллекта;
- моделировать и имитировать различные процессы;
- разрабатывать программное обеспечение.

Основу облачной среды Google Colaboratory составляет интерактивный блокнот Jupyter Notebook, в котором присутствует возможность работы как со статическим содержимым (текст, изображения, видео), так и с программным кодом на Python.

Сегодня у большинства обучающихся есть различные электронные устройства, поэтому облачные технологии могут активно использоваться в образовании, а их внедрение имеет определенные преимущества. Так,

применение облачных технологий открывает учителю широкие возможности для организации уровневой дифференциации и индивидуального подхода, а также появляется возможность создания учебного модуля с упражнениями различного уровня сложности [3].

Облачная среда Google Colaboratory может использоваться как учителями информатики, так и обучающимися в рамках изучения языка программирования Python. Привлекательность применения в образовательном процессе данного облачного сервиса состоит в том, что, имея только учетную запись Google и подключение к интернету, учителя могут использовать облачную среду Google Colaboratory для создания различных интерактивных средств обучения к урокам раздела «Программирование» в рамках школьного курса информатики, в частности, для создания обучающих интерактивных блокнотов. *Под интерактивными средствами обучения понимаются программные, аппаратно-программные и технические средства и устройства, которые функционируют на базе микропроцессорной и вычислительной техники и обеспечивают обучение в диалоговом взаимодействии пользователя с компьютером* [2].

Следует отметить, что программирование составляет основу содержательной линии «Алгоритмы и программирование», изучаемой на протяжении всего школьного курса информатики, поэтому задачи по программированию являются одним из тематических блоков Единого государственного экзамена по информатике. Актуальным способом подготовки к решению задач по программированию ЕГЭ является как непосредственное изучение языка программирования, так и изучение основных понятий языка с решением достаточного количества соответствующих задач, покрывающих основные идеи так называемых школьных алгоритмов [1]. Благодаря возможностям облачной среды Google Colaboratory интерактивные блокноты могут также использоваться и при подготовке обучающихся к решению задач по программированию ЕГЭ.

Целью данной статьи является описание инструментария и возможностей бесплатной интерактивной облачной среды Google Colaboratory, а также демонстрация примеров готовых интерактивных блокнотов — как для обучения программированию на базе языка Python в рамках школьного курса информатики, так и для подготовки обучающихся к решению задач по программированию единого государственного экзамена.

2. Инструментарий облачной среды Google Colaboratory для создания и использования обучающих интерактивных блокнотов при обучении программированию в школе

Проанализируем инструментарий облачной среды Google Colaboratory на примере обучающего интерактивного блокнота «Цикл с предусловием»:

https://colab.research.google.com/drive/13QBT_WRZimZFTpitJK3mUQumZ12D0whq

Этот блокнот может применяться в рамках изучения соответствующей темы раздела «Алгоритмы и программирование».

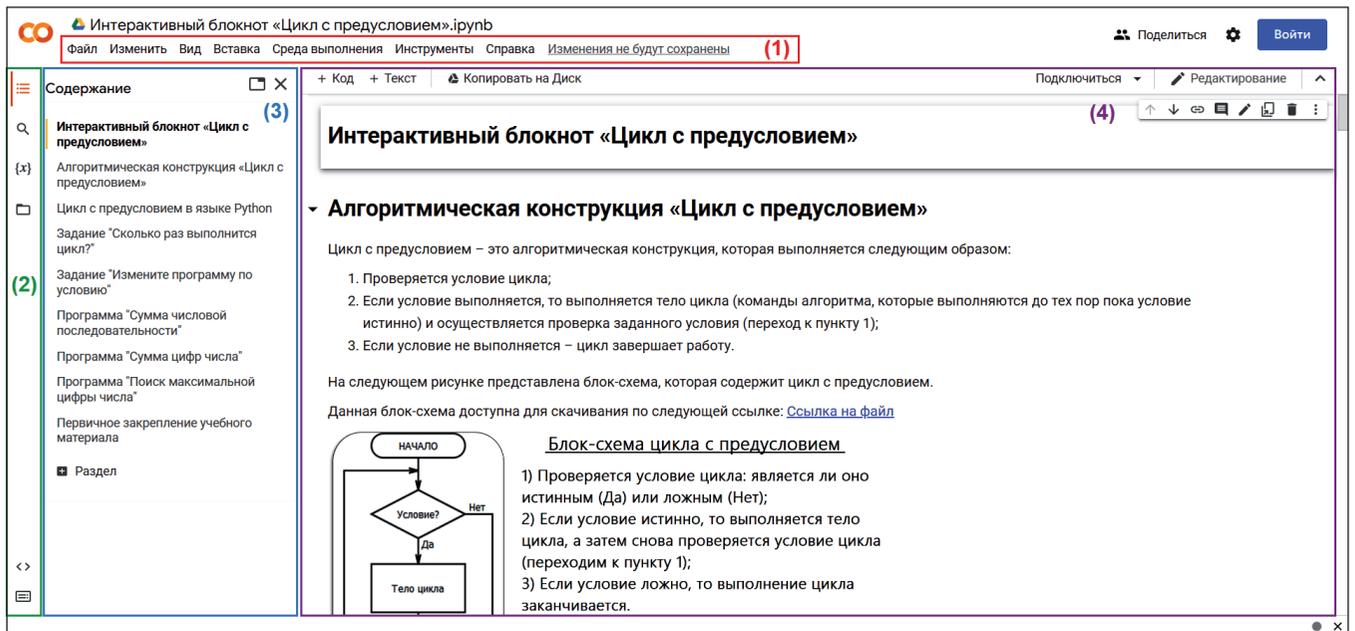


Рис. 1. Основные элементы облачной среды Google Colaboratory на примере интерактивного блокнота «Цикл с предусловием»

Рассмотрим основные элементы облачной среды Google Colaboratory (рис. 1). В верхней части окна расположено меню (1), над которым стандартно располагается название интерактивного блокнота. В левой части окна находится боковая панель (2). При нажатии на самую верхнюю кнопку данной боковой панели открывается колонка *Содержание* (3), которая позволяет перемещаться между разделами и подразделами интерактивного блокнота. Данную колонку можно закрыть при помощи соответствующей кнопки на боковой панели, подсвеченной оранжевым цветом. Оставшаяся часть окна представляет собой рабочую область среды (4) с различным интерактивным содержимым, которое расположено в текстовых ячейках (текст, изображения, видео) и кодовых ячейках (программы или фрагменты кода на языке Python).

Для внесения изменений в интерактивный блокнот необходимо войти в свою учетную запись Google и скопировать блокнот на свой Google Диск при помощи кнопки *Копировать на Диск*. Если не войти в учетную запись, внесенные в блокнот изменения будут удалены после закрытия страницы с интерактивным блокнотом. Об этом также информирует запись «Изменения не будут сохранены», расположенная справа от меню облачной среды.

Интерактивный блокнот «Цикл с предусловием» состоит из восьми разделов:

- «Алгоритмическая конструкция «Цикл с предусловием»»;
- «Цикл с предусловием в языке Python»;
- «Задание «Сколько раз выполнится цикл?»»;

- «Задание «Измените программу по условию»»;
- «Программа «Сумма числовой последовательности»»;
- «Программа «Сумма цифр числа»»;
- «Программа «Поиск максимальной цифры числа»»;
- «Первичное закрепление учебного материала».

Заголовок раздела «Алгоритмическая конструкция «Цикл с предусловием»» в **режиме редактирования** представлен на рисунке 2. Чтобы перейти в данный режим, достаточно дважды нажать на заданную ячейку. Для создания такого заголовка используется текстовая ячейка для заголовка раздела, которую можно создать при помощи команды меню *Вставка, Ячейка для заголовка раздела*. При создании ячейки открывается окно редактирования ячейки, в котором можно превратить раздел в подраздел при помощи кнопки (1), сделать текст заголовка жирным (2) или курсивным (3). Остальные кнопки используются в основном для работы с текстовыми ячейками. Кроме того, каждая ячейка имеет управляющее меню (4), с помощью команд которого можно:

- перемещать ячейки по блокноту (стрелки вверх-вниз);
- получить ссылку на ячейку (кнопка со скрепкой);
- добавить комментарий (кнопка со значком сообщения);
- завершить изменения в ячейке (кнопка с зачеркнутым карандашом);
- удалить ячейку (кнопка с изображением корзины).

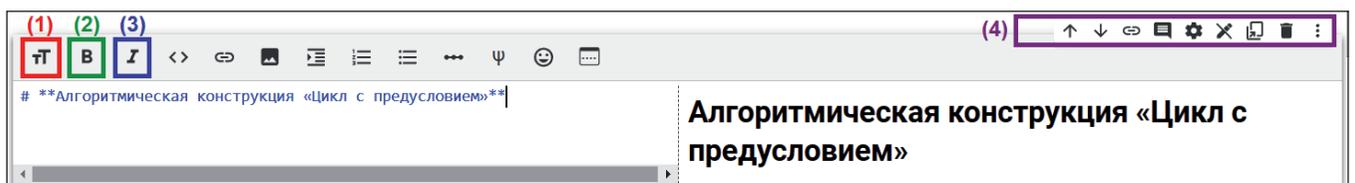


Рис. 2. Редактор ячейки для заголовка раздела «Алгоритмическая конструкция «Цикл с предусловием»»

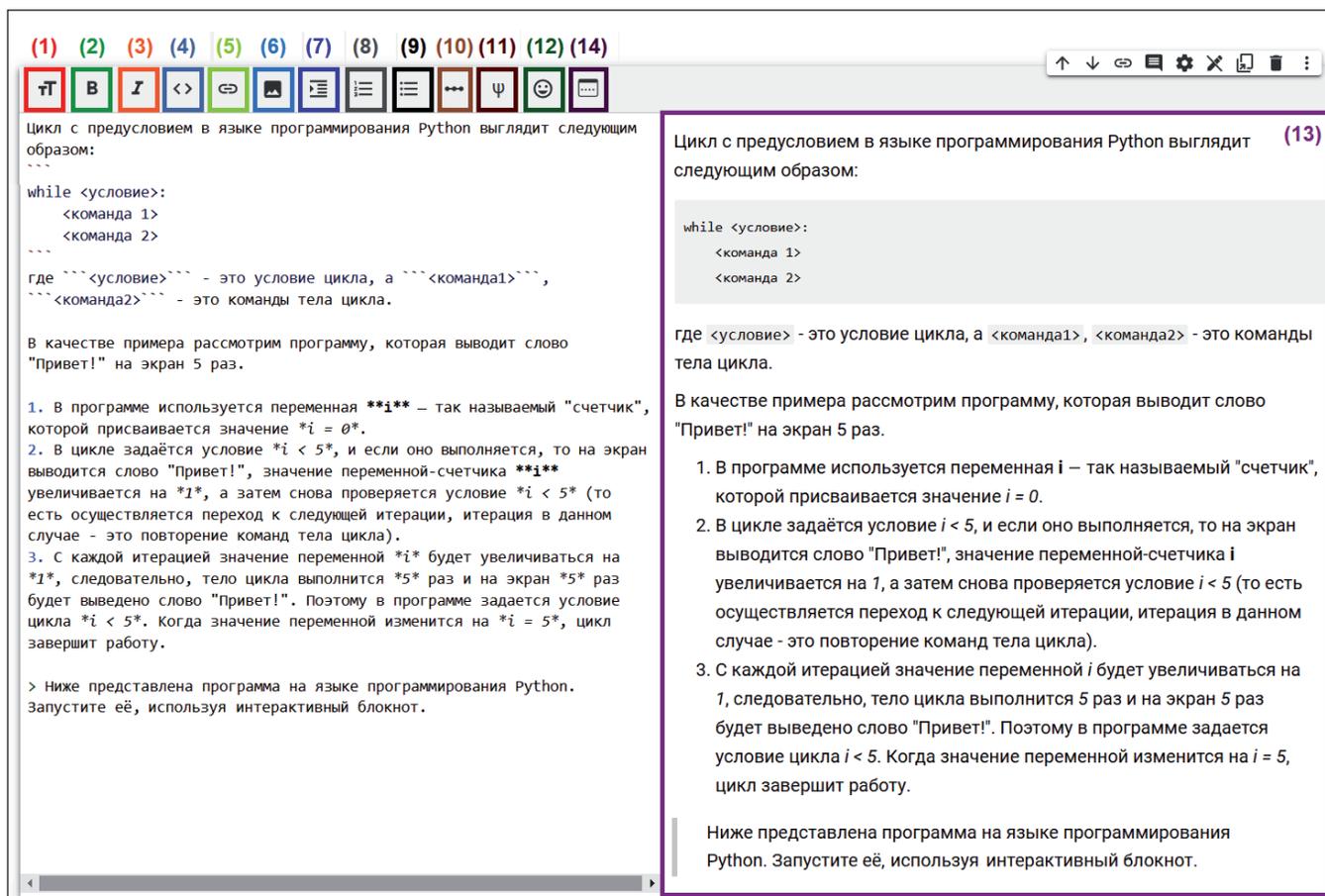


Рис. 3. Редактор текстовой ячейки раздела «Цикл с предусловием в языке Python»

В разделе «Алгоритмическая конструкция “Цикл с предусловием”» рассматривается алгоритм выполнения цикла с предусловием и приводится способ записи данной алгоритмической конструкции с использованием блок-схемы. В свою очередь, в разделе «Цикл с предусловием в языке Python» рассматривается структура цикла с предусловием в языке программирования Python и приводится простейший пример программы с использованием данного цикла, которая пять раз выводит на экран слово «Привет!»

Для отображения информации в облачной среде Google Colaboratory существует два основных типа ячеек: текстовые и кодовые.

В качестве примера рассмотрим **текстовую ячейку** раздела «Цикл с предусловием в языке Python» в режиме редактирования (рис. 3). Для создания такой ячейки можно использовать кнопку *Текст* в верхней части рабочей области среды или команду меню *Вставка, Текстовая ячейка*. При создании ячейки открывается окно редактирования ячейки, в котором, так же как и для ячейки заголовка раздела, можно превратить раздел в подраздел при помощи кнопки (1), сделать текст заголовка жирным (2) или курсивным (3), оформить текст в виде программного кода, используя кнопку (4). Кнопка (5) позволяет вставить в ячейку ссылку, а кнопка (6) — изображение. Кроме того, текст можно выделить при помощи отступа (7), который дополнительно обозначается серой вертикальной линией. В ячейке также можно использовать маркированные (8) и нумерованные (9) списки, горизонтальную линию (10), математические формулы

в формате LaTeX (11) и эмодзи (12). По умолчанию окно редактирования ячейки при помощи специальной разделительной линии отделяется от окна предварительного просмотра (13), в котором можно посмотреть, как будет выглядеть ячейка после внесения изменений. Положение данного окна можно изменить (справа или снизу от окна редактирования) при помощи кнопки (14).

Следует отметить, что использование кнопок для редактирования текста не является обязательным. Для превращения текста в заголовок раздела может использоваться символ «#». Чтобы сделать текст жирным, достаточно заключить его между символами «**», а чтобы сделать текст курсивным — между символами «*». Символ «*», отделенный пробелом, может использоваться для создания маркированного списка, а цифра с точкой, отделенная пробелом, — для создания нумерованного списка. В свою очередь, для создания отступа применяется символ «>», а для создания горизонтальной линии — символы «---». При этом отступы, так же как и горизонтальные линии, должны отделяться от остального текста отдельной строкой.

Рассмотрим возможности добавления ссылок и изображений в облачной среде Google Colaboratory. Для этого в качестве примера рассмотрим текстовую ячейку раздела «Алгоритмическая конструкция “Цикл с предусловием”» (рис. 4).

При нажатии на кнопку добавления ссылки предлагается создать ссылку в формате [Текст ссылки](https://link.com), где `https://link.com` — некоторая ссылка.

Чтобы добавить изображение в текстовую ячейку, необходимо нажать на кнопку добавления изображения, выбрать изображение со своего компьютера и нажать на кнопку *Открыть*. При этом в текстовую ячейку добавляется конструкция формата [Имя файла](Ссылка на изображение). Следует отметить, что при загрузке изображения с компьютера ссылка может быть очень длинной, поскольку изображение загружается на серверы Google. Кроме того, такие изображения загружаются с оригинальными размерами, без возможности их изменения. Однако редактор текстовых ячеек поддерживает html-теги, поэтому для добавления изображений рекомендуется использовать html-тег «img».

Например, тег: ``

позволит создать изображение размером 500 на 300 пикселей (вместо «ссылка на изображение» необходимо указать ссылку, которая появится после загрузки изображения с компьютера, либо использовать ссылку из сети Интернет). Для создания ссылок также можно использовать команду:

`Ссылка`,

где вместо «http://link.com» находится ссылка.

Кроме текстовых ячеек, интерактивные блокноты также могут содержать **кодовые ячейки**, содержащие

фрагменты кода или программы на языке программирования Python. Интерактивные блокноты Google Colaboratory позволяют использовать в кодовых ячейках все возможности языка Python версии 3.8.16, в том числе стандартные библиотеки языка Python (math, random и т. д.), а недостающие библиотеки можно импортировать при помощи команды «!pip install название библиотеки» (библиотеки с поддержкой графического интерфейса, такие как tkinter, не поддерживаются интерактивными блокнотами Google Colaboratory).

В качестве примера рассмотрим кодовую ячейку раздела «Цикл с предусловием в языке Python», которая представлена на рисунке 5. В ней содержится программа «Привет», которая пять раз выводит на экран слово «Привет!». Для того чтобы запустить данную программу, необходимо нажать на кнопку запуска (1), которая появляется при наведении курсора на символы «[]». При этом под кодовой ячейкой появятся результаты работы программы, левее ячейки — зеленая галочка (если программа выполнена успешно, без ошибок), а под ней — время выполнения программы в секундах (2).

Интерактивный блокнот «Цикл с предусловием» содержит различные типы практических заданий для закрепления полученных знаний по соответствующей теме.

Задание «Измените программу по условию» предлагает изменить заданные программы на языке Python так, чтобы цикл выполнялся определенное количество

Цикл с предусловием – это алгоритмическая конструкция, которая выполняется следующим образом:

1. Проверяется условие цикла;
2. Если условие выполняется, то выполняется тело цикла (команды алгоритма, которые выполняются до тех пор, пока условие истинно) и осуществляется проверка заданного условия (переход к пункту 1);
3. Если условие не выполняется – цикл завершает работу.

На следующем рисунке представлена блок-схема, которая содержит цикл с предусловием.

Данная блок-схема доступна для скачивания по следующей ссылке: [Ссылка на файл] (https://drive.google.com/file/d/1NzprJwjsW63HjzDkLk_ZlP1039W5jvstD/view?usp=sharing)

```
 b:
        a = a + b
        print(a)
```

Программа №4.

```
[ ] a = 4
    b = 7
    while a < b:
        b = a - b
        print(b)
```

Программа №5.

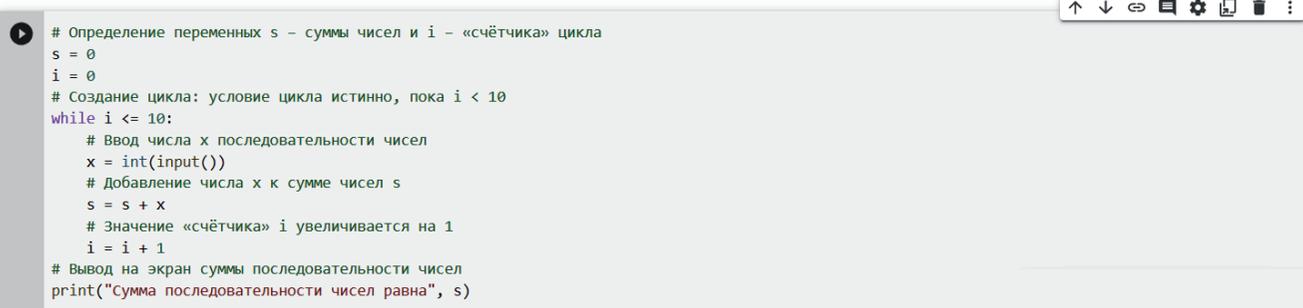
```
[ ] a = 4
    b = 7
    while a < b:
        a = a - 1
        print(a)
```

Рис. 6. Задание «Сколько раз выполнится цикл?»

▼ Программа "Сумма числовой последовательности"

Рассмотрим возможности циклов с предусловием для написания различных программ для работы с числами. Одними из самых простых задач с использованием данной алгоритмической конструкции являются алгоритмы обработки последовательностей чисел. В качестве примера такой программы рассмотрим следующую задачу.

Напишите программу на языке программирования Python, которая получает на вход последовательность из **10** целых чисел, вводимых с клавиатуры, находит их сумму и выводит её на экран.



```
# Определение переменных s - суммы чисел и i - «счётчика» цикла
s = 0
i = 0
# Создание цикла: условие цикла истинно, пока i < 10
while i <= 10:
    # Ввод числа x последовательности чисел
    x = int(input())
    # Добавление числа x к сумме чисел s
    s = s + x
    # Значение «счётчика» i увеличивается на 1
    i = i + 1
# Вывод на экран суммы последовательности чисел
print("Сумма последовательности чисел равна", s)
```

Для решения данной задачи необходимо задать переменные **s = 0** - сумма чисел, вводимых с клавиатуры и **i = 0** - "счетчик" цикла, который будет подсчитывать, сколько раз выполнялся цикл и, соответственно, сколько чисел было введено с клавиатуры.

Далее задается цикл с предусловием, который выполняется до тех пор, пока истинно условие $i \leq 10$, т.е. пока пользователь не ввел с клавиатуры 10 чисел.

Если условие выполняется, программа получает введенное с клавиатуры число **x** и добавляет его к сумме чисел **s**. Затем значение "счётчика" цикла **i** увеличивается на 1, чтобы подсчитать введенное с клавиатуры число.

После того как была найдена сумма 10 чисел, она выводится на экран.

Подумайте, как можно изменить данную программу, чтобы она находила произведение 5 чисел, вводимых с клавиатуры. Какое начальное значение должна иметь переменная произведения чисел? Какое условие необходимо задать в цикле программы?

Рис. 7. Программа «Сумма числовой последовательности»

вательность из 10 целых чисел, вводимых с клавиатуры, находит их сумму и выводит ее на экран;

- «Сумма цифр числа» — программа получает на вход натуральное число n и выводит на экран произведение его цифр;
- «Поиск максимальной цифры числа» — программа получает на вход натуральное число x и выводит на экран его максимальную цифру.

Данные программы обучающийся может запустить непосредственно в блокноте, без необходимости устанавливать среду программирования IDLE Python на компьютер или использовать онлайн-среды с возможностью написания программ на языке программирования Python. Таким образом, интерактивный блокнот «Цикл с предусловием» можно использовать на уроке открытия новых знаний, обретения новых умений и навыков непосредственно на этапе изучения нового материала.

В разделе «Первичное закрепление учебного материала» для закрепления полученных знаний обучающимся предлагается выполнить **практическую работу «Цикл с предусловием»**, представленную в форме интерактивного блокнота:

<https://colab.research.google.com/drive/12nrTVMBPFyB1oO-3PAEPaKbp9xcpUiiJ?usp=sharing>.

Данная практическая работа направлена на проверку знаний обучающихся по теме «Цикл с предусловием»,

полученных в рамках изучения теоретического материала по соответствующей теме, который был рассмотрен ранее. В работе обучающемуся необходимо выполнить **пять практических заданий**:

- «Исправь программу» (задание на исправление программы таким образом, чтобы цикл в программе выполнялся заданное число раз);
- «Повтори фразу» — задание на написание программы, которая получает на вход строку s , а затем выводит ее на экран четыре раза;
- «Произведение числовой последовательности» — задание на написание программы, которая получает на вход натуральное число n и последовательность из n целых чисел, вводимых с клавиатуры, находит их произведение и выводит его на экран;
- «Произведение цифр числа» — задание на написание программы, которая получает на вход натуральное число x и выводит на экран число m — произведение цифр числа x ;
- «Минимальная цифра числа» — задание на написание программы, которая получает на вход натуральное число x и выводит на экран число m — минимальную цифру числа x .

Первое задание практической работы представлено на рисунке 8. Обучающийся имеет возможность выполнить практическое задание непосредственно в блокноте, написав программу в соответствующем поле. Кроме того, он может проверить правильность работы написан-



Практическая работа «Цикл с предусловием»

Инструкция: познакомьтесь с интерактивным блокнотом "Цикл с предусловием" и выполните следующие практические задания. Прежде чем приступать к заданиям, нажмите на кнопку "Копировать на диск", чтобы сохранить результаты работы.

Задание №1. Исправь программу

Постановка задания: Ниже представлены 4 программы на языке программирования Python.

Измените одну команду программы на языке Python так, чтобы:

- для программы А: цикл выполнялся 3 раза;
- для программы Б: цикл выполнялся 1 раз;
- для программы В: цикл не выполнялся ни разу;
- для программы Г: цикл выполнялся бесконечно;

<pre>a = 5 b = 11 while a < b: a = a + 1 print(a)</pre> <p style="text-align: right; font-weight: bold; color: red;">А</p>	<pre>a = 20 b = 4 while a > b: a = a - b print(a)</pre> <p style="text-align: right; font-weight: bold; color: red;">Б</p>	<pre>a = 8 b = 7 while a >= b: b = b + 1 print(b)</pre> <p style="text-align: right; font-weight: bold; color: red;">В</p>	<pre>a = 18 b = 34 while a <= b: a = a + b print(a)</pre> <p style="text-align: right; font-weight: bold; color: red;">Г</p>
---	---	---	---

Задание может иметь несколько различных решений!

```
# Исправьте программу А, которая представлена ниже
a = 5
b = 11
while a < b:
    a = a + 1
print(a)
```

```
[ ] # Исправьте программу Б, которая представлена ниже
a = 20
b = 4
while a > b:
    a = a - b
print(a)
```

```
[ ] # Исправьте программу В, которая представлена ниже
a = 8
b = 7
while a >= b:
    b = b + 1
print(b)
```

```
[ ] # Исправьте программу Г, которая представлена ниже
a = 18
b = 34
while a <= b:
    a = a + b
print(a)
```

Рис. 8. Задание «Исправь программу» практической работы «Цикл с предусловием»

ной им программы, нажав на кнопку запуска, которая появляется при наведении курсора на символы «[]», и запустив программу. Таким образом, практическую работу «Цикл с предусловием» можно использовать при изучении соответствующей темы раздела «Алгоритмы и программирование» как на уроке открытия новых знаний, обретения новых умений и навыков непосредственно на этапе первичного закрепления нового материала, так и на уроке комплексного применения знаний и умений на этапе актуализации знаний.

Для того чтобы иметь возможность **отправить результат своей работы учителю**, обучающемуся сначала необходимо зарегистрироваться или войти в свою

учетную запись Google, нажать на кнопку *Копировать на диск* и только затем приступить к практическим заданиям работы, в противном случае после перезагрузки страницы с интерактивным блокнотом результаты работы будут утеряны. После выполнения заданий практической работы обучающийся может воспользоваться кнопкой *Поделиться* для того, чтобы создать ссылку на интерактивный блокнот.

Окно предоставления доступа к интерактивному блокноту представлено на рисунке 9. Чтобы создать ссылку на блокнот, необходимо во вкладке *Общий доступ* выбрать пункт *Все, у кого есть ссылка* и нажать на появившуюся кнопку *Копировать ссылку*. Кроме того,

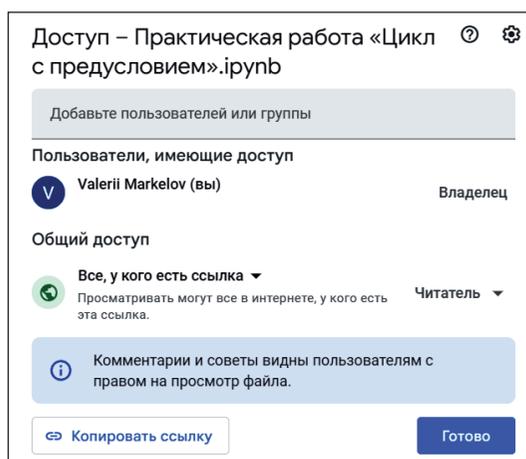


Рис. 9. Окно предоставления доступа к интерактивному блокноту



Рис. 10. Отображение комментария к ячейке интерактивного блокнота

используя текстовое поле *Добавьте пользователя или группы*, можно предоставить доступ к блокноту по адресу электронной почты преподавателя (если он привязан к учетной записи Google).

При проверке практической работы учитель может воспользоваться **функцией добавления комментариев к ячейке**. В качестве примера рассмотрим комментарий к заданию «Повтори фразу». Чтобы создать комментарий, необходимо в управляющем меню выбранной ячейки нажать на кнопку *Добавить комментарий* (кнопка со значком сообщения), в появившемся окне написать комментарий и нажать на кнопку *Комментировать*. Комментарий будет отображаться справа от ячейки, к которой он относится (рис. 10).

Учителя могут как использовать готовые обучающие блокноты интерактивной облачной среды Google Colaboratory, так и создавать собственные интерактивные блокноты для обучения программированию в рамках школьного курса информатики на базе языка Python.

3. Интерактивные блокноты как инструмент подготовки обучающихся к решению заданий по программированию ЕГЭ по информатике

Облачная среда Google Colaboratory может использоваться как в рамках изучения отдельных тем раздела «Алгоритмы и программирование», так и при подготовке обучающихся к ЕГЭ по информатике, в частности, в контексте заданий по программированию, направленных

на проверку умения создавать собственные программы. Одной из особенностей таких заданий является необходимость обработки информации из одного или нескольких текстовых файлов.

Рассмотрим возможности использования Google Colaboratory для обучения решению данных заданий на базе языка программирования Python на примере задания № 24 ЕГЭ по информатике.

Интерактивный блокнот «Задание № 24 ЕГЭ по информатике» (для обучающихся) (рис. 11) доступен по ссылке:

<https://colab.research.google.com/drive/1NyAca9zOlpXBztLiYyWS9-bAcl8u2GF5?usp=sharing>

Данный блокнот состоит из четырех разделов:

- «Теоретические сведения для решения задач»;
- «Примеры экзаменационных заданий»;
- «Список тренировочных заданий»;
- «Инструкция по отправке решения».

Раздел «Теоретические сведения для решения задач» содержит теоретические сведения, которые могут помочь обучающимся при решении задания № 24 ЕГЭ по информатике. В частности, в данном разделе рассматривается алгоритмическая конструкция для работы с файлами в языке программирования Python и приводятся различные методы работы со строками.

В разделе «Примеры экзаменационных заданий» представлены алгоритмы решения различных типов экзаменационных заданий № 24, в том числе из открытого банка заданий ЕГЭ.

Рассмотрим один из примеров экзаменационного задания (рис. 12). В данной задаче представлен тексто-

Интерактивный блокнот «Задание №24 ЕГЭ по информатике» (для обучающихся)

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения не будут сохранены

Поделиться ⚙ Войти

Содержание

Интерактивный блокнот «Задание №24 ЕГЭ по информатике»

Теоретические сведения для решения задач

Примеры экзаменационных заданий

Список тренировочных заданий

Инструкция по отправке решения

Раздел

+ Код + Текст Копировать на Диск Подключиться Редактирование

Интерактивный блокнот «Задание №24 ЕГЭ по информатике»

Задание №24 в едином государственном экзамене (ЕГЭ) по информатике проверяет умение обучающегося создавать собственные программы (10-20 строк) для обработки символьной информации.

Максимальный балл за выполнение задания: 1 балл.
Примерное время выполнения задания: 18 минут.

Теоретические сведения для решения задач

Прежде всего для решения задачи необходимо получить символьную информацию (одну или несколько строк) из текстового файла.

В языке программирования Python получить символьную информацию из файла **file.txt** (которая записана в одну строку) в переменную **s** можно следующим образом:

```
with open("file.txt", "r") as f:
    s = f.readline()
```

Для этого в Python используется алгоритмическая конструкция **with-as**, которая при помощи команды **open** открывает файл **file.txt** в режиме чтения ("r" - аргумент команды open, указывающий открытие в режиме чтения).

Рис. 11. Интерактивный блокнот «Задание № 24 ЕГЭ по информатике» (для обучающихся)

Интерактивный блокнот «Задание №24 ЕГЭ по информатике» (для обучающихся)

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохранены

Комментировать Поделиться ⚙

ОЗУ Диск Редактирование

Файлы

sample_data

24-01.txt (4)

Скачать (2)

Переименовать файл

Удалить файл

Скопировать путь

Обновить (5)

+ Код + Текст

Примеры экзаменационных заданий

Пример №1. Текстовый файл состоит из одной строки, которая содержит только буквы латинского алфавита. Определите, сколько раз в исходном файле встречается символ, стоящий в файле на первом месте. Для выполнения этого задания следует написать программу.

Текстовый файл задания доступен по следующей ссылке: <https://drive.google.com/file/d/1TOY-PuE03d5iyDTPQuM4VvGDit27-4NI/view?usp=sharing> (1)

Решение: Для проверки правильности решения задачи скачайте файл **24-01.txt**, для этого выберите пункт "Файлы" на боковой панели окна облачной среды и нажмите на кнопку "Загрузить в сессионное хранилище" (значок документа со стрелкой вверх). Обратите внимание, что ссылка на файл указывается в формате [/content/24-01.txt](#), где **24-01.txt** - название текстового файла задания.

```
[ ] # Открываем файл "24-01.txt" в режиме чтения
with open("/content/24-01.txt", "r") as f:
    # Читаем строку файла в переменную s
    s = f.readline()
    # Записываем искомый символ в переменную k
    k = s[0]
    # Создаем переменную count - сколько раз искомый символ встречается в строке s
    count = 0
    # Создаем цикл для прохождения по всем символам строки s
    for i in s:
        # Проверяем условие: совпадает ли искомый символ k с текущим символом i
        if k == i:
            # Если условие выполняется - значение переменной count увеличивается на 1
            count = count + 1
    # Выводим на экран количество вхождений первого символа k в строку s
    print(count)
```

Алгоритм решения задачи:

1. Открываем файл **24-01.txt** и считываем строку файла в переменную **s**.
2. Получаем из строки её первый символ **k**, используя срез **s[0]** - это искомый символ, количество которых необходимо подсчитать в строке.
3. Создаем переменную **count** со значением **0**, которая будет содержать количество вхождений символа **k** в строку **s**.
4. Перебираем все символы строки, используя цикл с параметром **for i in s**, где **s** - искомая строка, **i** - текущий символ строки.
5. Сравниваем текущий символ строки с искомым символом **k**: если они совпадают - подсчитываем данный символ, увеличивая значение **count** на 1.
6. Выводим значение переменной **count** на экран.

Диск Доступно: 70.73 GB.

Рис. 12. Пример экзаменационного задания № 1

вый файл, состоящий из одной строки, которая содержит только буквы латинского алфавита, и необходимо определить, сколько раз встречается символ, стоящий в файле на первом месте. В соответствующей кодовой ячейке представлена программа, позволяющая решить данную задачу, а ниже — алгоритм решения задачи с подробными комментариями.

Для того чтобы запустить программу, которая решает задачу, сначала необходимо прикрепить текстовый файл задачи к интерактивному блокноту. Для этого необходимо скачать данный файл (1) на свой компьютер под именем, которое указано в задаче (в данном случае — *24.01.txt*). Затем нужно нажать на кнопку *Файлы* (2) на боковой панели интерактивной облачной среды, в появившейся колонке выбрать кнопку *Загрузить в сессионное хранилище* (3), выбрать искомый текстовый файл и нажать на кнопку *Открыть*. При первой загрузке файла в интерактивный блокнот выводится предупреждение о том, что загруженные файлы будут удалены при перезапуске облачной среды. Загруженный файл *24.01.txt* (4) отобразится в списке файлов ниже папки *sample_data*. При нажатии на файл правой кнопкой мыши появится контекстное меню (5), используя которое можно скачать файл на свой компьютер, переименовать или удалить файл, а также скопировать путь к нему. Поскольку все файлы, загружаемые в интерактивный блокнот, находятся в папке *content*, текстовый файл будет доступен по следующему пути: */content/24.01.txt*. После загрузки файла обучающийся может запустить программу, нажав на кнопку запуска, которая появляется при наведении курсора на символы «[]», и увидеть результат работы программы.

Раздел «Список тренировочных заданий» представлен на рисунке 13. Данный раздел содержит тренировочные задания, направленные на подготовку обучающихся к решению задания № 24 ЕГЭ. Прежде чем приступить к работе, обучающемуся необходимо

войти в свою учетную запись Google и нажать на кнопку *Копировать на диск*, чтобы скопировать интерактивный блокнот на свой Google Диск. Для решения тренировочных заданий учащийся может воспользоваться теоретическим материалом интерактивного блокнота и текстовыми файлами заданий, которые можно скачать на свой компьютер, используя представленные в блокноте ссылки.

Для проверки ответов обучающихся и создания собственных тестовых текстовых файлов к тренировочным заданиям учитель может использовать **интерактивный блокнот «Задание № 24 ЕГЭ по информатике» (для учителя)** (рис. 14):

<https://colab.research.google.com/drive/1mpTM4FCZdF2YprLd7aQYMxPKkuVC6mWm?usp=sharing>.

Данный блокнот состоит из трех разделов:

- «Создание тестовых файлов к тренировочным заданиям»;
- «Ответы к тренировочным заданиям»;
- «Ответы к тренировочным заданиям для скачивания».

Последние два раздела содержат ответы к тренировочным заданиям блокнота «Задание № 24 ЕГЭ по информатике» (для обучающихся) как непосредственно в самом блокноте, так и в форме ссылки на архив с текстовыми файлами и решениями тренировочных заданий в виде программ на языке Python.

В разделе «Создание тестовых файлов к тренировочным заданиям» представлена программа, позволяющая создавать текстовые файлы к заданиям, подобным тем тренировочным заданиям, которые рассматривались в интерактивном блокноте «Задание № 24 ЕГЭ по информатике» (для учителя) (рис. 15). Чтобы сгенерировать текстовый файл, учителю достаточно изменить имя файла в константе `FILENAME`, в переменной `N` ввести количество символов, которые должен

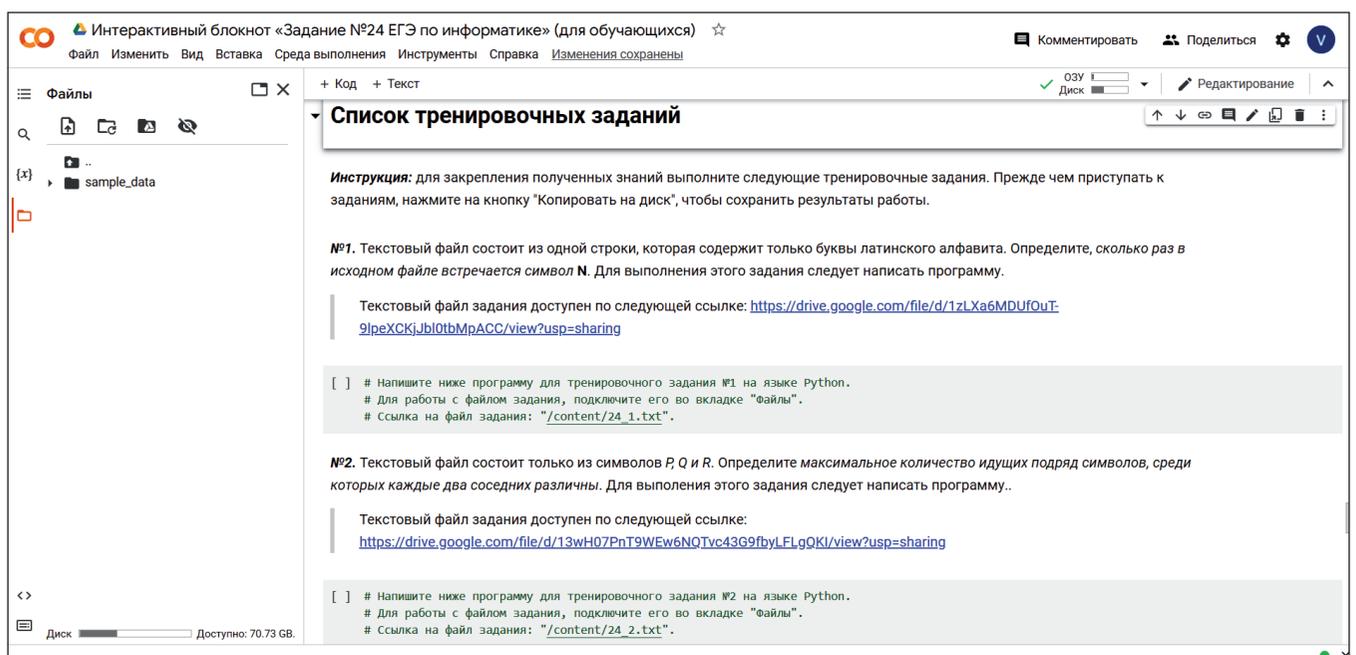


Рис. 13. Раздел «Список тренировочных заданий»

Интерактивный блокнот «Задание №24 ЕГЭ по информатике» (для учителя) ☆

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохранены

Содержание

Интерактивный блокнот «Задание №24 ЕГЭ по информатике»

Создание тестовых файлов к тренировочным заданиям

Ответы к тренировочным заданиям

Ответы к тренировочным заданиям для скачивания

Раздел

Интерактивный блокнот «Задание №24 ЕГЭ по информатике»

Инструкция: данный интерактивный блокнот содержит ответы к тренировочным заданиям интерактивного блокнота «Задание №24 ЕГЭ по информатике», который доступен по следующей ссылке: <https://colab.research.google.com/drive/1NyAca9z0lpxBztlLyvWS9-bAcl8u2GF5?usp=sharing>

Создание тестовых файлов к тренировочным заданиям

Кроме того, используя программу на языке Python, учителя могут создавать собственные файлы к тренировочным заданиям №24. Для этого необходимо скопировать программу, представленную ниже, в кодовую ячейку интерактивного блокнота, в котором находятся тренировочные задания (её можно создать при помощи кнопки "Код" в верхней части окна программы), изменить параметры программы и нажать на кнопку запуска. При успешном создании файла появится соответствующее сообщение.

Файл с заданным названием (по умолчанию, file.txt) можно найти, нажав на пункт "Файлы" на боковой панели окна облачной среды (значок папки). Чтобы скопировать путь к файлу для дальнейшего использования в программе, необходимо нажать на файл и выбрать пункт "Скопировать путь".

```
[ ] # Программа для генерации тестовых файлов к заданию №24
# Задайте значение константы "FILENAME" чтобы задать свое имя файла
```

Рис. 14. Интерактивный блокнот «Задание № 24 ЕГЭ по информатике» (для учителя)

Интерактивный блокнот «Задание №24 ЕГЭ по информатике» (для учителя) ☆

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохранены

Файлы

ОЗУ Диск

Редактирование

Файл с заданным названием (по умолчанию, file.txt) можно найти, нажав на пункт "Файлы" на боковой панели окна облачной среды (значок папки). Чтобы скопировать путь к файлу для дальнейшего использования в программе, необходимо нажать на файл и выбрать пункт "Скопировать путь".

```
# Программа для генерации тестовых файлов к заданию №24
# Задайте значение константы "FILENAME" чтобы задать свое имя файла
# По умолчанию, имя файла - file
FILENAME = "24.1"
# Задайте значение константы "N" чтобы задать количества символов в файле
# По умолчанию, файл содержит 1100000 символов
NUM = 1100000
# Задайте символы, которые могут использоваться в файле при помощи списка LETTERS
# По умолчанию, файл содержит все символы английского алфавита
LETTERS = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"]

import secrets

with open("/content/" + FILENAME + ".txt", "w") as F:
    for _ in range(NUM):
        F.write(secrets.choice(LETTERS))

print("Текстовый файл", FILENAME + ".txt", "успешно создан!")
```

Текстовый файл 24.1.txt успешно создан!

Диск Доступно: 70.73 GB

2 сек. выполнено в 21:49

Ответы к тренировочным заданиям

Рис. 15. Программа для генерации файлов к тренировочным заданиям

содержать файл, изменить список символов LETTERS, оставив в нем только те символы, которые должен содержать файл, а затем нажать на кнопку запуска программы. При успешном создании файла в качестве результата программа выведет сообщение «Текстовый файл успешно создан», а в колонке *Файлы* появится файл с заданным названием, который можно скачать, используя контекстное меню.

Кроме того, текст данной программы учитель может скопировать в собственный интерактивный блокнот, чтобы сгенерированные файлы могли использоваться для решения задач в облачной среде, без необходимости скачивания файлов на компьютер.

4. Заключение

Интерактивная облачная среда Google Colaboratory, с одной стороны, позволяет учителям использовать готовые обучающие интерактивные блокноты в рамках обучения программированию на базе языка Python в школьном курсе информатики — как на уроках открытия новых знаний, обретения новых умений и навыков, так и на уроках комплексного применения знаний и умений, для демонстрации теоретического материала по соответствующим темам раздела, а также для создания практических работ с целью формирования у обучающихся навыков программирования.

С другой стороны, возможности облачной среды Google Colaboratory позволяют учителям использовать интерактивные блокноты в качестве инструмента для подготовки обучающихся к заданиям по программированию ЕГЭ по информатике.

Список источников

1. *Андреева Е. В.* Задачи по программированию в ЕГЭ по информатике // Информатика и образование. 2015. № 3. С. 54–56. EDN: TQMHDR.
2. *Артюхина М. С.* Особенности современных средств обучения в контексте интерактивных технологий // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2014. № 2. С. 76–81. EDN: SCNMVX.
3. *Балабанова И. А.* Программирование в «облаке» // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы международной научно-практической интернет-конференции (Москва, 19–25 апреля 2021 года). М.: Московский педагогический государственный университет, 2021. С. 26–29. EDN: XSMPEQ.
4. *Босова Л. Л.* Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе // Информатика в школе. 2018. № 6. С. 3–11. EDN: XZOOJV.
5. *Газейкина А. И., Тупицына М. В.* Методика формирования у учащихся основной школы умения осуществлять учебное сотрудничество средствами облачных технологий // Педагогическое образование в России. 2017. № 6. С. 21–30. DOI: 10.26170/ro17-06-03. EDN: ZDLZWD.
6. *Голицына И. Н., Афзалова А. Н.* Использование облачных вычислений в образовательном процессе // Образовательные технологии и общество. 2014. № 2. С. 450–459. EDN: SEFPMZ.
7. *Касьянов С. Н.* Обучение языку программирования Python в онлайн-сообществе учащихся школ // Известия Волгоградского государственного педагогического университета. 2021. № 6. С. 50–54. EDN: FGYVWR.
8. *Маркелов В. К., Завьялова О. А.* Возможности языка Python 3.10 при обучении программированию в школе // Шуйская сессия студентов, аспирантов, педагогов, молодых ученых. Материалы XIV Международной научной конференции, посвященной Году науки и технологий Российской Федерации, 205-летию начала подготовки педагогов в Ивановской области (Москва — Иваново — Шуя, 06–07 октября 2021 года). Шуя: Шуйский филиал федерального государственного бюджетного образовательного учреждения высшего образования «Ивановский государственный университет», 2021. С. 96–97. EDN: XXZKLZ.
9. *Митин А. Н.* Облачные технологии в образовании // Вестник НГИЭИ. 2016. № 8. С. 41–47. EDN: WJDAPF.
10. *Фролов Ю. В., Сахнюк П. А.* Цифровые технологии и цифровые компетенции в образовательном процессе подготовки бизнес-информатиков // Актуальные проблемы теории и практики обучения математике, информатике и физике в современном образовательном пространстве. Курск: Курский государственный университет, 2018. С. 60–63. EDN: ZBUWQH.



DOI: 10.32517/2221-1993-2022-21-6-81-88

Т. В. Баракина

Омский государственный педагогический университет, г. Омск, Россия

СИСТЕМА ОБУЧЕНИЯ ДЕТЕЙ ДОШКОЛЬНОГО И МЛАДШЕГО ШКОЛЬНОГО ВОЗРАСТА ОСНОВАМ НАЧАЛЬНОГО КОНСТРУИРОВАНИЯ И РОБОТОТЕХНИКИ

Аннотация

Робототехника — это одно из самых популярных направлений в системе дополнительного образования детей дошкольного и младшего школьного возраста в настоящее время. Но возникают вопросы о том, что происходит на следующих уровнях образования: почему у более старших учащихся теряется интерес к робототехнике, почему подростки перестают посещать занятия кружка робототехники, почему они, как правило, не хотят продолжать заниматься этим направлением в будущей профессиональной деятельности? В чем причина? Как построить систему обучения дошкольников и младших школьников основам начального конструирования и робототехники так, чтобы впоследствии не просто сохранить, но и преумножить интерес учащихся к этому направлению?

В статье рассматриваются система подготовки дошкольников и младших школьников в области конструирования и образовательной робототехники, этапы формирования конструкторских умений у детей, описаны возможные средства обучения, варианты учебных заданий, показан постепенный переход от более простых моделей и конструкторов к более сложным, от игры — к учебной деятельности и выполнению проектов, исследовательских работ, что, на наш взгляд, позволит обеспечить сохранность интереса обучающихся к робототехнике до конца обучения в школе.

Ключевые слова: модель, моделирование, конструирование, конструктор, робототехника, образовательная робототехника, соревновательная робототехника, дошкольники, младшие школьники.

1. Введение

Роботы, робототехника, манипуляторы, умный дом... Какой современный человек не слышал этих слов? Дети с раннего детства играют с роботами и трансформерами, взрослые покупают таких помощников домой, ими оснащены производства. Роботы стали использоваться даже для организации досуга человека.

Робототехника — это одно из самых популярных направлений в системе дополнительного образования детей дошкольного и младшего школьного возраста в настоящее время. А что же дальше? Почему впоследствии у ребят теряется интерес? Почему подростки чаще всего перестают посещать занятия кружка и, как правило, не хотят продолжать заниматься этим направлением в будущей профессиональной деятельности?

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

THE SYSTEM OF TEACHING CHILDREN OF PRESCHOOL AND PRIMARY SCHOOL AGE TO THE BASICS OF INITIAL CONSTRUCTION AND ROBOTICS

Abstract

Robotics is one of the most popular areas in the system of additional education for children of preschool and primary school age at the present time. But questions arise about what happens at the following levels of education: why older students lose interest in robotics, why teenagers stop attending robotics classes, why they, as a rule, do not want to continue to engage in this area in their future professional activities? What is the reason? How to build a system for teaching preschoolers and younger schoolchildren the basics of initial construction and robotics in such a way that subsequently not only maintain, but also increase students' interest in this direction?

The article discusses the system of training preschoolers and junior schoolchildren in the field of construction and educational robotics, the stages of the formation of construction skills in children, describes possible teaching aids, variants of training tasks, shows a gradual transition from simpler models and constructors to more complex ones, from a game to a learning activities and implementation of projects, research work, which, in our opinion, will ensure the preservation of students' interest in robotics until the end of schooling.

Keywords: model, modeling, construction, constructor, robotics, educational robotics, competitive robotics, preschoolers, junior schoolchildren.

На наш взгляд, это связано с отсутствием последовательной системы подготовки будущих инженерных кадров в области робототехники. Ребенок, заинтересовавшись яркой презентацией, приходит на занятия и вдруг через некоторое время понимает, что это не развлечение с игрушками, это серьезная подготовка, которая основывается на точных теоретических положениях, на знании основ математики, физики, информатики и т. д. А у него или нет этих знаний, или он просто не умеет их применять на практике, не понимает, зачем ему это нужно.

Очень важно понимать, что интерес к конструированию в целом и к робототехнике в частности формируется не за один день. Кроме того, следует отметить, что начинать формировать этот интерес необходимо с дошкольного и младшего школьного возраста, постепенно переходя от игры к учебной деятельности, формируя потребность в использовании изученного далее не только на занятиях по конструированию, но и при изучении учебных предметов в школе, в повседневной жизни [2].

Как же построить систему обучения дошкольников и младших школьников основам начального конструирования и робототехники так, чтобы не просто сохранить, но и преумножить интерес ребенка? Попробуем разобраться.

2. Подготовка детей дошкольного возраста в области конструирования и образовательной робототехники

Дошкольное детство — это пора жизни, когда ребенок не просто знакомится с окружающим миром, а овладевает основными видами человеческой деятельности через манипулирование конкретными предметами, в играх, в общении со сверстниками и взрослыми.

Поэтому важно с раннего возраста окружать ребенка теми игрушками, которые будут соответствовать действительности и не искажать ее, знакомить с теми материалами и технологиями, которые встречаются в повседневной жизни.

Современный ребенок практически с рождения окружен разнообразными гаджетами, которые он легко, быстро, без страха, на интуитивном уровне осваивает, игрушками-трансформерами, роботами, конструкторами и т. п. Технологизация детства — это хорошо или плохо? Сложно сказать однозначно. На наш взгляд, необходимо соблюдать золотую середину, объединяя традиции и инновации.

Ознакомление дошкольников с элементарными основами моделирования, с конструированием является традиционным для системы образования в России. Еще в детском саду дети знакомятся с различными материалами (бумага, пластилин, глина, природный, бросовый материал) и технологиями их обработки (оригами, аппликация, бумагопластика, папье-маше, лепка и т. д.), учатся конструировать по образцу, по замыслу, в том числе с применением простейших конструкторов, осуществляется развитие у них пространственных представлений, мелкой моторики, графомоторных умений, общей культуры, элементов учебной деятельности.

Пространственные представления — это представления, в которых находят отражение пространственные

отношения предметов (величина, форма, местоположение, движение).

Мелкая моторика — совокупность скоординированных действий человека, направленных на выполнение точных мелких движений кистями и пальцами рук и ног.

К *графомоторике* относятся: зрительное восприятие заданного материала, сосредоточенность и внимание, правильное удержание карандаша или ручки, соответствующий нажим ручки при письме, ритмичность движений, точность в обведении линий, двигательные навыки и т. п.

Чтобы обеспечить развитие вышеуказанных качеств у ребенка, целесообразно использовать различные мозаики, пазлы, раскраски, задания со штриховками, игры на размещение объектов, конструкторы.

Первые конструкторы, с которыми знакомятся дошкольники, как правило, блочного типа с креплением и без крепления, затем вводятся магнитные пластинчатые, щеточные конструкторы, с креплением стык в стык, в завершение — винтовые (рис. 1). Такая последовательность определена не только необходимостью соблюдения техники безопасности, но и особенностями физического развития дошкольников: ребенок с плохо развитой мелкой моторикой рук не сможет успешно выполнять задания с винтовыми конструкторами, пользоваться инструментами.

Важно помнить, что детали конструкторов для дошкольников должны быть крупными, не иметь острых краев, легко соединяться, а конструкции, созданные из этих деталей, должны разбираться без особых затруднений самими детьми, без помощи взрослых. Поэтому, на наш взгляд, не стоит торопиться вводить в детском саду робототехнические конструкторы, содержащие мелкие, хрупкие и дорогостоящие элементы, тем более если нет возможности организовать индивидуальную работу или конструирование в малых группах по два-три человека.

Ведущий вид деятельности в дошкольном возрасте — игра, поэтому и процесс конструирования должен быть построен в игровой, занимательной форме. В сюжетно-ролевых играх дети сначала рассматривают ситуацию, создают из деталей конструкторов необходимые объекты, а потом разыгрывают сюжет. Таким образом решаются не только образовательные задачи, но и воспитательные, дети учатся взаимодействовать друг с другом и со взрослыми.

Какие задания могут быть предложены дошкольникам?

- На построение конструкций по образцу (в качестве образца выступает реальный объект или его изображение, конструкция, уже созданная из деталей конструктора, или ее изображение);
- на построение конструкций по схеме, алгоритму, данным вербально или графически;
- на дополнение, завершение частично данной конструкции;
- на преобразование, улучшение данных конструкций;
- на построение конструкций по теме или по собственному замыслу.

В ходе выполнения заданий дети знакомятся с конструктором, с названием его деталей, с особенностями их соединения, учатся представлять свой проект, разы-



Рис. 1. Виды конструкторов по способу крепления деталей

грывать жизненную ситуацию с применением созданных моделей. В свою очередь, это способствует обогащению словарного запаса ребенка,

Таким образом, в ходе выполнения конструкторской деятельности у дошкольников формируются первичные представления об объектах окружающего мира: форме, цвете, размере, материале, количестве и т. д. Дети учатся выделять признаки объектов, определять среди них существенные (позволяют однозначно определить объект) и несущественные. Развиваются элементы логического мышления. Так, например, определяя, из каких элементов состоит конструкция или какую конструкцию можно построить из данных элементов, дети выполняют аналитико-синтетическую деятельность, сравнивают конструкции со схемами сборки и с готовыми образцами, а также с объектами из окружающего мира, выполняют операцию классификации, разделяя элементы конструкторов на группы по цвету, размеру, назначению. У детей развиваются речь, коммуникативные умения.

3. Подготовка детей младшего школьного возраста в области конструирования и образовательной робототехники

Обучение младших школьников основам конструирования может реализовываться в рамках как основного (на уроках математики, технологии, информатики), так и дополнительного (кружок, факультатив) образования. Но, если дополнительное образование реализуется на добровольной основе, основное является обязательным для изучения каждым ребенком.

Что же предполагается рассматривать в области конструирования в начальной школе?

В результате изучения *начального курса математики* обучающиеся на уровне начального общего образования [8]:

- научатся использовать начальные математические знания для описания окружающих предметов, процессов, явлений, оценки количественных и пространственных отношений;
- овладеют основами логического и алгоритмического мышления, пространственного воображения и математической речи;
- научатся применять математические знания и представления для решения учебных, в том числе конструкторских, задач, приобретут начальный опыт применения математических знаний в повседневных ситуациях.

В рамках предметной области «Технология» младшие школьники должны не только получить общее представление о мире профессий, их социальном значении, истории возникновения и развития, но и научиться использовать приобретенные знания и умения для творческой самореализации.

Решение конструкторских, художественно-конструкторских и технологических задач должно обеспечить развитие основ творческой деятельности, конструкторско-технологического мышления, пространственного воображения, эстетических представлений, формирования внутреннего плана действий, мелкой моторики рук.

В программе выделяется отдельный **раздел «Конструирование и моделирование»**, в ходе изучения которого выпускник [8]:

научится:

- анализировать устройство изделия: выделять детали, их форму, определять взаимное расположение, виды соединения деталей;
- решать простейшие задачи конструктивного характера по изменению вида и способа соединения деталей: на достраивание, придание новых свойств конструкции;
- изготавливать несложные конструкции изделий по рисунку, простейшему чертежу или эскизу, образцу и доступным заданным условиям;

получит возможность научиться:

- соотносить объемную конструкцию, основанную на правильных геометрических формах, с изображениями их разверток;
- создавать мысленный образ конструкции с целью решения определенной конструкторской задачи или передачи определенной художественно-эстетической информации;
- воплощать этот образ в материале.

При этом **процесс формирования конструкторских умений у младших школьников может быть разбит на несколько этапов** [1].

Этап 1. Знакомство младших школьников с простейшими механизмами, различными конструкторами, в том числе робототехническими. Дети должны получить представления о мире профессий, о сферах применения роботов в жизни и производстве, их значении в жизни современного человека, научиться собирать конструкции по образцу, дополнять или преобразовывать данные устройства в соответствии с поставленным заданием, работать с изображениями, самостоятельно придумывать и создавать модели роботов с использованием простейших механизмов (рычаг, ролик, ось, зубчатое колесо и т. п.).

Этап 2. Конструирование и прототипирование (создание модели для решения учебных практических задач). Задача педагога на данном этапе состоит в том, чтобы научить детей применять полученные на предыдущем этапе знания в практической деятельности. Показать, какие задачи в области математики, информатики, технологии могут помочь решить роботы. От механического создания роботов дети должны перейти к осознанному и рациональному.

Этап 3. Алгоритмизация. Знакомство с алгоритмами, языками программирования. Обучение детей

созданию программы управления роботом. Сначала обучающиеся учатся работать с готовыми простейшими алгоритмами (выполнение, анализ, изучение последовательности команд, поиск и исправление ошибок), затем переходят к преобразованию алгоритмов по заданным условиям, к самостоятельному составлению алгоритмов.

Этап 4. Решение образовательных и жизненных задач с использованием программируемых роботов.

Таким образом, к окончанию начальной школы обучающиеся не только усвоят понятия модели и моделирования, но и научатся создавать конструкции различного уровня сложности с применением различных материалов и средств, в том числе с использованием конструкторов.

4. Робототехнические конструкторы, используемые в дошкольном и начальном общем образовании

Рассмотрим некоторые из робототехнических конструкторов, использующихся в настоящее время в дошкольном и начальном общем образовании.

4.1. Линейка наборов Huna — MRT — РОБОТРЕК

Линейка конструкторов Huna — MRT — РОБОТРЕК [4] достаточно широка и может обеспечить потребности учащихся от детского сада (с трех лет) до вуза, тем самым реализуя принцип преемственности (рис. 2). Во всех наборах используются блоки, соединяющиеся с шести сторон, делающие процесс конструирования и 3D-моделирования простым, быстрым и увлекательным. Строительные блоки и электроника всех наборов совместимы друг с другом, поэтому при переходе с одного набора на другой ребенку не приходится изучать конструктор заново.



Рис. 2. Городок собран из трех конструкторов HUNA — Kicky Basic, Kicky Junior и Fun&Bot Sensing

В линейке наборов MRT 1 (MRT — My Robot Time, конструкторы производятся южнокорейской компанией My Robot Time) с наборами идут распечатанные полноцветные карты сборки всех моделей. Все остальные наборы сопровождаются цифровыми книгами, в которых предлагаются большое количество моделей для сборки от простых к сложным и рабочие тетради.

То, что конструкторы начального уровня (MRT 1 и MRT 2 (Kicky)) не требуют программирования, обеспечивает их доступность и для детей, и для начинающих педагогов, что немаловажно с учетом дефицита кадров в области образовательной робототехники для младшего возраста. В сериях Class и MRT 3 (для начальной школы) используется простая визуализированная среда программирования (открытая, не требующая лицензий) с возможностью параллельного генерирования кода в язык программирования C.

Линейка делится на серии наборов, различных по составу деталей, электроники и сложности предлагаемых моделей в зависимости от возраста учащихся.

Для дошкольников предлагаются наборы:

- MRT 1 (Hand и Brain) — для детей 3–5 лет;
- MRT 2 (Kicky) — для детей 5–7 лет;

- тематические наборы для коллективной проектной деятельности.

Для начальной школы:

- Class 3 Full Kit (Class 2 Fullkit) — для детей 7–11 лет;
- MRT 3 (1+2+3+4) (MRT 3 (1+2)) — непрограммируемый уровень) — для детей 7–11 лет.

4.2. LEGO

Одними из лидирующих в мире конструкторов являются конструкторы LEGO. У компании LEGO множество различных линеек и наборов конструкторов. Для обучения компания предлагает образовательные наборы для разных возрастов под названием **LEGO Education** [3, 5].

LEGO «Городская жизнь» (рис. 3). Конструктор подходит для детей дошкольного возраста старше четырех лет. В комплект входит инструкция по сборке и около 2000 пластиковых деталей: прямоугольных блоков разных размеров и цветов (красный, зеленый, желтый, синий), фигурки людей, колеса, цветы, окна, двери и другие подобные детали. С помощью деталей обучающиеся не просто создают разнообразные модели,



Рис. 3. Конструктор LEGO «Городская жизнь»



Рис. 4. Конструктор LEGO «Простые механизмы»

а строят целые истории, которые обязательно потом представляют друг другу и взрослым. Таким образом, конструктор не только позволяет формировать конструкторские умения у детей, но и способствует развитию у детей речи и коммуникативных навыков, творческих способностей, фантазии и изобретательности.

LEGO «Простые механизмы» (рис. 4). Набор рекомендован для работы с детьми от семи лет, он знакомит детей с основными механизмами, которые встречаются в повседневной жизни. Комплект является образовательным и включает в себя пластиковые фигурки человечков, строительные блоки (пластины, кирпичики и т. д.), оси, шкивы, колеса, шестеренки, лопасти, крылья, шкалы, цветные чертежи для сборки моделей. Набор позволяет уже с семи лет в игровой форме объяснять явления и основные механизмы, разнообразить уроки математики, технологии, в дальнейшем набор можно использовать

на уроках физики. У обучающихся есть возможность собрать четыре основных и 16 стандартных моделей. К данному набору выпущены методические материалы для педагогов.

ПервоРобот LEGO WeDo (рис. 5). Рекомендуется для работы с детьми шести— восьми лет. В набор входят 158 деталей для сборки, сервомотор, датчики наклона и сборки, USB-коммутатор. Также в комплект входят программное обеспечение, с помощью которого обучающиеся смогут самостоятельно программировать модели и создавать новых роботов, инструкции по сборке и примеры программирования. Работая с данным конструктором, дети не только смогут на доступном уровне изучить теоретические основы информатики, математики и физики, но и получат первый опыт написания программ, изучат принципы работы моделей и построения алгоритмов.



Рис. 5. Конструктор ПервоРобот LEGO WeDo



Рис. 6. Конструктор VEX

Каждый набор LEGO сопровождается методическими рекомендациями. Все наборы предусматривают последовательное изучение основ робототехники и программирования, что позволяет эффективно работать с обучающимися и реализовывать задачи инженерно-политехнического образования.

Ученикам старше 10 лет можно предложить работу с конструктором LEGO MINDSTORMS.

4.3. VEX

Для детей от пяти лет и старше предлагается линейка конструкторов VEX фирмы HEXBUG [6] (рис. 6), предназначенная путем вовлечения в игру развивать инженерное воображение и помочь ребенку осуществить первый шаг к изучению науки, технологии, инжиниринга и математики (STEM).

Образовательные решения VEX Robotics для обучающихся начальных и средних классов представлены двумя линейками робототехнических наборов: пластиковыми конструкторами VEX IQ и металлическими комплектами VEX EDR.

Конструкторы VEX IQ, в которые входит большое количество пластиковых деталей, сенсоров и программируемый контроллер, очень просты в использовании — структурные элементы соединяются и разъединяются без специальных инструментов. А огромное количество шестеренок, колес и других соединительных механизмов

позволяет конструировать разнообразных мобильных роботов.

Наборы VEX EDR состоят из перфорированных металлических деталей (профилей и пластин), пластиковых элементов передач (зубчатых колес, шкивов, колес и пр.), микроконтроллера Cortex, сервомоторов и разнообразных датчиков. Среди комплектующих VEX есть элементы пневматики и линейные передачи, различные колеса и гусеничные траки.

4.4. «Роботология»

«Роботология» — российское оборудование для программирования и конструирования роботов [7] (рис. 7). Его особенность — в широком применении отечественных элементов, а также в возможности изучения различных российских и зарубежных процессоров. Оборудование спроектировано для поэтапного изучения робототехники, начиная с самых азов — изучения принципов работы радиоэлементов и механических узлов, из которых потом строятся роботы. Оборудование дает возможность программирования различных микропроцессоров в одном конструкторе. Стоит отметить облегченный переход между разными архитектурами микропроцессоров. Конструктор создавался ведущими инженерами-конструкторами, имеющими 20-летний опыт внедрения разработок в области автоматизации процессов на крупных российских промышленных



Рис. 7. Расширенный набор конструктора «Роботология», предназначенный для обучения детей от 6 лет

предприятиях. Создателям удалось передать преемственность российской инженерной традиции от дошкольных учреждений через школы, вузы к производственным предприятиям. Детали конструктора вандалоустойчивы и обладают высокой ремонтпригодностью. Исключена возможность короткого замыкания. Все это облегчает процесс использования конструктора в работе с детьми дошкольного и младшего школьного возраста.

И это далеко не весь перечень робототехнических конструкторов, которые можно использовать в системах дошкольного и начального общего образования.

5. Заключение

На наш взгляд, именно глубокое погружение детей в процесс конструирования и моделирования, постепенный переход от более простых моделей и конструкторов к более сложным, от игры — к учебной деятельности, использование игровых моментов, проектов и исследовательских работ позволят сохранить интерес обучающихся к робототехнике до конца обучения в школе.

Начальное конструирование и моделирование, а также образовательная робототехника способны быстро вовлечь ребенка в практическую деятельность, обеспечив наглядность, получение результата, взаимодействие со сверстниками и взрослыми [9, 10].

При этом следует отметить, что необходимо учитывать индивидуальный подход к обучающимся, скорость усвоения ими учебного материала. Возможен «прыжок» через ступень или программу, ускоренный курс. Важ-

но, чтобы ребенок не заскучал, не потерял интерес. Педагогу и родителям необходимо вовремя поддержать его и в случае необходимости перевести на следующую ступень подготовки.

Список источников

1. *Баракина Т. В.* Направления робототехники в системе основного и дополнительного образования детей // Информатика в школе. 2021. № 5. С. 44–49. DOI: 10.32517/2221-1993-2021-20-5-44-48. EDN: GKASZU.
2. *Баракина Т. В.* Политехническое образование с детского сада и начальной школы // Информатика в школе. 2018. № 8. С. 28–34. DOI: 10.32517/2221-1993-2018-17-8-28-33. EDN: YNHKGL.
3. *Белиовская Л. Г.* Использование ЛЕГО-роботов в инженерных проектах школьников. Отраслевой подход: учебное пособие. М.: ДМК Пресс, 2016. 88 с.
4. Конструктор Huna. <http://hunarobo.ru/oborudovanie.html>
5. Конструкторы LEGO. <https://www.lego.com/ru-ru>
6. Конструкторы VEX и VEX IQ. <https://robotgeeks.ru/collection/vex-iq>
7. Образовательные конструкторы Роботология. <http://robotologia.ru/>
8. Примерная основная образовательная программа начального общего образования (одобрена решением федерального учебно-методического объединения по общему образованию, протокол от 18 марта 2022 г. № 1/22). <https://fgosreestr.ru/poop/primernaia-osnovnaia-obrazovatelnaia-programmanachalnogo-obshchego-obrazovaniia-1>
9. *Тарапата В. В., Самылкина Н. Н.* Робототехника в школе: методика, программы, проекты. М.: Лаборатория знаний, 2018. 109 с.
10. *Филиппов С. А.* Уроки робототехники. Конструкция. Движение. Управление. М.: Лаборатория знаний, 2018. 190 с.



НАШ ЮБИЛЕЙ

Баракина Т. В. Формирование элементов медиаинформационной грамотности у младших школьников	4
Блинов Д. М. Развитие функциональной грамотности у обучающихся на уроке информатики при построении компьютерных моделей с помощью табличных процессоров	4
Богданова Д. А. О современных подходах к дизайну пользовательского интерфейса	4
Босова Л. Л. О новых подходах к изучению школьной информатики в условиях цифровой трансформации общества	4
Букина Е. Ю. STEAM-проект «Тайна эоловой арфы. По следам братьев Бернардацци»	4
Заславская О. Ю., Любутов О. Д. Особенности методики преподавания цикла For в языке Python в средней школе	4
Зубрилин А. А., Зубрилина М. С. Использование игровых элементов во внеурочной деятельности по информатике	4
Климина Н. В. Программирование дизайна машинной вышивки	4
Кошелева Л. А., Пешкова Е. А., Егорькова И. А. Интегрированный урок по физике, обществознанию и информатике «Радиевые девушки. Скандальное дело работниц фабрик, получивших дозу радиации от светящейся краски»	4
Курганова Н. А. Использование устройств для погружения в виртуальную реальность при изучении раздела «Устройства ввода и вывода» на уроке информатики	4
Меньшиков В. В. Урок на тему «Элементы схемотехники. Логические схемы»	4
Федосов А. Ю. Вопросы воспитания в школьном курсе информатики базового уровня	4

АКТУАЛЬНЫЕ ВОПРОСЫ МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ В СОВРЕМЕННОЙ ШКОЛЕ

Белоконова С. С., Яковлева А. Н. Чат-боты как элемент геймификации для повышения вовлеченности учащихся в образовательный процесс	5
Босова Л. Л., Филиппов В. И., Босова А. Ю. Программирование графики и анимации в курсе информатики основной школы	5
Векслер В. А. Реализация концепции STEAM-образования при решении задач о движении роботов	5
Добровольская Н. Ю. Мотивация школьников к учебной деятельности как опора организации самостоятельной работы по информатике	5
Заводчикова Н. И., Быкова И. А. Поэтапное формирование у учащихся навыка применения алгоритма перевода чисел между системами счисления с основаниями p и q , где q является степенью p	5

Корнилов П. А., Федотова Н. П. Олимпиада по информатике и информационным технологиям для абитуриентов ЯГПУ	5
Лопушанская Н. Д. Интегрированный курс «Методы приближенных вычислений» в классах физико-математического и инженерного профилей	5
Панишева О. В. Наглядные средства обучения математике в условиях компьютеризации образования	5
Плясунова У. В., Заводчикова Н. И. Опыт разработки и проведения Всероссийского дистанционного командного квеста по информатике	5
Пузиновская С. Г., Счеснович О. А. Лучше один раз увидеть, чем сто раз услышать, или Визуализация учебной информации на уроках информатики	5
Рыжова Н. И., Трубина И. И. Актуализация изучения этических проблем искусственного интеллекта современными школьниками	5
Самылкина Н. Н., Кашапова Р. Ф. Изучение 3D-моделирования и прототипирования в предпрофессиональных классах с помощью программы T-FLEX CAD	5
Федорова Г. А., Соломина К. А. Дистанционная олимпиада для учащихся VII—IX классов «Этические и правовые основы взаимодействия в социальных сетях»	5

КОНКУРС ИНФО-2021

Баракина Т. В. Формирование знаково-символической деятельности у младших школьников в процессе ознакомления с информационными процессами	1
Блинов Д. М. Образовательные анимационные видеоролики в школьном образовании	1
Итоги XVIII Всероссийского конкурса научно-практических работ ИНФО-2021	1
Климина Н. В. Веб-подход к созданию VR-контента при проведении занятий в VR-квантуме мини-технопарка «Кванториум»	1
Курганова Н. А. Использование технологии цифрового сторителлинга при изучении темы «Компьютерные вирусы» на уроке информатики	1
Фоминых И. А., Довранов А. Р. О разработке курса по выбору «Основы программирования на Python» для предпрофильной подготовки школьников	1

ИНТЕГРИРОВАННЫЕ УРОКИ

Букина Е. Ю. Интегрированный урок по литературе и информатике «Мужичок с ноготок» (VI класс)	1
Сальникова О. С., Селяничев О. Л. Особенности занятий по геоинформационным технологиям со школьниками в рамках дополнительного образования	1

МЕТОДИЧЕСКАЯ КОПИЛКА

Александрова Н. А., Тимонин А. Н. Развитие аналитического мышления обучающихся на уроках информатики средствами имитационного моделирования	2
--	---

Богданова А. Н., Федорова Г. А. Чат-боты как компонент содержания обучения основам искусственного интеллекта в школе	2
Босова Л. Л., Аквилянов Н. А. Обработка символьных данных: от простого к сложному	6
Водопьян Г. М. На пути к смарт школе: взгляд из классной комнаты	2
Гербут С. С. Пропедевтика программирования: как подготовить учеников к восприятию объектно-ориентированного программирования	6
Дженжер В. О., Денисова Л. В. Динамические массивы и списки в PascalABC.NET	1
Дронова Е. Н. Реализация межпредметных связей информатики и математики при изучении темы «Обработка числовой информации в электронных таблицах»	6
Златопольский Д. М. Что такое «десятичная система счисления»? (О двух терминах информатики)	2
Маркушевич М. В. Элементы методики преподавания компьютерной трехмерной графики на уровне основного общего образования с использованием КОМПАС-3D LT V12	3
Маркушевич М. В. Элементы методики преподавания темы «Растровая компьютерная графика» с помощью свободного графического редактора GIMP на уровне основного общего образования	6
Маркушевич М. В. Элементы методики преподавания темы «Создание и обработка звуковой информации» с использованием свободного звукового редактора Audacity	1
Махмутова М. В., Шевцова М. А. Некоторые аспекты разработки курса графического дизайна для дополнительного образования учащихся основной школы	2
Паравина А. С. Чек-лист по теме урока как средство обратной связи для учителя	6
Полякова А. Ю. Цифровая трансформация математического образования как преодоление цифрового разрыва (на примере обучения стохастике)	2
Розов К. В., Шрайнер Б. А. Дистанционное обучение школьников технологиям искусственного интеллекта	6

ЗАДАЧИ

Долинский М. С. Рекурсивное решение задач с помощью метода «Разделяй и властвуй»	2
Долинский М. С. Ускорение рекурсивных решений при помощи мемоизации	6
Магомедов А. М., Раджабова Н. Ш. Замощение клетчатой полосы шириной 4	1
Рузаков А. А. Система дифференцированных заданий для формирования у обучающихся универсальных учебных действий по теме «Базы данных. Поиск информации»	2

ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Карманова Е. В., Медведева П. И. Подготовка школьников старших классов к решению олимпиадных заданий по интеллектуальному анализу данных	2
---	---

Самылкина Н. Н., Калинин И. А. Решение задач отборочного и основного этапов первой всероссийской олимпиады по искусственному интеллекту для учащихся VIII—XI классов	3
---	---

СРЕДСТВА ИКТ В УЧЕБНОМ ПРОЦЕССЕ

Завьялова О. А., Маркелов В. К. Возможности онлайн-сред программирования при обучении языку Python в школе	3
Кузьмин О. В., Лавлинский М. В. Применение иммерсивных информационных технологий в образовании	2
Маркелов В. К., Завьялова О. А. Инструментарий облачной среды Google Colaboratory для обучения программированию и подготовки к ЕГЭ по информатике	6
Морох Е. А., Шаулина Д. С., Юдина Е. Н. Навигатор сред блочного программирования	2

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Баракина Т. В. Система обучения детей дошкольного и младшего школьного возраста основам начального конструирования и робототехники	6
Князева Н. К. Формирование навыков коммуникации и кооперации у младших школьников при создании анимации на уроках математики	3

КОНКУРСЫ

Алексеева А. Путешествие в школу будущего	3
Батракова В. О., Веснин С. Н., Корепанова А. А., Лаптева Т. Д., Саврасов А. И. Элективный курс «Применение методов моделирования систем при работе с текстовой информацией»	3
Жумашева С. Завтра — уже сегодня	3
Итоги XXII Всероссийского конкурса цифровых изображений и фотографий ФОТО 1-2022	1
Итоги XXIII Всероссийского конкурса цифровых изображений и фотографий ФОТО 2-2022	4
Итоги Всероссийского конкурса «Школьная информатика: вчера, сегодня, завтра»	3
Климина Н. В. Виртуальная робототехника на базе робосимуляторов VEXcode VR и Open Roberta Lab	3
Курганова Н. А. Использование квадрокоптеров Tello EDU при изучении раздела «Основы алгоритмизации» школьного курса информатики	3
Перминов Н. Информатике быть!	3
Плаксин М. А. Игры-ходилки для изучения блок-схем	3
Шакуров А. Информационные технологии в школе будущего	3
Шевчук Е. Пусть меня научат!	3

КОНКУРСЫ

В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов:



Кристина Постникова,
студентка Барнаульского государственного педагогического колледжа имени Василия Константиновича Штильке, Россия



Рамзия Кадырова,
студентка Мамадышского политехнического колледжа, Республика Татарстан, Россия



В. А. Гаевская,
учитель информатики средней школы № 206, г. Минск, Беларусь



Е. В. Павлова,
учитель информатики Академической гимназии № 56, Санкт-Петербург, Россия



М. Е. Крекин,
учитель информатики школы № 1228 «Лефортово», г. Москва, Россия



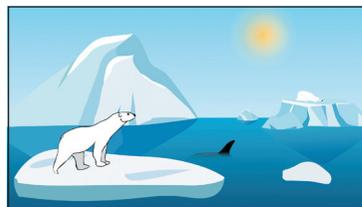
Вячеслав Баракин,
ученик лицея № 64, г. Омск, Россия



Леонид Черепенин,
ученик лицея № 6, г. Качканар, Свердловская область, Россия



Даниил Лбов,
ученик лицея № 6, г. Качканар, Свердловская область, Россия



Никита Рютин,
ученик Оренбургского президентского кадетского училища, Россия

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» в первом полугодии 2023 года.

Подробности — на сайте издательства «Образование и Информатика»: <http://www.infojournal.ru/>

ПОДПИСКА

Журнал «Информатика в школе»

Индекс подписки
на 1-е полугодие 2023 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: +7 (495) 140-19-86

ХІХ ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ПО МЕТОДИКЕ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ ИНФО-2022

НОМИНАЦИИ КОНКУРСА

1. Инновации в методике обучения информатике.
2. Технологии искусственного интеллекта в сфере образования.
3. Цифровизация образовательной организации и учебного процесса.
4. Развитие профессиональных компетенций педагогов в контексте цифровизации системы образования.

СРОКИ И ЭТАПЫ ПРОВЕДЕНИЯ КОНКУРСА

1. Работы на конкурс принимаются с 1 октября по 15 декабря 2022 года включительно.
2. Итоги конкурса будут подведены до 1 февраля 2023 года и опубликованы на сайте издательства «Образование и Информатика».
3. Лучшие работы будут опубликованы в журналах «Информатика и образование» и «Информатика в школе» в 2023 году.

Заявки на участие в конкурсе принимаются только через заполнение формы на сайте издательства «Образование и Информатика».

К участию в конкурсе 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/info-2022/>

31.01.2023

—

01.02.2023

XXIII МЕЖДУНАРОДНАЯ
НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

ОСНОВНЫЕ НАПРАВЛЕНИЯ РАБОТЫ:

- Перспективы развития технологий 1С для развития образования, мировых и отечественных практик цифровизации бизнеса.
- Экосистема 1С для поддержки педагогической деятельности в цифровой среде образовательной организации.
- Развитие форм и содержания сотрудничества «1С» с образовательными организациями. Участие представителей индустрии в системе профессионального образования.
- Применение передовых технологий и методик обучения при подготовке высококвалифицированных специалистов для экосистемы 1С.

В 2022 году в конференции приняли участие более 7500 человек.

Подробнее о тематиках конференции и условиях участия см. сайт educonf.1c.ru

МЕРОПРИЯТИЯ В РАМКАХ КОНФЕРЕНЦИИ:

- Пленарные и секционные заседания
- Мастер-классы по использованию программных продуктов фирмы «1С»
- Вернисаж программных и методических разработок

Традиционно 1 февраля проводится 1С:День студента. Это отличный шанс для студентов оценить свои силы и попасть на стажировку в ИТ-компанию!

Участие бесплатное для всех сотрудников образовательных организаций и органов управления образованием.

Обязательная предварительная регистрация открыта до 31 января 2023 года на сайте educonf.1c.ru

ФИРМА «1С»

Оргкомитет конференции:

Тел./факс: +7 (495) 688-90-02

Email: npk@1c.ru. Web: educonf.1c.ru

1С[®]
ФИРМА "1С"

