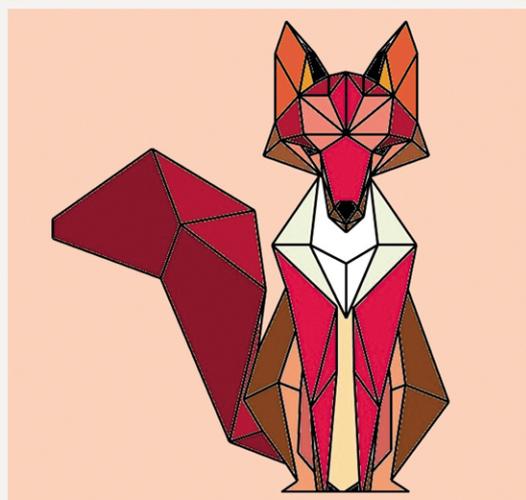
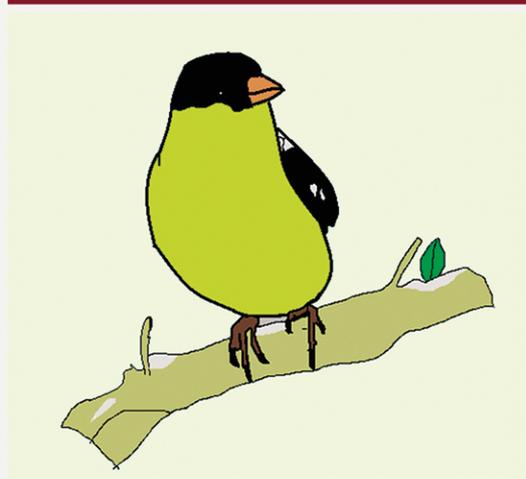
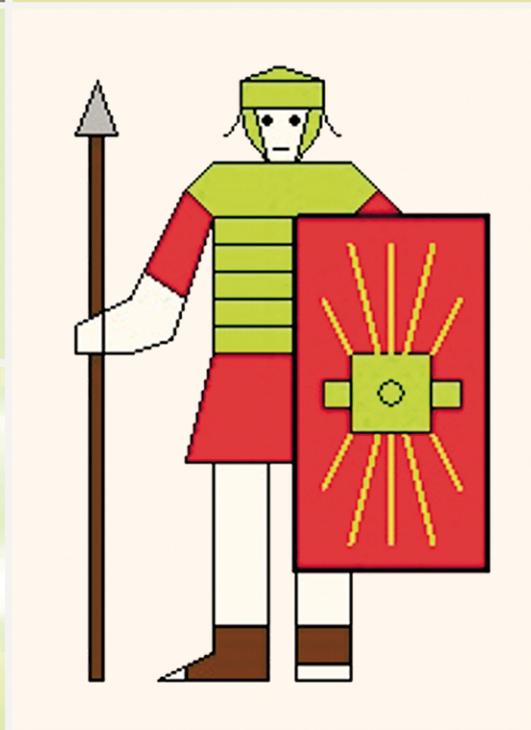


ИНФОРМАТИКА В ШКОЛЕ

№ 7'2018

ISSN 2221-1993

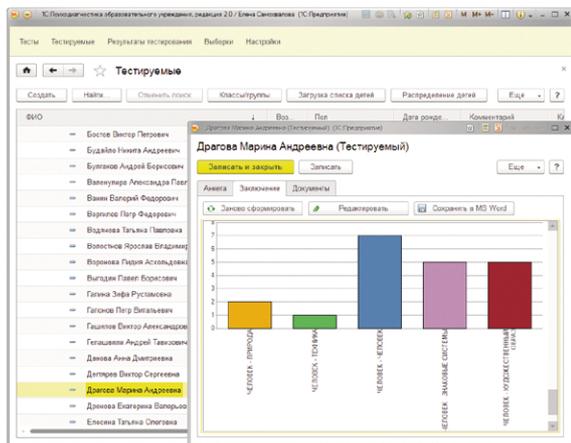
www.infojournal.ru



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.



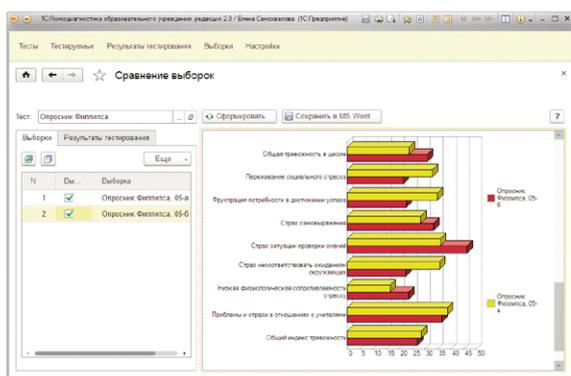
Функциональные возможности

- 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	Возраст	Возраст от	Возраст до	Время тестирования	Для младших групп
Спринг: жалоб ребенка	Адаптация в коллективе	5	14	15	
Спринг: Катетера. Подрос...	Общие	12	15	40	
Спринг: креативности Дюк...	Креативность	7	10	15	
Спринг: Стейнберга - За...	Общие	15	99	15	
Спринг: толерантности То...	Общие	3	7	15	✓
Спринг: толерантности	Толерантность	15	99	10	
Спринг: Томаса	Общие	14	99	15	
Спринг: Феликса	Адаптация в коллективе	7	17	10	
Спринг: Шварца	Мотивация	11	99	15	
Оценочное к сверстникам	Адаптация в коллективе	3	7	5	✓
Оценочное к взрослому	Адаптация в коллективе	3	7	5	✓
Оценка нерепетации	Адаптивное поведение	13	99	20	
ПДО	Общие	14	18	40	
Попытка	Исследования	4	7	20	
Пословицы	Мотивация	11	99	15	
Проба на зрелость	Мотивация	5	7	15	
Провальные агрессия	Адаптация в коллективе	3	7	5	✓
Психолог-педагогическая кв	Общие	3	10	5	✓
Расшифрование кружков	Общие	5	9	30	
Расшифрование	Плечевость в школе	5	7	15	
САН	Общие	7	10	5	

Блоки методик

- Профориентация.
- 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.

Редакционная коллегия

Баракина Т. В.
Бешенков С. А.
Босова Л. Л.
Воеводин Вл. В.
Дергачева Л. М.
Заславская О. Ю.
Захарова Т. Б.
Зенкина С. В.
Кириченко И. Б.
Кравцова А. Ю.
Кузнецов А. А.
Лаптев В. В.
Левченко И. В.
Рыбаков Д. С.
Слинкина И. Н.

Редакция

Босова Л. Л.
главный редактор
Кириченко И. Б.
*заместитель
главного редактора*
Дергачева Л. М.
научный редактор
Губкин В. А.
Коптева С. А.
Кузнецова Е. А.
Федотов Д. В.
Шарапова Л. М.

Адрес редакции:
119261, г. Москва,
Ленинский пр-т, д. 82/2, комн. 6

Почтовый адрес:
119270, г. Москва, а/я 15

Телефон/факс: (495) 140-19-86

E-mail: readinfo@infojournal.ru

URL: http://www.infojournal.ru

*Подписные индексы
в каталоге «Роспечать»:*
для индивидуальных подписчиков – 81407
для предприятий и организаций – 81408

Подписано в печать 27.09.2018.
Формат 60×90^{1/8}. Усл. печ. л. 8,0.
Тираж 2000 экз. Заказ № 541.

Отпечатано в типографии
ООО «Принт сервис групп»
105187, г. Москва, Борисовская ул.,
д. 14, стр. 6, тел./факс: (499) 785-05-18,
e-mail: 3565264@mail.ru

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

Все права защищены. Никакая часть журнала
не может быть воспроизведена в любой форме
или любыми средствами, электронными или
механическими, включая фотографирование,
сканирование, магнитную запись, размещение
в интернете или иные средства копирования
или сохранения информации, без письменного
разрешения издательства.

**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**

© «Образование и Информатика», 2018

Содержание

ОБЩИЕ ВОПРОСЫ

Зенько С. И. О проблеме классификации понятий информатики, изучаемых в средней школе..... 4

МЕТОДИЧЕСКАЯ КОПИЛКА

Слинкин Д. А. Использование языка программирования Free Pascal и программной библиотеки RubiRobotLib для управления роботами на платформе LEGO MINDSTORMS EV3 8

Маркушевич М. В. Использование отечественного программного обеспечения в учебном процессе общеобразовательной школы как фактор гражданско-патриотического воспитания учащихся 13

Комшина А., Телибаев С., Михлин Б. С. Сборка RS-триггера на микросхемах, содержащих элементы «ИЛИ-НЕ», «И-НЕ» 17

Зайка И. В. Практические работы по созданию прикладных программ в Visual C++ 26

СУПЕРКОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ШКОЛЕ

Тюшнякова И. А. Изучение сортировки подсчетом и слиянием в школьном курсе информатики 32

Лосик С. Н. Пропедевтика параллельных вычислений в начальной школе с использованием среды Kodu Game Lab 36

ЗАДАЧИ

Травкин И. Ю. Элементы функционального программирования на языке JavaScript: две задачи из теории вероятностей 40

СРЕДСТВА ИКТ В УЧЕБНОМ ПРОЦЕССЕ

Радионова А. В., Падчина Я. В. Применение 3D-технологий при создании тактильных книг для детей с нарушениями зрения 46

Глузман Н. А. Проектирование электронных образовательных ресурсов для обучения математике в начальной школе средствами Adobe Flash и HTML5 49

Королев А. Л. Компьютерное моделирование и программные комплексы поддержки изучения моделирования в школе 58

ЮБИЛЕЙНЫЙ

Уважаемые коллеги!

В начале 2019 года редакцию ИНФО и всех наших читателей ожидают сразу два знаменательных события: во-первых, выйдет в свет юбилейный, 300-й, выпуск журнала «Информатика и образование», и, во-вторых, в нем будут подведены итоги юбилейного, пятнадцатого, конкурса ИНФО.

Первый выпуск научно-методического журнала «Информатика и образование» вышел в свет в августе 1986 года. Создание журнала было вызвано требованием времени — учителям была остро необходима методическая поддержка при решении актуальных задач внедрения в педагогическую практику нового общеобразовательного курса «Основы информатики и вычислительной техники».

Первым главным редактором ИНФО стал Владимир Андреевич Мельников — академик РАН (в то время — АН СССР), в первых номерах журнала были опубликованы статьи академиков Евгения Павловича Велихова и Андрея Петровича Ершова. Участие академии в жизни журнала подчеркивало то значение, которое придавала научная общественность новому школьному предмету, — уже в те годы было понимание того, что информатика будет играть важнейшую роль в развитии образования в XXI веке.

Статьи ведущих ученых, стоявших у истоков школьной информатики, — Александра Андреевича Кузнецова, Михаила Павловича Лапчика, Вадима Семеновича Леднева, Вадима Макарьевича Монахова, Алексея Львовича Семенова, Александра Юрьевича Уварова — не только оказывали существенную помощь учителям в их повседневной практической работе, но и раскрывали значимость и перспективы использования информационных технологий в образовании.

На страницах журнала информатика всегда рассматривалась как основа информатизации, поэтому в нем всегда было два равноправных направления: методика преподавания информатики и информатизация образования. Журнал не только задавал ориентиры в развитии методики преподавания информатики, давал образцы грамотного построения уроков по этому предмету, но и предлагал материалы, в которых известные ученые-теоретики и специалисты-практики формировали и развивали теоретические основы и практические аспекты информатизации образования, использования средств информационно-коммуникационных технологий в образовательном процессе.

За время своего существования школьная информатика претерпела колоссальные изменения, проделав огромный путь от «курса компьютерной грамотности» до полноценного общеобразовательного учебного предмета. И на всей этой длинной дистанции журнал не только отражал те перемены, которые происходили в учебном предмете «Информатика», но и принимал непосредственное участие в его успешном становлении. Авторы программ и школьных учебников по информатике, разработчики электронных образовательных ресурсов, учителя-практики постепенно пополняли авторский коллектив журнала.

Огромное число учителей-новаторов стали авторами журнала благодаря конкурсу ИНФО, который издательство «Образование и Информатика» проводит с ноября 2003 года. Многие педагоги, однажды решив прислать свои работы на конкурс ИНФО, в дальнейшем становились его постоянными участниками, а некоторые — и неоднократными лауреатами и дипломантами. Среди имен авторов нынешних выпусков журналов «Информатика и образование» и «Информатика в школе» (который в 2002 году начал издаваться как приложение к журналу «Информатика и образование», а затем вырос в самостоятельное издание) можно встретить немало тех, кто в свое время становился победителем конкурса.

В настоящее время перед школьной информатикой стоят новые задачи, определяемые новыми направлениями развития, модернизации российской школы. Новые стандарты образования и существенное расширение представлений о требованиях к образовательным результатам, совершенствование ЕГЭ и процедур оценивания учебных достижений школьников, создание новой информационно-образовательной среды и организация сетевого взаимодействия участников образовательного процесса — все эти аспекты обновления находят отражение на страницах журналов «Информатика и образование» и «Информатика в школе».

Наши журналы — верные помощники не только для тех, кто занимается непосредственно школьной информатикой. Современная школа немыслима без информационно-образовательной среды, современное образование невозможно представить без информационно-коммуникационных технологий, поэтому к страницам наших изданий обращаются учителя самых разных предметов, руководители системы образования, разработчики новых средств ИКТ, создатели цифровых образовательных ресурсов.

Мы уверены, что все разнообразные направления развития образовательной системы XXI века найдут отражение в работах нового конкурса ИНФО, и приглашаем вас к участию в нем.

КОНКУРС ИНФО-2018

Издательство «Образование и Информатика»,
Всероссийское научно-методическое общество педагогов
объявляют о проведении в 2018 году юбилейного

XV Всероссийского конкурса научно-практических работ по методике обучения информатике и информатизации образования ИНФО-2018, посвященного выходу в свет 300-го номера журнала «Информатика и образование»

Руководит конкурсом **Организационный комитет** (далее — Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов Всероссийского научно-методического общества педагогов, членов редакционных коллегий 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.

Условия участия в конкурсе

1. **К участию в конкурсе могут быть представлены любые работы по методике обучения информатике и информатизации образования.**
2. Участником конкурса может стать любой человек, связанный с работой в системе образования.
3. Возраст участников не ограничен.
4. Участником конкурса может быть индивидуальный заявитель или группа авторов.
5. Участниками конкурса могут быть как граждане России, так и граждане других стран, приславшие свои материалы на русском языке.
6. Заявки на участие в конкурсе принимаются только через заполнение формы на сайте издательства «Образование и Информатика».
7. Форма участия в конкурсе — заочная.
8. **В дополнение к основному конкурсу** каждая работа может быть представлена автором для онлайн-голосования на сайте издательства «Образование и Информатика». Победители онлайн-голосования будут отмечены **специальными дипломами**.

Сроки и этапы проведения конкурса

1. **Работы на конкурс принимаются** с 1 июня по 1 ноября 2018 года включительно. Работы, присланные позже 1 ноября 2018 года, к участию в конкурсе допускаться не будут.
2. **Голосование на сайте** за работы, представленные для онлайн-голосования, будет проходить с 1 декабря 2018 года по 1 января 2019 года включительно.
3. **Итоги конкурса** будут подведены до 1 февраля 2019 года и опубликованы на сайте издательства «Образование и Информатика», а также в журналах «Информатика и образование» № 1-2019 и «Информатика в школе» № 1-2019.
4. **Лучшие работы** будут опубликованы в журналах «Информатика и образование» и «Информатика в школе».

Победители конкурса получат (бесплатно):

1. Диплом от Всероссийского научно-методического общества педагогов и издательства «Образование и Информатика».
2. Электронную подписку на журналы «Информатика и образование» и «Информатика в школе» на 2019 год.
3. По одному печатному экземпляру журналов «Информатика и образование» № 1-2019 и «Информатика в школе» № 1-2019, в которых будут опубликованы итоги конкурса.
4. Авторский печатный экземпляр журнала с опубликованной работой.

**Подробную информацию о конкурсе
вы можете найти на сайте ИНФО:**
<http://infojournal.ru/competition/info-2018/>

Контакты Оргкомитета:
Телефон: +7 (495) 140-1986
E-mail: readinfo@infojournal.ru
<http://www.infojournal.ru/>

С. И. Зенько,

Белорусский государственный педагогический университет имени Максима Танка, г. Минск, Республика Беларусь

О ПРОБЛЕМЕ КЛАССИФИКАЦИИ ПОНЯТИЙ ИНФОРМАТИКИ, ИЗУЧАЕМЫХ В СРЕДНЕЙ ШКОЛЕ

Аннотация

В статье поднимается проблема классификации понятий информатики, изучаемых в средней школе. От ее решения зависит эффективность построения методик обучения учащихся данным понятиям. Автором предлагается рассматривать классификации понятий школьной информатики с четырех позиций: по межпредметному основанию, по содержательным линиям предмета «Информатика», по логико-структурным взаимосвязям и взаимодействиям изучаемых понятий, по этимологии иноязычных и переводных слов в определениях понятий информатики. В результате применения первой классификации выделяются общеучебные и специальные понятия; второй классификации — межсодержательные и внутрисодержательные понятия; третьей классификации — стабильные (устойчивые), расширяющиеся, ключевые и вспомогательные понятия; четвертой классификации — понятия-существительные, понятия-глаголы, понятия-прилагательные и понятия — комбинации частей речи.

Ключевые слова: методика обучения информатике, классификация понятий, понятия школьной информатики, общеучебные понятия, специальные понятия, межсодержательные понятия, внутрисодержательные понятия, стабильные понятия, устойчивые понятия, расширяющиеся понятия, ключевые понятия, вспомогательные понятия, понятия-существительные, понятия-глаголы, понятия-прилагательные, понятия — комбинации частей речи.

DOI: 10.32517/2221-1993-2018-17-7-4-7

Контактная информация

Зенько Сергей Иванович, канд. пед. наук, доцент, доцент кафедры информатики и методики преподавания информатики Белорусского государственного педагогического университета имени Максима Танка, г. Минск, Республика Беларусь; *адрес:* 220030, Республика Беларусь, г. Минск, ул. Советская, д. 18 (корп. 2), БГПУ, физ.-мат. факультет, каб. 511; *телефон:* +375 (017) 327-77-83; *e-mail:* sergey.zenko@tut.by

S. I. Zenko,
Belarusian State Pedagogical University named after Maxim Tank, Minsk,
The Republic of Belarus

ABOUT THE PROBLEM OF CLASSIFICATION OF THE CONCEPTS OF INFORMATICS STUDIED AT SECONDARY SCHOOL

Abstract

The article raises the problem of classification of the concepts of computer science and informatics studied at secondary school. The efficiency of creation of techniques of training of pupils in these concepts depends on its solution. The author proposes to consider classifications of the concepts of school informatics from four positions: on the cross-subject basis, the content lines of the educational subject "Informatics", the logical and structural interrelations and interactions of the studied concepts, the etymology of foreign-language and translated words in the definition of the concepts of informatics. As a result of the first classification general and special concepts are allocated; the second classification — inter-content and intra-content concepts; the third classification — stable (steady), expanding, key and auxiliary concepts; the fourth classification — concepts-nouns, concepts-verbs, concepts-adjectives and concepts — combinations of parts of speech.

Keywords: methods of teaching informatics, technique of training in informatics, classification of concepts, concepts of school informatics, general educational concepts, special concepts, inter-content and intra-content concepts, stable concepts, steady concepts, expanding concepts, key concepts, auxiliary concepts, concepts-nouns, concepts-verbs, concepts-adjectives, concepts — combinations of parts of speech.

Содержание учебного предмета «Информатика» базируется на изучении основ соответствующей науки. Однако существенным отличием науки информатики от одноименного учебного предмета является то, что в последнем основное содержание науки представляется педагогически адаптированной системой знаний и умений (компетенций). От того, насколько обоснованно, правильно и удачно будет выстроена система понятий, изучаемых учащимися, зависит успешность подготовки обучающихся в целом.

В научных, педагогических и методических исследованиях отдельные аспекты классификации понятий информатики поднимались С. А. Бешенковым, Л. Л. Босовой, Г. А. Бордовским, В. А. Извозчиковым, Ю. В. Исаевым, Н. В. Матвеевой, Л. В. Милохиной, В. В. Морозовым и др.

В книге Г. А. Бордовского, В. А. Извозчикова, Ю. В. Исаева и В. В. Морозова «Информатика в понятиях и терминах» [7] систематизация понятий представлена в тематическом порядке:

- *общие понятия и термины информатики* («информатика», «информация», «информационное поле», «информационная сфера», «информационное общество», «компьютерная грамотность» и др.);
- *понятия и термины, характеризующие вычислительную технику* («программа», «кодирование», «компьютер», «процессор», «дисплей», «устройство ввода информации», «триггер», «быстродействие», «производительность» и др.);
- *понятия и термины, относящиеся к алгоритмическим и программным средствам информатики* («преобразование (обработка) информации», «алгоритмизация», «алгоритм», «эвристика», «свойства алгоритма», «операционная система», «компилятор», «интерпретатор»,



«программирование», «бит», «байт», «база знаний» и др.).

По результатам анализа работы можно прийти к выводу о том, что авторы указанной книги [7] выделяют **три группы понятий**:

- *понятия, соответствующие программе школьного курса информатики;*
- *понятия, ориентированные на курс повышенного уровня (факультатив);*
- *понятия, ориентированные на специальные курсы, читаемые в профильных классах программистов.*

В диссертационном исследовании С. А. Бешенкова «Развитие содержания обучения информатике в школе на основе понятий и методов формализации» рассматривается развитие содержания обучения информатике, базирующееся, в первую очередь, на имеющихся для того времени технических и программных возможностях, отмечается, что программирование можно рассматривать как «машинную» реализацию основных аспектов формализации. А именно: «Понятие данных связано с понятием знаковой системы, понятие алгоритма — с преобразованием знаковых систем, понятие программы — с проблемой коммуникации “человек-машина”» [1, с. 14]. Таким образом, автором выделяются **три группы понятий в соответствии с тремя видами информационной деятельности**:

- *сбор и хранение информации;*
- *обработка (преобразование) информации;*
- *интерпретация (получение) информации.*

«Первая группа понятий в минимальном объеме раскрывает основной тезис о разделении знакового и содержательного аспектов изучаемого объекта. К ним относятся: “предмет”, “имя”, “алфавит”, “порождающие правила”.

Вторая группа дает трактовку понятия “алгоритмического языка” с более общей, логической точки зрения. В этом смысле эта группа понятий не отличается от понятий перовой и представляет собой перенос основных понятий синтаксиса: “алфавита”, “порождающих правил” на предметную область, состоящую из предписаний.

Третья группа понятий вводит в рассмотрение “интерпретатора” и “исполнителя” и показывает возможность интерпретации (реализации) основных представлений о семантике и коммуникации» [1, с. 31].

В дальнейшем **С. А. Бешенков, Е. А. Ракитина, Н. В. Матвеева, Л. В. Милохина** в своей книге **«Непрерывный курс информатики»** [2] выделили в предметной области «Информатика» **два вида понятий: основные (базовые) и вспомогательные.**

К основным они относят понятия, на которых строится содержание непрерывного курса учебного предмета. Это, в первую очередь, понятия «информационные процессы», «информационное моделирование» и «информационные основы управления», а также производные от них: «данные» («закодированная информация»), «виды информации», «хранение информации», «обработка информации», «носитель информации», «источник информации» и т. д.

К вспомогательным исследователи относят общенаучные и межпредметные понятия, используемые для раскрытия сущности основных понятий информатики, например, «объект», «система» и т. д.

С точки зрения авторов [2], основные понятия предметной области «Информатика» определяют содержание

школьного предмета «Информатика» и основные виды информационной деятельности, которыми должны овладеть учащиеся.

Однако, анализируя представленное в работе [2] примерное распределение учебного материала для II—IX классов, убеждаемся, что вышеуказанное подразделение понятий явно не отражено, поскольку все они представлены в одном разделе под названием «Основные понятия». В частности, в VII классе это понятия:

- «задача», «решение задачи», «этапы решения задачи»;
- «объект», «информационный объект», «модель», «информационная модель», «моделирование», «формализация»;
- «алгоритм», «алгоритмическая конструкция», «алгоритмический язык»;
- «программа», «отладка», «тестирование программы», «компьютерный эксперимент».

Л. Л. Босова в диссертационном исследовании **«Развитие методической системы обучения информатике и информационным технологиям младших школьников»** на основе анализа и обобщения педагогического опыта среди целей изучения информатики и ИКТ в V—VII классах выделяет «пропедевтическое (предварительное, вводное, ознакомительное) изучение понятий основного курса школьной информатики, обеспечивающее целенаправленное формирование **общеучебных понятий**, таких как “объект”, “система”, “модель”, “алгоритм”» [3, с. 234]. Выделение такого вида понятий оправданно, но недостаточно для изучения информатики в старших классах.

Проанализированные выше исследования свидетельствуют о том, что наличие классификации понятий информатики необходимо. Однако можно констатировать, что целенаправленных исследований проблемы взаимосвязи классификации понятий информатики, изучаемых в школе, и рассматриваемых методик обучения данным понятиям нет. Это создает сложности для современного учителя информатики при попытках формирования осознанной целостной системы изучаемых понятий у учащихся.

Мы предлагаем рассматривать классификации понятий школьного предмета «Информатика» с четырех позиций:

- 1) *на межпредметной основе при изучении предмета «Информатика» и других школьных предметов;*
- 2) *на основе содержательных линий предмета «Информатика»;*
- 3) *на основе логико-структурных взаимосвязей и взаимодействий изучаемых понятий информатики;*
- 4) *на основе этимологии иноязычных и переводных слов в определении понятий информатики.*

1. Классификация понятий информатики, изучаемых в средней школе, на межпредметной основе.

Рассматривая понятия информатики с позиции межпредметности, мы выделяем **общеучебные** и **специальные понятия**.

Общеучебные понятия — это понятия, которые изучаются и используются для раскрытия сущности содержания предметов и явлений с различных сторон в рамках нескольких (от двух и более) учебных предметов, включая предмет «Информатика».

Это такие понятия, как: «алгоритм», «модель», «память», «архив», «запрос», «программа», «множества», «цикл» и др.

Специальные понятия — это понятия, которые изучаются, формируются и используются в рамках только предмета «Информатика» для раскрытия сущности содержания предметов и явлений.

К специальным понятиям предмета «Информатика» мы относим: «компьютер», «среда программирования», «процессор», «двоичный код», «байт», «принтер», «вспомогательный алгоритм» и др.

2. Классификация понятий информатики, изучаемых в средней школе, на основе содержательных линий предмета «Информатика».

При классификации понятий информатики на основе содержательных линий предмета «Информатика» нами выделяются *межсодержательные* и *внутрисодержательные* понятия.

В Республике Беларусь в курсе школьной информатики выделяются шесть содержательных линий:

- информация и информационные процессы;
- аппаратное и программное обеспечение компьютеров;
- основы алгоритмизации и программирования;
- компьютерные информационные технологии;
- коммуникационные технологии;
- информационное моделирование.

Межсодержательные понятия — это понятия, которые используются при рассмотрении учебного материала в нескольких (от двух и более) содержательных линиях предмета «Информатика» в школе. Примеры таких понятий: «информация», «данные», «алгоритм», «компьютер» и др.

Внутрисодержательные понятия — это понятия, которые используются при рассмотрении учебного материала только в одной содержательной линии предмета «Информатика» в школе. К внутрисодержательным понятиям относятся: «байт», «адаптер», «алгоритмическая конструкция “следование”», «среда программирования», «компьютерный видеомонтаж», «интернет», «сетевая модель» и др.

3. Классификация понятий информатики, изучаемых в средней школе, на основе логико-структурных взаимосвязей и взаимодействий изучаемых понятий информатики.

При классификации понятий предмета «Информатика» на основе логико-структурных взаимосвязей и взаимодействий изучаемых понятий нами выделяются *стабильные (устойчивые)* и *расширяющиеся* понятия, *ключевые (основные)* и *вспомогательные* понятия.

Стабильные (устойчивые) понятия — это понятия, которые с момента их введения используются в неизменном виде на протяжении всего времени изучения предмета «Информатика» в школе. Пример такого понятия — «информационный процесс». Различают следующие информационные процессы: хранение, передача, обработка и поиск информации.

Расширяющиеся понятия — это понятия, которые на протяжении изучения предмета «Информатика» в школе масштабируются (укрупняются, уточняются, развиваются) за счет рассмотрения компонентов нового учебного материала — новых объектов, процессов, свойств, типов данных, видов информации и т. д., при этом доступность для понимания понятия не снижается. Пример такого понятия — «процессор». Его уточнение и развитие осуществляются на протяжении учебной деятельности с шестого по десятый класс за счет рассмотрения того, для чего это устройство ис-

пользуется — «для обработки информации» (VI класс), «для обработки информации, выполнения вычислений» (VII класс), «для преобразования данных и управления другими устройствами компьютера» (X класс).

Ключевые (основные) понятия — это понятия, которые наиболее часто используются в содержании учебного материала при изучении различных тем предмета «Информатика» в школе.

Вспомогательные понятия — это понятия, которые вводятся с целью предоставления дополнительного уточнения характеристик, свойств, признаков для лучшего понимания отдельных стабильных и расширяющихся понятий.

В таблице представлены отдельные примеры понятий предмета «Информатика», иллюстрирующие взаимосвязи и взаимодействия, возникающие между стабильными (устойчивыми) понятиями и ключевыми (основными) и вспомогательными понятиями, а также расширяющимися понятиями и соответствующими им ключевыми (основными) и вспомогательными понятиями [9].

4. Классификация понятий информатики, изучаемых в средней школе, на основе этимологии иноязычных и переводных слов в определении понятий информатики.

Информатика содержит большое число иноязычных понятий. В связи с тем что компьютерные, цифровые, сетевые, мобильные, интернет- и др. технологии стремительно развиваются, количество таких понятий быстро возрастает. Поэтому для учителей информатики проблема обоснованной, равноценной и целесообразной терминологии, передающей сущность понятий с учетом иноязычности происхождения, специфики русского и белорусского языков, остается актуальной. На основании анализа значений, этимологии иноязычных слов и их переводных эквивалентов, систематизации аналогов русскоязычных понятий в английском языке и выявления сущности понятий предмета проведена дифференциация понятий на основе этимологии по частям речи: **понятия-существительные, понятия-глаголы, понятия-прилагательные и понятия — комбинации частей речи**. Оценка степени соответствия между частями речи понятий предмета «Информатика» и частями речи иноязычных и переводных эквивалентов позволяет вносить коррективы в методическую работу учителя при подготовке к обучению учащихся понятиям информатики.

Например, понятие курса информатики «буфер (обмена)» — существительное, а иноязычный эквивалент «buffer» происходит от глагола «buff» (англ.) — «смягчать удар». В VI классе понятие «буфер» вводится как «специальная область памяти компьютера для временного хранения информации при выполнении операций *Вырезать* и *Копировать*» [8].

* * *

Подготовка будущих учителей информатики к использованию вышерассмотренных классификаций при обучении учащихся на основе деятельностно-семантического подхода [4–6], как показал педагогический эксперимент во время практики студентов в школе, способствует формированию у учащихся осознанного восприятия изучаемых понятий и облегчает освоение программного материала. Это связано с тем, что в процессе подготовки к урокам студенты сознательно рассматривали понятия предмета «Информатика» с различных

Таблица

Примеры из классификации понятий информатики на основе их логико-структурных взаимосвязей и взаимодействий

Стабильные (устойчивые) понятия		Расширяющиеся понятия	
<i>Информационный процесс</i> — любая деятельность человека, связанная с информацией (VII класс)		1. <i>Процессор</i> — одно из самых важных устройств компьютера, расположенное в системном блоке и предназначенное для обработки информации (VI класс)	
Ключевые понятия	Вспомогательные понятия	Ключевые понятия	Вспомогательные понятия
Хранение информации	Оперативная память	Устройства компьютера	Системный блок
	Долговременная память	Информация	Обработка информации
	Компьютерные носители информации	2. <i>Процессор</i> — важнейшее устройство компьютера, его мозг. Он обрабатывает информацию, выполняя вычисления (VII класс)	
Передача информации	Источник информации	Ключевые понятия	Вспомогательные понятия
	Приемник информации	Устройства компьютера	—
	Канал связи	Информация	Обработка информации
Обработка информации	Редактирование информации	3. <i>Процессор</i> — устройство, обеспечивающее преобразование данных и управление другими устройствами компьютера (X класс)	
	Форматирование информации	Ключевые понятия	Вспомогательные понятия
Поиск информации		Данные	Преобразование данных
		Устройства компьютера	Управление устройствами

позиций и учитывали их специфические особенности с методической точки зрения.

Список использованных источников

1. *Бешенков С. А.* Развитие содержания обучения информатике в школе на основе понятий и методов формализации: автореф. дис. ... д-ра пед. наук: 13.00.02. М., 1994.

2. *Бешенков С. А., Ракина Е. А., Матвеева Н. В., Михолина Л. В.* Непрерывный курс информатики. М.: БИНОМ. Лаборатория знаний, 2015.

3. *Босова Л. Л.* Развитие методической системы обучения информатике и информационным технологиям младших школьников: дис. ... д-ра пед. наук: 13.00.02. М., 2010.

4. *Зенько С. И.* Деятельностно-семантический подход к профессиональной направленности формирования понятийной компетентности учителя информатики в педагогическом университете // *Вестні БДПУ. Серія 3 «Фізика. Математика. Інформатика. Біологія. Географія»*. 2018. № 2.

5. *Зенько С. И.* Деятельностно-семантический подход как условие повышения эффективности методической подготов-

ки будущего учителя информатики // *Вестні БДПУ. Серія 3 «Фізика. Математика. Інформатика. Біологія. Географія»*. 2017. № 4.

6. *Зенько С. И.* Методические аспекты подготовки будущего учителя информатики к работе с понятиями содержательной линии «Аппаратное и программное обеспечение компьютеров» на основе деятельностно-семантического подхода // *Вестник Полоцкого государственного университета. Серия Е «Педагогические науки»*. 2017. № 15.

7. *Информатика в понятиях и терминах: книга для учащихся старших классов средней школы / Г. А. Бордовский, В. А. Извозчиков, Ю. В. Исаев, В. В. Морозов; под ред. В. А. Извозчикова. М.: Просвещение, 1991.*

8. *Макарова Н. П., Лано А. И., Войтехович Е. Н.* Информатика: учебное пособие для 6 кл. учреждений общего среднего образования с русским языком обучения. Мн.: Народная асвета, 2018.

9. *Новик И., Зенько С.* К вопросу классификации понятий информатики в средней школе // *Aktualne i przyszle wyzwania w edukacji: monografia wieloautorstwa. Siedlce, 2018.*



Д. А. Слинкин,

Шадринский государственный педагогический университет, Курганская область

ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ FREE PASCAL И ПРОГРАММНОЙ БИБЛИОТЕКИ RUBIROBOTLIB ДЛЯ УПРАВЛЕНИЯ РОБОТАМИ НА ПЛАТФОРМЕ LEGO MINDSTORMS EV3

Аннотация

В статье рассматриваются робототехническая платформа LEGO MINDSTORMS EV3, достоинства и недостатки среды разработки LEGO MINDSTORMS Education EV3, возможности программирования робота с использованием альтернативных языков программирования и сторонних прошивок. Особое внимание уделяется авторскому проекту RubiRobot и библиотеке RubiRobotLib, позволяющей разрабатывать программы для LEGO MINDSTORMS EV3 на языке программирования Free Pascal.

Ключевые слова: LEGO MINDSTORMS EV3, образовательная робототехника, алгоритм, программирование, Free Pascal.

DOI: 10.32517/2221-1993-2018-17-7-8-12

Контактная информация

Слинкин Дмитрий Анатольевич, канд. пед. наук, доцент кафедры программирования и автоматизации бизнес-процессов Шадринского государственного педагогического университета, Курганская область; *адрес:* 641800, Курганская область, г. Шадринск, ул. К. Либкнехта, д. 3; *телефон:* (35253) 3-63-69; *e-mail:* xdsl@list.ru

D. A. Slinkin,
Shadrinsk State Pedagogical University,
Kurgan Region

USING THE FREE PASCAL PROGRAMMING LANGUAGE AND THE RUBIROBOTLIB SOFTWARE LIBRARY TO CONTROL ROBOTS ON THE LEGO MINDSTORMS EV3 PLATFORM

Abstract

The article examines the LEGO MINDSTORMS EV3 robotic platform, the advantages and disadvantages of LEGO MINDSTORMS Education EV3 development environment, the possibilities of programming a robot using alternative programming languages and third-party firmware. Particular attention is paid to the RubiRobot authoring project and the RubiRobotLib software library which allows developing programs for LEGO MINDSTORMS EV3 on the Free Pascal programming language.

Keywords: LEGO MINDSTORMS EV3, educational robotics, algorithm, programming, Free Pascal.

Робототехнические системы различных типов прочно вошли в образовательный процесс школ, ссузов и вузов. Предмет «Робототехника» и робототехнические кружки позволяют изучить конструирование и программирование роботов, оценить их возможности и практическую применимость алгоритмических конструкций, использовать роботов в решении прикладных задач. Широко распространены российские и международные робототехнические соревнования: World Robot Olympiad [2, 26], «РобоФинист» [7], «Робофест» [3], соревнования JuniorSkills [18] в компетенции «Мобильная робототехника» и многие другие.

Одним из лидеров — производителей образовательных робототехнических систем заслуженно считается компания LEGO, особенностью работы которой является комплексный подход к продвижению своей продукции — это не только производство и продажа конструкторов, но и целая образовательная экосистема: курсы для учителей, печатная и электронная образовательная продукция, организация и спонсирование робототехнических олимпиад.

В последние годы широко распространение в России получила робототехническая платформа **LEGO MINDSTORMS Education EV3** [8] с сопутствующим программным обеспечением. «Сердцем» платформы является программируемый блок-микрокомпьютер, на порядок превосходящий по своим характеристикам блок предыдущей серии LEGO MINDSTORMS NXT. Однако подход к программированию блока особо не изменился, LEGO продолжает продвигать графический язык программирования, теперь уже в новой среде разработки LEGO MINDSTORMS Education EV3, которая поддерживает работу и с предыдущей платформой. В помощь учителям, школьникам и энтузиастам разработано множество самоучителей, учебников и методических рекомендаций, предназначенных для изучения самой платформы [1, 9, 11], обучения основным алгоритмам [1, 10] и программированию в среде разработки [1, 10, 11].

Среда разработки от компании LEGO имеет низкий порог вхождения, может использоваться школьниками и студентами всех ступеней обучения, однако, обладает и рядом существенных **недостатков:**

- Создание сложных и объемных программ в данной среде крайне затруднено, уже при десятке собственных блоков резко увеличивается латентность отклика на любое действие, загрузка больших программ в память среды может длиться до нескольких минут, переключение между проектами или программами в одном проекте происходит с длительными задержками. Указанная проблема воспроизводилась нами на нескольких де-

сятках компьютеров, на операционных системах Microsoft Windows XP, 7, 8.1, 10, с различными блоками EV3 и разными версиями среды разработки. При создании простых алгоритмов проблема, естественно, не проявляется, но при решении серьезных задач, например, во время подготовки к робототехническим соревнованиям и во время них, работа крайне замедляется.

- В среде невозможна разработка некоторых видов алгоритмов, например, с использованием прямой или косвенной рекурсии. Данный факт является прямым препятствием к эффективному использованию среды в робототехнических соревнованиях.
- Данное программное обеспечение не функционирует ни на одной операционной системе российского производства из внесенных в Единый реестр российских программ для электронных вычислительных машин и баз данных (<https://reestr.minsvyaz.ru/>). Таким образом, с учетом реализуемой в России программы импортозамещения возможность дальнейшего применения среды в государственных образовательных учреждениях находится под вопросом.

Существует возможность заменить среду разработки LEGO MINDSTORMS Education EV3 на другие среды и языки программирования, как функционирующие на оригинальной прошивке LEGO EV3, так и требующие для корректной работы загрузки сторонних прошивок и операционных систем. Специфика контроллера EV3 позволяет осуществлять загрузку с MicroSD-карты, оставляя оригинальную прошивку без изменений, что, в свою очередь, позволяет вернуться к штатной загрузке при возникновении каких-либо ошибок.

Среди сторонних ОС следует выделить две:

- leJOS [20], представляющую собой минимальный дистрибутив Linux с запущенной виртуальной машиной Java,
- ev3dev [17] — полноценный дистрибутив Linux Debian с возможностью использования любых языков программирования, доступных для ОС Linux.

Существует множество сред и языков программирования для EV3 — как текстовых, так и графических. Среди графических следует выделить TRIK Studio [25], которая кроме собственного контроллера TRIK поддерживает и контроллер LEGO EV3, является свободным программным обеспечением и функционирует в большинстве операционных систем. Среди учебных сред стоит обратить внимание на Scratch [12] и Open Roberta [23], среди коммерческих профессиональных — на LabVIEW [19] и его адаптированный для образования клон ROBOLAB [22].

Для текстовых языков программирования важным является наличие программных библиотек поддержки аппаратных подсистем и подключаемых устройств LEGO EV3. Указанными качествами обладают: Java проекта leJOS, EV3 Basic [16], такие языки проекта ev3dev, как Java, Python, Go, C, C++ и многие другие.

В то же время в РФ для обучения программированию в большинстве школ используются разные диалекты языка программирования Pascal (Delphi [15], PascalABC [14], Free Pascal [21] и т. п.). До настоящего времени не существовало средств разработки на языке Pascal для EV3 (за исключением экспериментальной генерации кода на Pascal в TRIK Studio), что затрудняло приме-

ние школьниками полученных на уроках информатики знаний и умений для программирования робототехнических систем.

Указанную проблему решают **проект RubiRobot** [13] и свободно распространяемая **программная библиотека RubiRobotLib**, доступная по адресу: <http://rubirobot.ru/files/src/RubiRobotLib.tar.gz>.

Библиотека RubiRobotLib разработана на языке программирования Free Pascal и представляет собой программный интерфейс в виде набора функций, классов и объектов с поддержкой всех внутренних и подключаемых устройств LEGO EV3. RubiRobotLib требует загрузки робота с оригинального или модифицированного в рамках проекта RubiRobot дистрибутива Linux Debian проекта ev3dev. Оригинальный дистрибутив доступен на сайте проекта ev3dev по адресу: <https://www.ev3dev.org/downloads>, модифицированный — на сайте проекта RubiRobot по адресу: <http://rubirobot.ru/files/img/ev3dev-rubirobot.img.bz2>. Библиотека содержит подробную документацию [6] в формате PDF/A, несколько десятков примеров различной степени сложности, консольный менеджер, позволяющий ускорить и упростить передачу данных между хостовым компьютером и роботом.

Подготовка текста программы с использованием библиотеки и ее компиляция могут производиться непосредственно на LEGO EV3 в ssh-сессии без установки специализированного ПО на персональный компьютер. Для подключения к роботу по протоколу ssh можно использовать произвольный ssh-клиент, который в ОС российского производства присутствует обычно изначально, а в ОС Windows может быть установлен дополнительно [24]. Запуск полученного в результате компиляции исполняемого файла может также производиться в ssh-сессии либо с помощью управляющего LEGO EV3 приложения Brickman. В первом случае штатные средства ввода-вывода Free Pascal будут обмениваться с пользователем данными в рамках терминала ssh-соединения. Во втором — взаимодействовать с пользователем через дисплей LEGO EV3, кнопки блока и клавиатуру (в случае ее подключения к USB-порту контроллера). С учетом невысокой (по сравнению с персональным компьютером) производительности LEGO EV3 более эффективным вариантом является создание программ с использованием средств кросскомпиляции, что дает значительный выигрыш в скорости разработки. Пакет программ RubiRobotCross проекта RubiRobot, доступный по адресу: <http://rubirobot.ru/files/img/setupRubiRobot.tgz>, содержит в себе все необходимое для организации кросскомпиляции и предназначен для использования на российских операционных системах Alt Linux, седьмой [4] и восьмой [5] платформах. В частности, работоспособность RubiRobotCross была протестирована на дистрибутивах Simply Linux 7.0, Альт Линукс 7.0 Кентавр, Альт Образование 8.2 и некоторых других. Ведутся работы по модификации RubiRobotCross для установки на другие дистрибутивы Linux, такие как Debian, Arch, Calculate. С учетом реализуемой в России программы импортозамещения модификация RubiRobotCross для ОС Windows признана на текущий день автором проекта нецелесообразной. Но все компоненты проекта выпускаются под лицензией LGPLv3, что позволяет портировать RubiRobotCross на ОС Windows любому стороннему разработчику.

Рассмотрим некоторые возможности управления устройствами контроллера LEGO EV3 с использованием библиотеки RubiRobotLib. Предполагаем, что разработка программ ведется в ssh-сессии непосредственно на контроллере LEGO EV3 загруженного с модифицированного в рамках проекта RubiRobot дистрибутива Linux Debian проекта ev3dev. Все нижеприведенные примеры входят в состав библиотеки RubiRobotLib, их исходный код может быть получен совместно с библиотекой. В качестве текстового редактора предполагаем использование встроенного редактора файлового менеджера Midnight Commander. В то же время исключительно для удобства визуализации листинги программ в данной статье будут представлены в виде скриншотов из среды программирования Geany.

1. Светоиндикация.

Блок EV3 имеет красный и зеленый светоиндикаторы слева от центральной кнопки и аналогичную пару — справа от центральной кнопки. Яркость светоиндикаторов определяется числовым значением от 0 до 255. При одновременном свечении красного и зеленого индикаторов возникает эффект смешения цветов, что позволяет получать отдельные оттенки желтого и оранжевого. Данный эффект демонстрирует программа, представленная в листинге 1.

```
led0.pp x
1 {$mode objfpc}
2 uses uev3, rubiroLeds;
3 var i:integer;
4 begin
5   ev3Init();
6   for i:=0 to 255 do ev3Leds.all(255-i,i,i,255-i);
7   for i:=0 to 255 do ev3Leds.all(i,255-i,255-i,i);
8 end.
```

Листинг 1. Программа демонстрации светоиндикации

Программа пригодна для запуска как в ssh-сессии, так и через Brickman. Рассмотрим ее более подробно.

Строка 1 отключает установленный по умолчанию режим поддержки синтаксиса Турбо Паскаля (tp) и включает режим поддержки расширенного синтаксиса Free Pascal с объектными возможностями (objfpc). Если не указать данный режим, то в более сложных программах возможно появление различных видов ошибок — от синтаксических при компиляции до логических при исполнении, так как библиотека RubiRobotLib разработана с использованием возможностей режима objfpc.

В строке 2 программа подключает необходимые для работы модули. Наличие базового модуля uev3 является обязательным при работе программы на LEGO EV3, он содержит необходимые средства инициализации библиотеки RubiRobotLib. Модуль rubiroLeds содержит объект ev3Leds для управления светоиндикацией контроллера.

В строке 5 процедура ev3init() инициализирует основные объекты библиотеки. Ее вызов должен быть осуществлен первым действием в программе. До этого момента попытка обращения к любым возможностям библиотеки может завершиться ошибкой и аварийным остановом программы. Исключением из этого правила является набор специальных переменных, предназначенных для тонкой настройки возможностей библиотеки и дающих необходимый эффект только при изменении перед вызовом ev3Init(). Например, чтобы «форсировать» чтение данных с устройств, достаточно до вызова ev3Init() установить в 0 значение переменной getInterval из модуля uev3sysfs.

В строке 6 первый цикл программы обеспечивает «переливание» цветов: левый ярко-красный индикатор плавно изменяет цвет на ярко-зеленый, правый ярко-зеленый индикатор плавно изменяет цвет на ярко-красный.

В строке 7 второй цикл проделывает обратную операцию.

Стоит обратить внимание, что подобного эффекта цветоиндикации невозможно добиться при использовании среды программирования LEGO MINDSTORMS Education EV3, так как в ней невозможно управлять уровнями яркости светоиндикаторов — они могут находиться только в одном из двух состояний: либо выключенном, либо включенном на максимальной яркости.

2. Робот-путешественник.

В этом примере робот с закрепленным впереди датчиком расстояния (рис. 1) путешествует по открытому пространству с набором препятствий. При обнаружении препятствия в пределах 50 сантиметров робот случайным образом изменяет направление движения и продолжает свое путешествие.



Рис. 1. Робот-путешественник

Обязательными требованиями к роботу является наличие двух больших моторов (EV3 или NXT) и ультразвукового датчика расстояния (EV3). Левый мотор должен быть подключен к порту с меньшим номером, правый — к порту с большим номером (например: А и В, С и D, В и D и т. д.). Датчик расстояния может быть подключен к любому порту.

Программа управления роботом представлена в листинге 2.

```
sns7_1.pp x
1 {$mode objfpc}
2 uses uev3, rubiroSensors, rubiroMotors, rubiroButtons;
3 begin
4   ev3Init();
5   ev3buttons.waitCenter;
6   while not ev3buttons.back do begin
7     if ev3Ultra.lenCM>50 then
8       ev3Rule.Run(0,20)
9     else
10      ev3Rule.RunTurnWait(100-200*random(2),
11                          0.2+random(2,15));
12   end;
end.
```

Листинг 2. Программа управления роботом-путешественником

Как и в предыдущем случае, программа пригодна для запуска и в ssh-сессии, и через Brickman. Рассмотрим ее более подробно.

В строке 2 последовательно подключаются базовый модуль, модуль поддержки датчиков, модуль поддержки двигателей и модуль поддержки кнопок блока EV3.

Инициализирует библиотеку строка 4.

В строке 5 робот блокируется вплоть до нажатия кнопки CENTER блока, сигнализируя об ожидании миганием цветоиндикаторов.

В строке 6 начинается цикл, выход из которого предусмотрен по нажатию на кнопку BACK блока. Метод `ev3buttons.back` является асинхронным, не блокирующим выполнение программы. Он возвращает текущее состояние соответствующей кнопки.

Строки 7–10 цикла представляют собой ветвление, регламентирующее поведение робота. Если расстояние до препятствия больше 50 сантиметров, то робот продолжает движение вперед вызовом асинхронного метода `Run` объекта рулевого управления со скоростью в 20 % от максимальной. В противном случае робот осуществляет поворот вокруг оси синхронным методом `RunTurnWait` влево или вправо от 0,2 до 0,7 оборота со скоростью 15 % от максимальной.

3. Дистанционное управление роботом с помощью инфракрасных датчика и маяка.

Конструкция робота с дистанционным управлением (рис. 2) отличается от робота-путешественника заменой датчика расстояния на инфракрасный датчик, который будет использоваться для управления роботом совместно с инфракрасным маяком. Предполагается использование первого канала маяка, красные кнопки которого отвечают за скорость правого мотора, синие — за скорость левого мотора. В простейшем случае инфракрасный датчик можно установить на те же самые крепления, которые использовались датчиком расстояния (как показано на рисунке 2). Однако для увеличения радиуса приема сигнала рекомендуется вынести инфракрасный датчик на верхнюю часть робота.



Рис. 2. Робот с дистанционным управлением

Программу можно запускать и в `ssh`-сессии, и через `Brickman`. Рассмотрим программу управления роботом (листинг 3), опуская очевидные моменты.

В строке 4 определяются два объекта класса `TEv3LargeMotor`, которые затем в строке 8 ассоциируются с левым и правым моторами EV3.

В строке 5 определяется переменная — множество типа `TEv3IRButtons`, которая будет содержать набор кнопок, нажатых на инфракрасном маяке.

Строки 9–17 представляют собой основной цикл программы, выход из которого производится по нажатию на кнопку BACK блока.

В строке 10 из объекта инфракрасного датчика считывается множество нажатых в настоящий момент на маяке кнопок.

```

1  {$mode objfpc}
2  uses uev3,rubiroSensors,rubiroMotors,rubiroButtons;
3  var
4    LM,RM:TEv3LargeMotor;
5    s:TEv3IRButtons;
6  begin
7    ev3Init();
8    LM:=ev3motor1; RM:=ev3motor2;
9    while not ev3buttons.back do begin
10   s:=ev3infra.anybuttons;
11   if [irRU,irRD]*s=[] then RM.stop;
12   if [irBU,irBD]*s=[] then LM.stop;
13   if irRU in s then RM.Run(100);
14   if irRD in s then RM.Run(-100);
15   if irBU in s then LM.Run(100);
16   if irBD in s then LM.Run(-100);
17 end;
18 end.

```

Листинг 3. Программа дистанционного управления роботом

В строках 11 и 12 при отпущенных кнопках управления выполняется остановка.

Строки 13–16 используются для организации прямого или обратного движения каждого из моторов в зависимости от нажатых кнопок блока. Положительное значение скорости дает движение вперед, отрицательное — назад.

4. Дистанционное управление роботом с помощью клавиатуры.

Если доступ к роботу организован через беспроводную сеть и программа запускается в `ssh`-сессии, то появляется возможность управления движением робота с клавиатуры (листинг 4). Конечно, клавиатурное управление возможно и при других способах доступа к EV3, но проводное подключение через USB ограничивает движение робота длиной USB-провода, а запуск программы с использованием `Brickman` требует подключения клавиатуры непосредственно к роботу, что ограничивает перемещение робота теперь уже длиной провода клавиатуры. Единственным приемлемым вариантом использования `Brickman` будет являться подключение беспроводной клавиатуры.

Конструкция робота аналогична любой из вышеописанных, наличие или отсутствие датчиков значения не имеет.

```

1  {$mode objfpc}
2  uses crt,uev3,rubiroMotors,sysutils;
3
4  var ch:char; command:char=#0; speed:integer=20;
5  begin
6    writeln('Робот с ручным управлением');
7    writeln('Клавиша ESC - завершение работы программы');
8    writeln('Клавиши курсора ↑,←,→,↓ - движение робота');
9    writeln('Клавиши от 1 до 9 - скорость робота');
10   writeln('Остальные клавиши - остановка робота');
11   ev3init();
12   repeat
13     ch:=readkey;
14     if ch in ['1'..'9'] then speed:=10*strToInt(ch)
15     else if ch=#0 then command:=readkey
16     else command:=#0;
17     case command of
18       #72:ev3rule.Run(0,speed);
19       #80:ev3rule.Run(0,-speed);
20       #75:ev3rule.Run(-100,speed);
21       #77:ev3rule.Run(100,speed)
22     else ev3rule.Stop();
23     end;
24   until ch=#27;
25 end.

```

Листинг 4. Программа управления роботом с помощью клавиатуры

Для получения информации о нажатых клавишах можно использовать разнообразные средства. Например, Free Pascal предлагает использовать кроссплатформенный модуль `keyboard` с поддержкой Unicode, функциональных клавиш, комбинаций клавиш и т. д. Однако в нашем случае возможности данного модуля останутся в основном невостребованными. Поэтому предлагаем для указанной задачи использовать модуль `ctrl`, портированный во Free Pascal из Turbo Pascal и остающийся неизменным уже много лет. Рассмотрим подробно код программы, опуская очевидные моменты.

В строке 4 определены три основные переменные: `ch` — код нажатой клавиши, `command` — сканкод нажатой клавиши/команда роботу, `speed` — скорость движения робота.

Строки 12–24 содержат основной цикл получения/обработки нажатий клавиш и реакций на нажатия. Выход из цикла — по нажатию на клавишу ESC (код 27).

Строки 13–16 используются для чтения значений нажатых клавиш, формирования команды роботу и установки скорости его движения.

В строках 17–23 применяется сформированная ранее команда. В зависимости от команды обеспечивается старт моторов вперед или назад с заданной скоростью, вращение робота вокруг оси влево или вправо. Если команда не соответствует сканкодам клавиш курсора, производится остановка робота.

* * *

Представленная программная библиотека может применяться при изучении программирования на уроках информатики, в программировании роботов на предмете «Робототехника», на занятиях в робототехнических кружках, при подготовке к робототехническим соревнованиям, на любых робототехнических соревнованиях, где отсутствуют ограничения на используемое программное обеспечение, а также при самостоятельном программировании LEGO EV3. Автор библиотеки не ограничивает возможностей ее модификации, разрешает ее распространение, изучение и применение в любых целях в соответствии с лицензией LGPLv3. Проект RubiRobot динамично развивается, ведутся работы по расширению его возможностей. Автор приглашает всех заинтересованных лиц к использованию библиотеки и участию в развитии данного проекта.

Список использованных источников

1. Валк Л. Большая книга LEGO MINDSTORMS EV3: подробное руководство для начинающих по постройке и программированию роботов / пер. с англ. С. В. Черникова. М.: Издательство «Э», 2017.

2. Всероссийская робототехническая олимпиада. <http://robolymp.ru/>

3. Всероссийский робототехнический фестиваль «Робофест». <http://www.russianrobofest.ru/>

4. Дистрибутивы и репозиторий 7-й платформы Alt Linux // Файловый архив Alt Linux. <http://ftp.altlinux.ru/pub/distributions/ALTLinux/p7/>

5. Дистрибутивы и репозиторий 8-й платформы Alt Linux // Файловый архив Alt Linux. <http://ftp.altlinux.ru/pub/distributions/ALTLinux/p8/>

6. Документация проекта RubiRobot // Проект RubiRobot (о роботах и не только). <http://rubirobot.ru/files/src/rubiro.manual.archive.pdf>

7. Ежегодный Международный фестиваль робототехники «РобоФинист». <https://robofinist.ru/>

8. Знакомство с EV3 // LEGO.com RU — Вдохновлять и учить строителей будущего. <https://www.lego.com/ru-ru/mindstorms/about-ev3>

9. Исогава Й. Книга идей LEGO MINDSTORMS EV3. 181 удивительный механизм и устройство / пер. с англ. О. В. Обручева. М.: Издательство «Э», 2017.

10. Овсяницкая Л. Ю., Овсяницкий Д. Н., Овсяницкий А. Д. Алгоритмы и программы движения по линии робота LEGO MINDSTORMS. М.: Перо, 2015.

11. Овсяницкая Л. Ю., Овсяницкий Д. Н., Овсяницкий А. Д. Курс программирования робота LEGO MINDSTORMS EV3 в среде EV3: основные подходы, практические примеры, секреты мастерства. Челябинск: ИП Мякотин И. В., 2014.

12. Программирование робота LEGO MINDSTORMS EV3 с помощью Scratch 2.0 // ПроХаус. <http://www.proghouse.ru/article-box/52-scratch2-ev3>

13. Проект RubiRobot (о роботах и не только). <http://rubirobot.ru>

14. Язык программирования PascalABC.NET. <http://pascalabc.net>

15. Delphi — Обзор // Fast Cross-Platform App Development Software — Embarcadero. <https://www.embarcadero.com/ru/products/delphi>

16. EV3 Basic // The official site for tutorials on EV3 Basic (Microsoft Small Basic with the EV3 extension). <https://sites.google.com/site/ev3basic>

17. ev3dev is your EV3 re-imagined. <http://www.ev3dev.org>

18. JuniorSkills — программа ранней профессиональной подготовки и профориентации школьников. <https://juniorskills.ru/>

19. LabVIEW — программное обеспечение для системного проектирования // National Instruments. <http://www.ni.com/ru-ru/shop/labview.html>

20. leJOS. Java for LEGO MINDSTORMS. <http://www.lejos.org>

21. Open source compiler for Pascal and Object Pascal. <http://freepascal.org>

22. Platform ROBO LAB // LEGO Engineering, Center for Engineering Education and Outreach. <http://www.legoengineering.com/platform/robolab>

23. Programming environment Open Roberta Lab. <https://lab.open-roberta.org>

24. PuTTY is an SSH and telnet client. <http://www.putty.org/>

25. TRIK. Learn to invent the future // ООО «КиберТех». <http://blog.trikset.com>

26. World Robot Olympiad. <https://wro-association.org/>

М. В. Маркушевич,
школа № 1352, г. Москва

ИСПОЛЬЗОВАНИЕ ОТЕЧЕСТВЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УЧЕБНОМ ПРОЦЕССЕ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЫ КАК ФАКТОР ГРАЖДАНСКО-ПАТРИОТИЧЕСКОГО ВОСПИТАНИЯ УЧАЩИХСЯ

Аннотация

В статье рассматривается возможность гражданско-патриотического воспитания учащихся общеобразовательных российских школ через процесс миграции учебно-воспитательного процесса на отечественное программное обеспечение. Также уточняются некоторые недостатки и риски применения иностранного проприетарного программного обеспечения в образовательных организациях Российской Федерации, в том числе конкретизируются виды потенциальных угроз информационной инфраструктуре нашей страны, возникающих в связи с применением иностранного программного обеспечения.

Ключевые слова: гражданско-патриотическое воспитание, отечественное программное обеспечение, свободное программное обеспечение, импортозамещение.

DOI: 10.32517/2221-1993-2018-17-7-13-16

Контактная информация

Маркушевич Михаил Владимирович,
учитель информатики школы № 1352, г. Москва; адрес: 107497, г. Москва, ул. Байкальская, д. 44; телефон: (495) 467-77-55; e-mail: mihael11@yandex.ru

M. V. Markushevich,
School 1352, Moscow

USING DOMESTIC SOFTWARE IN THE EDUCATIONAL PROCESS OF A COMPREHENSIVE SCHOOL AS A FACTOR OF CIVIL-PATRIOTIC EDUCATION OF STUDENTS

Abstract

The possibility of civil-patriotic education of pupils of comprehensive Russian school through the process of migration of the educational process to the domestic software is considered in the article. Also some shortcomings and risks of using foreign proprietary software in educational organizations of the Russian Federation are specified, including the types of potential threats to the information infrastructure of our country arising from the use of foreign software.

Keywords: civil-patriotic education, domestic software, free software, freeware, import substitution.

В настоящее время в России гражданско-патриотическому воспитанию подрастающего поколения уделяется особое внимание, оно, безусловно, во многом определяет духовно-нравственное становление школьников. Более того, надо отметить, что Концепция духовно-нравственного развития и воспитания личности гражданина России [6] является методологической основой разработки и реализации Федеральных государственных образовательных стандартов общего образования.

Согласно Концепции, в сфере общественных отношений духовно-нравственное развитие и воспитание обучающихся должно обеспечить в том числе:

- осознание себя гражданином России на основе принятия общих национальных нравственных ценностей;
- готовность граждан солидарно противостоять внешним и внутренним вызовам;
- развитость чувства патриотизма и гражданской солидарности;
- заботу о благосостоянии многонационального народа Российской Федерации, поддержание межэтнического мира и согласия;
- законопослушность и сознательно поддерживаемый гражданами правопорядок.

В качестве одного из возможных способов патриотического воспитания учащихся, обеспечивающего реализацию практически всех вышеуказанных задач, можно предложить применение отечественного программного обеспечения в учебном процессе общеобразовательной российской школы. К сожалению, можно констатировать, что большая часть используемого сегодня в школе программного обеспечения относится к импортному проприетарному ПО, что, с точки зрения автора, негативно влияет на духовно-нравственную сферу школьников. В статье [17] мы подробно рассмотрели воздействие, которое оказывает на духовно-нравственное воспитание учащихся начальной школы миграция учебного процесса на свободное программное обеспечение. В данной статье предлагается изучить, в свою очередь, влияние применения отечественного ПО на такой элемент духовно-нравственного становления, как гражданско-патриотическое воспитание.

Отрицательное влияние применения импортного ПО заключается в том, что учащиеся (особенно младшие школьники), вынужденные в процессе учебы постоянно использовать закрытое проприетарное программное обеспечение иностранного производства, в духовно-нравственной сфере получают некую скрытую деформацию, или повреждение, связанное с навязанным им искаженным стереотипом действий. Причем данное повреждение имеет двойную природу:

- детей приучают к использованию платного программного обеспечения, формируют из них послушных потребителей продукции транснациональных корпораций по производству проприетарного ПО;
- в детском сознании прочно утверждается идея о том, что все используемые ими товары, в том числе программное обеспечение, должны быть именно иностранного производства и никак иначе.

Тенденция к преимущественному использованию импортного проприетарного ПО в российских школах обусловлена, с точки зрения автора, следующими причинами:

- Существует фактор инерции у педагогов в плане использования привычного для них типа программного обеспечения. Данная инерция возникает в результате того, что большинство преподавателей за свою многолетнюю педагогическую практику выработали и отработали сценарии уроков по основным темам курса информатики, которые стали для них привычными и удобными.
- Ранее, до прецедента с директором школы в Пензенской области А. М. Поносовым [12], имевшего место в 2006 году, вопросам соблюдения авторских и смежных прав в области программного обеспечения школьными преподавателями, к сожалению, уделялось недостаточно внимания.
- Вышеупомянутый прецедент привел в том числе к принятию Распоряжения Правительства Российской Федерации № 1447-р от 18.10.2007 года «О лицензионной поддержке программного обеспечения, используемого в общеобразовательных учреждениях (с изменениями на 10 марта 2009 года)» [13], в рамках которого для нужд образовательных учреждений была закуплена лицензия на импортное программное обеспечение на срок до 31 декабря 2010 года [8]. Далее данная лицензия несколько раз продлевалась. Только после этого начался процесс внедрения отечественного программного обеспечения в виде пакета свободного программного обеспечения (ПСПО), содержащего различные дистрибутивы операционной системы Alt Linux.
- Большинство школьных преподавателей используют проприетарное импортное программное обеспечение на домашних компьютерах, так как приобретают персональные компьютеры с предустановленной операционной системой Windows и офисным пакетом Microsoft Office и данный опыт переносят на учебный процесс.
- Технология применения проприетарного импортного программного обеспечения чаще проще, чем отечественного ПО, основанного на свободной операционной системе семейства Linux, она требует меньшей квалификации и ИКТ-компетенций от использующих ее лиц.
- Существует определенный дефицит в области методической и учебно-методической литературы, посвященной методике преподавания информатики и других учебных предметов на базе применения отечественного программного обеспечения.

Таким образом, для миграции учебно-воспитательного процесса на отечественное программное обеспечение от школьных учителей и администрации образовательных организаций требуются определенные дополнительные усилия, связанные с достаточно серьезными временными и некоторыми финансовыми затратами, а также со сломом наработанного в отношении методики преподавания информатики стереотипа. Данные усилия, с точки зрения автора, в первую очередь должны быть направлены на следующее:

- повышение квалификации педагогов в области методики применения отечественного ПО в учебном процессе;
- разработку собственных элементов методики преподавания информатики на базе отечественного программного обеспечения;
- выбор, приобретение лицензии и установку отечественного программного обеспечения в классах информатики на первых порах в качестве второй операционной системы, что позволит провести плавную миграцию с импортного проприетарного ПО;
- разработку и апробацию сценариев уроков всего курса информатики на базе вновь установленного отечественного ПО, их отработку на практике;
- изучение уже существующих элементов методики преподавания информатики на базе отечественного ПО.

К сожалению, в настоящее время количество литературы данного типа весьма ограничено. Из немногих существующих можно рекомендовать следующие источники:

- *Методическое пособие В. Б. Волкова «Линукс Юниор: книга для учителя»* [1]. В книге описываются не только операционная система, но и все основные свободные прикладные программы, входящие в дистрибутив Линукс Юниор. Описание каждой программы сопровождается несколькими примерами, в зависимости от сложности и насыщенности программы функциональными возможностями. Все примеры приближены к тем задачам, которые выполняет учитель в своей повседневной деятельности: организация, документирование, методическая работа и проведение учебных занятий, а также составление разного рода отчетности. Книга находится в свободном доступе, ее можно скачать в формате PDF по адресу: <https://www.altlinux.org/Books:Junior>
- *Учебно-методический комплект Л. Ф. Соловьевой «Информатика и ИКТ. Работаем в Windows и Linux» для VIII и IX классов* (издательство «БХВ-Петербург», 2011 год) [15] содержит программу курса, подробные поурочные разработки, дидактические и технологические приемы и формы организации учебных материалов. Изложены рекомендации по изучению информатики в VIII—IX классах в соответствии с действовавшим на тот момент времени образовательным стандартом. Описаны методики организации самостоятельной деятельности учащихся на уроке с использованием электронных учебников на DVD, входящих в комплект, для работы в операционных системах Windows и Linux.
- Планируется к изданию в библиотеке Alt Linux книга А. Панюковой «ALT Linux Children: графика, анимация, видео» [10]. Основная идея книги — показать, как можно быстро, просто и эффективно создать что-либо красивое с помощью различных графических и видеоредакторов. Книга ориентирована на детей и взрослых, занимающихся творчеством. Кроме того, в книге есть методическая часть, которая может быть полезна тем, кто собирается организовывать занятия с детьми на базе дистрибутива ALT Linux 4.0 Children.
- *Учебное пособие А. Н. Мартынова, Н. Н. Мартынова «Работа в операционной системе Альт Линукс 5 Школьный Юниор»* (издательство «БИНОМ. Лаборатория знаний», 2010 год) [9] предназначено для учителей информатики средних школ Российской Федерации. В нем содержатся

сведения о рабочем столе GNOME, наборе прикладных и системных программ дистрибутива Школьный Юниор, таких как текстовый редактор gedit, файловый менеджер Nautilus. Кроме того, в нем изложены основы программирования на языке JavaScript.

- Учебное пособие С. С. Задорожного, Н. Н. Мартынова «Компьютерная сеть кабинета информатики на базе Альт Линукс 5 Школьный Сервер / Юниор» (издательство «БИНОМ. Лаборатория знаний», 2010 год) [5] предназначено для учителей информатики, разрабатывающих локальные компьютерные сети школьных кабинетов информатики на базе семейства операционных систем Linux. Данное небольшое по объему пособие может использоваться в учебном процессе в старшей школе в ходе кружковых или факультативных занятий по тематике, связанной с установкой и настройкой операционной системы персонального компьютера или сервера, а также с администрированием локальных компьютерных сетей.

Результатом приложения вышеописанных усилий может стать постепенное успешное внедрение отечественного программного обеспечения в учебный процесс российской общеобразовательной школы. Надо отметить, что *развитие данного направления в педагогике соответствует стратегическим интересам нашего государства, Концепции духовно-нравственного развития и воспитания личности гражданина России [6] и Доктрине информационной безопасности Российской Федерации [4], а также поддерживает процесс импортозамещения в области программного обеспечения, используемого в образовательном процессе.*

Хотелось бы отметить тот факт, что тенденция к отказу от проприетарного программного обеспечения вообще характерна для образовательных учреждений большого количества развитых и динамически развивающихся стран, в том числе Бразилии, Великобритании, Индии, Италии, США, Франции, Швейцарии, Японии [11]. Таким образом, предлагаемый автором в настоящей статье подход не является для мировой практики чем-то революционным, скорее, это устойчивый тренд. Новизна заключается в предложении не просто использовать свободное программное обеспечение, на основе которого создается российское ПО, но и задействовать его воспитательный потенциал.

Рассмотрим духовно-нравственные воспитательные аспекты использования отечественного программного обеспечения в учебном процессе образовательной организации, к которым можно отнести патриотический и гражданско-правовой аспекты.

Патриотический аспект.

С точки зрения автора, уже сам факт применения школьниками в учебной и досуговой деятельности отечественного программного обеспечения будет демонстрировать им самостоятельность и конкурентоспособность российского сектора IT-экономики, связанного с разработкой и тестированием ПО.

Параллельно с внедрением российского ПО в учебный процесс крайне важно информировать учащихся о наличии перманентных потенциальных угроз информационной инфраструктуре Российской Федерации со стороны ряда иностранных государств. К наиболее очевидным угрозам можно отнести следующие [16]:

- прекращение технической поддержки пользователей иностранного программного обеспечения на территории России;
- выпуск обновлений, которые могут заблокировать или усложнить использование программного обеспечения именно для российских пользователей;
- блокировка доступа к репозиториям пакетов программ для российских пользователей (для операционных систем семейства Linux);
- возможное наличие скрытых программных модулей, производящих сбор информации об активности российских пользователей в интересах третьей стороны, например, спецслужб тех государств, на территории которых зарегистрированы производители ПО;
- кибератаки на компьютеры и локальные сети, находящиеся под управлением операционных систем иностранного производства.

К сожалению, перечисленные выше виды угроз не только являются потенциально возможными, но и подтверждаются реальными фактами, такими как, например, кибератака неизвестных злоумышленников на коммутаторы Cisco, имевшая место весной 2018 года [7]. «Лаборатория Касперского» сообщила, что некая группировка использует уязвимость в программе под названием Cisco Smart Install Client, чтобы получить возможность исполнять произвольный код на устройстве. Злоумышленники перезаписывают образ системы Cisco IOS и меняют конфигурационный файл, оставляя в нем послание «Do not mess with our elections» («Не вмешайтесь в наши выборы»).

Кроме того, можно привести пример вирусной атаки на сеть компьютеров ИСОД МВД России, в ходе которой все персональные компьютеры под управлением операционной системы Windows были заблокированы. Отголоски хакерской атаки почувствовали во многих территориальных подразделениях МВД России — отсутствовал доступ к базам данных, наблюдались перебои с телефонной связью [2].

В свою очередь, применение отечественного программного обеспечения в информационной инфраструктуре российских государственных организаций минимизирует вышеуказанные угрозы, что, в частности, подтверждается фактом успешного использования операционной системы Гослинукс (продукт компании «Ред Софт») в отделении Федеральной службы судебных приставов по Владимирской области [3].

С точки зрения автора, в результате миграции учебного процесса на отечественное программное обеспечение и ведения преподавателями информатики соответствующей разъяснительной работы в области информационной безопасности у учащихся *должен формироваться образ России как государства, способного противостоять не только военной, но и информационной агрессии, факт наличия которой в наше время подтверждается в том числе вышеприведенными фактами.*

Гражданско-правовой аспект.

Офисное отечественное программное обеспечение в подавляющем большинстве не создается разработчиком на пустом месте, обычно в его основе находится свободное ПО. Под *свободным программным обеспечением* надо понимать такое программное обеспечение, пользователи которого имеют права («свободы») на его неограниченную установку, запуск, свободное ис-

пользование, изучение, распространение и изменение (совершенствование), а также распространение копий и результатов изменения [14].

Отечественные операционные системы могут быть как полностью свободным ПО (например, ROSA R9), так и частично свободным (например, Альт Образование 8). Для нужд образовательных организаций и учащихся актуален вопрос стоимости лицензии на использование операционной системы и прикладного программного обеспечения. В плане стоимости для частных пользователей и образовательных организаций как ROSA R9, так и Альт Образование 8 являются бесплатными операционными системами.

Таким образом, учащиеся, работающие на персональных компьютерах под управлением легально приобретенных лицензионных отечественных операционных систем, к тому же бесплатных, вырабатывают свою гражданскую позицию в плане соблюдения российского законодательства в области авторских и смежных прав. В результате данного подхода к выбору типа программного обеспечения можно надеяться на то, что наши школьники, получившие соответствующий опыт в процессе своего обучения, в дальнейшей своей взрослой жизни не позволят себе использовать контрафактные копии проприетарного программного обеспечения, а будут совершать квалифицированный сознательный выбор в пользу бесплатного отечественного ПО.

Подводя итоги, необходимо еще раз отметить, что миграция учебно-воспитательного процесса в общеобразовательной школе на отечественное программное обеспечение является важным аспектом процесса импортозамещения в области народного образования и одним из существенных факторов гражданско-патриотического воспитания учащихся.

Список использованных источников

1. Волков В. Б. Линукс Юниор: книга для учителя. М.: ALT Linux; Издательский дом ДМК-пресс, 2009. <https://www.altlinux.org/Books:Junior>
2. Голодков Ю. Э., Ларионова Е. Ю. Современные подходы к формированию компетенций в области информационной безопасности обучающихся вузов МВД России // Информатика и образование. 2018. № 2.
3. Гослинукс — операционная система ФССП России. <http://goslinux.fssprus.ru/>

4. Доктрина информационной безопасности Российской Федерации // Информационно-правовой портал Гарант.ру. <http://www.garant.ru/products/ipo/prime/doc/71456224/>

5. Задорожный С. С., Мартынов Н. Н. Компьютерная сеть кабинета информатики на базе Альт Линукс 5 Школьный Сервер / Юниор. М.: БИНОМ. Лаборатория знаний, 2010.

6. Концепция духовно-нравственного развития и воспитания личности гражданина России // Сайт ГМЦ г. Москвы. <http://mosmetod.ru/metodicheskoe-prostranstvo/nachalnaya-shkola/inklyuzivnoe-obrazovanie/fgos/kontseptsiya-dukhovno-nravstvennogo-razvitiya-i-vospitaniya-lichnosti-grazhdanina-rossii.html>

7. «Лаборатория Касперского»: атаки на коммутаторы Cisco нацелены на Рунет // РИА Новости. 07.04.2018. <https://ria.ru/technology/20180407/1518127365.html>

8. Малахов А. «Компьюлинк» включили в школьную программу // Коммерсантъ. 13.11.2007. <https://www.kommersant.ru/doc/824583>

9. Мартынов А. Н., Мартынов Н. Н. Работа в операционной системе Альт Линукс 5 Школьный Юниор. М.: БИНОМ. Лаборатория знаний, 2010.

10. Панюкова А. ALT Linux Children: графика, анимация, видео // ALT Linux Team. <https://www.altlinux.org/Books:Children>

11. Перечень школьного программного обеспечения. <https://dic.academic.ru/dic.nsf/ruwiki/310737>

12. Поносов Александр Михайлович // Википедия. https://ru.wikipedia.org/wiki/Поносов,_Александр_Михайлович

13. Распоряжение Правительства РФ № 1447-р от 18 октября 2007 года «О лицензионной поддержке программного обеспечения, используемого в общеобразовательных учреждениях (с изменениями на 10 марта 2009 года)». <http://docs.cntd.ru/document/902066443>

14. Свободное программное обеспечение // Википедия. https://ru.wikipedia.org/wiki/Свободное_программное_обеспечение

15. Соловьева Л. Ф. Информатика и ИКТ. Работаем в Windows и Linux. 8–9 кл. СПб.: БХВ-Петербург, 2011.

16. Федосов А. Ю., Маркушевич М. В. Актуальные вопросы применения отечественного программного обеспечения в учебно-воспитательном процессе общеобразовательной школы // Информатика и образование. 2018. № 6.

17. Федосов А. Ю., Маркушевич М. В. Использование отечественного и свободного программного обеспечения в образовательном процессе как фактор духовно-нравственного воспитания младших школьников // Герценовские чтения. Начальное образование. 2018. Т. 9. № 1.

НОВОСТИ

Рособрнадзор: апробация ЕГЭ по информатике на компьютере пройдет в 2019 году

Массовая апробация Единого государственного экзамена по информатике на компьютере может состояться в 2019 году. Об этом в ходе круглого стола «Итоги мониторинга общественного мнения по проблемам подготовки к ЕГЭ-2018» сообщил заместитель руководителя Федеральной службы по надзору в сфере образования и науки (Рособрнадзор) Анзор Музаев. «Давно мы шли к этому — к ЕГЭ по информатике на компьютерах,

надеемся этот процесс закончить в 2020/2021 учебном году. Потому что в век цифровой технологии сдавать ЕГЭ по информатике на бумаге — не совсем правильно. Нас сдерживала материально-техническая база регионов, чтобы сильно их не нагружать. Я думаю, что в следующем году будет массовая апробация, в 2020/2021 году мы надеемся выйти на финишную прямую», — сообщил А.Музаев.

(По материалам Агентства новостей «Москва»)

Анастасия Комшина, Севастьян Телибаев, Б. С. Михлин,
школа № 1925, г. Москва

СБОРКА RS-ТРИГГЕРА НА МИКРОСХЕМАХ, СОДЕРЖАЩИХ ЭЛЕМЕНТЫ «ИЛИ-НЕ», «И-НЕ»*

Аннотация

В статье представлена общая информация о триггерах, их типах, об асинхронном RS-триггере. Дано подробное описание практической работы по сборке RS-триггера в двух вариантах — на базе микросхемы, содержащей четыре элемента «2ИЛИ-НЕ», и на базе микросхемы, содержащей четыре элемента «2И-НЕ». При сборке используются детали из электронного конструктора «Микроник» фирмы «Амперка» и детали, предоставляемые сайтом «Сотворим вместе».

Ключевые слова: триггер, RS-триггер, тумблер, реле, микросхема, таблица истинности.

DOI: 10.32517/2221-1993-2018-17-7-17-25

Контактная информация

Михлин Борис Самуилович, учитель информатики школы № 1925, г. Москва; *адрес:* 111673, г. Москва, Новокосинская ул., д. 13А; *телефон:* (495) 702-66-76; *e-mail:* bmichlin@yahoo.com

A. Komshina, S. Telibaev, B. S. Mikhlin,
School 1925, Moscow

ASSEMBLING THE RS FLIP-FLOP ON CHIPS CONTAINING ELEMENTS OF "OR-NOT", "AND-NOT"

Abstract

The article provides general information about flip-flops, their types, about an asynchronous RS flip-flop. A detailed description of the practical work on assembling the RS flip-flop in two variants is given, based on a chip containing four "2OR-NOT" elements, and based on a chip containing four "2AND-NOT" elements. The details from the electronic set "Micronik" ("Amperka") and the details provided by the site "Let's create together" are used in assembling.

Keywords: trigger, flip-flop, RS flip-flop, toggle switch, relay, chip, truth table.

Электроника — это искусство, которое основано на нескольких основных законах и включает в себя большое количество практических правил и приемов.

П. Хоровиц, У. Хилл [6]

1. Введение

Происхождение термина.

Слово «триггер»** происходит от английского слова «trigger» — защелка, спусковой крючок.

Что такое триггер.

Триггер — это электронное устройство, которое может длительно находиться в одном из двух устойчивых состояний*** (их можно обозначить как 0 и 1) и способно почти мгновенно переходить из одного состояния в другое под воздействием внешних электрических сигналов [1, 5]. Триггер хранит один бит информации.

Электромеханической моделью триггера может служить обычный электрический выключатель, тумблер**** или реле*****. На рисунке 1 показана группа тумблеров, а на рисунке 2 — группа реле с выставки школы № 1925.

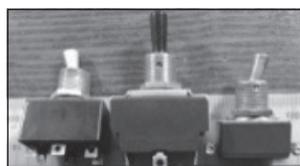


Рис. 1



Рис. 2

Когда контакты выключателя или реле замкнуты — это состояние логической единицы (ток через контакты проходит), разомкнуты — состояние логического нуля (ток не проходит)*****.

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_7-2018/

** В английском языке триггер чаще называют «flip-flop».

*** Бывают триггеры и с более чем двумя устойчивыми состояниями, но они используются редко (см.: <https://ru.wikipedia.org/wiki/Триггер>).

**** Тумблер (от *англ.* tumble — опрокидываться, переворачиваться) — малогабаритный механический переключатель.

***** Реле (*фр.* relais) — элемент автоматических устройств, который при воздействии на него внешних физических явлений скачкообразно принимает конечное число значений выходной величины (<https://ru.wikipedia.org/wiki/Реле>).

***** Обычно тумблеры (так же как и реле или другие переключатели) имеют две или более пар контактов. Когда одна пара контактов замыкается, то другая (инверсная к этой паре) автоматически размыкается, и наоборот.

История возникновения триггера.

Триггер изобрели независимо друг от друга в 1918 году:

- М. А. Бонч-Бруевич (1888–1940) — основоположник радиоламповой промышленности в СССР;
- англичане У. Г. Икклз (W. H. Eccles) и Ф. У. Джордан (F. W. Jordan).

Применение.

Триггер — это один из базовых (основных) элементов цифровой техники, и некоторые исследователи включают его в сто величайших изобретений человечества.

В компьютерах триггеры используются в регистрах, счетчиках и в устройствах памяти.

Типы триггеров.

Существует много типов триггеров: RS-триггеры, D-триггеры, T-триггеры, JK-триггеры и др.

В учебниках информатики для IX–X классов в основном рассматривается наиболее простой и широко распространенный **асинхронный RS-триггер**. Его мы и будем собирать и изучать.

Работа RS-триггера на логических элементах «ИЛИ-НЕ» (вентилях, реализующих логическую операцию «стрелка Пирса», «пог») описана (на уровне функциональных схем) как в учебниках А. В. Горячева [1], К. Ю. Полякова [5], так и во многих других учебниках информатики.

Работа RS-триггера на логических элементах «И-НЕ» (вентилях, реализующих логическую операцию «штрих Шеффера», «pand») описана (на уровне функциональных схем) в работе Е. А. Еремина [2].

Обозначение асинхронного RS-триггера*.

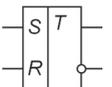
Условное обозначение на схеме: 	Входы: <i>S</i> (Set) — установка триггера в 1 ($Q = 1$), <i>R</i> (Reset) — сброс триггера в 0 ($Q = 0$). Выходы: Q и \bar{Q} (выход с кружком — инверсный выход, НЕ Q)
---	---

Таблица истинности RS-триггера.

<i>S</i>	<i>R</i>	Q	\bar{Q}	Состояние
0	0	Q	\bar{Q}	Хранение бита
0	1	0	1	Сброс в 0 ($Q = 0$)
1	0	1	0	Установка в 1 ($Q = 1$)
1	1	0	0	Запрещено**
		1	1	

Триггеры могут быть собраны на реле, радиолампах, транзисторах, микросхемах и других элементах. Примеры представлены на рисунках 3–5:

- на рисунке 3 — триггер на радиолампах и других отечественных деталях, СССР (экспонат музея школы № 1530, г. Москва***);

- на рисунке 4 — триггер на радиолампе 6К4П (6К4П пентод, СССР, 1971 год; резисторы и конденсатор импортные) (экспонат выставки школы № 1925, г. Москва);
- на рисунке 5 — триггеры, собранные на макетной плате: слева на плате — на двух транзисторах, справа на плате — на ТТЛ-микросхеме****.

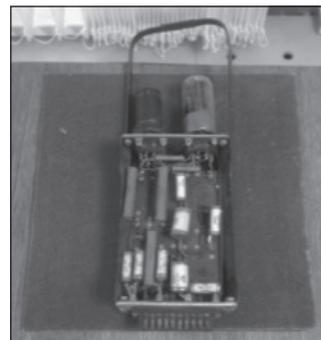


Рис. 3



Рис. 4

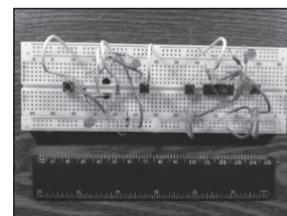


Рис. 5

Устройство асинхронного RS-триггера на логических элементах (вентилях) «ИЛИ-НЕ», «И-НЕ» (функциональные схемы) приведено на рисунках 6, 7 соответственно.

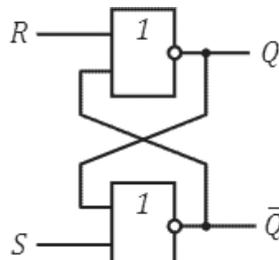


Рис. 6

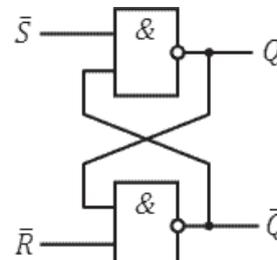


Рис. 7

На рисунке 6 активным сигналом на входах *R* и *S* является высокий логический уровень, на рисунке 7 — низкий логический уровень (о чем говорят черточки инверсии над входами *S* и *R*). То есть на входах у вентилях «И-НЕ» используется инверсная логика (у таких входов также принято ставить кружочки).

Уровни напряжений ТТЛ.

Логический ноль (низкий логический уровень)	От 0 В до 0,4 В
Логическая единица (высокий логический уровень)	От 2,4 В до 5 В

* Отечественные и зарубежные обозначения логических элементов: <http://go-radio.ru/bazovie-logicheskie-elementi.html>

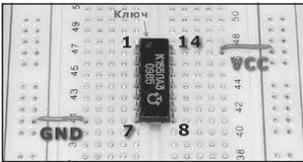
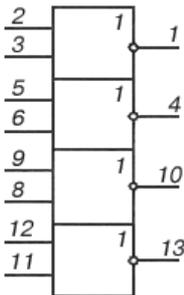
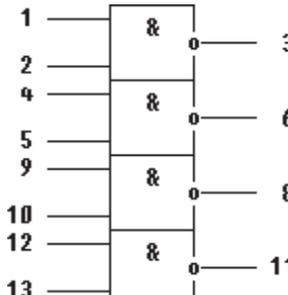
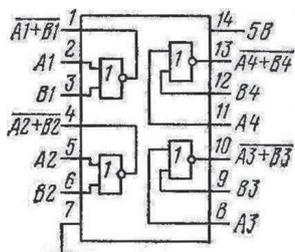
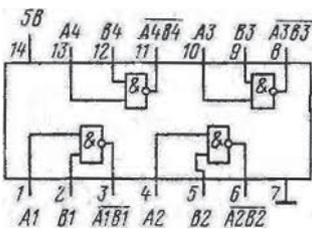
** В этом состоянии схема теряет свои триггерные свойства: на выходах Q и НЕ Q могут одновременно быть два логических нуля (для вентилях «ИЛИ-НЕ») или две логические единицы (для вентилях «И-НЕ») (см. ниже проверку работы собранных триггеров).

*** Музей «История вычислительной техники» школы № 1530 (основатель и директор музея Д. М. Златопольский): http://gym1530.mskobr.ru/nashi_muzei/muzej_istorii_vychislitel_noj_tehniki/

**** ТТЛ — транзисторно-транзисторная логика (https://ru.wikipedia.org/wiki/Транзисторно-транзисторная_логика/).

Диапазон входных напряжений от 0,4 В и до 2,4 В является запрещенным (нерабочим*).

Расположение выводов на микросхеме (цоколёвка**).

Триггер на элементах «ИЛИ-НЕ»	Триггер на элементах «И-НЕ»
 <p>Ножка 7 — нулевой (общий, земля (GND)) Ножка 14 — +5 В (V_{CC})</p>	 <p>Ножка 7 — нулевой (общий, земля (GND)) Ножка 14 — +5 В (V_{CC})</p>
	
	
<p>K155ЛЕ1. Зарубежные аналоги: SN7402, SN74HC02N. 4 логических элемента (вентили) «ИЛИ-НЕ»</p>	<p>K155ЛА3 (К1ЛБ553). Зарубежный аналог: SN7400***. 4 логических элемента (вентили) «И-НЕ»</p>

2. Сборка триггера

Для упрощения сборки триггеров будем проводить ее без пайки на макетных платах****.

Чтобы не покупать детали россыпью, будем использовать готовые наборы деталей (возможно, с некоторыми минимальными дополнениями).

Рассмотрим два варианта сборки RS-триггера.

Вариант 1. На базе микросхемы, содержащей четыре элемента «ИЛИ-НЕ».

В качестве базового набора элементов будем использовать детали из электронного конструктора «Микро-ник» фирмы «Амперка»***** (рис. 8).

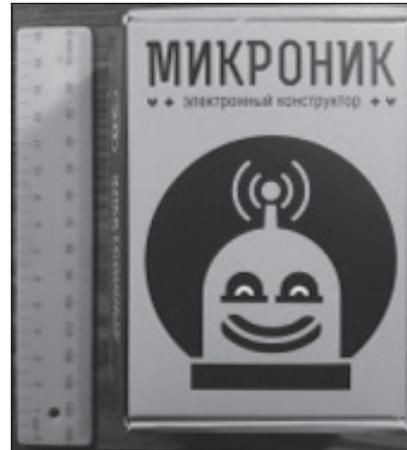


Рис. 8

Для удобства сборки макетную плату конструктора «Микро-ник» мы заменили платой большего размера и добавили проводов синего и зеленого цветов.

Вариант 2. На базе микросхемы, содержащей четыре элемента «И-НЕ».

В качестве набора элементов будем использовать детали, бесплатно предоставленные сайтом «Сотворим вместе»***** (рис. 9).

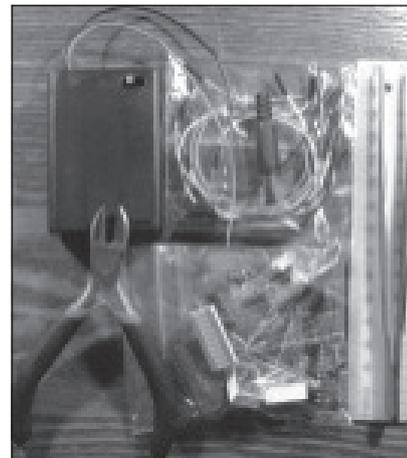


Рис. 9

* Диапазоны логического нуля (от 0 до 0,4 В) и логической единицы (от 2,4 до 5 В) предусмотрены для того, чтобы электрические помехи в цепях не приводили к ложному опознанию логического нуля как единицы и наоборот. Так как цифровые сигналы имеют форму, близкую к прямоугольной, с крутыми фронтами (подъемами и спусками), вентили микросхем находятся в запрещенном состоянии очень короткое время.

** Цоколёвка («распиновка»), или разводка выводов — описание каждого контакта электрического соединения в электронной аппаратуре, например, разъема, микросхемы. Цоколёвка необходима для задач сборки и ремонта устройств, содержащих несколько выводов, она идентифицирует все контакты соединения. Цоколёвка может быть представлена в виде чертежа или таблицы. Описание обычно включает номер контакта, краткое наименование и другие параметры — такие как цвет оболочки провода, функциональное назначение, электрические характеристики. На чертеже, там, где это необходимо, развязывают сторону соединения (например, изображен ли разъем со стороны кабеля или со стороны подключаемого устройства). <https://ru.wikipedia.org/wiki/Цоколевка>. У микросхем корпус с двухрядным расположением выводов (DIP-корпус: Dual-In-Line Package).

*** Серия микросхем 74xx фирмы Texas Instruments, США. Отечественный аналог — серия 155.

**** Макетная плата без пайки: <http://sotvorimvmeste.ru/viewtopic.php?f=12&t=8>

***** Конструктор «Микро-ник» фирмы «Амперка»: <http://amperka.ru/collection/kits>

***** Сайт «Сотворим вместе» (<http://sotvorimvmeste.ru/viewtopic.php?f=34&t=96&p=426>) совместно со спонсорами (НИЯУ МИФИ и др.) оказывает содействие в организации и работе технических кружков в учебных заведениях.

Перед сборкой схемы надо сначала нарисовать электрическую принципиальную схему триггера. Это можно сделать либо вручную (в тетрадке в клетку или на миллиметровке*) или с помощью компьютерных редакторов электрических схем, например, редактора sPlan 7.0**, который мы и будем использовать.

2.1. Общие замечания по сборке

Учащиеся выполняют один из вариантов сборки триггера в группах по два-три человека.

Перед практической работой группы должны сдать зачет-допуск, для которого нужно знать не только тему «Логика» учебника (см., например, [1, 5]), но и данное описание, а также ответить на контрольные вопросы (см. Приложение).

Монтажные схемы рисовать не следует, чтобы ученики бездумно не собирали схемы как пазлы.

Желательно, чтобы на практической работе перед учащимися был лист с электрической принципиальной схемой, на которой они карандашом (чтобы была возможность стереть) обводили собранные участки схемы.

Для питания схем будем использовать батарейные отсеки на три батарейки по 1,5 В типа АА («пальчиковые»), которые соединены последовательно.

Фактически такие батарейные отсеки дают около 4,5 В ($3 \times 1,5$ В). По справочникам у микросхемы питание должно быть $5 \text{ В} \pm 5\%$, т. е. от 4,75 В до 5,25 В, но при свежих батарейках все нормально работает***.

Для подключения к шине +5 В (4,5 В) будем использовать красные провода, а для подключения к нулевой шине (общий, земля (GND, ground), -5 В) — синие провода (такие цвета приняты в электронике).

Особенно надо быть внимательным при подключении батареек в варианте 1 (в схеме с вентилями «ИЛИ-НЕ» на базе конструктора «Микроник»). Там отсутствует разъем питания (который есть в варианте 2), и существует опасность ошибочно подключить плюс батарейки (красный провод) к общей синей шине макетной платы. И хотя кажется очевидным, что красный провод от батарейки надо подключать к красной плюсовой шине монтажной платы, но лучше установить разъем по питанию, как в варианте 2, и полностью исключить вероятность такой ошибки.

Остальные соединения будем выполнять зелеными проводами (они хорошо видны на фоне белой макетной платы и деталей)****.

Длину проводов не будем брать с запасом, иначе за клубком проводов не будет видно схемы, сложно будет искать ошибки в соединениях, кроме того, провода будут выдергиваться, если их случайно задеть.

Более длинная ножка светодиодов (анод) (на рисунке 10 это правая ножка) должна подключаться к плюсу питания (в данной схеме — к выходам вентиляей) через ограничивающий ток резистор, а более короткая ножка (катод) — к минусу (в данной схеме — к нулевой шине)*****.

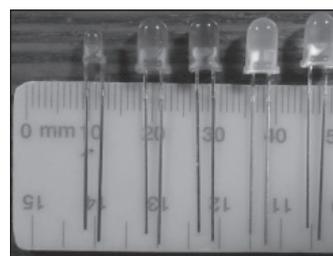


Рис. 10

Красный светодиод будем подключать к выходу Q , а зеленый светодиод — к выходу \overline{Q} (Q инвертированному).

Для измерения всех электрических величин (например, сопротивления и напряжения) будем использовать недорогой, компактный и с защитой от перегрузок цифровой мультиметр (тестер), например, DT838*****.

Номиналы (значения сопротивлений) резисторов можно определить по их цветным полоскам***** (рис. 11) или с помощью мультиметра.

Цветовая маркировка			
Не полосы и значение			
1 и 2	число		
3	кол-во нулей		
4	допуск		
1	2	3	4
1	0	00	5%
1000 Ом = 1 кОм 5%			
Черный	0		
Коричневый	1		
Красный	2		
Оранжевый	3		
Желтый	4		
Зеленый	5		
Синий	6		
Фиолетовый	7		
Серый	8		
Белый	9		
Золотой	5%		
Серебро	10%		
www.sotvorimvmeste.ru			

Рис. 11

При определении величины сопротивления резистора с помощью тестера необходимо, чтобы резистор не был подключен в схему.

При этом измерительный прибор покажет реальное сопротивление резистора, которое может отличаться в большую или меньшую сторону от номинального сопротивления (в пределах допуска, указанного на резисторе).

Например, для резистора номиналом в 1 кОм и допуском $\pm 5\%$ величина сопротивления может быть от 950 до 1050 Ом. Как видно на рисунке 12, при измерении сопротивления конкретного резистора мультиметр (DT838) показывает 969 Ом.

В качестве вентиляей (элементов «ИЛИ-НЕ», «И-НЕ») для триггера можно использовать любые два вентиля из четырех входящих в состав используемых микросхем.

***** Использование стрелочных (аналоговых) тестеров менее удобно.

***** Цветовая маркировка резисторов: <http://sotvorimvmeste.ru/viewtopic.php?f=33&t=80#p268>

* Раньше для таких целей даже продавались специальные трафаретки для радиоинженеров.

** Редактор электрических схем sPlan 7.0: <http://splansoft.ru/>

*** Для «подсевших» батареек можно использовать батарейные отсеки на четыре батарейки.

**** О проводах и монтаже на макетной плате см.: <http://sotvorimvmeste.ru/viewtopic.php?f=12&t=131>

***** Если перепутать выводы светодиода, то он будет «заперт» (т. е. не будет пропускать ток) и, следовательно, не будет светиться.



Рис. 12

Для индикации включения питания слева вверху на макетных платах установим красный светодиод с ограничивающим ток резистором в 1 кОм.

Чтобы легче выполнить сборку схем, разобьем весь процесс сборки на несколько этапов.

2.2. Инструменты для сборки

При сборке схем мы использовали следующие инструменты (см. также [3], гл. 2) (рис. 13):

- бокорезы — для обрезания проводников и зачистки их от изоляции*;
- пинцет;
- длинногубцы — для изгибания выводов деталей и проводов;
- лупа складная — для чтения маркировок на резисторах и микросхемах (достаточно двух-трех штук на класс).



Рис. 13. Инструменты для сборки схем

Инструменты должны быть небольшими, так как выводы деталей и провода имеют диаметр меньше 1 мм. При этом также уменьшается травмоопасность инструмента, а также его стоимость.

Для зачистки концов проводов от изоляции и их обрезки удобно использовать специальные клещи (стрипперы) с минимальным диаметром проводов 0,5–0,6 мм.

Нож по соображениям безопасности лучше не использовать. На всякий случай у учителя под рукой всегда должны быть йод и вата.

* Бокорезы и пинцет прилагаются к набору деталей от сайта «Сотворим вместе».

2.3. Вариант 1. Порядок сборки RS-триггера на микросхеме 4×«2ИЛИ-НЕ»

Электрическая принципиальная схема.

По функциональной схеме триггера (см. рис. 6) нарисован в редакторе sPlan электрическую принципиальную схему RS-триггера на микросхеме 4×«2ИЛИ-НЕ»** (рис. 14).

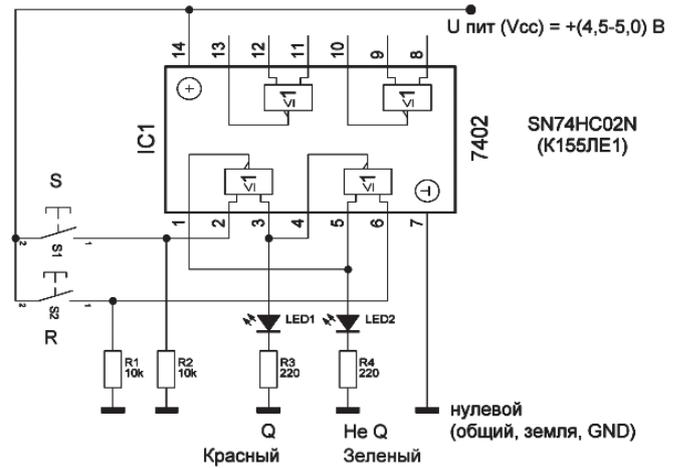


Рис. 14

Этапы сборки триггера.

1. Установка элементов на монтажную плату и подключение деталей к шинам питания (рис. 15).

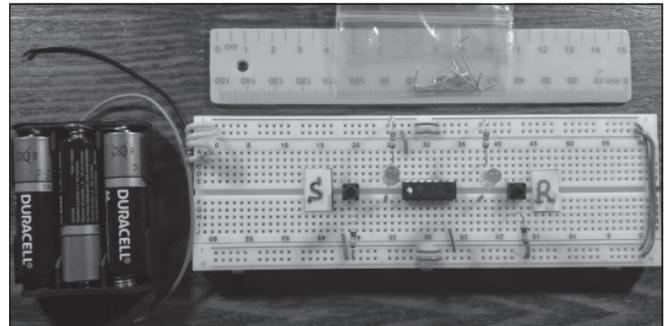


Рис. 15

Батарейки отключены от схемы.

2. Соединение установленных элементов проводами (перемычками) и проверка работы схемы (рис. 16).

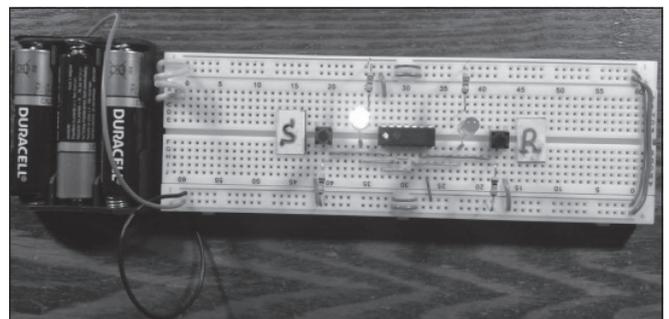


Рис. 16

** В редакторе sPlan вентили «2ИЛИ-НЕ» обозначаются по стандарту Великобритании (значок «≥1» соответствует логической операции ИЛИ). Резисторы R1 и R2 можно брать и других номиналов (например, от 1 до 2,2 кОм, но в конструкторе «Микроник» таких номиналов нет).

В используемой импортной микросхеме SN-74HC02N (SN7402) (отечественный аналог — микросхема К155ЛЕ1) четыре двухходовых вентиля «ИЛИ-НЕ» (4×«2ИЛИ-НЕ»).

Ключ у микросхемы (выемка на корпусе) расположен слева. Нумерация выводов микросхемы идет от ключа по кругу против часовой стрелки.

На схеме в качестве *первого* логического элемента «ИЛИ-НЕ» будем использовать, например, вентиль с входами на ножках 2, 3 и выходом на ножке 1 (см. цоколёвку микросхемы на с. 19).

В качестве *второго* логического элемента «ИЛИ-НЕ» будем использовать, например, вентиль с входами на ножках 5, 6 и выходом на ножке 4.

Питание на схему можно подавать (подключением проводов батарейного отсека) только после завершения этапа сборки и проверки схемы. При этом красный провод (+4,5 В) подключается к красной шине платы, а черный провод — к синей (общей) шине. *Их нельзя перепутать!*

Если схема собрана правильно, то при нажатии на левую кнопку *S* (Set) загорается левый красный светодиод (выход *Q* устанавливается в 1), а правый зеленый светодиод не горит (выход НЕ *Q* = 0) (рис. 17).

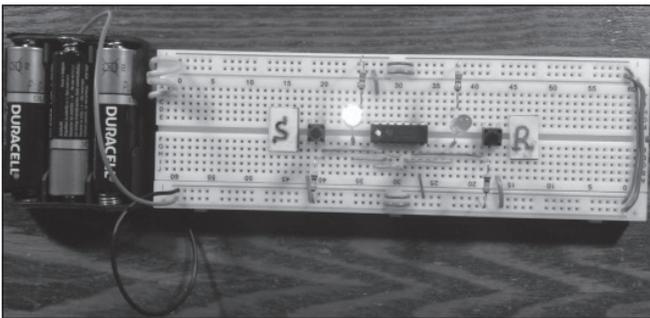


Рис. 17

При повторных нажатиях кнопки *S* ничего не меняется.

При нажатии на правую кнопку *R* (Reset) левый красный светодиод гаснет (выход *Q* сбрасывается в 0) и, соответственно, загорается правый зеленый светодиод (выход НЕ *Q* = 1) (рис. 18).

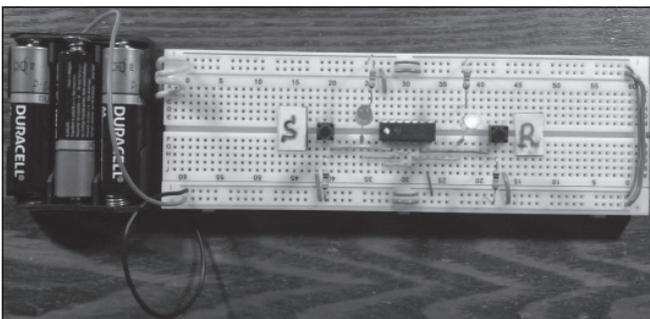


Рис. 18

Повторные нажатия кнопки *R* ничего не меняют.

Одновременное нажатие кнопок *S* и *R* запрещено. Схема не сгорит, но в этом режиме она не работает как триггер (оба светодиода не горят, т. е. $Q = 0$ и НЕ $Q = 0$) (рис. 19).

3. У нас остались неиспользованными в микросхеме еще два вентиля «ИЛИ-НЕ» (т. е. на этой микросхеме можно при желании собрать еще, например, один триггер).

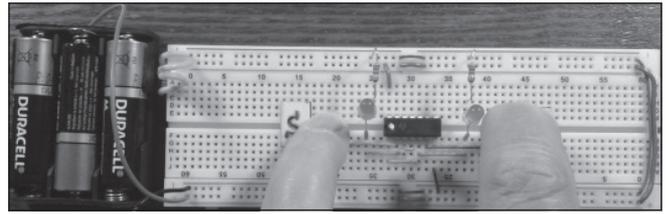


Рис. 19

Сборка и проверка работы RS-триггера закончены.

Если схема все-таки не работает (или работает неправильно), то надо проверить все соединения по электрической принципиальной схеме или проверить схему тестером.

2.4. Вариант 2. Порядок сборки RS-триггера на микросхеме 4×«2И-НЕ»

Электрическая принципиальная схема RS-триггера на микросхеме 4×«2И-НЕ» приведена рисунке 20.

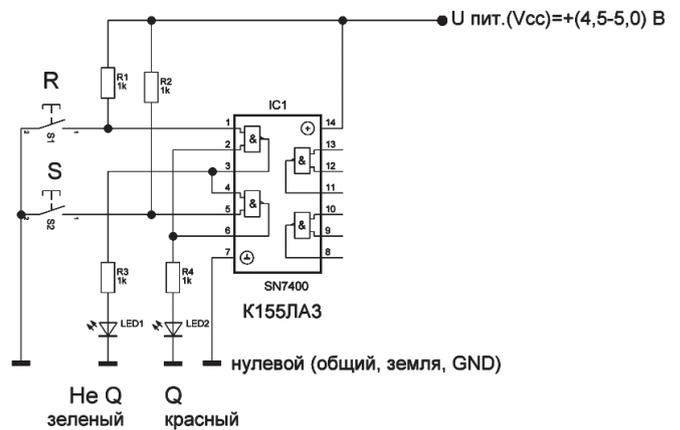


Рис. 20

Мы немного изменили схемы, приведенные в [2] и на сайте «Сотворим вместе»:

- В схеме, приведенной в [2], для более надежной работы триггера (защиты от помех), как это и полагается по документации на микросхему (см., например, в справочнике М. Тули [7]), мы подключили входы вентилях 1 и 5 через резисторы 1 кОм ($R1$ и $R2$) к шине питания +5 В. Дело в том, что, когда кнопки $S1$ и $S2$ разомкнуты (не нажаты), входы вентилях 1 и 5 «висят в воздухе» и воздействующие на них электромагнитные и статические электрические помехи могут спонтанно переключить триггер. Если уровень помех невысокий и надежность хранения триггером информации (одного бита) не слишком важна (что соответствует школьным условиям), то эти резисторы ($R1$ и $R2$) можно не ставить и тем самым упростить схему. Без них и так все будет нормально работать. Номиналы резисторов, ограничивающих ток у светодиодов, заменили с 330 Ом на 1 кОм.
- Для схемы, приведенной на сайте «Сотворим вместе»*, мы к выходу НЕ Q подключили зеленый светодиод с ограничивающим ток резистором 1 кОм.

* Сборка RS-триггера: <http://sotvorimvmeste.ru/viewtopic.php?f=9&t=11>

Этапы сборки триггера.

1. Установка элементов на монтажную плату и подключение деталей к шинам питания (рис. 21).

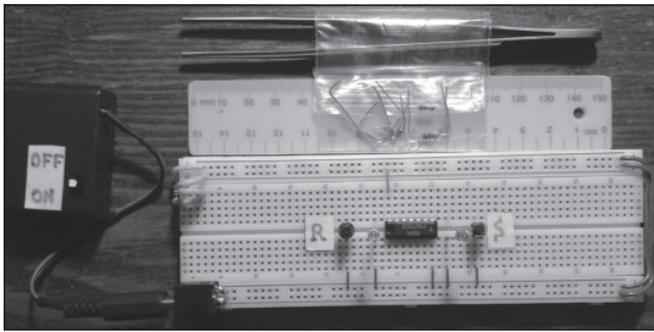


Рис. 21

Выключатель батарейного отсека находится в состоянии OFF («Отключено»), т. е. питание на схему не подается.

В используемой микросхеме (отечественная микросхема К155ЛА3 или импортный аналог SN7400) четыре двухвходовых вентиля «И-НЕ» (4×«2И-НЕ»).

Ключ у микросхемы (выемка на корпусе) расположен слева. Нумерация выводов микросхемы идет от ключа по кругу против часовой стрелки.

На схеме в качестве первого логического элемента «И-НЕ» используется, например, вентиль с входами на ножках 1, 2 и выходом на ножке 3 (см. цоколевку микросхемы на с. 19).

В качестве второго логического элемента «И-НЕ» будем использовать, например, вентиль с входами на ножках 12, 13 и выходом на ножке 11.

Питание на схему можно подавать (установкой переключателя на батарейном отсеке в положение ON («Включено»)) только после завершения этапа сборки и проверки схемы.

2. Соединение установленных элементов проводами (перемычками) и проверка работы схемы.

Резисторы R_1 и R_2 , повышающие надежность работы схемы, пока не устанавливаем (на рисунке 22 они лежат на линейке).

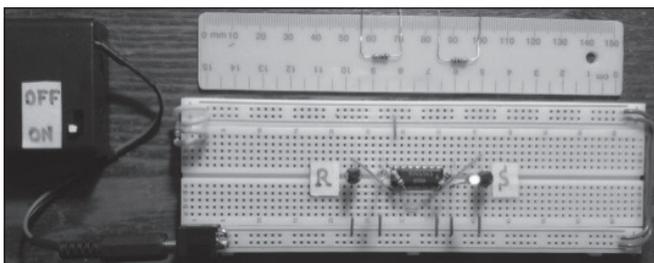


Рис. 22

Если схема собрана правильно, то при нажатии на правую кнопку S (Set) загорается правый красный светодиод (выход Q устанавливается в 1), а левый зеленый светодиод не горит (выход \overline{Q} устанавливается в 0) (рис. 22).

При повторных нажатиях кнопки S ничего не меняется.

При нажатии на левую кнопку R (Reset) правый красный светодиод гаснет (выход Q сбрасывается в 0) и, соответственно, загорается левый зеленый светодиод (выход \overline{Q} устанавливается в 1) (рис. 23).

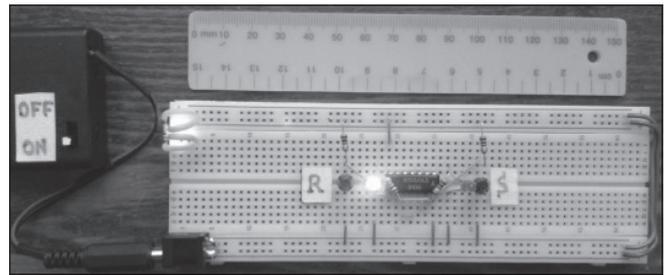


Рис. 23

Повторные нажатия кнопки R ничего не меняют. Одновременное нажатие кнопок S и R запрещено. Схема не горит, но в этом режиме она не работает как триггер (оба светодиода горят, т. е. $Q = 1$ и $\overline{Q} = 1$) (рис. 24).

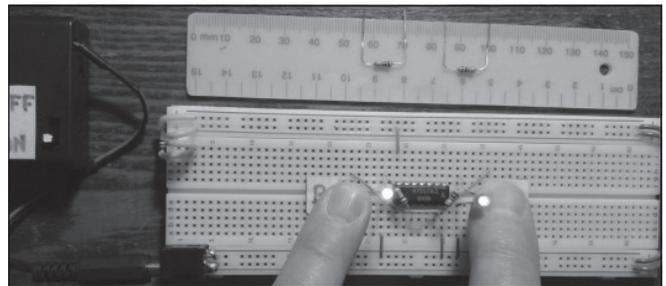


Рис. 24

3. Убедившись, что схема работает правильно, устанавливаем резисторы R_1 и R_2 и снова проверяем работу схемы (см. этап 2 сборки) (рис. 25).

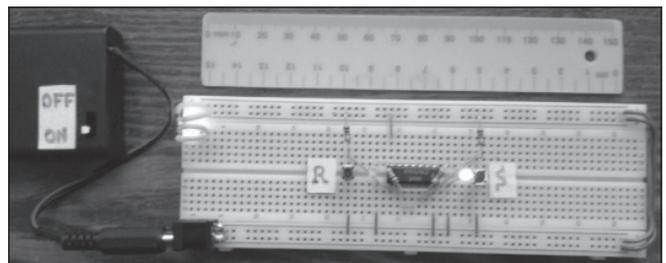


Рис. 25

4. У нас остались неиспользованными в микросхеме еще два вентиля «И-НЕ» (т. е. на этой микросхеме можно при желании собрать, например, еще один триггер).

Сборка и проверка работы RS-триггера закончены.

Если схема все-таки не работает (или работает неправильно), то надо проверить все соединения по электрической принципиальной схеме или проверить схему тестером.

3. Выводы

Сборка триггеров из реальных деталей с использованием монтажных плат без пайки оказалась несложной и вполне доступной для учащихся IX—X классов в течение урока после предварительной теоретической подготовки.

Рассмотренные схемы триггеров содержат небольшое количество недорогих деталей, которые можно взять, например, из конструктора «Микроник» фирмы «Амперка» или получить бесплатно на сайте «Сотворим вместе».

Использование реальных электронных деталей для сборки:

- знакомит учеников с увлекательным миром цифровой электроники (раньше многие люди увлекались радиолобительством!) и развивает техническое творчество;
- углубляет, расширяет и интегрирует знания по информатике, физике, технологии и отчасти черчению (при использовании, например, редактора sPlan).

Представленная работа уменьшает существующую в нашем образовании диспропорцию между изучением программного и аппаратного обеспечения на уроках информатики (на изучение аппаратного обеспечения вместе с практикой по нему обычно выделяется недостаточно времени) и является особенно актуальной в инженерных и профильных классах.

Подробное описание в данной работе вариантов сборки триггеров должно помочь учителям и ученикам, мало знакомым с электроникой, в успешном проведении этих практических работ.

Работа фактически не зависит от используемого в школе учебника информатики и может стать первым шагом для дальнейшего применения более сложной электроники — Arduino и др.

Список использованных источников*

1. Горячев А. В. Информатика. 9 класс. Кн. 2. М.: Баласс, 2013.
2. Еремин Е. А. Популярная лекция об устройстве компьютера. СПб.: БХВ-Петербург, 2003.
3. Платт Ч. Электроника для начинающих / пер. с англ. 2-е изд. СПб.: БХВ-Петербург, 2017.
4. Поляков К. Ю. Информатика. 9 класс: учебник. М.: БИНОМ. Лаборатория знаний, 2016.
5. Поляков К. Ю. Информатика. Углубленный уровень. Учебник для 10 класса: в 2 ч. Ч. 1, 6-е изд. М.: БИНОМ. Лаборатория знаний, 2017.
6. Ревич Ю. В. Занимательная электроника. 2-е изд. СПб.: БХВ-Петербург, 2010.
7. Тули М. Карманный справочник по электронике / пер. с англ. М.: Энергоатомиздат, 1993.

Полезные ссылки

1. Сайт для поиска деталей и оборудования по минимальной цене: <http://www.chipfind.ru/>
2. Фирма Амперка: <http://amperka.ru/>
3. Сайт «Сотворим вместе»: <http://sotvorimvmeste.ru/>
4. О проводе и монтаже на макетной плате: <http://sotvorimvmeste.ru/viewtopic.php?f=12&t=131#p406>
5. Способ соединения гнезд на макетной плате: http://robocraft.ru/files/starter_kit/breadboard_wires/breadboard_cennection.jpg
6. Цветовая маркировка резисторов: <http://www.radiant.su/rus/articles/?action=show&id=335>.
7. Базовые логические элементы (И, ИЛИ, НЕ): <http://go-radio.ru/bazovie-logicheskie-elementi.html>
8. Что такое триггер: <https://ru.wikipedia.org/wiki/Триггер>
9. Сборка RS-триггера на микросхеме: <https://www.youtube.com/watch?v=RH7c0ZUB6GI>; <http://go-radio.ru/rs-trigger.html>
10. Портал для радиолобителей: <http://www.radioman-portal.ru/>

* В качестве ссылок на профильные учебники информатики для X класса здесь приведена ссылка только на учебник К. Ю. Полякова [5]. Это сделано для сокращения списка литературы, тем более что данная работа практически не зависит от используемого в школе учебника информатики.

Приложение

Контрольные вопросы и задания для проекта «Сборка RS-триггера на микросхемах, содержащих элементы «ИЛИ-НЕ», «И-НЕ»»

1. Как обозначают «ложь» (False) и «истину» (True) в алгебре логики? Какие значения может иметь логическая переменная? Нарисуйте модель логической переменной в виде переключателя.

2. Нарисуйте таблицу истинности логических функций И, ИЛИ, НЕ. Какой у них приоритет? Почему их называют полной (базовой) системой логических функций?

3. Нарисуйте схемы на выключателях, реализующих функции И, ИЛИ, НЕ.

4. Нарисуйте отечественные и зарубежные обозначения на схемах элементов И, ИЛИ, НЕ. Как называются эти элементы?

5. Нарисуйте таблицу истинности логических функций «штрих Шеффера» и «стрелка Пирса». Почему их называют полной (базовой) системой логических функций?

6. Как соединить элементы, реализующие логическую функцию «штрих Шеффера», чтобы получить конъюнктор, дизъюнктор и инвертор?

7. Как соединить элементы, реализующие логическую функцию «стрелка Пирса», чтобы получить конъюнктор, дизъюнктор и инвертор?

8. Каковы отечественные и импортные обозначения вентиля «И-НЕ», «ИЛИ-НЕ»?

9. Каковы уровни напряжений ТТЛ-логики? Каков запрещенный диапазон напряжений? Почему для логических 0 и 1 выделены не два конкретных напряжения, а целые диапазоны напряжений?

10. Что такое инверсная логика? Как обозначаются инверсные входы на схемах?

11. Каковы правила нумерации выводов микросхем? Что такое ключ микросхемы? Куда подключаются выводы 7 и 14 ТТЛ-микросхем?

12. Что такое светодиод? Как он обозначается на схемах? Что означает полярность включения светодиода? Какой потенциал должен иметь длинный вывод светодиода по отношению к короткому выводу? Можно ли включать светодиод без ограничивающего сопротивления?

13. Как маркируются малогабаритные резисторы? Определите номинал какого-либо резистора. Как связана величина сопротивления резистора с его номиналом и погрешностью (например, для резистора в 1 кОм и погрешности 5%)? Сколько Ом в 1 кОм, в 1 МОм? Как определить величину сопротивления резистора по закону Ома?

14. Как устроена макетная плата без пайки? Какие гнезда соединены между собой? Какие гнезда подключены к плюсу питания, а какие к нулевой шине? Какого цвета провода надо использовать для подключения к плюсу питания, а какие — для подключения к нулевой шине? Как еще называются и обозначаются плюс питания и нулевая шина?

15. Что такое триггер? Что является электромеханическим аналогом триггера? Когда и кем он был изобретен? Для чего применяется триггер? На каких электронных элементах его можно собрать? Как обычно называют триггер на английском языке?

16. Что представляет собой RS-триггер? Как он обозначается на схеме? Что обозначают кружки на выходе и входах? Какие еще бывают типы триггеров?

17. Сколько надо триггеров, чтобы можно было хранить 1 байт, 1 Кбайт?

18. Объясните принцип работы RS-триггера по его функциональным схемам для вентилях «ИЛИ-НЕ», «И-НЕ». Что будет на выходах Q и «НЕ Q » при подаче одновременно на входы R и S активного сигнала?

19. По цоколевке микросхемы с вентилями «ИЛИ-НЕ» нарисуйте электрическую принципиальную схему триггера.

20. По цоколевке микросхемы с вентилями «И-НЕ» нарисуйте электрическую принципиальную схему триггера.

21. Почему неиспользуемые входы рабочих вентилях не должны «висеть в воздухе»? Куда их рекомендуется подключать и через какое сопротивление?

22. Когда к схеме можно подключать питание? Что надо сделать с питанием, если схема не работает или работает неправильно?

23. Как измерить напряжение питания и величину сопротивления резистора цифровым мультиметром? Почему надо сначала отключить резистор от схемы, чтобы измерить его сопротивление?

НОВОСТИ

Две золотые и две серебряные медали завоевали российские школьники на Международной олимпиаде по информатике

5 сентября стали известны результаты 29-й Международной олимпиады по информатике (International Olympiad in Informatics — IOI), которая проходила в Цукубе (Япония). Российская сборная в составе четырех человек завоевала две золотые и две серебряные медали. Всего за призовые места на IOI-2018 состязались более 900 школьников из 85 стран мира.

Золотые медали на олимпиаде завоевали Рамазан Рахматуллин из Республики Татарстан и Владимир Романов из Москвы. Серебряные медали получили Михаил Анопренко из Санкт-Петербурга и Егор Лифарь из Москвы.

Министр просвещения Российской Федерации О. Ю. Васильева поздравила участников сборной и их наставников.

— Наша сборная по информатике возвращается домой после успешного выступления. Ребята завоевали четыре медали, показав высокий уровень владения информатикой. Когда цифровизация проникает во все сферы жизни человека, виртуозное владение навыками программирования имеет важнейшее значение. Про это

(По материалам, предоставленным пресс-службой Министерства просвещения РФ)

образно говорил академик Виктор Михайлович Глушков: без умения пользоваться компьютером человек XXI века будет «подобен человеку XX века, не умевшему ни читать, ни писать». И наша команда, безусловно, подтвердила свою высокую компьютерную грамотность. Важно, что ребята, не забывая про традиционное обучение, активно развиваются в той области, которая реально формирует наше будущее. И в этом заключается успех преемственности и последовательного развития российского образования. Может быть, именно сегодняшние победители олимпиады станут авторами тех прорывов, которые определяют развитие страны в самых разных сферах. Еще раз поздравляю ребят и их наставников и желаю им дальнейших успехов! — сказала глава Минпросвещения России.

Руководителем сборной выступил А. В. Малеев, проректор по международным программам и технологическому предпринимательству Московского физико-технического института. Главными тренерами сборной стали М. Тихомиров и А. Станкевич.

Детские технопарки «Кванториум» откроются в регионах к 2024 году

Детские технопарки «Кванториум» откроются в каждом регионе РФ с численностью населения более 60 тысяч жителей в рамках национального проекта в сфере образования, сообщила и. о. директора департамента государственной политики в сфере общего образования Минобрнауки Жанна Садовникова.

Национальный проект в сфере образования был утвержден майским указом президента в 2018 году. Планируется, что его реализация охватит период до 2024 года. В нем указывается на необходимость обеспечить глобальную конкурентоспособность российского образования, а также вхождение России в число 10 ведущих стран мира по качеству общего образования.

«В рамках реализации национального проекта в сфере общего образования мы продолжим реализацию успешных направлений — это сеть детских технопарков «Кванториум». И к концу реализации национального

проекта они будут доступны каждому региону численностью населения более 60 тысяч человек», — сказала Садовникова на съезде представителей общественно-профессиональных сообществ учителей и преподавателей учебных предметов в Российской академии образования.

Помимо этого, по ее словам, будет запущена полномасштабная работа с детьми вне школы — во всех регионах будут созданы Центры выявления и поддержки одаренных детей по опыту «Сириуса».

Президент РФ Владимир Путин 1 марта 2018 года в послании Федеральному Собранию заявил о необходимости продолжения укрепления целостной системы поддержки и развития творческих способностей и талантов детей. Такая система должна охватить всю территорию страны, интегрировать возможности таких площадок, как «Сириус», «Кванториумы», центры дополнительного образования и детского творчества во всех регионах России.

(По материалам «РИА Новости»)

И. В. Заика,

Таганрогский институт имени А. П. Чехова (филиал) Ростовского государственного экономического университета (РИНХ)

ПРАКТИЧЕСКИЕ РАБОТЫ ПО СОЗДАНИЮ ПРИКЛАДНЫХ ПРОГРАММ В VISUAL C++*

Аннотация

В статье представлены практические работы по созданию прикладных программ в среде Visual C++, в частности, рассматриваются примеры создания калькулятора с помощью Visual C++. Практические работы могут быть использованы при изучении объектно-ориентированного программирования в школьном курсе информатики.

Ключевые слова: программирование, Visual C++, создание прикладных программ, калькулятор.

DOI: 10.32517/2221-1993-2018-17-7-26-31

Контактная информация

Заика Ирина Викторовна, канд. тех. наук, доцент, доцент кафедры информатики Таганрогского института имени А. П. Чехова (филиала) Ростовского государственного экономического университета (РИНХ); *адрес:* 347936, Ростовская область, г. Таганрог, ул. Инициативная, д. 48; *телефон:* (8634) 60-18-07; *e-mail:* irin-zaik@yandex.ru

I. V. Zaika,

Taganrog Institute named after A. P. Chekhov (branch) of Rostov State University of Economics

PRACTICAL WORKS FOR CREATING APPLICATION PROGRAMS IN VISUAL C++

Abstract

The article presents practical works on creating application programs in the Visual C++ environment. In particular, examples of how to create a calculator using the Visual C++ environment are considered. Practical works can be used in the study of object-oriented programming in the school course of informatics.

Keywords: programming, Visual C++, creating application programs, calculator.

С помощью методов и средств объектно-ориентированного программирования можно осуществлять моделирование различных объектов и процессов [1, 3, 4], интересных для школьников. Разработка визуальных прикладных программ позволяет повысить мотивацию, а также познавательную активность учащихся.

Ниже представлены практические работы для пошагового создания приложения «Калькулятор» средствами Visual C++, которые могут быть использованы при изучении объектно-ориентированного программирования в школьном курсе информатики [2, 5]. Все программы создаются в Microsoft Visual C++ 2008 Express Edition.

Практическая работа 1 «Разработка программы простых вычислений средствами Visual C++»

Задание.

Создать программу для:

- перевода десятичного целого числа в двоичную систему счисления;
- вычисления факториала;
- вычисления значения тригонометрической функции $\sin(x)$.

При запуске программы должно появляться окно, где при нажатии на форме одной из трех кнопок реализуются соответствующие вычисления (рис. 1).

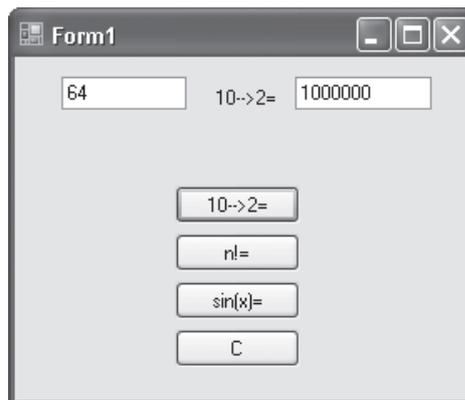


Рис. 1

Решение.

1. Запустить Microsoft Visual C++ 2008 Express Edition.
2. Создать новый проект. Для создания нового проекта необходимо:
 - выбрать в меню *File* пункт *New Projects*;
 - в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;
 - на закладке *Templates* выбрать *Windows Forms Application*;
 - в строке *Name* ввести имя, например *kalkulyat*;
 - щелкнуть мышью на кнопке *OK*.
3. Откроется форма *Form1*.
4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox (Текстовое поле)* и перетянуть *textBox1* на форму *Form1*, затем разместить *textBox2* таким об-

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_7-2018/

разом, чтобы *textBox1* и *textBox2* находились на одной линии (см. рис. 1). Обратите внимание: название объектов (компонент), перетянутых на форму, отличается от названия элементов, указанных на панели инструментов *Toolbox*. Например, на панели инструментов *Toolbox* элемент *TextBox* начинается с прописной буквы, а после размещения на форму *Form1* объекта *textBox1* название будет начинаться со строчной буквы.

5. На панели инструментов *Toolbox* во вкладке *Alle Windows Forms* выбрать элемент *Label (Метка)* и разместить *label1* на форме.

Чтобы обращаться к элементам из *Toolbox*, каждому объекту (например, *textBox1*, *textBox2* или *label1* и т. д.) присваивается уникальное имя или номер. Каждый

объект можно охарактеризовать присущим именно ему набором свойств (*Properties*), а также объект может реагировать на события (*Events*).

6. Для *label1* нужно назначить текст «=».

Для этого необходимо в меню *View* выбрать *Other Windows, Properties Window*. Откроется окно *Properties* для объекта *label1*. В окне *Properties* с помощью меню *Properties* и *Events* можно соответственно менять свойства объектов и назначать события для объектов. Выделить объект *label1* и в окне *Properties* в меню *Properties* назначить текст «=» в свойстве *Text*.

Для реализации программы *kalkulyat* в таблице 1 приводится пошаговая инструкция по написанию кода программы.

Таблица 1

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Перевод десятичного целого числа в двоичную систему счисления			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 10->2=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=((button^ sender) ->Text; long a,d, ostatok, razryad, dvoichnoe_chislo; a=Convert::ToInt32(textBox1->Text); d=a; ostatok=0; razryad=1; dvoichnoe_chislo=0; while(d!=0){ostatok=a%2; dvoichnoe_chislo=dvoichnoe_chislo+ostatok*razryad; razryad=razryad*10; d=a/2; a=d;} textBox2->Text=dvoichnoe_chislo.ToString();</code>
Вычисление факториала n!			
<i>button2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: n!=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=((button^ sender) ->Text; int k=1; for (int i=1; i<=Convert::ToInt32(textBox1->Text); i++) k=k*i; textBox2->Text=k.ToString();</code>
Вычисление значения тригонометрической функции sin(x) (для работы с математическими функциями необходимо подключение библиотеки <Math.h>)			
<i>button3</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: sin(x)=
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=((button^ sender) ->Text; float k; k=sin(Convert::ToDouble(textBox1->Text)); textBox2->Text=k.ToString();</code>
<i>button4</i> (кнопка сброса — при нажатии на кнопку <i>button4</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: С
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>textBox1->Text=""; textBox2->Text=""; label1->Text="";</code>

7. Сохранить программу *kalkulyat*. Для этого в меню *File* выбрать *Save All*.

8. Запустить программу *kalkulyat*. Для этого в меню *Debug* выбрать *Start Debugging*.

Ниже представлен фрагмент листинга программы *kalkulyat*.

```
private: System::Void button1_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
long a,d,ostatok,razryad,dvoichnoe_chislo;
a=Convert::ToInt32(textBox1->Text); d=a;
ostatok=0;
razryad=1;
dvoichnoe_chislo=0;
while(d!=0) {ostatok=a % 2; dvoichnoe_chislo=
dvoichnoe_chislo+ostatok*razryad;
razryad=razryad*10;
d=a/2;
a=d;}
textBox2->Text=dvoichnoe_chislo.ToString();}

private: System::Void button2_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
int k=1;
for(int i=1; i<=Convert::ToInt32(textBox1
->Text); i++) k=k*i;
textBox2->Text=k.ToString();}

private: System::Void button3_Click
(System::Object^sender, System::EventArgs^e)
{label1->Text=((button^sender)->Text;
float k;
k=sin(Convert::ToDouble(textBox1->Text));
textBox2->Text=k.ToString();}

private: System::Void button4_Click
(System::Object^sender, System::EventArgs^e)
{textBox1->Text="";
textBox2->Text="";
label1->Text="";}
```

Полный листинг программы приведен в *Приложении 1*, которое можно скачать на сайте ИНФО.

Далее рассматривается возможность создания кнопочного калькулятора с помощью Visual C++. При разработке калькулятора необходимо учесть все нюансы при работе с целыми и вещественными типами данных в C++ [6–8]. В частности, при реализации частного от деления двух чисел все данные должны быть вещественного типа, в противном случае результат от деления будет выводиться только как целое число. Поэтому необходимо уметь оперировать не только целыми, но и вещественными типами данных [10].

Практическая работа 2 «Создание приложения Калькулятор средствами Visual C++»

Задание.

Создать форму по образцу (рис. 2).

Решение.

1. Запустить Microsoft Visual C++ 2008 Express Edition.
2. Создать новый проект. Для создания нового проекта необходимо:

- выбрать в меню *File* пункт *New Projects*;
- в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;

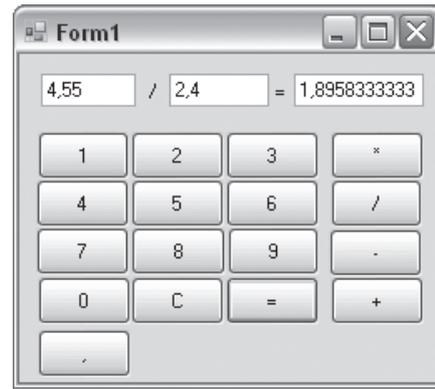


Рис. 2. Калькулятор

- на закладке *Templates* выбрать *Windows Forms Application*;
- в строке *Name* ввести имя, например *kalkulyator*;
- щелкнуть мышью на кнопке *OK*.

3. Откроется форма *Form1*.

4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox* (*Текстовое поле*) и разместить *textBox1* на форме *Form1*, затем на форму разместить *textBox2* и *textBox3* таким образом, чтобы они находились на одной линии (см. рис. 2).

5. Выбрать на панели инструментов *Toolbox* элемент *Label* (*Метка*) и перетащить *label1* на форму, чтобы метка была расположена между объектами *textBox1* и *textBox2*. Вторую метку *label2* расположить между *textBox2* и *textBox3* (образец представлен на рисунке 2).

В меню *View* выбрать *Other Windows, Properties Window*. Откроется окно *Properties*. Выделить объект *label2* и в окне *Properties* в меню *Properties* назначить текст «=» в свойстве *Text*. Выделить *label1* и в окне *Properties* в меню *Properties* очистить текст в свойстве *Text*.

6. Выбрать в *Toolbox* элемент *Button* (*Кнопка*) и перетащить *button1* на форму. Скопировать 16 раз кнопку *Button* на форму *Form1*.

7. Переименовать кнопки — соответственно на 0, 1, 2 и т. д.

Для реализации программы *kalkulyator* в таблице 2 приводится пошаговая инструкция по написанию кода программы.

8. Сохранить программу *kalkulyator*. Для этого в меню *File* выбрать *Save All*.

9. Запустить программу *kalkulyator*. Для этого в меню *Debug* выбрать *Start Debugging*.

Полный листинг программы приведен в *Приложении 2*, которое можно скачать на сайте ИНФО.

Для **самостоятельной работы** можно предложить дополнительное задание, которое позволит закрепить изученный материал.

Задание.

Добавить кнопку для возведения числа в степень: N^m .

Решение.

В C++ для возведения N в степень m используется функция *pow*(N, m).

Для обращения к функции возведения в степень необходимо в самом начале программы подключить библиотеку функций `<math.h>`.

1. Запустить Microsoft Visual C++ 2008 Express Edition.

Таблица 2

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Чтобы после ввода цифр в <i>textBox1</i> осуществлялся ввод во вторую компоненту — <i>textBox2</i> , необходимо перед областью описания класса в редакторе кода формы ввести логическую переменную <code>bool Z = false</code>			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Очистить текст
<i>label2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 1
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>if (Z==false)textBox1->Text+= ((button^)sender)->Text; else textBox2->Text+= ((button^)sender)->Text;</code>
Переименовать кнопки <i>button2</i> , <i>button3</i> , ..., <i>button11</i> соответственно в 2, 3, ..., 0, «,»			
<i>button2</i> , ... <i>button11</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 2, ..., 0, «,»
	<i>Events</i>	<i>Click</i>	Для каждой кнопки <i>button2</i> , <i>button3</i> , ..., <i>button11</i> назначить событие кнопки <i>button1</i> Чтобы назначить событие кнопки <i>button1</i> на другие кнопки, нужно в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button2</i> , <i>button3</i> , ..., <i>button11</i> выбрать из списка код: <code>button1_Click</code>
<i>button12</i> (кнопка для умножения)	<i>Properties</i>	<i>Text</i>	Ввести текст: *
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text=((button^)sender)->Text; Z=true;</code>
<i>button13</i> , <i>button16</i> , <i>button17</i> (кнопки для сложения, вычитания, деления соответственно)	<i>Properties</i>	<i>Text</i>	Соответственно ввести текст: +, -, /
	<i>Events</i>	<i>Click</i>	Нужно назначить событие кнопки <i>button12</i> на другие кнопки с арифметическими операциями. Для этого в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button13</i> , <i>button16</i> , <i>button17</i> выбрать из списка код: <code>button12_Click</code>
<i>button14</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>double k; switch (label1->Text[0]) {case '*': k=Convert::ToDouble(textBox1->Text) * Convert::ToDouble(textBox2->Text); break; case '/': k=Convert::ToDouble(textBox1->Text) / Convert::ToDouble(textBox2->Text); break; case '+': k=Convert::ToDouble(textBox1->Text) + Convert::ToDouble(textBox2->Text); break; case '-': k=Convert::ToDouble(textBox1->Text) - Convert::ToDouble(textBox2->Text); break; } textBox3->Text=k.ToString();</code>
<i>button15</i> (кнопка сброса — при нажатии на кнопку <i>button15</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: C
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>Z=false; textBox1->Text=""; textBox2->Text=""; textBox3->Text=""; label1->Text="";</code>

2. Создать новый проект. Для создания нового проекта необходимо:

- выбрать в меню *File* пункт *New Projects*;
- в открывшемся окне на закладке *Projects types* выбрать тип проекта *CLR*;
- на закладке *Templates* выбрать *Windows Forms Application*;
- в строке *Name* ввести имя, например *Pow1*;
- щелкнуть мышью на кнопке *OK*.

3. Откроется форма *Form1*.

4. В меню *View* открыть панель инструментов *Toolbox*, во вкладке *Alle Windows Forms* выбрать элемент *TextBox* (*Текстовое поле*) и разместить *textBox1*

на форме *Form1*, затем на форму разместить *textBox2* и *textBox3* таким образом, чтобы они находились на одной линии.

5. Выбрать на панели инструментов *Toolbox* элемент *Label* (*Метка*) и перетащить *label1* на форму, чтобы метка была расположена между объектами *textBox1* и *textBox2*. Вторую метку *label2* расположить между *textBox2* и *textBox3*.

Для реализации программы *Pow1* в таблице 3 приводится пошаговая инструкция по написанию кода программы.

6. Сохранить программу *Pow1*. Для этого в меню *File* выбрать *Save All*.

Таблица 3

Выделенный объект	Окно <i>Properties</i>	Свойство или событие	Описание действий для выделенного объекта
Разместить на форме компоненты <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> . Чтобы после ввода цифр в <i>textBox1</i> осуществлялся ввод во вторую компоненту — <i>textBox2</i> , необходимо перед областью описания класса в редакторе кода формы ввести логическую переменную <code>bool Z = false</code>			
<i>label1</i>	<i>Properties</i>	<i>Text</i>	Очистить текст
<i>label2</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
<i>button1</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 1
	<i>Events</i>	<i>Click</i>	Вести текст в редактор кода: <code>if (Z==false) textBox1->Text+= ((button^ sender) ->Text);</code> <code>else textBox2->Text+= ((button^ sender) ->Text);</code>
Переименовать кнопки <i>button2</i> , <i>button3</i> , ..., <i>button10</i> , соответственно в 2, 3, ..., 0			
<i>button2</i> , ... <i>button10</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: 2, ..., 0
	<i>Events</i>	<i>Click</i>	Для каждой кнопки <i>button2</i> , <i>button3</i> , ..., <i>button10</i> назначить событие кнопки <i>button1</i> Нужно назначить событие кнопки <i>button1</i> на другие кнопки. Для этого в меню <i>Events</i> напротив вкладки <i>Click</i> для кнопок <i>button2</i> , <i>button3</i> , ..., <i>button10</i> выбрать из списка код: <code>button1_Click</code>
<i>button11</i> (кнопка для возведения в степень)	<i>Properties</i>	<i>Text</i>	Ввести текст: P
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>label1->Text= ((button^ sender) ->Text); Z=true;</code>
При нажатии на кнопку <i>button12</i> происходит возведение в степень целых чисел. Число <i>N</i> вводится в <i>textBox1</i> , число <i>m</i> — в <i>textBox2</i> , и результат вычисления выводится в <i>textBox3</i>			
<i>button12</i>	<i>Properties</i>	<i>Text</i>	Ввести текст: =
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>double k;</code> <code>switch (label1->Text [0])</code> <code>{case 'P': k=pow(Convert::ToInt32(textBox1->Text),</code> <code>Convert::ToInt32(textBox2->Text));break;}</code> <code>textBox3->Text=k.ToString();</code>
<i>button13</i> (кнопка сброса — при нажатии на кнопку <i>button13</i> будет очищен текст в <i>textBox1</i> , <i>textBox2</i> , <i>textBox3</i> , <i>label1</i>)	<i>Properties</i>	<i>Text</i>	Ввести текст: C
	<i>Events</i>	<i>Click</i>	Ввести текст в редактор кода: <code>Z=false;</code> <code>textBox1->Text="";</code> <code>textBox2->Text="";</code> <code>textBox3->Text="";</code> <code>label1->Text="";</code>

7. Запустить программу *Pow1*. Для этого в меню *Debug* выбрать *Start Debugging*.

* * *

Создание прикладных программ в средах объектного программирования развивает у учащихся умения самостоятельно переносить знания и умения в новую ситуацию, видеть новую проблему в знакомой ситуации, устанавливать новые свойства объекта изучения. Разработка калькулятора помогает начинающему пользователю разобраться с особенностями работы с разными типами представления данных в C++, познакомиться с библиотеками математических функций и их подключением при разработке прикладных приложений [5–7, 9].

Обучение программированию в средах объектно-ориентированных языков помогает достигнуть эффективного результата в формировании основных познавательных процессов у школьников: воображения, памяти, абстрактного и логического мышления.

Список использованных источников

1. Айра П. Объектно-ориентированное программирование на C++. М.: БИНОМ; Невский Диалект, 2001.

2. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технология разработки программного обеспечения: учебное пособие для СПО. М.: Юрайт, 2018.

3. Зыков С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для академического бакалавриата. М.: Юрайт, 2018.

4. Комлев Н. Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей. М.: СОЛОН-Пресс, 2014.

5. Лафоре Р. Объектно-ориентированное программирование в C++. СПб.: Питер, 2015.

6. Лесневский А. С. Объектно-ориентированное программирование для начинающих. М.: БИНОМ. Лаборатория знаний, 2010.

7. Огнева М. В., Кудрина Е. В. Программирование на языке C++: практический курс: учебное пособие для СПО. М.: Юрайт, 2018.

8. Павловская Т. А. Процедурное и объектно-ориентированное программирование. C/C++: учебник. СПб.: Питер, 2015.

9. Трофимов В. В., Павловская Т. А. Алгоритмизация и программирование: учебник для академического бакалавриата / под ред. В. В. Трофимова. М.: Юрайт, 2018.

10. Тузовский А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата. М.: Юрайт, 2018.

НОВОСТИ

HP создала симуляцию марсианского поселения в виртуальной реальности

HP объединила силы с Nvidia и мировым творческим сообществом, чтобы реализовать проект HP Mars Home Planet — масштабную VR-симуляцию того, как могла бы выглядеть жизнь одного миллиона поселенцев на Марсе. Виртуальная симуляция HP Mars Home Planet — кульминация годового сотрудничества с дизайнерами, инженерами и изобретателями, в результате которого была смоделирована жизнь на Красной планете.

Разработка проекта привлекла более 90 тыс. представителей различных творческих профессий, архитекторов, инженеров и студентов из более чем 150 стран. Самые инновационные идеи были реализованы в виртуальной реальности от Technicolor, которая будет представлена на SIGGRAPH, крупнейшей в мире конференции, посвященной компьютерной графике.

«Максимальный эффект погружения в этой VR-симуляции достигнут благодаря кооперации творческого сообщества и технологий, которые предоставила HP и наши партнеры, — сказала Гвен Кобл (Gwen Coble), директор направления рабочих станций, тонких клиентов, решений для розничной торговли и иммерсивных вычислений в регионе EMEA, HP Inc. — Передовые технологии и богатая история инноваций HP в сочетании с творческим подходом позволяют воплотить в жизнь самые потрясающие идеи».

Виртуальная симуляция программы HP Mars Home Planet демонстрирует центр адаптации новых поселенцев Марса, в котором показаны инновации в сферах архитектуры, техники и транспорта, которые позволяют людям осваивать Марс. Компания Technicolor разработала визу-

альные образы с помощью игрового движка Unreal Engine Epic Games, в результате чего была создана первая в мире VR-программа с шестью степенями свободы для игровых кресел с поддержкой движения и гарнитур HP Windows Mixed Reality. Окружающая среда и основные объекты из VR будут доступны творческому сообществу уже этой осенью. Их можно будет скачать через Unreal Engine.

«Мы живем в интересное время, когда технологический прогресс идет рука об руку с множеством фундаментальных научных и планетарных исследований. Надеемся, что это сочетание приблизит тот момент, когда человек начнет осваивать Марс, — сказала доктор Дарлин Лим (Darlene Lim), научный руководитель проекта NASA Biologic Analog Science. — В наши дни люди по всему миру мечтают об инновациях, которые сделают такое будущее возможным, а этот удивительный проект — своего рода виртуальное окно, через которое мы можем увидеть, как будет выглядеть жизнь на Марсе для миллиона поселенцев».

Университеты и школы по всему миру также подхватили инициативу изобретений, проектирования и моделирования жизни в виртуальном поселении на Марсе. При поддержке технологий HP и ведущих специалистов в сфере образования в этих учреждениях были созданы команды энтузиастов, которые вошли в образовательную программу HP Mars Home Planet. В рамках этого проекта соревновались 10 европейских и два австралийских университета, а в Японии был проведен отдельный конкурс со 110 участниками. В результате были выбраны победители, чьи работы вошли в проект HP Mars Home Planet.

(По материалам CNews)



И. А. Тюшнякова,

Таганрогский институт имени А. П. Чехова (филиал)

Ростовского государственного экономического университета (РИНХ)

ИЗУЧЕНИЕ СОРТИРОВКИ ПОДСЧЕТОМ И СЛИЯНИЕМ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Аннотация

Статья посвящена вопросам изучения сортировки в школьном курсе информатики. Изложены модификации методов сортировки подсчетом и слиянием, характеризующиеся визуально понятным и наглядным представлением алгоритмов. Описана возможность познакомить учащихся с основами параллельного программирования при изучении данных методов сортировки.

Ключевые слова: сортировка подсчетом, сортировка слиянием, алгоритм, программирование.

DOI: 10.32517/2221-1993-2018-17-7-32-35

Контактная информация

Тюшнякова Ирина Анатольевна, канд. тех. наук, доцент, доцент кафедры информатики Таганрогского института имени А. П. Чехова (филиала) Ростовского государственного экономического университета (РИНХ); адрес: 347936, Ростовская область, г. Таганрог, ул. Инициативная, д. 48; телефон: (8634) 60-18-07; e-mail: Solovyova_Irina@mail.ru

I. A. Tyushnyakova,
Taganrog Institute named after A. P. Chekhov
(branch) of Rostov State University of Economics

STUDYING THE COUNTING SORT AND MERGE SORT IN THE SCHOOL COURSE OF INFORMATICS

Abstract

The article is devoted to the study of sorts in the school course of informatics. Modifications of the sorting and merging methods are described, characterized by a visually understandable and visual representation of the algorithms. An opportunity to acquaint students with the basics of parallel programming in the study of sorting data is described.

Keywords: counting sort, merge sort, algorithm, programming.

Построение алгоритмов, изучение языков программирования являются неотъемлемыми частями освоения курса информатики. В процессе программирования у учащихся развивается логическое и алгоритмическое мышление, вырабатываются творческие способности. Авторские программы по информатике и ИКТ для среднего общего образования различных авторов предполагают изучение алгоритмов сортировки в рамках раздела «Алгоритмы и элементы программирования».

Сортировка относится к одной из базовых процедур нечисловой обработки данных и заключается в упорядочении элементов в списках [1, 2]. Алгоритмы сортировки имеют достаточно широкий спектр применения и используются в разнообразных задачах, которые связаны с системами автоматизированного управления (САУ), информационно-поисковыми системами и т. д. Сортировка может применяться для эффективной обработки в больших наборах данных (например, для поиска); для представления массивов данных в удобной для анализа форме; с целью группировки элементов по определенному признаку и т. п. В работах [3, 4] нами было рассмотрено использование алгоритмов сортировки для решения задач вычислительной математики и распознавания изображений, в частности, подробно (с листингами и результатами численных экспериментов) были описаны авторские схемы применения сортировок для нахождения экстремумов, корней и полюсов функций с разнообразными приложениями.

В школьном курсе информатики учащиеся знакомят, как правило, с наиболее простыми (в плане программной реализации) и известными алгоритмами сортировки: методом выбора, методом обмена («пузырька»), быстрой сортировкой.

В качестве одного из интересных алгоритмов можно отметить один из частных случаев класса сортировки m -путевым слиянием — сортировку подсчетом.

Суть **сортировки подсчетом** состоит в определении номеров мест элементов массива, расстановка по которым даст отсортированный массив. Метод основан на том, что j -ключ упорядоченной последовательности больше $(j - 1)$ -ключа массива. Сортировка подсчетом представляется с помощью **матрицы сравнений**. Модификация метода, характеризующаяся визуально понятным, наглядным представлением алгоритма, разработана доктором технических наук, профессором Я. Е. Роммом [5].

Рассмотрим массив неупорядоченных чисел $A = (1, -5, 7, -1, 4, -5)$, состоящий из $N = 6$ элементов, расположим его в первой строке и в первом столбце матрицы сравнений (рис. 1).

Матрица сравнений заполняется знаком разности элементов, расположенных горизонтально в строке и вертикально в столбце. С целью определения места элемента в отсортированном массиве следует просуммировать число нулей и плюсов в каждом столбце матрицы сравнений над диагональю (вместе с диагональным элементом) и сложить получившееся

	1	-5	7	-1	4	-5
1	0	-	+	-	+	-
-5	+	0	+	+	+	0
7	-	-	0	-	-	-
-1	+	-	+	0	+	-
4	-	-	+	-	0	-
-5	+	0	+	+	+	0
	↓	↓	↓	↓	↓	↓
	4	1	6	3	5	2

Рис. 1. Матрица сравнений для массива A = (1, -5, 7, -1, 4, -5)

число с количеством только плюсов ниже диагонали. Полученные значения запоминаются в счетчик k, значение k получается адресом вставки A1[k] = A[j], при этом можно запомнить и номер элемента в исходном массиве (входной индекс вставленного элемента) e[k] = j. На выходе сортировки A1[k] и e[k] имеют равный номер k.

На рисунке 1 подсчитанные номера элементов в отсортированном массиве представлены ниже таблицы цифрами рядом со стрелками. Структурная схема алгоритма представлена на рисунке 2.

Программа, реализующая сортировку подсчетом на языке Pascal, представлена на рисунке 3.

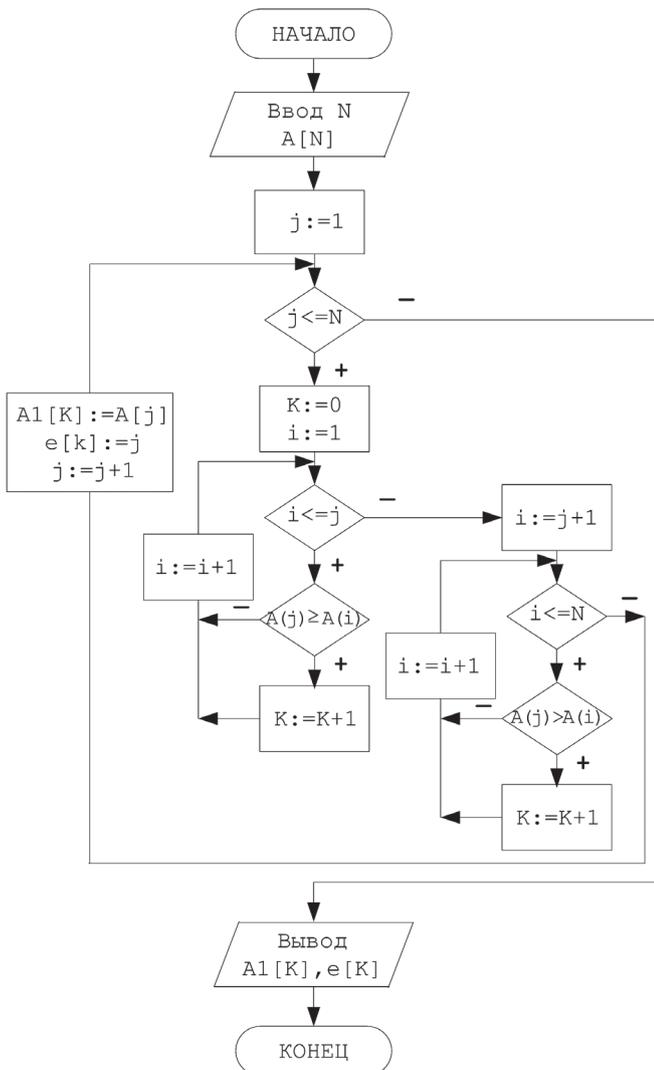


Рис. 2. Структурная схема алгоритма сортировки подсчетом

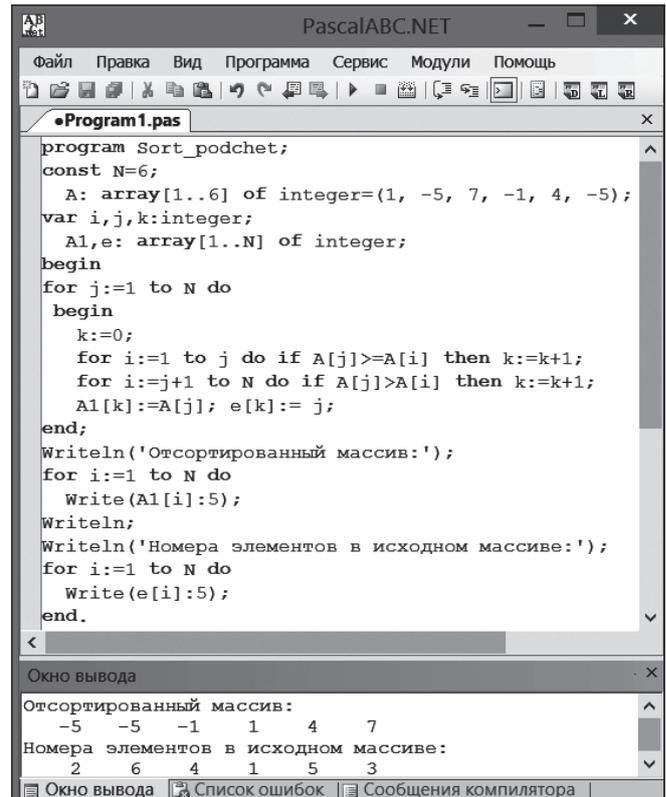


Рис. 3. Реализация сортировки подсчетом в среде PascalABC.NET

Данная сортировка допускает эффективное распараллеливание с единичной оценкой временной сложности. Эта оценка получается из следующих соображений. На основании взаимной независимости элементов матрицы сравнений можно установить знак всех ячеек матрицы за время одного сравнения. В силу известной диагонали (с нулевыми элементами) и симметричности матрицы сравнений для этого потребуется количество процессоров $\frac{N^2 - N}{2} \leq \frac{N^2}{2}$; время выполнения шага $O(1)$, т. е. $T\left(\frac{N^2}{2}\right) = O(1)$, где N — число сортируемых элементов.

Помимо сортировки подсчетом интерес может представлять сортировка слиянием. Она основана на адресном слиянии двух ранжированных массивов по матрице сравнений.

В качестве примера на рисунке 4 приведена матрица сравнений для двух сливаемых массивов a = (2, 3, 13, 14) и b = (3, 4, 4 5). Массив a располагается в матрице сравнений горизонтально сверху, массив b — вертикально

	-∞	2	3	13	14	∞
-∞	0	+	+	+	+	+
3	-	→	→	0	+	+
4	-	-	→	→	+	+
4	-	-	-	→	+	+
5	-	-	-	-	→	+
∞	-	-	-	-	-	0

Рис. 4. Матрица сравнений для слияния двух массивов с указанием направления обхода

слева, к матрице сравнений добавляются окаймляющие строки и столбцы. Ячейки матрицы сравнений заполняются аналогично вышеописанной сортировке подсчетом.

Слияние двух массивов a и b организуется путем поэлементного обхода слева направо и сверху вниз по неотрицательной части матрицы сравнений. Направление обхода указано стрелками на рисунке 4. Выходной массив c формируется в соответствии со следующими соотношениями:

$$c_{i+j} = \begin{cases} b_i, & \text{если } \alpha_{ij} = -1, \alpha_{i(j+1)} \geq 0, \\ a_j, & \text{если } \alpha_{ij} \geq 0, \alpha_{(i+1)j} = -1, \end{cases} \quad \alpha_{ij} = \begin{cases} 1, & a_j > b_i, \\ 0, & a_j = b_i, \\ -1, & a_j < b_i. \end{cases}$$

Здесь:

i и j — произвольно взятые индексы при ограничениях $0 \leq i \leq n+1, 0 \leq j \leq m+1$;

n и m — количество элементов в ранжированных массивах b и a ;

α_{ij} — элемент i -й строки и j -го столбца матрицы сравнений.

Программная реализация слияния двух упорядоченных массивов на языке Pascal представлена на рисунке 6.

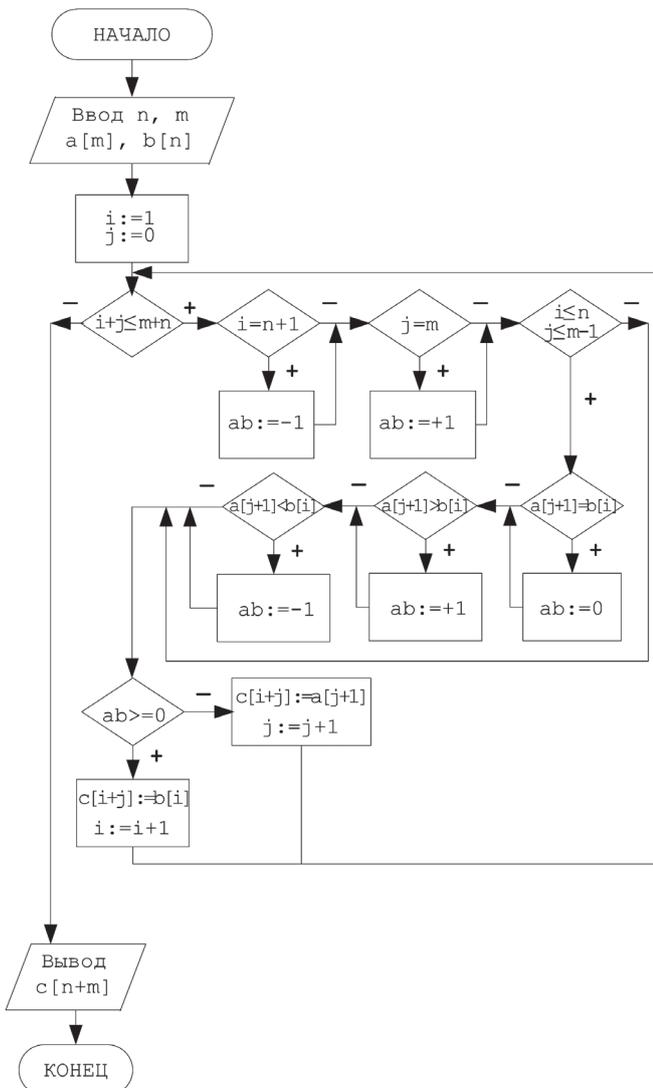


Рис. 5. Структурная схема слияния двух упорядоченных массивов

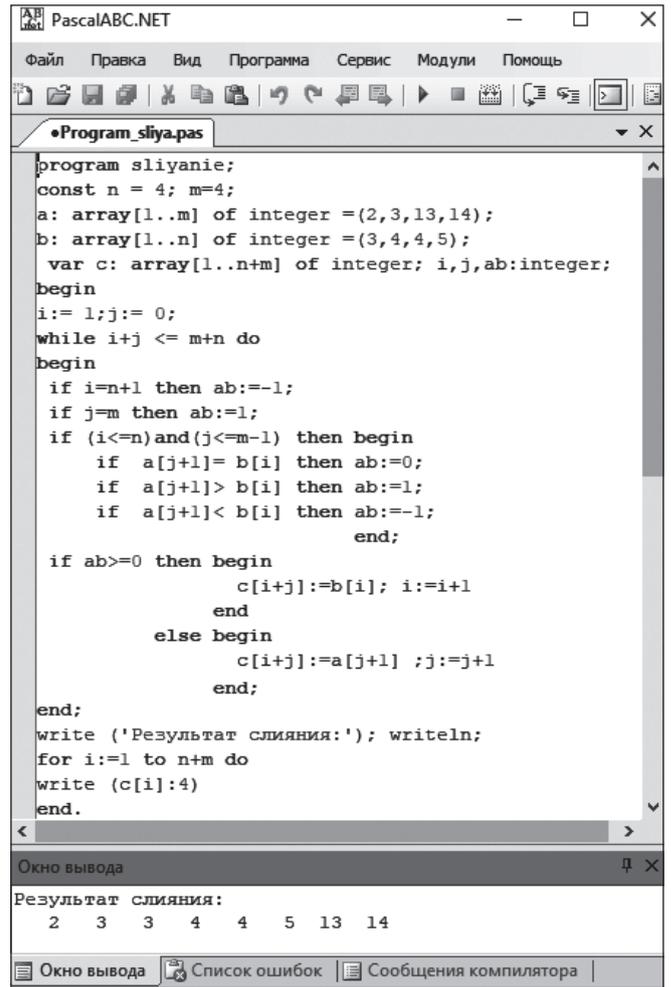


Рис. 6. Реализация слияния двух массивов в среде PascalABC.NET

Сортировка слиянием представляется в виде рекурсивного алгоритма, выполняющего свою работу, двигаясь вверх по рекурсии. При этом исходный неупорядоченный массив необходимо рассматривать как множество, состоящее из N подмножеств, каждое из которых состоит из одного элемента. Слияния начнутся с подмножеств, состоящих из одного элемента. Парно сливаются смежные подмножества, образуя подмножества из двух элементов, и так далее до тех пор, пока не образуется одно отсортированное множество. Количество подмножеств на каждом этапе слияния уменьшается вдвое, а их размеры удваиваются. Кратко охарактеризовать сортировку слиянием можно следующим образом: первоначально производится слияние каждой пары элементов с запоминанием обратных адресов, затем слияние упорядоченных пар, затем упорядоченных четверок и т. д. Полный листинг программы сортировки слиянием представлен в [3].

Время выполнения максимально *параллельно*-го варианта сортировки слиянием оценивается как $T\left(\frac{N^2}{4}\right) = O(\log_2 N)$, где $\frac{N^2}{4}$ — количество процессоров. *Последовательный* вариант сортировки слиянием выполняется за время $O(\log_2 N)$ [4]. Первая оценка получается из следующих соображений. Одно слияние на i -м шаге сортировки получается за время одного сравнения τ в силу взаимной независимости элементов матрицы сравнений. Такое слияние требует $(2^{i-1})^2$ про-

цессоров. Число взаимно независимых слияний на i -м шаге равно $\frac{N}{2^i}$, откуда число процессоров на шаге — $\frac{N}{2^i} \cdot (2^{i-1})^2 \leq \frac{N^2}{4}$; время выполнения шага $O(1)$. С учетом числа шагов $\log_2 N$ получается представленная оценка временной сложности параллельной сортировки слиянием.

Алгоритмы сортировки подсчетом и слиянием отличаются от известных аналогов параллелизмом, устойчивостью (сохранением порядка равных элементов), а также прямой и обратной адресностью.

Помимо визуально интересного представления алгоритмов сортировки подсчетом и слиянием их изучение в рамках уроков информатики либо факультативных или элективных курсов дает возможность познакомить учащихся с основами параллельного программирования. Представленные оценки временной сложности параллельного выполнения сортировки подсчетом и слиянием показывают возможность эффективного распараллеливания сортировки с помощью компьютер-

ных MIMD-систем*, когда мы имеем дело с несколькими процессорами, каждый из которых способен выполнять свою инструкцию, при этом существует несколько потоков данных и каждый процессор может работать со своим набором.

Список использованных источников

1. Вурт Н. Алгоритмы и структуры данных. М.: ДМК-Пресс, 2016.
2. Кнут Д. Искусство программирования. В 3 т. Т. 3. Сортировка и поиск (второе издание). М.: Вильямс, 2017.
3. Ромм Я. Е., Тюшнякова И. А. Применение сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений: учебное пособие. Таганрог: Изд-во Таганрог. гос. пед. ин-та, 2009.
4. Тюшнякова И. А. Разработка и исследование схем применения сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений: дис. ... канд. тех. наук. Таганрог, 2006.
5. Romm Ya. E. Sorting-based calculation of zeros and extrema of functions as applied to search and recognition // Cybernetics and Systems Analysis. 2001. Vol. 37. Is. 4.

* MIMD (англ. Multiple Instruction Stream, Multiple Data Stream — множественный поток команд, множественный поток данных) — концепция архитектуры компьютера, используемая для достижения параллелизма вычислений. Машины имеют несколько процессоров, которые функционируют асинхронно и независимо.

НОВОСТИ

«Траектория будущего» подготовила 150 ИТ-тьюторов в помощь московским учителям

Агентство «Траектория будущего» (АТБ) к началу нового учебного года подготовило команду из 150 ИТ-тьюторов, которые помогут московским учителям освоить новые технологии — инструменты для ведения качественного образовательного процесса.

Школьный ИТ-тьютор — это специалист, который помогает участникам образовательного процесса, преимущественно педагогам, адаптироваться к инновационным решениям в их профессиональной деятельности. Школьный ИТ-тьютор проводит консультации по использованию сетевой инфраструктуры, которая включает в себя электронный дневник, электронные учебники и другие функции. Также ИТ-тьютор знакомит преподавателя с возможностями интерактивных панелей, пришедших на смену классическим школьным доскам.

В школьной сетевой инфраструктуре хранится информация об успеваемости учеников и другие данные по учебному процессу. Благодаря консультациям ИТ-тьютора педагог с легкостью заходит в свой виртуальный рабочий кабинет, просматривает активность учеников, ведет отчетность по успеваемости, которая доступна для директора школы, родителя и ученика в режиме онлайн.

После работы с ИТ-тьютором педагог в полной мере использует возможности интерактивной панели, ориентируется в электронной библиотеке, самостоятельно готовит сценарии уроков, превращая занятие в увлекательный процесс и при этом экономя время на разработке темы.

ИТ-тьютор сопровождает педагога в течение такого количества времени, которое необходимо учителю для полноценной адаптации к новым решениям.

«У нашего проекта нет аналогов в России. Мы выявили потребность в консультантах по внедрению инновационных технологий в различные сферы и создали новую профессию — ИТ-тьютор, — сказала генеральный директор АТБ Юлиана Науменко. — Этой профессии не обучают в нашей стране. В университетах только появляются первые программы дополнительного обучения на позицию “тьютор”, в которой совмещены технологии, социология и психология. Но потребность в междисциплинарных специалистах актуальна уже сейчас. Внедрение инновационных технологий проходит массово, затрагивает многие сферы и огромные пласты населения, и для каждой профессии, будь то врач или учитель, нужна адаптация к новым инструментам работы. Пройти адаптацию в кратчайшие сроки, не нарушая рабочего процесса, помогает ИТ-тьютор».

В течение летнего периода 2018 года АТБ рассмотрели заявки 1,5 тыс. претендентов, желающих пройти стажировку на позицию «ИТ-тьютор». Отбор кандидатов проходил в несколько этапов: конкурс, тесты, обучение, аттестация. Среди соискателей отдавалось предпочтение студентам и недавним выпускникам с техническим или педагогическим образованием. Около 150 кандидатов прошли отбор и получили возможность освоить новую профессию на стыке наук.

(По материалам CNews)

С. Н. Лосик,

школа № 45 имени Л. И. Мильграма, г. Москва

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В НАЧАЛЬНОЙ ШКОЛЕ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ KODU GAME LAB

Аннотация

В статье рассматривается опыт преподавания пропедевтического курса параллельных вычислений в третьем классе общеобразовательной школы с использованием среды Kodu Game Lab. Данный курс опирается на УМК «Информатика для всех» для I–IV классов начальной школы Д. И. Павлова под редакцией А. В. Горячева. Приводятся примеры заданий, которые ученики выполняли на уроках.

Ключевые слова: параллельные вычисления, УМК по информатике, программирование, «Информатика для всех», статические объекты, динамические объекты.

DOI: 10.32517/2221-1993-2018-17-7-36-39

Контактная информация

Лосик Сергей Николаевич, учитель информатики школы № 45 имени Л. И. Мильграма, г. Москва; адрес: 117447, г. Москва, Новочеремушкинская ул., д. 11А; телефон: (499) 129-71-55; e-mail: losiksn@ms45.edu.ru

S. N. Losik,
School 45, Moscow

PROPAEDEUTICS OF PARALLEL COMPUTING IN THE PRIMARY SCHOOL USING THE KODU GAME LAB

Abstract

The article discusses the experience of teaching a propaedeutic course of parallel computing in the 3rd grade of primary school using the Kodu Game Lab environment. This course is based on the textbook set "Informatics for All" by D. I. Pavlov for grades 1–4 of primary school (edited by A. V. Goryachev). The examples of the tasks are given that students performed in the classroom.

Keywords: parallel computing, textbook set on informatics, programming, "Informatics for All", static objects, dynamic objects.

Истоки информатики традиционно принято искать в работах западных математиков, таких как Алонзо Черч, Алан Тьюринг и др. И конечно же, в выпущенной в 1948 году книге «Кибернетика» Норберта Винера. Первое время информатика рассматривалась как прикладная область математики и кибернетики. И только с развитием электронных вычислительных систем информатика получила признание как самостоятельная область научного знания и, как следствие, учебная дисциплина. В этой связи появление информатики в учебном процессе было направлено на развитие навыков автоматизации вычислений. Что, в свою очередь, положило начало такому разделу информатики, как программирование.

В СССР вопросы теоретической информатики на государственном уровне долгое время игнорировались, а разработки С. А. Лебедева и В. М. Глушкова в области создания вычислительных машин и автоматизации народного хозяйства, скорее, «терпели», нежели оказывали им существенную поддержку [11]. И только на фоне взрывообразного распространения средств автоматизации вычислений, сформировавшего социальный заказ на специалистов-программистов, учебная дисциплина «Информатика» появилась в школе.

Неудивительно, что информатика пришла в школу под лозунгом «Программирование — вторая грамотность». А содержательная линия программирования, в отличие от сформировавшихся позже линий формализации и моделирования, информационных технологий и иных, по сей день является одной из самых объемных в школьном курсе информатики. А. П. Ершов отмечал, что «грамотность и программирование не только выстраиваются в параллель, соединяясь мостиками аналогий, но и дополняют друг друга, формируя новое представление о гармонии человеческого ума» [3].

Но если относительно значимости линии программирования для курса информатики сомнений обычно не возникает, то возраст начала обучения программированию долгое время был предметом дискуссий. Тот же А. П. Ершов неоднократно ссылался по этому вопросу на работы Ж. Пиаже, отмечавшего, что любое знание связано с действием и, как следствие, любая деятельность может вызвать интерес ребенка и иметь для него смысл, если изучение объекта заменить его конструированием.

Соглашаясь с этой мыслью, А. П. Ершов замечал: «Хотелось бы подчеркнуть, что речь идет не о том, чтобы навязать детям новые, не свойственные им навыки и знания, а о том, чтобы проявить и сформулировать те стороны мышления и поведения, которые реально существуют, но формируются стихийно, неосознанно» [3].

Этими же идеями руководствовался и С. Пейперт, разработавший уникальный для своего времени язык LOGO. Этот язык был графичен и состоял из команд, понятных ученикам, команд, которые дети могли выполнить и сами, без компьютера. Кроме того, результатом работ также становилось что-то очень наглядное. Сам С. Пейперт воспринимал LOGO, скорее, как некую моделью языка, реализующего визуально-ориентированный подход, и признавал, что с течением времени появится множество более современных и более продуманных языковых концепций [10].

Время доказало правоту С. Пейперта. Сегодня школам предложен целый ряд различных учебных языков программирования, в основном пиктографического типа. Ю. А. Первин провел большую работу по сравнительному анализу подобных языков и отметил тенденцию к созданию языков для конструирования бестекстовых программ, что реализовано в средах «ПиктоМир», «ПервоЛого», Scratch, «Азбука Роботландии» [5].

Вопросам пропедевтики программирования в начальной школе уделяли внимание множество ведущих отечественных специалистов. Так, А. В. Горячев [2] и А. Л. Семенов [12] много работали над развитием у младших школьников основ алгоритмического мышления, Л. Л. Босова [1] уделила внимание изучению особенностей применения интерактивных сред, а А. В. Могилев [6] — вопросам геймификации при обучении программированию.

Между тем с момента введения ФГОС НОО подходы к обучению информатике на уровне начальной школы существенно пересматриваются [7]. И речь идет не только об обновлении целей и подходов к реализации курса [4], но и об обновлении содержания образования [9].

В вопросах раскрытия содержательной линии «Программирование» на уровне начального образования важно не только следовать положениям ФГОС НОО, но и учитывать социальный заказ общества системе образования. И в этой связи важно обратить внимание на некоторую однобокость реализации линии программирования, а именно на рассмотрение на уровне начального образования темы «Алгоритмы для конкретного исполнителя» в различных вариациях.

Программирование сегодня не укладывается только в рамки алгоритмизации «поведения» конкретного исполнителя. *Одна из главных тенденций современного программирования — параллельные вычисления.*

Под параллельными вычислениями мы будем подразумевать способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно).

Тема эта сложна и, разумеется, на уровне начального образования рассматривается упрощенно. Но, учитывая, что период 7–10 лет у ребенка мы уверенно можем считать сенситивным по отношению к обучению основам программирования, вовсе не рассматривать такую важную тему, как «Параллельные вычисления», мы не можем.

Однако на уровне начальной школы проблематика должна рассматриваться скорее как «алгоритмы, управляемые событиями». Основные задачи реализации этого раздела заключаются в формировании у учеников представления об алгоритмах, управляемых событиями, и о взаимодействии нескольких одновременно функционирующих алгоритмов. Также важно раскрыть основные сложности взаимодействия нескольких исполнителей, а именно подходы к обеспечению правильной последовательности взаимодействий между различными вычислительными процессами.

Первое представление об алгоритмах, управляемых событиями, ученики начальной школы, в логике С. Пейперта, получают еще при работе с простыми бытовыми алгоритмами вроде алгоритма перехода через дорогу. Именно события и поведение сторонних исполнителей (светофора и участников дорожного движения) определяют поведение программируемого исполнителя, превращая линейный алгоритм его поведения в условный и даже (при некотором стечении обстоятельств) циклический.

Однако одних бытовых примеров и графического отображения алгоритмических конструкций в тетради для правильного представления об алгоритмах, управляемых событиями, недостаточно.

Опираясь на мнение ведущих специалистов в области пропедевтического обучения программированию,

таких как Ю. А. Первин, Л. Л. Босова, А. В. Могилев, Н. К. Нателаури и ряда других, мы в школе № 45 имени Л. И. Мильграма (г. Москва) при выборе подходов к обучению младших школьников информатике приняли решение иметь обязательную компьютерную часть. Обучение информатике младших школьников на базе нашей школы организовано в форме внеурочных занятий. Дидактическая основа определяется УМК «Информатика для всех» Д. И. Павлова под редакцией А. В. Горячева. В соответствии с имеющейся программой реализация проблематики «Алгоритмы, управляемые событиями» с помощью компьютера с использованием среды Kodu Game Lab начинается в третьем классе.

Разработанная корпорацией Microsoft свободно распространяемая среда **Kodu Game Lab** ориентирована на применение технологии геймификации при обучении программированию. Среда Kodu разворачивается на персональных компьютерах или консолях XBOX.

Kodu Game Lab изначально предлагает обучающимся взглянуть на программирование с нетрадиционной стороны. Каждый пользовательский алгоритм формируется по схеме «условие — действие», что превращает алгоритм в алгоритм, наступлением которого управляет то или иное событие. А это говорит о том, что выбранная среда подходит для реализации заявленной задачи — *пропедевтики параллельных вычислений с помощью компьютера на уровне начальной школы.*

Создание любого проекта в среде Kodu состоит из двух основных частей:

- первая часть — *создание среды* (ландшафта). В этой части среда особо интересна ученикам, так как похожа на любимую ими компьютерную игру Minecraft;
- вторая часть создания проекта — *работа с объектами.*

Объекты в среде Kodu также условно делятся на две группы — объекты и исполнители.

Объекты статичны. Исполнители могут вступать с ними во взаимодействие, сами же объекты могут выполнять какие-то команды, но не могут перемещаться по миру.

Исполнители мобильны. Они также делятся на две группы, которые условно можно назвать аватары и NPC.

Под *аватаром* мы понимаем исполнителя, движениями и иными действиями которого частично или полностью руководит пользователь.

NPC (Non Playable Creature) — это исполнитель, способный взаимодействовать с пользователем или средой, управление им осуществляет компьютер.

Работа с исполнителями на уроке проводится сначала в тетради: ученики формулируют правила действия для исполнителей и записывают их в тетради. После этого ребята реализуют алгоритмы средствами среды Kodu.

Ниже приведены **примеры заданий, в рамках которых обучающиеся осваивают практическую составляющую темы «Алгоритмы, управляемые событиями»**. В этих заданиях параллельность проявляется через взаимодействие двух и более исполнителей, реализующих каждый свой алгоритм и имеющих область пересекающихся задач.

Задание 1. Эмоции и взаимодействия.

Это одно из первых заданий, когда обучающиеся еще не создают полноценных игр, а только осваивают некоторые элементы среды.

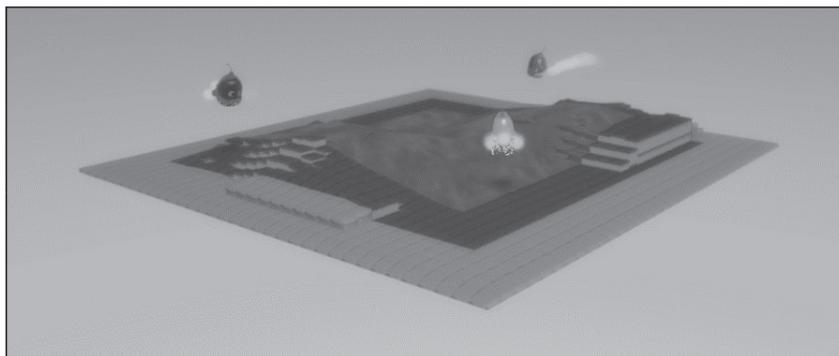


Рис. 1

Для начала ребята готовят простой «мир» и размещают в нем трех исполнителей (рис. 1).

Ученики получают задачу преобразовать двух исполнителей в NPC, один из которых перемещается по некоторому заданному пути, а второй — свободно. Третий исполнитель управляется с клавиатуры (рис. 2). К этому уроку школьники уже знают, как это делать, что превращает данный этап урока в этап актуализации.

Далее каждый ученик получает задание: настроить взаимодействие персонажей. При приближении к другому Kodu любой персонаж должен говорить ту или иную фразу и испытывать эмоцию, при этом фраза и эмоция зависят от того, к какому именно Kodu приближается

персонаж, т. е. от конкретного события. На примере NPC-1 такая программа выглядит, как представлено на рисунке 3.

К концу урока ученики получают небольшой игровой этюд, где поведение трех исполнителей меняется в зависимости от той или иной ситуации. Это позволяет учащимся наглядно проанализировать работу простейших алгоритмов, управляемых событиями, и заложить основу для реализации параллельных вычислений в дальнейшем.

Задание 2. Двое в поисках яблок.

Данный пример представляет собой преобразование изложенной в образовательной программе «Информа-



Рис. 2



Рис. 3

тика для всех» [8] задачи «Байер и яблоки». Ученикам предлагается изменить игру, создав в ней соревновательный эффект — с компьютерным исполнителем или другим игроком.

В данном случае на карте периодически появляются яблоки красного и синего цветов. Первый исполнитель получает +1 балл при поедании красного яблока и –1 балл при поедании синего яблока. Второй исполнитель, наоборот, получает +1 балл при поедании синего яблока и –1 балл при поедании красного яблока. Кроме того, на карте присутствуют 2 NPC, которые, свободно перемещаясь, поедают любые яблоки в пределах видимости. Но им баллы не начисляются.

«Событие победы» в игре — набранные 20 баллов. Стратегий победы получается несколько, и стратегия зависит как от поведения второго исполнителя, так и от поведения NPC. Игрок может не только поедать «свои» яблоки, но и постараться не допустить к яблокам нужного цвета своего соперника.

При выполнении этого задания ученики получают пример практического применения алгоритмов, управляемых событиями, при создании компьютерных игр. Кроме того, часто занятие посвящается обсуждению возможных победных тактик, в том числе тех, когда рационально съесть «яблоко –1». Также здесь реализуется ряд исследований. Например, перед учениками ставится вопрос: «Если два исполнителя одновременно коснутся объекта “яблоко”, кто именно его съест?» В рамках этого исследования ученики выявляют основные закономерности «принятия решений» в «конфликтной ситуации» между двумя исполнителями.

* * *

Рассмотренные задания — только примеры. На начальном этапе обучения ученики осваивают их немало, и с каждым следующим уроком у них все больше преобладает самостоятельная проектная составляющая деятельности.

Результаты внедрения Kodu в учебный процесс показывают, что ученики уже в третьем классе показывают высокие результаты в рамках акции «Час кода», а в четвертом классе справляются с более сложными проектами в среде Scratch. Что касается далеко идущих оценок влияния использования среды Kodu на освоение

параллельного программирования в школе, то два года — недостаточный срок для подобных оценок. Но совместно с коллегами в школе № 45 имени Л. И. Мильграма мы разработали ряд модулей по освоению параллельного программирования в V—VI классах и в течение ближайших двух лет сможем проанализировать динамику освоения этой темы у учеников, прошедших пропедевтический курс, по сравнению с учениками, которые начали освоение темы «с нуля».

Список использованных источников

1. Босова Л. Л., Сорокина Т. Е. Методика применения интерактивных сред для обучения младших школьников программированию // Информатика и образование. 2014. № 7.
2. Горячев А. В. Методика обучения информатике в начальной школе, реализующая объектно-информационный подход: дис. ... канд. пед. наук. СПб., 2004.
3. Ершов А. П. Программирование — вторая грамотность // Архив А. П. Ершова. http://www.ershov.ras.ru/russian/second_literacy/article.html
4. Кузнецов А. А. Реализация требований нового ФГОС в практике школьного образования // Муниципальное образование: инновации и эксперимент. 2008. № 2.
5. Леонов А. Г., Первин Ю. А. Элементы программирования в непрерывном курсе школьной информатики // Ярославский педагогический вестник. 2013. Т. 3. № 1.
6. Могилев А. В. Что наша жизнь? Геймификация! // Народное образование. 2015. № 6.
7. Павлов Д. И. Новая редакция федерального государственного образовательного стандарта начального общего образования — место информатики в начальной школе // Педагогическая информатика. 2017. № 3.
8. Павлов Д. И. Информатика. 2–4 классы: методическое пособие / под ред. А. В. Горячева. М.: БИНОМ. Лаборатория знаний, 2016.
9. Павлов Д. И. Раскрытие содержательных линий «Представления информации» и «Информационных процессов» на уровне начального общего образования // Открытое и дистанционное образование. 2018. № 1.
10. Пейперт С. Переворот в сознании. Дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.
11. Ревич Ю. В., Малиновский Б. Н. Информационные технологии в СССР. Создатели советской вычислительной техники. СПб.: БХВ-Петербург, 2014.
12. Семенов А. Л. Концептуальные проблемы информатики, алгоритмы и программирование в школе // Вестник кибернетики. 2016. № 2.

НОВОСТИ

Google сообщила об открытии раздела «Видеоигры» на YouTube

Google представила раздел «Видеоигры» на Youtube. В 2015 году был запущен «YouTube гейминг» — отдельное приложение для геймеров, в котором тестирование проходили новые функции. Среди них: страницы игр, суперчат, спонсорские подписки, ночной режим и даже новая платформа для трансляций на основе отзывов геймеров. Эти функции понравились игровому сообществу, поэтому они были перенесены из приложения на YouTube. Лучшие трансляции и загруженные видео собраны в одном месте.

К тому же добавлены разделы для трансляций и набирающих популярность видео. Теперь можно найти видео о конкретной игре на обновленных игровых страницах. Сюда входят популярные видео, трансляции и другие игры от того же издателя или разработчика.

В марте 2019 года прекратится поддержка приложения «YouTube гейминг», а Google полностью сосредоточится на игровых видео на YouTube, где их аудитория станет намного больше.

(По материалам CNews)



И. Ю. Травкин,

Сахалинский государственный университет, г. Южно-Сахалинск

ЭЛЕМЕНТЫ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVASCRIPT: ДВЕ ЗАДАЧИ ИЗ ТЕОРИИ ВЕРОЯТНОСТЕЙ

Аннотация

В статье подробно рассмотрены два примера составления программ в функциональном стиле, знакомство с которым является частью подготовки к решению современных задач обработки больших данных. Выбранные задачи, приведенный исходный код и объяснения автора могут служить материалом для одного или двух элективных занятий на соответствующую тему.

Ключевые слова: функциональное программирование, функциональный стиль, JavaScript, теория вероятностей, занимательные задачи.

DOI: 10.32517/2221-1993-2018-17-7-40-45

Контактная информация

Травкин Иван Юрьевич, ст. преподаватель кафедры математики Сахалинского государственного университета, г. Южно-Сахалинск; *адрес:* 693008, Сахалинская область, г. Южно-Сахалинск, Коммунистический пр-т, д. 33; *телефон:* (4242) 45-23-73; *e-mail:* iwannt@gmail.com

I. Yu. Travkin,
Sakhalin State University, Yuzhno-Sakhalinsk

FUNCTIONAL STYLE PROGRAMMING IN JAVASCRIPT: TWO PROBABILITY PROBLEMS

Abstract

The article contains detailed explanations for two examples of solving simple probability problems with simulations written in JavaScript using functional programming paradigm. Provided material is designed to be used in teaching introductory level of functional programming in upper secondary school.

Keywords: functional programming, functional style, JavaScript, probability theory, entertaining tasks.

Введение

Функциональное программирование предлагает непривычный для большинства программистов подход к пониманию и формулированию вычисления: вместо списка инструкций, означающих строго заданную последовательность действий исполнителя, приходится иметь дело с «декларативными» выражениями — программа перестает быть пошаговой инструкцией («рецептом») и становится совокупностью определений, содержащих описание требуемого результата вычисления (часто такое описание дается в рекурсивной форме).

Знакомство с функциональным программированием еще в школе может вооружить будущего студента инструментарием, актуальность которого возрастает в связи с задачами обработки так называемых *больших данных* (Big Data). Часто приводимый пример — модель MapReduce, разработанная инженерами компании Google с целью упростить создание приложений для распределенной обработки колоссальных объемов хранимой на серверах информации и основанная на принципах, заимствованных из языков функционального программирования [8]. В задачах машинного обучения (еще один пример), также связанных с обработкой больших данных и требующих от программиста более серьезной математической подготовки, трудно переоценить роль инструмента, позволяющего формулировать программу в терминах математических преобразований, не заботясь, например, о непосредственном управлении памятью вычислительной машины.

Язык программирования JavaScript, не являясь функциональным в том же смысле, в котором эту характеристику обычно применяют к языкам Haskell, ML или LISP, перенимает некоторые общие принципы, что позволяет составлять программы в так называемом *функциональном стиле*. Кроме того, JavaScript является самым распространенным на планете языком программирования [7] — любой современный интернет-браузер имеет встроенные инструменты, позволяющие экспериментировать с небольшими программами на JavaScript и таким образом изучать язык. Цель данной статьи: используя «подручное средство» — интернет-браузер со встроенной средой отладки, дать конкретный и готовый к использованию учителем пример составления программ в функциональном стиле.

Важные замечания

Статья адресована главным образом учителям информатики, при этом автор постарался сделать изложение настолько доступным, чтобы его мог усвоить обычный старшеклассник. Предполагается наличие у читателя опыта составления программ на уровне школьных задач и начальных представлений из теории вероятностей. При чтении важно последовательно

знакомиться с предложенными задачами: идея вычислительного эксперимента и исходный код, рассмотренные в первой задаче, используются и далее.

По поводу строгости изложения математических вопросов необходимо заметить, что автором выбран неформальный стиль, опирающийся на то интуитивное представление о вероятности, которое должно формироваться в старших классах школы. Понятия случайной величины и математического ожидания как «среднего» изложены в сильно упрощенной форме, дающей лишь самое необходимое представление.

В качестве среды для исполнения исходного кода удобно использовать любой веб-браузер с функцией «инструменты разработчика». Автор рекомендует Google Chrome версии 65 или более новый (можно загрузить «переносную» версию для установки на USB-диск [9]).

Чтобы начать работу в Google Chrome, откройте новую вкладку (сочетание клавиш Ctrl+T), активируйте инструменты разработчика (клавиша F12) и перейдите в открывшейся панели на вкладку консоли JavaScript (*Console*). Исходный код, приводимый в тексте статьи на отдельных строках, вводится в консоль JavaScript в соответствующей изложению последовательности.

Автор рекомендует предварительно познакомиться с документацией к «стрелочным» функциям [2], а также к методам *map()*, *reduce()*, *filter()* и *slice()* объекта *Array* [3–6] и иметь ее под рукой. В ряде случаев совместного применения *map* и *reduce* можно обойтись одним *reduce*, но такой подход кажется автору менее наглядным с учетом рассматриваемых задач. Аналогичный аргумент можно привести в пользу выбранного способа работы с массивами — предпочтение отдано наглядности, т. е. возможности без дополнительных усилий сопоставить структуру данных и выполняемые с ними преобразования с общей идеей излагаемого решения.

Практика составления программ в функциональном стиле — именно стиле, так как речь идет об отдельных приемах, умело сочетаемых с более традиционным для JavaScript подходом, — стала популярной относительно недавно, в том числе благодаря внедрению языкового стандарта ECMAScript 2015 (ES6). У читателя, знакомого с «традиционным» JavaScript, может возникнуть представление, что все сводится к использованию новых форм синтаксиса и надстроек, делающих язык неузнаваемым для тех, кто не интересовался его развитием в последние годы [10]. На самом деле, функциональный стиль выражается в стремлении к декомпозиции алгоритмов на элементарные функции, сочетание которых дает требуемый результат [11]. Именно эту идею желает передать автор на материале выбранных им задач.

Парадокс дней рождения

Представим следующую ситуацию. В некотором месте собрались совершенно незнакомые и никак не связанные друг с другом люди, всего n человек. Не исключено, что среди них есть родившиеся в один день года (т. е. одного числа, возраст может не совпадать). Каким должно быть минимальное число незнакомцев, чтобы вероятность таких совпадений была не меньше 50 %?

Вероятность того, что два случайно выбранных незнакомца родились в один день, равна $1/365 \approx 0,003$, т. е. всего 0,3 % (для простоты считаем, что в году всегда 365 дней, и все дни равновероятны). Если $n > 365$, т. е. когда незнакомцев больше, чем дней в году, то совпаде-

ния становятся неизбежными, и вероятность достигает 100 %. Можно предположить, что число незнакомцев, соответствующее вероятности 50 %, должно быть около половины от общего числа дней в году, т. е. примерно 182, но вычисления (см., например, [1]) дают неожиданный ответ: при $n = 22$ имеем вероятность 47,6 %, а уже при $n = 23$ вероятность становится 50,7 %, т. е. достаточно всего 23 человек!

Результат кажется удивительным даже после того, как лично выполнишь соответствующие расчеты (отсюда название «парадокс»). Непосредственным способом принять контринтуитивный факт служит эксперимент — проверка опытом. Если бы мы могли организовать описанную встречу для 23 совершенно незнакомых людей, причем повторили этот эксперимент достаточно большое число раз, то примерно в половине случаев среди незнакомцев нашлись хотя бы двое, родившиеся в один день года. Такой эксперимент, основанный на свойстве относительной частоты случайного события приближаться к его вероятности при достаточно большом числе испытаний, можно смоделировать на компьютере.

Чтобы разыграть встречу незнакомцев на компьютере, достаточно сгенерировать n чисел — случайных дней года, как если бы это были дни рождения наших незнакомцев, и проверить на наличие совпадений.

В JavaScript случайные числа получают с помощью функции *Math.random()*, которая возвращает действительные значения в интервале (0; 1). Если умножить такое число на 365 и взять целую часть от результата с помощью функции *Math.floor()*, то мы получим случайное целое число от 0 до 364, т. е. ровно 365 возможных значений (пусть счет дней начинается с нуля).

Чтобы получить n случайных значений, подготовим массив на n «мест» и заполним их случайными значениями:

1) создадим массив из n ячеек, заполненных нулями (необходимо, чтобы в ячейках что-то было до того, как мы начнем преобразовывать значения):

```
Array(n).fill(0)
```

2) преобразуем с помощью *map* значения в ячейках, заменив их случайными номерами от 0 до 364:

```
...map(_ => Math.floor(Math.random()*365))
```

Полное определение функции для получения n случайных дней рождения имеет вид:

```
random_days = (n) => Array(n).fill(0).map(_ =>
  Math.floor(Math.random()*365));
```

Например, при $n = 10$ мы можем получить следующий результат:

```
> random_days(10);
[25, 215, 237, 340, 102, 179, 282, 68, 308, 137]
```

Теперь необходимо реализовать проверку на совпадения. Сначала сформулируем критерий: в массиве есть совпадения, если массив содержит больше одного элемента и либо первый элемент равен одному из последующих, либо совпадения есть среди последующих. Переводя с естественного языка на язык JavaScript, мы получим рекурсивное определение функции, вычисляющей логическое значение ИСТИНА или ЛОЖЬ:

```
has_same = (days) => days.length > 1 &&
  (days.slice(1).includes(days[0]) ||
  has_same(days.slice(1)));
```

Здесь:

`days.length` возвращает число элементов в массиве сравниваемых дней рождения;

`days.slice(1)` возвращает массив без первого элемента («последующие»);

метод `includes()` проверяет наличие в массиве конкретного значения (совпадение «первого» с одним из «последующих»).

Проверим на примерах:

```
> has_same([10, 12, 31, 14]);
false
> has_same([10, 12, 31, 12]);
true
```

Эксперимент предполагает многократное (N раз) разыгрывание встречи незнакомцев, и если представить результат отдельного розыгрыша одним числом: 1 — в случае «успеха» (есть незнакомцы, родившиеся в один день) и 0 — в случае «неудачи» (все незнакомцы родились в разные дни), то общее число «успехов» будет равно сумме чисел — результатов всех розыгрышей:

$$M = X_1 + X_2 + \dots + X_N = \sum_{i=1}^N X_i,$$

где

$$X_i = \begin{cases} 1, & \text{если «успех» в } i\text{-м розыгрыше,} \\ 0, & \text{если «неудача» в } i\text{-м розыгрыше.} \end{cases}$$

Преобразовать результат розыгрыша в число, т. е. получить значение X_i (для всех $i = 1, 2, \dots, N$), можно с помощью тернарной условной операции:

$$\begin{aligned} & \text{условие} ? \text{результат1} : \text{результат2} = \\ & = \begin{cases} \text{результат1, если условие} = \text{ИСТИНА,} \\ \text{результат2, если условие} = \text{ЛОЖЬ.} \end{cases} \end{aligned}$$

Тогда

$$X_i = \text{has_same}(\text{random_days}(n)) ? 1 : 0.$$

Поступим следующим образом: разместим результаты N розыгрышей в ячейках массива (каждая ячейка — как бы отдельный розыгрыш), а затем найдем общую сумму:

1) подготовим массив из N ячеек, заполненных нулями:

```
Array(N).fill(0)
```

2) преобразуем с помощью `map` значения в ячейках, заменив их результатами розыгрышей:

```
...map(_ => has_same(random_days(n))
? 1 : 0)
```

3) найдем с помощью `reduce` сумму значений из всех ячеек, т. е. общее число «успехов»:

```
...reduce((x, y) => x + y)
```

4) получим окончательный результат как отношение M/N .

Все вычисление результата численного эксперимента примет вид функции:

```
simulate = (n, N) => Array(N).fill(0).map(_ =>
has_same(random_days(n)) ? 1 : 0).
reduce((x, y) => x + y) / N;
```

Например, при $n = 100$, $N = 10\,000$ мы можем увидеть следующий результат (повторив действие, читатель по-

лучит близкое, но отличающееся значение частоты, так как эксперимент случайный):

```
> simulate(100, 10000);
0.9999
```

Полученное значение превышает 0,5 (50 %), что позволяет заключить: ста незнакомцев достаточно для того, чтобы вероятность совпадения дней рождения хотя бы у двоих была не меньше 50 %. Нас же интересует «пороговое» значение, т. е. минимальное n , при котором выполняется условие. Мы не станем искать n полным перебором, а повторим эксперимент для значений, выбранных из заданного промежутка с постоянным шагом (например, для $n = 10, 20, \dots, 100$).

Определим функцию, которая позволит получить выборку результатов экспериментов с различными значениями n :

```
experimental_data = (a, b, s, N) => {
let K = Math.floor((b - a) / s) + 1;
return Array(K).fill(0).map((_, i) =>
[a + s * i, simulate(a + s * i, N)])
};
```

Значения n выбираются из промежутка от a (включая) до b с шагом s ; всего таких значений $K = \lfloor (b - a) / s \rfloor + 1$. Результатом вычисления является массив из K элементов, в каждой ячейке которого записана пара значений вида: $[n, M/N]$, где $n = a + s \cdot i$ (число незнакомцев), $i = 0, 1, \dots, K - 1$ (порядковый номер ячейки в массиве), а M/N — относительная частота, т. е. результат эксперимента при заданном n .

Произведем вычисление для n от 10 до 100 с шагом 10 и $N = 1000$:

```
> data = experimental_data(10, 100, 10, 1000);
[[10, 0.118], [20, 0.408], [30, 0.692],
[40, 0.901], [50, 0.971], [60, 0.997],
[70, 0.999], [80, 1], [90, 1], [100, 1]]
```

Последний штрих: преобразуем полученные данные (массив пар) в формат текста-таблицы, который легко воспринимается при непосредственном изучении.

«Сцепим» значения внутри пар символом табуляции (`join(«\t»)`), а между парами вставим переход на новую строку (`join(«\n»)`):

```
print_data = (pairs) => pairs.map(pair => pair.
join("\t")).join("\n").replace(/\.\/g, ",");
```

Последнее преобразование `replace(/\.\/g, «,»)` нужно для того, чтобы заменить точку, разделяющую целую и дробную части числа в JavaScript, на привычную запятую.

Теперь имеем:

```
> print_data(data);
"10 0,118
20 0,408
30 0,692
40 0,901
50 0,971
60 0,997
70 0,999
80 1
90 1
100 1"
```

Анализ таблицы позволяет заключить, что «пороговое» значение n , т. е. минимальное число незнакомцев, при котором вероятность совпадения дней рождения хотя бы у двоих становится не меньше 50 %, действительно находится между $n = 20$ и $n = 30$. А уже

при $n = 50$ вероятность почти достигает 100 %. Кажется невероятным, но когда мы едем в автобусе в час пик, среди пассажиров с большой вероятностью есть двое, родившиеся в один день!

Задача про перестановки

Пусть дана перестановка из n чисел, т. е. взятые в некотором порядке числа от 1 до n . Например: (1, 5, 3, 4, 2) — перестановка длины $n = 5$. Числа, входящие в перестановку, будем называть *элементами данной перестановки*. *Максимум* — это такой элемент перестановки, который больше двух соседних с ним элементов. Например, в перестановке (1, 5, 3, 4, 2) есть два максимума: 5 (так как $5 > 1$ и $5 > 3$) и 4 (так как $4 > 3$ и $4 > 2$). Заметим, что первый и последний элементы не могут быть максимумами (максимум определен для «внутренних» элементов, имеющих два соседних с ними).

Вопрос: сколько в среднем максимумов содержит перестановка длины n ?

Например, существует всего шесть различных перестановок длины $n = 3$:

(1, 2, 3), (2, 1, 3), (1, 3, 2), (3, 2, 1), (3, 1, 2), (2, 3, 1).

Указанным перестановкам соответствуют следующие числа максимумов:

0, 0, 1, 0, 0, 1.

Получаем, что среднее число максимумов в перестановке длины $n = 3$ равно:

$$(0 + 0 + 1 + 0 + 0 + 1) / 6 = 2 / 6 = 1 / 3.$$

Легко догадаться, что наименьшее возможное число максимумов в перестановке любой длины равно нулю: если расположить элементы строго по возрастанию, то максимумов не будет. Наибольшее возможное число максимумов не может превышать половины длины перестановки, так как максимумы должны быть отделены друг от друга хотя бы одним элементом.

Таким образом, *среднее число максимумов* k в перестановке длины n удовлетворяет условию $0 \leq k \leq n/2$.

Найдем зависимость k от n с помощью случайного эксперимента.

В отличие от эксперимента к первой задаче, где разыгрывалось *случайное событие*, здесь мы будем разыгрывать *случайную величину*. Случайная величина — это переменная, которая в отдельном испытании принимает одно, и только одно из своих возможных значений, причем каждому значению соответствует определенная вероятность. Важнейшая характеристика случайной величины — ее *математическое ожидание*, равное сумме произведений возможных значений на соответствующие им вероятности. *Число максимумов в случайно выбранной перестановке заданной длины есть случайная величина, математическое ожидание которой равно среднему числу максимумов.*

Например, число максимумов в перестановке длины $n = 3$ равно 0 или 1 в зависимости от того, какая перестановка выбрана: в двух из шести случаев, т. е. с вероятностью $2/6$, число максимумов равно 1, в остальных четырех случаях, т. е. с вероятностью $4/6$, число максимумов равно 0 (см. выше). Тогда математическое ожидание числа максимумов в случайной перестановке длины $n = 3$ равно:

$$1 \cdot 2/6 + 0 \cdot 4/6 = 1/3,$$

т. е. найденному ранее среднему числу максимумов.

Эксперимент, основанный на статистическом смысле математического ожидания, позволит нам оценить среднее число максимумов без перебора всех возможных перестановок. Для этого произведем N испытаний, в каждом из которых выберем случайную перестановку длины n и определим число максимумов в ней, получив тем самым N значений случайной величины, а затем найдем среднее арифметическое этих значений, которое и будет статистической оценкой математического ожидания.

Начнем с определения числа максимумов в данной перестановке.

Пусть дан массив xs , содержащий n элементов (т. е. перестановка). Ни первый, ни последний элементы массива не могут быть максимумами, требуется проверить остальные $n - 2$ элементов:

1) возьмем «усеченный» массив, т. е. элементы со второго по $(n - 1)$ -й:

```
xs.slice(1, xs.length - 1)
```

Помним, что для элементов массива счет начинается с нуля;

2) преобразуем выбранные элементы следующим образом: запишем 1, если данный элемент — максимум, и 0 — в противном случае. Учтем, что нумерация элементов в исходном и «усеченном» массивах отличается на единицу!

```
...map((x, i) => x > xs[i] && x > xs[i+2] ? 1 : 0)
```

3) сложим записанные значения:

```
...reduce((x, y) => x + y)
```

Полное определение функции для нахождения числа максимумов в перестановке имеет вид:

```
count_maxes = (xs) => xs.slice(1, xs.length - 1).
  map((x, i) => x > xs[i] && x > xs[i+2] ? 1 : 0).
  reduce((x, y) => x + y);
```

Проверим:

```
> count_maxes([1, 2, 3, 4, 5]);
0
> count_maxes([1, 2, 3, 5, 4]);
1
> count_maxes([1, 3, 2, 5, 4]);
2
```

Теперь займемся получением случайной перестановки.

Возьмем массив, состоящий из расположенных по порядку чисел от 1 до n , и «перетасуем» его. Основное здесь — это «тасовка», т. е. случайное перемешивание, гарантирующее одинаковые шансы получить любую из возможных перестановок. Реализуем это следующим образом: будем брать по одному элементу из упорядоченного массива, выбирая очередной элемент в случайном порядке, и помещать в новый массив.

Сначала определим вспомогательные функции, решающие следующие задачи:

- выбор случайного элемента из массива — функция *take_random*;
- исключение из массива элементов, уже помещенных в новый массив, — функция *exclude*;
- помещение элементов исходного массива, взятых по одному в случайном порядке, в новый массив — функция *take_all* (использует *take_random* и *exclude*).

Рассмотрим перечисленные функции в соответствующем порядке:

```
take_random = (xs) => xs[Math.floor(Math.random() * xs.length)];
```

Здесь `Math.floor(Math.random()*xs.length)` позволяет получить случайный порядковый номер элемента в массиве `xs`. С `Math.random()` мы уже знакомы: умножая случайное значение от 0 до 1 на `xs.length`, мы получаем случайное значение от 0 до `xs.length`, а `Math.floor()` переводит его в целое (порядковый номер — целое число).

```
exclude = (xs, ys) =>
  xs.filter(x => !ys.includes(x));
```

С помощью `filter` мы отбираем лишь те элементы массива `xs`, которых еще нет в массиве `ys`. Помним, что восклицательный знак означает отрицание: если `ys.includes(x)` истинно для любого значения `x`, содержащегося в массиве `ys`, то `!ys.includes(x)` истинно только для таких `x`, которые *не* содержатся в массиве `ys`.

```
take_all = (xs, ys) => ys.length == xs.length ?
  ys : take_all(xs, ys.concat(take_random(exclude(xs, ys))));
```

Массив `ys` сначала пуст, а в процессе вычисления наполняется элементами из `xs`. Если число элементов в `ys` становится равно числу элементов в `xs`, т. е. все элементы из `xs` уже помещены в `ys`, вычисление завершается ответом — `ys`. Иначе продолжаем наполнение `ys`, вычисляя `take_all` (рекурсия) с тем же `xs`, но *обновленным* `ys`: добавляем с помощью `concat` очередной элемент, взятый с помощью `take_random` из числа таких элементов `xs`, которых нет в `ys` (используем `exclude`).

Проверим:

```
> take_all([1, 2, 3, 4, 5], []);
[1, 5, 4, 3, 2]
> take_all([1, 2, 3, 4, 5], []);
[2, 4, 1, 5, 3]
```

Теперь мы готовы определить функцию, вычисляющую случайную перестановку:

```
shuffle = (xs) => take_all(xs, []);
permutation = (n) => shuffle(Array(n).fill(0).map((_, i) => i + 1));
```

Здесь `Array(n).fill(0).map((_, i) => i + 1)` позволяет получить массив с числами от 1 до `n` (т. е. массив, который «тасуется»).

Наконец, используя функции `permutation` и `count_maxes`, мы можем сгенерировать `N` случайных перестановок заданной длины и определить среднее число максимумов в них, а затем, изменяя длину перестановки, выявить зависимость среднего числа максимумов от длины перестановки. Здесь можно снова воспользоваться функциями `simulate`, `experimental_data` и `print_data` (см. исходный код к первой задаче), слегка изменив одну из них:

```
simulate = (n, N) => Array(N).fill(0).map(_ => count_maxes(permutation(n))).reduce((x, y) => x + y) / N;
```

Произведем окончательное вычисление и получим данные для анализа:

```
> data = experimental_data(10, 100, 10, 1000);
[[10, 2.66], [20, 5.954], [30, 9.306],
 [40, 12.658], [50, 15.979], [60, 19.357],
 [70, 22.559], [80, 26.032], [90, 29.246],
 [100, 32.628]]
> print_data(data);
"10 2,66
20 5,954
30 9,306
40 12,658
50 15,979
60 19,357
70 22,559
80 26,032
90 29,246
100 32,628"
```

Для наглядности скопируем выведенную в консоль таблицу на новый лист Microsoft Excel и отобразим данные с помощью графика *Точечная диаграмма* (см. рис.).

Приведенный на рисунке график позволяет предположить, что среднее число максимумов в перестановке прямо пропорционально ее длине. И хотя точки графика не лежат в точности на одной прямой, такую прямую можно получить примерно, используя метод наименьших квадратов (подробности метода опускаем):

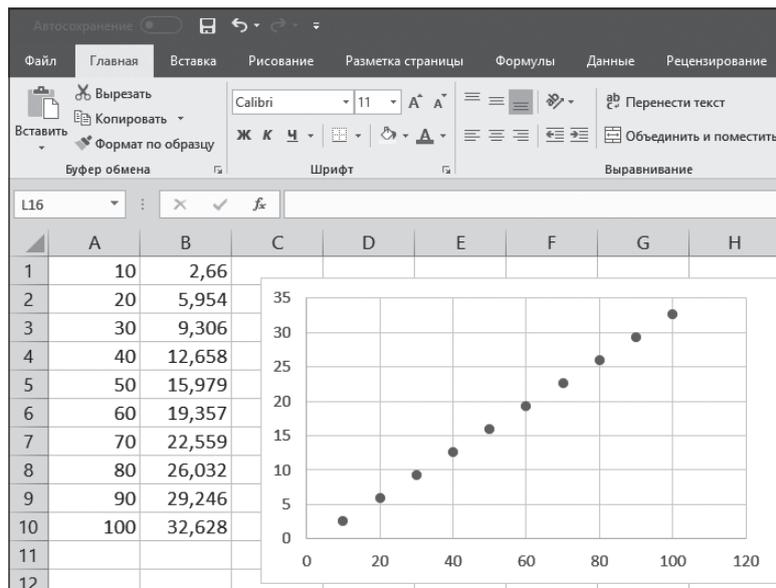


Рис. График экспериментального распределения данных. По горизонтальной оси отложены значения n (длина перестановки), по вертикальной — «экспериментальное» среднее число максимумов k

$$k = 0,33301n - 0,67767 \approx 1/3n - 2/3.$$

Найденная экспериментально зависимость совпадает с точным решением: среднее число максимумов перестановки линейно зависит от ее длины и выражается указанным выше соотношением.

Подробно о точном решении можно узнать из курса лекций по теории вероятностей Дж. Блитцштейна [12], у которого автор позаимствовал эту задачу в слегка измененной постановке.

Случайный эксперимент, как было сказано выше, не требует перебора всех возможных перестановок: чтобы получить достаточно точный результат, нам понадобилось всего по 1000 случайных перестановок каждой длины. Заметим, что если различных перестановок длины $n = 10$ существует $10! = 3\,628\,800$, то различных перестановок длины $n = 20$ будет уже $20! \approx 2,4 \cdot 10^{18}$, а при $n = 100$ имеем $100! \approx 9,3 \cdot 10^{157}$.

Заключение

Рассмотренные в статье задачи, исходный код и предложенное объяснение могут служить материалом для одного или двух элективных занятий, имеющих целью познакомить старшеклассников с элементами функционального программирования, быстро проникающими в популярные языки. Автор считает, что использованного им «подручного средства» — браузерной консоли JavaScript — достаточно и для более глубокого погружения в функциональное программирование, однако требуется дополнительная методическая работа

по переложению примеров из «классических» функциональных языков и отбору конкретных задач.

Список использованных источников

1. Аль-Халили Дж. Парадокс. Девять великих загадок физики. СПб.: Питер, 2017.
2. Стрелочные функции. https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Functions/Arrow_functions
3. Array.prototype.filter(). https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
4. Array.prototype.map(). https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/map
5. Array.prototype.reduce(). https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/Reduce
6. Array.prototype.slice(). https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/slice
7. Crockford D. The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language. <https://www.crockford.com/javascript/popular.html>
8. Dean J., Ghemawat S. MapReduce: simplified data processing on large clusters // Communications of the ACM. 2008. N 51(1). https://www.usenix.org/legacy/publications/library/proceedings/osdi04/tech/full_papers/dean/dean_html/
9. Google Chrome Portable. https://portableapps.com/apps/internet/google_chrome_portable
10. Jang P. Modern JavaScript Explained For Dinosaurs. <https://medium.com/the-node-js-collection/modern-javascript-explained-for-dinosaurs-f695e9747b70>
11. Nakoryakov V. Two Years of Functional Programming in JavaScript: Lessons Learned. <https://hackernoon.com/two-years-of-functional-programming-in-javascript-lessons-learned-1851667c726>
12. Stat 110 on YouTube. <https://projects.iq.harvard.edu/stat110/youtube>

НОВОСТИ

Мировой рынок платформ «умного города» через пять лет вырастет до \$755 миллионов

Мировой рынок платформ «умного города» к 2027 году достигнет \$755 млн. Такие данные приводятся в последнем отчете Navigant Research. По мнению аналитиков, концепция создания таких платформ становится все более понятной. Крупнейшие мировые вендоры, такие как Nokia, Microsoft и Cisco, разрабатывают проекты, в состав которых входит множество решений, покрывающих самые разные области — от видеонаблюдения и создания сетей до парковки и мониторинга окружающей среды в «умных городах».

Интеграция сервисов «умного города» неизбежна и необходима, также как и их централизованная координация, говорят аналитики Navigant Research. Заказчики все чаще стремятся выработать единую стратегию подключения различных городских служб к интернету вещей. Таким образом, появление единой платформы позволит реализовать концепцию «город как услуга», а также будет способствовать организации эффективного взаимодействия между различными городами, выстраиванию партнерских отношений, необходимых для успешного развития городской среды в будущем.

«Платформенный подход к созданию «умного города» способствует внедрению инноваций и обеспечивает непрерывность предоставления городских сервисов. Он позволяет городам стать полноценными участниками современного мира, основанного на бизнес-платформ-

мах», — говорит Эрик Вудс, директор по исследованиям Navigant Research.

В качестве образцов управления ресурсами и предоставления услуг «умные города» могли бы использовать модели Uber, Airbnb, Amazon, Google и Facebook, считают аналитики.

Платформенный подход к реализации концепции «умного города» позволит сделать его более удобным для рядового пользователя, будет способствовать созданию интуитивно понятного интерфейса. А значит, предлагаемые технологии станут востребованы. Тем не менее вопросы обеспечения информационной безопасности по-прежнему остаются одними из ключевых в этой сфере. Особенно в случае, если созданием городских сервисов будут активно заниматься коммерческие компании.

Сегодня на рынок «умных городов» стремится выйти множество разработчиков платформ. По мнению Navigant Research, это может привести к тому, что сама идея создания «умных городов» может быть скомпрометирована чрезмерными PR-активностями, в то время как граждане ждут от нее настоящих инновационных преобразований. В таких условиях, по мнению аналитиков, правительствам городов необходимо разрабатывать платформенную стратегию, основанную на принципах открытости, использования стандартов и согласованности с другими технологиями и бизнес-программами.

(По материалам CNews)



А. В. Радионова,
Петрозаводский государственный университет, Республика Карелия,

Я. В. Падчина,
ООО «Соробан», г. Петрозаводск, Республика Карелия

ПРИМЕНЕНИЕ 3D-ТЕХНОЛОГИЙ ПРИ СОЗДАНИИ ТАКТИЛЬНЫХ КНИГ ДЛЯ ДЕТЕЙ С НАРУШЕНИЯМИ ЗРЕНИЯ

Аннотация

В статье представлен опыт применения 3D-технологий при изготовлении тактильной книги с рельефными изображениями для слепых и слабовидящих детей. Разрабатываемые для тактильной книги 3D-модели должны соответствовать требованиям, предъявляемым к тифлографике, а также должны быть пригодны для печати на 3D-принтере. Работа по созданию книги выполнена в рамках студенческого проекта.

Ключевые слова: тактильная книга, 3D-технологии, 3D-печать, Blender.

DOI: 10.32517/2221-1993-2018-17-7-46-48

Контактная информация

Радионова Анна Владимировна, преподаватель кафедры теории и методики обучения математике и ИКТ в образовании Петрозаводского государственного университета, Республика Карелия; *адрес:* 185910, Республика Карелия, г. Петрозаводск, ул. Ленина, д. 33; *телефон:* (8142) 71-10-80; *e-mail:* radionovaann@gmail.com

Падчина Яна Викторовна, тренер ментальной арифметики школы устного счета «Соробан», г. Петрозаводск, Республика Карелия; *адрес:* 185031, Республика Карелия, г. Петрозаводск, ул. Варкауса, д. 27, корп. 1; *телефон:* +7 (911) 407-20-10; *e-mail:* yana.luchkina.ya@yandex.ru

A. V. Radionova,
Petrozavodsk State University, The Republic of Karelia,

Ya. V. Padchina,
Soroban Ltd., Petrozavodsk, The Republic of Karelia

APPLYING 3D TECHNOLOGY TO CREATE TACTILE BOOKS FOR CHILDREN WITH VISUAL IMPAIRMENTS

Abstract

The article describes the experience of making tactile picture book for blind and visually impaired children using 3D printing. 3D models for the tactile book should be made with allowance for tactile perception and should be printable. The article describes the result of the student's project.

Keywords: tactile book, 3D technology, 3D printing, Blender.

В разных странах все чаще появляются проекты, где 3D-печать используется в помощь слепым и слабовидящим людям. Одним из направлений применения 3D-технологий стало создание тактильных книг для слепых и слабовидящих детей. Широко известен проект Колорадского университета в Боулдере (США) Tactile Picture Book [6], а также российский проект Политехнического музея TouchBook [3]. Подобные тактильные книги содержат рельефные изображения, полученные путем 3D-моделирования и печати на 3D-принтере. Ребенок воспринимает образы, исследуя пальчиками рельефные иллюстрации. Существуют различные технологии изготовления тактильных книг, например, ручное изготовление с применением техник рукоделия. Применение 3D-технологии упрощает тиражирование книг.

В рамках выпускной квалификационной работы студенткой Института математики и информационных технологий Петрозаводского государственного университета Яной Падчиной была создана тактильная сказка «Колобок» для слепых и слабовидящих детей с использованием 3D-печати. Тематика данной работы носит междисциплинарный характер на стыке информатики и тифлопедагогики. При разработке 3D-моделей необходимо учесть требования к тактильным изданиям для детей с нарушениями зрения дошкольного и младшего школьного возраста. В процессе работы над книгой была организована встреча со специалистами в области тифлопедагогики — сотрудниками Карельской республиканской библиотеки для слепых. Кроме того, был выполнен анализ файлов с 3D-моделями для двух версий тактильной книги «Гарольд и фиолетовый мелок», разработанных в рамках проектов Tactile Picture Book и TouchBook и размещенных в открытом доступе [2, 5].

Этапы изготовления тактильной книги с использованием 3D-технологий (на примере тактильной сказки «Колобок»).

1. Проектирование книги.

На этапе проектирования необходимо определить параметры книги, используемые для ее изготовления материалы, проанализировать содержание литературного произведения, чтобы определить, какие рельефные изображения необходимы.

2. Моделирование объектов.

На данном этапе следует начать работу с создания эскизов моделируемых объектов с учетом требований тифлопедагогики, а именно:

- Необходимо определить оптимальный размер объекта. Размер одной модели не должен быть больше ладони ребенка, чтобы ребенок мог правильно представить образ.
- Следует учитывать естественные пропорции объектов относительно друг друга, а также объекта и составляющих его элементов.

- Модели должны быть реалистичными, что способствует узнаванию объектов.
- Модель не должна быть перегружена несущественными деталями, это затруднит восприятие.
- Следует обратить внимание на позу персонажа. Например, все конечности, уши должны быть явно прорисованы и отделены друг от друга, чтобы сформировать у ребенка верные представления.
- Необходимо определить тактильно ощутимые элементы объекта и их высоту [4].

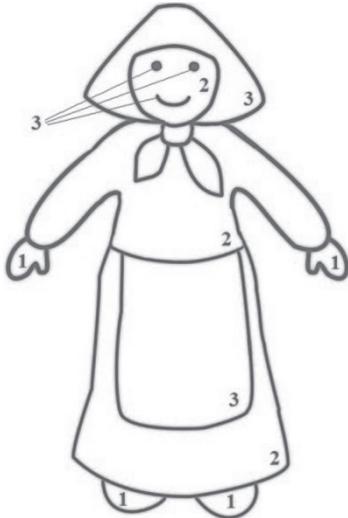


Рис. 1. Эскиз модели
(высота элементов 1 — 2 мм,
элементов 2 — 4 мм,
элементов 3 — 5 мм)

В графическом редакторе были созданы эскизы (пример эскиза приведен на рисунке 1), на основе которых в пакете трехмерной графики Blender было разработано одиннадцать 3D-моделей: колобок, бабушка, дедушка, заяц, волк, медведь, лиса, голова лисы крупным планом, печь, дом, ель. Blender является свободно распространяемым кроссплатформенным редактором 3D-графики с открытым исходным кодом. В дистрибутив Blender входит дополнение для подготовки моделей к печати — 3D Print Toolbox. При создании моделей для тактильной книги использовалось полигональное моделирование. При моделировании были учтены требования к 3D-моделям для печати. Требования, предъявляемые к размерам, толщине элементов и дизайну модели, зависят от используемого 3D-принтера и материала печати. Ошибкой при моделировании для 3D-печати является неманифолдная геометрия. Понятие «неманифолдная геометрия» определяет полигональную модель, для которой нельзя построить развертку без самопересечений. Примеры неманифолдной геометрии:

- ребра или вершины не принадлежат ни к одной грани;
- ребра соединены с тремя и более гранями;
- вершины принадлежат к несоседним граням.

Таким образом, модель должна представлять собой замкнутую поверхность, где каждое ребро связывает ровно две грани. Кроме того, модель не должна содержать инвертированных нормалей. Нормаль — вектор, перпендикулярный к поверхности. Нормали позволяют программному обеспечению 3D-принтера определить внутреннюю и внешнюю поверхности модели. Использо-

вая дополнение Blender 3D Print Toolbox, можно найти и исправить перечисленные ошибки.

Созданные модели были экспортированы в формат STL, пригодный для 3D-печати.

3. Печать моделей на 3D-принтере и сборка книги.

Печать 3D-моделей выполнялась на 3D-принтере Dimension Elite от компании Stratasys по технологии FDM (моделирование методом послойного наплавления). Для 3D-печати моделей использовался пластик ABSplus синего цвета. Параметры печати: 100-процентное заполнение, толщина слоя — 178 микрон, печать выполнялась на подложке, которая затем удалялась (рис. 2).

Для тактильной сказки было получено 17 моделей, модель «Колобок» распечатана в семи экземплярах.

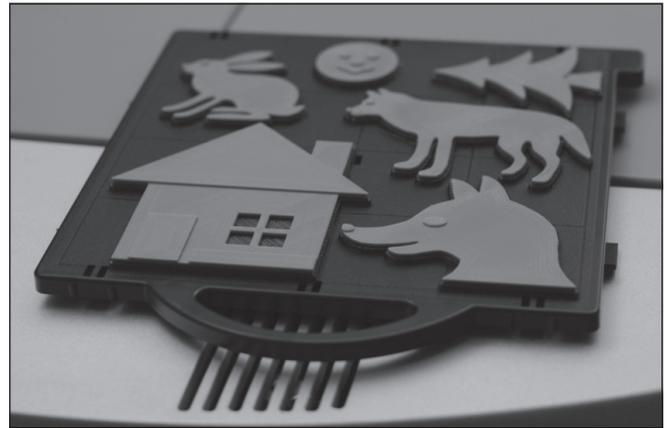


Рис. 2. Модели, извлеченные из 3D-принтера

На завершающем этапе необходимо прикрепить модели и дополнительные элементы к страницам книги, добавить текст сказки, набранный укрупненным шрифтом и/или шрифтом Брайля (рис. 3).

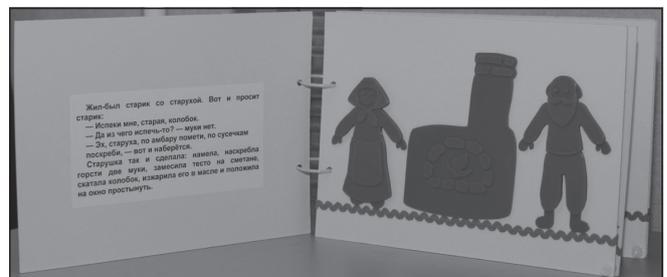


Рис. 3. Тактильная сказка «Колобок». Разворот

Разработанная книга содержит восемь страниц формата А4, ориентация страниц — альбомная. Таким образом, книга имеет оптимальный для ребенка вес и размер. Кроме того, в правом нижнем углу каждой страницы размещена пуговица, чтобы облегчить определение правильного расположения книги (рис. 4).

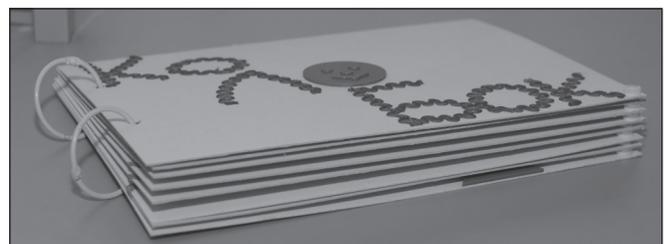


Рис. 4. Тактильная сказка «Колобок»

Тактильная сказка «Колобок» была передана Национальной библиотеке Республики Карелия, в состав которой входит Карельская республиканская библиотека для слепых [1].

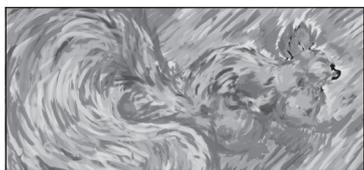
Авторы выражают благодарность А. С. Штыкову, заместителю начальника управления по инновационно-производственной деятельности ПетрГУ, П. Н. Середову, инженеру лаборатории прототипирования образцов и корпусов приборов Центра коллективного пользования научным оборудованием ПетрГУ, а также сотрудникам Карельской республиканской библиотеки для слепых.

Список использованных источников

1. Национальная библиотека Республики Карелия. Тактильная книга в подарок. http://library.karelia.ru/news/Taktil_naja_kniga_v_podarok/
2. Объемные иллюстрации к книге для слепых и слабовидящих детей «Гарольд и фиолетовый мелок». Проект TouchBook. <https://www.thingiverse.com/thing:666122>
3. Политех. TouchBook, книга для слепых детей. <https://polymus.ru/ru/pop-science/blogs/channels/kraudfanding/124728/>
4. Российская государственная библиотека для слепых. Тактильные рукодельные издания. <http://www.rgbs.ru/professionals/docs/taktilnye-rukodelnye-izdaniya/>
5. 3D «Harold and Purple Crayon» tactile book. Tactile Picture Book Project. <https://www.thingiverse.com/thing:397762/comments>
6. Tactile Picture Book Project. <https://tactilepicturebooks.org/>

КОНКУРСЫ

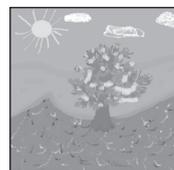
В оформлении обложки данного номера журнала «Информатика в школе» использованы работы следующих авторов — победителей конкурса цифровых изображений и фотографий:



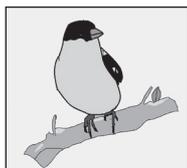
Алина Белова,
ученица гимназии № 1, г. Новосибирск



Дарья Каленицкая,
ученица средней общеобразовательной школы № 186, г. Новосибирск



Анастасия Каменская,
студентка Барнаульского государственного педагогического колледжа, Алтайский край



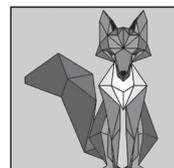
Тимерхан Хадиуллин,
студент Мамадышского политехнического колледжа, Республика Татарстан



Юлия Цзин,
ученица средней общеобразовательной школы сельского поселения Поселок Молодежный Комсомольского муниципального района Хабаровского края



Ирина Равина,
ученица средней общеобразовательной школы № 2, г. Невельск, Сахалинская область



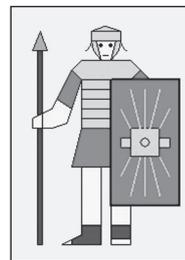
Андрей Годунов,
ученик Наро-Фоминской средней общеобразовательной школы № 6 с углубленным изучением отдельных предметов, Московская область



Валерий Вергола,
ученик Наро-Фоминской средней общеобразовательной школы № 1, Московская область



Владислав Крышталь,
ученик средней общеобразовательной школы № 8, г. Ноябрьск, Ямало-Ненецкий автономный округ



Сергей Ахсанов,
ученик Наро-Фоминской средней общеобразовательной школы № 1, Московская область

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 2019 года.

Подробности — на сайте издательства «Образование и Информатика»: <http://www.infojournal.ru/>

Н. А. Глузман,

Евпаторийский институт социальных наук (филиал) Крымского федерального университета имени В. И. Вернадского

ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ ДЛЯ ОБУЧЕНИЯ МАТЕМАТИКЕ В НАЧАЛЬНОЙ ШКОЛЕ СРЕДСТВАМИ ADOBE FLASH И HTML5

Аннотация

Для современного образовательного пространства в реалиях информационного общества особую значимость приобретают вопросы, касающиеся обеспечения высокого уровня информатизации образования, что предполагает овладение педагогами необходимыми компетенциями, умениями внедрять в учебно-воспитательную практику электронные образовательные ресурсы. Одной из самых популярных платформ для создания веб-приложений и мультимедийных презентаций, в том числе среди учителей, является Adobe Flash. Однако в связи с анонсом завершения в 2020 году поддержки Adobe Flash вопрос о выборе аналога для создания веб-приложений и презентаций для использования в учебных целях становится особо актуальным. В данной статье проводится комплексный анализ проектирования электронных образовательных ресурсов средствами Adobe Flash и HTML5 учителями для обучения математике в начальной школе.

Ключевые слова: проектирование интерактивных образовательных приложений, Adobe Flash, HTML5, будущий учитель начальной школы.

DOI: 10.32517/2221-1993-2018-17-7-49-57

Контактная информация

Глузман Неля Анатольевна, доктор пед. наук, профессор, зав. кафедрой методик начального и дошкольного образования Евпаторийского института социальных наук (филиала) Крымского федерального университета имени В. И. Вернадского; адрес: 297408, Республика Крым, г. Евпатория, ул. Просмушкиных, д. 6; телефон: (06569) 3-02-48; e-mail: gluzman_n@mail.ru

N. A. Gluzman,

Yevpatoria Institute of Social Sciences (branch) of V. I. Vernadsky Crimean Federal University

DESIGNING E-LEARNING RESOURCES FOR TEACHING MATHEMATICS IN PRIMARY SCHOOL WITH ADOBE FLASH AND HTML5 SYSTEMS

Abstract

In the modern educational space regarding the realities of the information society special importance is attached to issues related to the provision of a high level of informatization of education, which implies teachers' mastering the necessary competencies and the ability to introduce e-learning resources into educational and training practice. Adobe Flash as one of the platforms for creating web applications and multimedia presentations enjoys greatest popularity with users including teachers. However, in connection with the announcement of discontinuing Adobe Flash support in 2020, the issue of choosing an analog to create web applications and presentations for use in teaching purposes is becoming particularly relevant. The article provides a comprehensive analysis of developing electronic educational resources by teachers using Adobe Flash and HTML5 for teaching math in primary school.

Keywords: designing interactive educational applications, Adobe Flash, HTML5, future primary school teacher.

Стремительное развитие информационных технологий вызывает изменения в содержании и организации труда, в требованиях к уровню сформированности личностных качеств выпускников школы, которые должны уметь критически мыслить, иметь системные знания, владеть навыками сотрудничества в команде, управлять динамическими процессами. Сказанное акцентирует внимание на необходимости:

- развития образования на основе новых прогрессивных концепций;
- внедрения современных информационно-коммуникативных технологий и научно-методических достижений в учебный процесс;
- подготовки нового поколения педагогических кадров, начиная с учителя начальных классов.

Современная начальная школа не может оставаться в стороне от процессов информатизации образования. Начальной школы, как важного и неотъемлемого этапа образования каждой личности, касаются все мировые тенденции и инновации: компетентностный подход, личностно-ориентированное обучение, вариативность, информатизация, интеграция и др. К таким тенденциям относится и **повышение уровня готовности будущих учителей к проектированию электронных образовательных ресурсов** (А. А. Папышев и др. [2]).

Это актуализирует объективную потребность в разработке новых электронных средств обучения, в частности для обучения математике в начальной школе, на основе современных компьютерных технологий с учетом имеющегося педагогического опыта и достижений психолого-педагогических наук. От решения этой задачи зависит, насколько полно и эффективно будут использованы возможности электронных образовательных ресурсов в обучении младших школьников.

Исследования R. Mayer [17] и M. Grimley [13] показывают, что использование мультимедийных средств помогает более наглядно и схематично знакомить школьников с изучаемым материалом.

Одним из популярных среди учителей средств для создания образовательных мультимедийных презентаций, сайтов и игр является технология Flash. Данная платформа подтвердила свою эффективность в процессе обучения школьников различным дисциплинам: математике (E. G. Bakhom [5]), биологии (M. Teplá, H. Klímová [21]), химии (R. M. Gregorius и др. [12]), социальным наукам (Y. Weinstein [24]).

К сожалению, компания Adobe в июле 2017 года объявила о завершении в 2020 году поддержки технологии Flash (см.: The Blog.Adobe [22]). Основной заменой технологии Flash является открытый стандарт разработки веб-контента HTML5, стандарт которого был опубликован сравнительно недавно — в 2014 году (см.: World Wide Web Consortium [25]).

Под HTML5 понимается не только собственно язык разметки, который сам по себе не определяет способы создания анимации и интерактивности на веб-странице, но и дополнительные связанные специальные языки программирова-

ния JavaScript и CSS3. Adobe Flash также представлена в связке с языком программирования для создания интерактивности ActionScript.

Следует отметить, что HTML5 — это технология, призванная сделать интернет быстрее, проще и доступней для каждого пользователя, а в связке с CSS3 она успешно может заменить Adobe Flash во многих ее нишах, в том числе в образовательной. Дополнительно к этому HTML5 является единственным унифицированным языком разметки, полноценно работающим наравне с родными языками программирования большинства современных операционных систем, таких как Android, iOS, Windows Mobile, BlackBerry OS, в то время как операционная система компании Apple iOS не поддерживает Flash. На 4 января 2018 года только 4,3 % сайтов были созданы с использованием Flash и 94,9 % — с использованием JavaScript (см.: Web Technology Surveys [23]).

В противовес сказанному следует отметить, что Adobe Flash уже зарекомендовала себя как эффективное средство для создания мультимедийных презентаций и игр для обучения математике в начальных классах (см., например: http://www.learninggamesforkids.com/math_multiplication_games.html, <http://www.primaryinteractive.co.uk/math4kids.htm>, <https://www.coolmath4kids.com/>), в то время как HTML5 является новой технологией, овладение которой требует дополнительного исследования и анализа.

В данной статье мы делимся опытом обучения будущих учителей начальных классов технологиям создания мультимедийных и веб-приложений, в частности на основе технологий Adobe Flash и HTML5. Опыт основывается на преподавании спецкурса «Методика использования компьютерной техники на уроках в начальных классах», в котором изучались Adobe Flash с 2013 по 2016 год и HTML5 в 2017–2018 годах.

Поскольку современные ЭОР могут передавать информацию в сети, отмечают Н. Geng, M. Tian [9], а информация отображается различными способами и удобно сохраняется, то **ИКТ должны быть рационально интегрированы в цели, принципы и методы математического образования.** Использование данных технологий в процессе обучения вызовет появление новых целей и обновление содержания математического образования, позволит достичь значительно больших результатов учебной деятельности, обеспечит для каждого учащегося формирование и развитие его собственных информационных способностей (К. Ruthven, S. Hennessy [20]).

Современные информационные технологии позволяют развивать системное научное мышление, конструктивное образное мышление, пространственное и ассоциативное мышление, вариативность мышления, чувство нового, воображение, творческие способности. По мнению Y. M. Huang, S. H. Huang, T. T. Wu [14], цифровое игровое обучение (*англ.* Digital Game-Based Learning — DGBL) рассматривается как разумная обучающая стратегия в повышении мотивации учащегося и его интереса ко многим дисциплинам, в том числе к математике. В исследовании указанных авторов для курса математики начальной школы используется модель DGBL с применением стратегии диагностического механизма с целью изучения влияния методов обучения на уровень тревожности младшего школьника, мотивацию обучения, удовлетворенность обучением с точки зрения внимания, релевантности, доверия. Доказано, что общение с компьютером вызывает у детей живой

интерес — сначала как игровая деятельность, а затем и как учебная. Этот интерес лежит в основе таких важных структур, как познавательная мотивация, произвольная память и внимание, а именно эти качества обеспечивают высокий уровень когнитивного развития и креативности учащихся.

В зависимости от возраста обучающихся и применяемых технологий компьютер может выступать в роли противника в игре, быть рассказчиком, репетитором, экзаменатором. М. М. Klawe в своих исследованиях рассматривает проблему целесообразности использования компьютерных игр в обучении младших школьников. Полученные ею результаты свидетельствуют о том, что компьютерные игры могут быть очень эффективными в обучении детей математике [15]. Однако результативность применения таких игр зависит от многих факторов, в том числе от особенностей дизайнера программного обеспечения, от ожиданий учителей и учеников, от уровня интеграции с другими школьными мероприятиями. Важны и гендерные различия в отношении игр, обогащенных математикой: игры, в которые предпочитают играть мальчики, — это игры, требующие визуальных/пространственных представлений; игры, в которые предпочитают играть девочки, — это игры, требующие решения проблем, количественных вычислений и интерпретации графиков (Т. Lowrie, R. Jorgensen [16]).

Современная начальная школа, бесспорно, нуждается в учителе, который способен к творческому поиску креативных путей в профессиональной деятельности, работает в новом формате и является конкурентоспособным. Учитель сегодня должен быть не только психологически, но и технически подготовленным к проектированию и использованию информационных технологий в своей профессиональной деятельности.

В течение многих лет нами велись наблюдения за внедрением информационных технологий в обучение математике учителями начальной школы [10]. Индуктивный качественный анализ показал, что учителя фокусировались в основном на использовании таких информационных технологий, как презентации и интерактивные доски. Тем не менее исследования показали, что учителя начальной школы максимально пытаются интегрировать компьютерные технологии в обучение, что подтверждает дидактическую значимость этих технологий.

Нами были проанализированы **актуальные исследования отечественных и зарубежных ученых, имеющие непосредственное отношение к профессиональной подготовке будущих учителей начальной школы и технологиям их обучения:**

- Р. Drijvers, С. Kieran, М. А. Mariotti [8] рассматривают теоретические основы, которые используются в научно-технических исследованиях в области математического образования, и анализируют теоретические предпосылки улучшения профессиональной подготовки учителей;
- А. Bennison, М. Goos [6] описывают крупномасштабное исследование опыта профессионального развития учителей средних школ города Квинсленда (Австралия), связанное с использованием цифровых технологий для эффективности обучения учащихся математике;
- С. Н. Фортыгина [4] исследует влияние мультимедийного образовательного программного обеспечения на успешность овладения ИТ-навыками будущими учителями начальной и средней школы

с целью использования компьютерных технологий в процессе обучения и в профессиональной деятельности;

- М. Bétrancourt [7] раскрывает методический потенциал компьютерной анимации для визуализации динамических явлений, связанных со временем (например, биологических процессов, физических явлений, исторического развития), что может быть эффективно в обучении студентов и в использовании ими полученных навыков в педагогической практике;
- М. Ю. Табачкова, И. П. Борискина [3] убеждают, что, выбирая между электронными образовательными ресурсами и педагогической интерактивностью, профессиональный учитель всегда будет искать более рациональные технологические и дидактические возможности обучения.

Проведенный нами анализ вопросов подготовки учителей начальной школы к использованию ЭОР позволяет утверждать, что **основными проблемами в подготовке и повышении квалификации учителей по применению электронных образовательных ресурсов в школьном образовании являются:**

- недостаточная информированность относительно применения ЭОР в процессе обучения;
- отсутствие целенаправленного формирования необходимых компетенций, умений работы с электронными образовательными ресурсами и создания условий для их повседневного использования в учебном процессе;
- неуверенность учителей в том, что электронные образовательные ресурсы расширяют их профессиональные возможности.

На разных этапах исследования были использованы следующие методы:

- *эмпирические*: педагогическое анкетирование, опрос, беседы с учителями, учащимися, наблюдения за процессом и результатами проектирования электронных образовательных ресурсов по математике для начальной школы с использованием Adobe Flash и HTML5 с целью повышения качества образования, активизации познавательной деятельности учеников начальных классов, повышения уровня готовности будущих учителей начальных классов к профессиональной деятельности;
- *методы математической статистики* для количественного и качественного анализа результатов педагогического эксперимента.

Педагогический эксперимент проводился на базе Евпаторийского института социальных наук (филиала) и Гуманитарно-педагогической академии (г. Ялта) ФГАОУ ВО «Крымский федеральный университет имени В. И. Вернадского».

Опытно-экспериментальная работа по проектированию электронных образовательных ресурсов для проведения уроков математики в начальных классах проводилась с 2013 по 2018 год со студентами направления подготовки «Педагогическое образование» профиля подготовки «Начальное образование». Всего в экспериментальной работе участвовали 420 студентов. В эксперименте участвовали четыре группы: две экспериментальные (Э1 — 110 студентов, Э2 — 100 студентов) и две контрольные (К1 — 107 студентов, К2 — 103 студента).

Компетенция будущих учителей начальных классов в области проектирования ЭОР анализировалась по комплексным критериям:

- мотивационно-целевому;
- когнитивному (учебно-содержательному);
- технологическому (операционно-деятельностному);
- рефлексивно-прогностическому.

Для этих критериев была разработана система показателей, отражающих различные направления влияния нового подхода к проектированию системы педагогической диагностики на учебный процесс в соответствии с гипотезой исследования (Н. А. Глузман [1], N. A. Gluzman, T. V. Sibgatullina, A. A. Galushkin, I. A. Sharonov [10]).

Мотивационно-целевой критерий отражает профессионально-личностное самоопределение в направлении проектирования и использования ЭОР в начальных классах, позволяет определить ценностное отношение будущего педагога к проектированию и использованию ЭОР на уроках математики в начальных классах, осознанию значения умения проектировать ЭОР для успешной профессиональной деятельности и самореализации в будущей педагогической деятельности. Будущий учитель начальных классов понимает важность проектирования ЭОР и их использования в профессиональной деятельности (А. N. Privalov, Y. I. Bogatyreva, V. A. Romanov, V. N. Kormakova [18]).

Когнитивный (учебно-содержательный) критерий на формирующем этапе педагогического эксперимента определялся по наличию объема знаний, необходимых для проектирования ЭОР для начальной школы, при этом учитывались полнота, оперативность, гибкость и систематичность знаний. Учтено понимание студентами целей и задач и видение способов и результатов применения ЭОР в обучении математике в начальных классах.

Технологический (операционно-деятельностный) критерий — умение разрабатывать авторские ЭОР и использовать их при проведении пробных уроков по математике в начальных классах, умение использовать созданные ЭОР на факультативных занятиях по математике в начальных классах. Учитываются соответствие характера и объема учебных задач возможностям студента к самостоятельному их выполнению на достаточно высоком уровне трудности.

Рефлексивно-прогностический критерий — развитие будущим учителем начальных классов самооценки личности и проведение им самоанализа относительно созданных ЭОР. Умение анализировать учебно-воспитательный процесс по математике в начальных классах и его результаты; умение определять эффективность авторского электронного образовательного ресурса. Способность к субъективной оценке результатов своей деятельности, осознание своих конкурентных преимуществ и стремление к их наращиванию, умение оценивать соответствие педагогических технологий с использованием ЭОР современным требованиям.

По критериям и показателям были определены уровни сформированности компетенции проектирования и применения электронных образовательных ресурсов будущими учителями начальной школы — достаточный, средний, низкий и изучено состояние сформированности этой компетенции у студентов.

Целью опытно-экспериментальной работы стало выполнение исследовательских задач для эмпирического подтверждения (или отрицания) гипотезы о том,

что участие студентов в разработке ЭОР по математике для начальной школы с использованием Adobe Flash и HTML5 позволит сформировать у будущих учителей начальных классов компетенцию по проектированию ЭОР для обучения математике и углубить уровень профессиональной подготовки будущих учителей.

Эксперимент проводился на занятиях учебной дисциплины «Методика использования компьютерной техники на уроках в начальных классах». Одной из задач зачетной работы является проектирование ЭОР по математике для начальной школы с целью дальнейшего использования на уроках математики в начальных классах. При проведении студентами пробных уроков математики в начальной школе перед методистами института, академиками и учителями базовых школ ставилась задача максимально использовать возможности ИКТ.

Педагогический эксперимент проходил в три этапа: констатирующий, поисковый, формирующий.

На констатирующем этапе эксперимента проанализирована философская и психолого-педагогическая литература по проблеме подготовки студентов педагогического вуза к проектированию электронных образовательных ресурсов. Выяснены степень осведомленности преподавателей и студентов о проектировании ЭОР, уровень сформированности ИКТ-компетенции как предпосылки проектирования ЭОР по математике для начальной школы. Были использованы следующие методы: анализ, тестирование, опрос, анкетирование.

При этом необходимо отметить следующее: проведенный опрос практикующих учителей школ Республики Крым (150 респондентов) об использовании Adobe Flash и HTML5 в практике обучения математике младших школьников показал, что, по мнению основной части этих педагогов, технологии Adobe Flash и HTML5, несущие с собой новые комплексные способы представления, структурирования, хранения, передачи и обработки образовательной информации, позволяющие перейти к более эффективным формам организации учебной деятельности обучающихся и могут сыграть большую, а может, и определяющую роль в изменении традиционных образовательных технологий. Однако реальный процесс внедрения Adobe Flash и HTML5 в учебный процесс школ носит сегодня скорее стихийный и спонтанный характер. Учителя отмечают, что нередко новые информационные технологии помещают в учебные технологии ради самого их использования. Поэтому в современной школе существует противоречие между объективной потребностью в использовании универсальных и доступных ЭОР в рамках традиционной аудиторной системы и невозможностью применять для этого компьютеризированные системы (например, Adobe Flash и HTML5), предлагаемые разработчиками.

В ходе анкетирования учителей было установлено, что 97 % из них используют различные электронные образовательные ресурсы при проведении уроков, оформлении и защите аналитических отчетов. На вопрос: «Какие средства разработки мультимедийных программных приложений вы использовали в процессе обучения?», мы получили следующие ответы (в скобках указано количество выбравших то или иное средство из общего количества в 150 человек):

- Microsoft PowerPoint — 135;
- электронные книги — 5;
- видеохостинги (например, YouTube) — 21;
- приложения, разработанные в Adobe Flash, — 9.

Полученные результаты логично обосновывают положительный ответ практически всех респондентов на вопрос: «Считаете ли вы, что необходима подготовка студентов педвуза к проектированию и применению электронных образовательных ресурсов?»

На этом этапе педагогического эксперимента выявлено, что большинство педагогов недостаточно четко представляют специфику ЭОР для обучения математике младших школьников, не ориентируются на использование технологий Adobe Flash и HTML5 в профессиональной деятельности, хотя опрошенные учителя-практики и студенты считают предложенную компетенцию проектирования ЭОР средствами Adobe Flash и HTML5 и ее применения необходимым и обязательным компонентом профессиональной деятельности учителя начальной школы.

Результаты констатирующего этапа подтвердили предположение о том, что традиционная система профессиональной подготовки будущих учителей к электронному обучению математике младших школьников в недостаточной степени содействует формированию их профессиональной компетенции по использованию технологий Adobe Flash и HTML5 в педагогической деятельности, что определило необходимость целенаправленного формирования вышеупомянутого качества у студентов в области проектирования и применения электронных образовательных ресурсов.

На поисковом этапе эксперимента были выявлены педагогические условия, которые способствовали бы эффективной подготовке студентов педагогического вуза к проектированию ЭОР средствами Adobe Flash и HTML5:

- формирование ценностно-мотивационных установок для необходимости проектирования ЭОР средствами Adobe Flash и HTML5 для учеников начальных классов;
- организация обучения студентов проектированию ЭОР средствами Adobe Flash и HTML5;
- создание студентами электронных образовательных ресурсов для уроков математики в начальной школе средствами Adobe Flash и HTML5;
- проведение пробных уроков с использованием ЭОР;
- самоподготовка студентов по проектированию ЭОР средствами Adobe Flash и HTML5.

Были выполнены обобщение материалов данного этапа эксперимента, а также проверка и уточнение научно-методических выводов, полученных в ходе его проведения.

С целью повышения эффективности работы студентов было создано электронное учебное пособие «Информационно-коммуникационные технологии на уроках математики в начальных классах» и учебные пособия «Разработка учебных ресурсов с помощью Flash-технологии» и «Разработка учебных ресурсов с помощью HTML5-технологии».

В спецкурсе «Методика использования компьютерной техники на уроках в начальных классах» отводилось 10 лекционных и пять практических занятий на обучение студентов технологиям проектирования ЭОР.

В 2013–2016 годах спецкурс основывался на изучении технологии Flash в связке с языком программирования ActionScript.

Спецкурс состоял из следующих разделов:

1. Общая модель проектирования ЭОР для обучения математике в начальной школе. Цели и задачи интегративных платформ.

2. Введение в ActionScript. Использование окна Actions.
3. События в Flash. Событие ENTER_FRAME.
4. Подключение для интерактивного использования мыши и клавиатуры.
5. Обнаружение столкновений: команда hitTestObject.
6. Обнаружение столкновений: окружность-окружность, окружность-точка и др.
7. Использование переменных, условных выражений и циклов.
8. Одномерные и двумерные массивы.
9. Работа с фоном и графикой.
10. Создание и публикация веб-сайта онлайн. Введение в анимацию.

В 2016–2017 годах спецкурс основывался на изучении технологии HTML5 в связке с CSS3 и JavaScript.

Спецкурс состоял из следующих разделов:

1. Общая модель проектирования ЭОР для обучения математике в начальной школе. Цели и задачи интегративных платформ.
2. Знакомство с HTML5 как языком разметки. Каркас страницы.
3. Введение в Canvas.
4. Введение в CSS.
5. Взаимодействие с Canvas.
6. Обнаружение столкновений.
7. Использование переменных, условных выражений и циклов.
8. Одномерные и двумерные массивы.
9. Работа с фоном и графикой и публикация веб-сайта онлайн.
10. Введение в анимацию с использованием SVG.

Были отобраны однородные по составу группы студентов, в которых планировался эксперимент. В начале эксперимента студенты контрольных и экспериментальных групп не отличались по уровню сформированности отдельных показателей.

Поисковый этап эксперимента предусматривал выяснение уровня компетенции будущих учителей в области проектирования ЭОР для обучения математике в начальной школе средствами Adobe Flash и HTML5.

Программа *формирующего этапа эксперимента* предполагала реализацию методики формирования у будущих учителей компетенции по проектированию ЭОР для обучения математике в начальной школе средствами Adobe Flash и HTML5.

Проведение формирующего этапа педагогического эксперимента проходило по двум вариантам.

По первому варианту экспериментального обучения (1–4-й курсы, группа Э1) профессиональная подготовка студентов профиля подготовки «Начальное образование» к использованию технологии Adobe Flash в обучении математике младших школьников осуществлялась в ходе изучения дисциплины «Информатика и информационные технологии в начальной школе». В экспериментальной группе в практикумы были включены задачи, направленные на разработку будущими учителями начальных классов ЭОР с помощью Flash-технологий. Система HTML5 не изучалась. Контрольная группа (К1) осваивала технологию Adobe Flash в процессе самообразования с использованием электронного учебного пособия «Информационно-коммуникационные технологии на уроках математики в начальных классах» и учебных пособий «Разработка учебных ресурсов с помощью Flash-технологий».

По второму варианту экспериментального обучения (1–4-й курсы, группа Э2) использовалось учебное пособие «Разработка учебных ресурсов с помощью HTML5-технологий» и изучался авторский интегрированный спецкурс «Методика использования компьютерной техники на уроках в начальных классах». Контрольная группа (К2) осваивала технологию HTML5 в процессе самообразования с использованием учебного пособия «Разработка учебных ресурсов с помощью HTML5-технологий». Система Adobe Flash не изучалась.

Определено, что целью внедрения Adobe Flash и HTML5 как средств проектирования электронных образовательных ресурсов для начальной школы является формирование специалиста, способного к успешному проектированию ЭОР и их использованию в своей профессиональной деятельности.

Выявлено, что целью методики проектирования ЭОР с использованием Adobe Flash и HTML5 является формирование профессиональных компетенций будущих учителей начального образования. Умение проектировать ЭОР позволит углубить фундаментальную подготовку студентов — будущих учителей начальной школы и даст им возможность успешно адаптироваться к требованиям информационного общества в будущей профессиональной деятельности.

Разработана общая модель проектирования ЭОР для обучения математике в начальной школе с использованием Adobe Flash и HTML5 (см. рис.).

На данной модели можно увидеть, что Adobe Flash и HTML5 применяются на этапе технического воплощения проекта — будущий учитель создает запланированный ЭОР, например, презентацию к уроку, или интерактивные таблицы, или дидактические компьютерные игры, или тесты. Когда разработано достаточное количество ЭОР, то возникает возможность создания электронного учебного пособия, чтобы систематизировать материал, расположить его в одном месте и иметь возможность удобно использовать в дальнейшей работе.

Обучение студентов проходило в три этапа:

- *теоретический* этап был связан с усвоением студентами теории проектирования ЭОР и методики их применения в обучении младших школьников;
- *практический* — с формированием умений проектирования ЭОР;
- *апробационный* — с:
 - приобретением студентами опыта деятельности по подготовке и проведению учебных занятий с применением разработанных средств обучения;
 - самооценкой и экспертной оценкой опыта внедрения результатов проектирования в учебный процесс;
 - подготовкой публикаций по результатам проектирования;
 - участием студентов в семинарах и конференциях.

Обработка результатов эксперимента и оценка эффективности разработанной методики осуществлялись методами математической статистики (D. C. Miller, N. J. Salkind [18]). Результаты формирующего этапа эксперимента по определению уровня сформированности компетенции в области проектирования электронных образовательных ресурсов у будущих учителей начальных классов представлены в таблицах 1, 2.

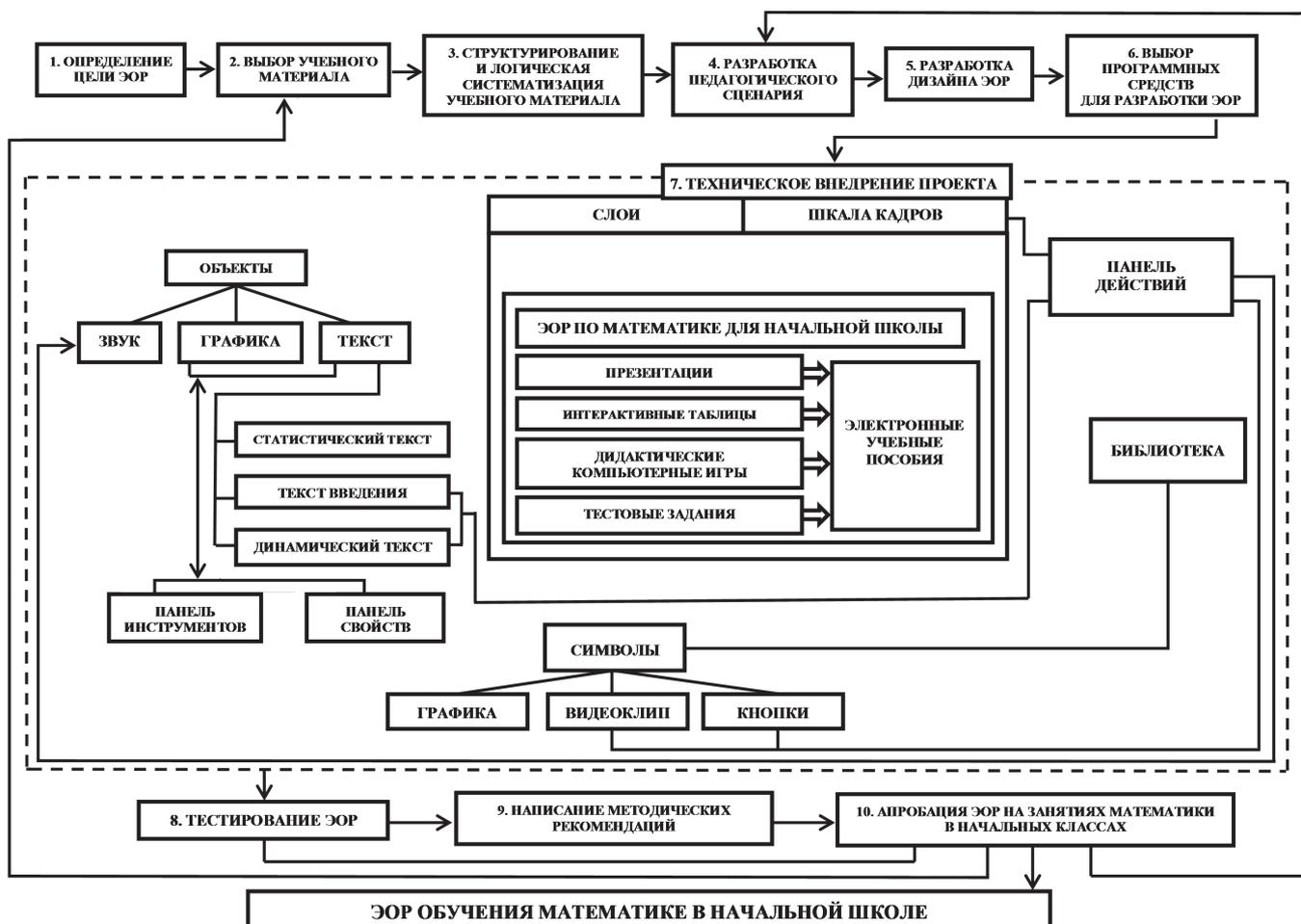


Рис. Общая модель проектирования ЭОР для обучения математике в начальной школе с использованием Adobe Flash и HTML5

Таблица 1

Уровни сформированности компетенции будущих учителей начальных классов в области проектирования электронных образовательных ресурсов средствами Adobe Flash в экспериментальной (Э1) и контрольной (К1) группах (вариант 1) в конце эксперимента

Экспериментальная группа (Э1)		Контрольная группа (К1)	
Уровень	Кол-во студентов, %	Уровень	Кол-во студентов, %
Достаточный	27	Достаточный	11
Средний	58	Средний	51
Низкий	15	Низкий	38

Таблица 2

Уровни сформированности компетенции будущих учителей начальных классов в области проектирования электронных образовательных ресурсов средствами HTML5 в экспериментальной (Э2) и контрольной (К2) группах (вариант 2) в конце эксперимента

Экспериментальная группа (Э2)		Контрольная группа (К2)	
Уровень	Кол-во студентов, %	Уровень	Кол-во студентов, %
Достаточный	26	Достаточный	8
Средний	54	Средний	48
Низкий	20	Низкий	44

В целом количественно-качественный анализ данных позволяет сделать вывод о положительных результатах формирования компетенции будущего учителя начальных классов по проектированию ЭОР по математике для начальной школы с использованием Adobe Flash и HTML5, что свидетельствует об эффективности разработанной нами методики.

Это нашло подтверждение в результатах анкетирования студентов в конце эксперимента с открытыми вопросами:

1. Считаете ли вы возможным самостоятельно выучить Adobe Flash (для первого варианта) и HTML5 (для второго варианта)? Если нет, то почему?
2. Какая тема в процессе обучения показалась вам наиболее сложной?
3. Намереваетесь ли вы использовать полученные знания в профессиональной деятельности?

В экспериментальных группах дополнительно задавался следующий вопрос:

4. Какими источниками вы пользовались при самостоятельном изучении Adobe Flash (для варианта 1) и HTML5 (для варианта 2)?

Результаты можно свести и интерпретировать следующими вариантами ответов, которые представлены в таблице 3.

Таким образом, экспериментальная проверка доказала преимущества предложенной методики обучения будущих учителей проектированию электронных образовательных ресурсов.

На основе теоретического и экспериментального исследования феномена формирования компетенции будущих учителей начальных классов по проектированию электронных образовательных ресурсов для обучения математике с использованием Adobe Flash и HTML5 и методики формирования компетенции будущих

учителей начальных классов по проектированию таких ресурсов получены следующие результаты.

Анализ научно-методических подходов к подготовке будущих учителей начальных классов показал, что не в полной мере представлены вопросы по проектированию и использованию электронных образовательных ресурсов, требующие разработки методики обучения, авторской учебной программы, которая бы отражала персональное видение преподавания учебного материала по математике в начальных классах

Установлено, что задача информатизации начальной школы, внедрения в ее учебный процесс современных электронных образовательных ресурсов по учебным дисциплинам, в частности по математике, требует разработки теоретических и методических основ проектирования данных ресурсов будущими учителями

Разработана общая модель проектирования электронных образовательных ресурсов для обучения математике в начальной школе с использованием Adobe Flash, HTML5 и следующие модели проектирования ЭОР для обучения математике в начальной школе с использованием Adobe Flash и HTML5:

- проектирование презентаций;
- проектирование интерактивных таблиц;
- проектирование дидактических игровых программ;
- проектирование тестовых заданий;
- проектирование электронных пособий с использованием Adobe Flash и HTML5.

Выявлены преимущества Adobe Flash перед Microsoft Office PowerPoint и Microsoft Office Excel, а именно: система содержит встроенные интерактивные образовательные компоненты как средства разработки авторских электронных образовательных ресурсов и является мощной средой для разработки авторских электронных

Таблица 3

Результаты анкетирования будущих учителей начальной школы в конце эксперимента

Вопрос	Группа			
	К1	Э1	К2	Э2
1. Считаете ли вы возможным самостоятельно выучить Adobe Flash (для первого варианта) и HTML5 (для второго варианта)? Если нет, то почему?	62 % считают, что возможно самостоятельно овладеть Adobe Flash	82 % считают, что возможно самостоятельно овладеть Adobe Flash	51 % считают, что возможно самостоятельно овладеть HTML5	81 % считают, что возможно самостоятельно овладеть HTML5
	Основные затруднения вызвали: «отсутствие «понятных» учебников», «малое количество примеров», «огромный объем материала», «отсутствие специальной литературы для будущих учителей»			
2. Какая тема в процессе обучения показалась вам наиболее сложной?	Обнаружение столкновений, создание и публикация веб-сайта	Обнаружение столкновений, массивы, работа с мышью	Обнаружение столкновений, CSS	Обнаружение столкновений, JavaScript, работа с графикой
3. Намереваетесь ли вы использовать полученные знания в профессиональной деятельности?	55 % намерены использовать полученные знания на практике	85 % намерены использовать полученные знания на практике	61 % намерены использовать полученные знания на практике	79 % намерены использовать полученные знания на практике
4. Какими источниками вы пользовались при самостоятельном изучении Adobe Flash (для варианта 1) и HTML5 (для варианта 2)?	Обучающими видео на YouTube, различными блогами		Обучающими видео на YouTube, http://htmlbook.ru , онлайн-новыми курсами на Coursera	

средств обучения, что обеспечит будущим учителям возможность проектировать презентации к урокам, интерактивные электронные таблицы, дидактические игровые программы, тестовые задания и электронные учебные пособия. Встроенные графические средства Adobe Flash предоставляют широкие возможности для разработки дизайна электронных образовательных ресурсов, а язык программирования ActionScript обеспечивает реализацию эффективного управления программным продуктом.

Обосновано то, что технология HTML5 в ближайшей перспективе станет мощным инструментом, способным обогатить опыт использования как обычного веб-приложения, так и любой системы электронного обучения. Помимо этого данная технология облегчает сам процесс создания веб-приложений. Популяризацией HTML5 занимаются ведущие IT-компании мира, такие как Apple, Google, Facebook, Twitter и Microsoft. Благодаря тому что HTML5 является унифицированным языком разметки, присутствующим в большинстве операционных систем современных мобильных платформ, его применение в целях предоставления более качественного опыта пользователю во время процесса дистанционного обучения является одним из направлений развития сферы проектирования ЭОР.

Разработаны рекомендации по формированию у будущих учителей компетенции по проектированию ЭОР для обучения математике в начальной школе с использованием Adobe Flash и HTML5, которые могут применяться как студентами, так и слушателями курсов повышения квалификации и учителями-практиками при осуществлении профессиональной деятельности.

В результате проведенного исследования установлено, что готовые программные продукты не предусматривают их полной адаптации к методической системе каждого учителя, поскольку разработчикам невозможно предусмотреть уникальность учителя и класса, в котором он будет работать. Существующие программные продукты требуют определенного приспособления учителей к методике, заложенной разработчиками. Педагоги остаются простыми пользователями готовых средств, «исполнителями» методических рекомендаций разработчиков.

Технология Adobe Flash во многом определила лицо современного интернет-видеообразования, создав целые новые классы контента. Хотя пока еще далеко до полного исчезновения Adobe Flash, но, по мнению многих специалистов, нужно быть готовым к тому, что в ближайшем будущем эта технология полностью исчезнет. Поэтому доказана необходимость подготовки учителей к использованию в учебном процессе как Adobe Flash, так и более современной технологии HTML5.

Разработана модель проектирования ЭОР в Adobe Flash и HTML5 для обучения математике в начальной школе, которая способствует повышению компетенции будущих учителей в области проектирования электронных образовательных ресурсов.

В процессе исследования создана собственная технология проектирования интерактивных электронных таблиц с использованием Adobe Flash и HTML5 и разработана методика их использования.

С использованием Adobe Flash и HTML5 будущий учитель получает возможность проектировать такие дидактические игровые программы, в которых ученик незаметно для себя может решить значительное количество примеров, выучить учебный материал, запланиро-

ванный учителем, с использованием подсказок, которые младший школьник получает, когда неправильно отвечает на поставленные вопросы.

Системы Adobe Flash и HTML5 являются средством создания программных приложений, предназначенных для тестового контроля по математике для начальной школы. В среде Adobe Flash и HTML5 можно создавать тесты с выбором одного правильного ответа из нескольких предложенных, тесты с введением ответа с клавиатуры и тесты с множественным выбором.

Доказано, что Adobe Flash и HTML5 являются средством моделирования и создания электронных учебных пособий по математике для начальной школы. Электронные учебные пособия — эффективное дидактическое средство, которое можно использовать на разных этапах обучения: с целью усвоения новых знаний, закрепления и совершенствования знаний, умений и навыков, а также проверки и коррекции учебных достижений учеников. Использование электронных учебных пособий способствует интенсификации труда как учителя, так и ученика.

Качественные и количественные данные, полученные в результате эксперимента, свидетельствуют о том, что уровень сформированности компетенции по проектированию ЭОР для обучения математике в начальной школе средствами Adobe Flash и HTML5 повысился в экспериментальной группе по сравнению с контрольной как в первом, так и во втором варианте обучения. В то же время уровень сформированности выделенной компетенции в экспериментальной группе (Э2) по варианту 2 выше по сравнению с показателями варианта 1 (Э1), что свидетельствует об эффективности методики ее формирования. Формирование рассматриваемой компетенции оказалось более продуктивным, так как на ее развитие было отведено больше аудиторного времени, также студенты имели возможность применять полученные теоретические знания в рамках всех видов педагогических практик. По варианту 1 на совершенствование компетенции проектирования ЭОР средствами Adobe Flash отводилась лишь производственно-стажерская педагогическая практика, так как в ходе обработки дидактико-методических учебных курсов на базовом этапе экспериментального обучения этому процессу не уделялось должного внимания.

Кроме того, у студентов экспериментальной группы, изучающих HTML5, отмечена более высокая мотивация как к процессу обучения, так и к будущей профессиональной деятельности. Необходимо также отметить, что экспериментальное обучение способствовало повышению формирования готовности студентов к самостоятельной работе, что дало толчок развитию умений приобретать и использовать новые знания в течение всей профессиональной деятельности.

Значительно увеличилось количество студентов с достаточным и средним уровнем сформированности компетенции проектирования ЭОР по математике для начальной школы; отмечено снижение количества студентов с низким уровнем сформированности компетенции проектирования ЭОР по математике для начальной школы в экспериментальных группах.

Результаты педагогического исследования являются основанием для **вывода**: экспериментальная проверка методики использования систем Adobe Flash и HTML5 как средств проектирования электронных образовательных ресурсов для обучения математике в начальной

школе показала, что предлагаемая методика является эффективной и может быть внедрена в учебный процесс высших учебных заведений для обучения данной категории студентов.

Проведенное исследование не исчерпывает решения всего спектра проблем, связанных с проектированием ЭОР для обучения математике в начальной школе. Считаем перспективным использование предложенной методики для проектирования ЭОР для обучения математике в дошкольных образовательных организациях и в учебных заведениях с инклюзивным обучением, а также ее применение для проектирования ЭОР для обучения другим дисциплинам в начальной школе.

Список использованных источников

1. Глузман Н. А. Становление и развитие информационной культуры будущих учителей начальных классов // Информатика и образование. 2018. № 2.
2. Папышев А. А., Сафонова Л. А., Сафонов В. И., Молчанова Е. А., Жамков А. А. Формирование ИКТ-компетентности и социальной активности студентов педагогического вуза посредством интерактивных форм обучения // Интеграция образования. Т. 19. № 2. 2015.
3. Табачкова М. Ю., Борискина И. П. Интерактивные методы обучения в математике // Интеграция образования. 2014. № 3 (76).
4. Фортыхина С. Н. Модель информационно-образовательной среды как средство формирования проекторочной компетенции у будущих учителей начальных классов // Современные проблемы науки и образования. 2015. № 5.
5. Bakhoum E. G. Animating an equation: A guide to using FLASH in mathematics education // International Journal of Mathematical Education in Science and Technology. 2008. No. 39 (5).
6. Bennison A., Goos M. Learning to teach mathematics with technology: a survey of professional development needs, experiences and impacts // Mathematics Education Research Journal. 2010. No. 22 (1).
7. Bétrancourt M. The animation and interactivity principles in multimedia learning. The Cambridge handbook of multimedia learning. New York: Cambridge University Press, 2005.
8. Drijvers P., Kieran C., Mariotti M. A. Integrating technology into mathematics education: Theoretical perspectives / C. Hoyle, J.-B. Lagrange (Eds.) // Mathematics Education and Technology — Rethinking the Terrain. Springer-Verlag, 2010.
9. Geng H., Tian M. Integration of modern information technologies and mathematics teaching // Advanced Materials Research. 2011. Vol. 225–226.
10. Gluzman N. A., Sibgatullina T. V., Galushkin A. A., Sharonov I. A. Forming the basics of future mathematics teachers' professionalism by means of multimedia technologies // Eurasia Journal of Mathematics, Science and Technology Education. 2018. Vol. 14. No. 5.
11. Gluzman N. A. Методико-математична компетентність майбутніх учителів початкових класів: монографія. Київ, 2010.
12. Gregorius R. M., Santos R., Dano J. B., Gutierrez J. J. Can animations effectively substitute for traditional teaching methods? Part II: Potential for differentiated learning // Chemistry Education Research and Practice. 2010. No. 11.
13. Grimley M. Learning from multimedia materials: the relative impact of individual differences // Educational Psychology. 2007. Vol. 27. No. 4.
14. Huang Y. M., Huang S. H., Wu T. T. Embedding diagnostic mechanisms in a digital game for learning mathematics // Educational Technology Research and Development. 2014. No. 2.
15. Klawe M. M. When does the use of computer games and other interactive multimedia software help students learn mathematics? // Journal of Curriculum Studies. 1998. No. 30 (3).
16. Lowrie T., Jorgensen R. Gender differences in students' mathematics game playing // Computers & Education. 2011. No. 57.
17. Mayer R. Multimedia learning. New York: Cambridge University Press, 2001.
18. Miller D. C., Salkind N. J. Handbook of Research Design and Social Measurement. Thousand Oaks, CA: Sage Publications Ltd, 2002.
19. Privalov A. N., Bogatyreva Y. I., Romanov V. A., Kormakova V. N. Safe information environment as a quality indicator of educational institution management // Naukova Visnyk Natsionalnogo Girnichogo universitetu. 2017. No. 4 (160).
20. Ruthven K., Hennessy S. A practitioner model of the use of computer-based tools and resources to support mathematics teaching and learning // Educational Studies in Mathematics. 2002. No. 49 (1).
21. Teplá M., Klímová H. Using Adobe Flash animations of electron transport chain to teach and learn biochemistry // Multimedia in Biochemistry and Molecular Biology Education. 2015. No. 43 (4).
22. The Blog.Adobe. Flash & The Future of Interactive Content. <https://theblog.adobe.com/adobe-flash-update/>
23. Web Technology Surveys. Usage of client-side programming languages for websites. https://w3techs.com/technologies/overview/client_side_language/all/
24. Weinstein Y. Flash programming for the social & behavioral sciences: A simple guide to sophisticated online surveys and experiments. Thousand Oaks, CA: Sage Publications Ltd, 2013. DOI: 10.4135/9781452244099
25. World Wide Web Consortium. HTML5 is a W3C recommendation. <https://www.w3.org/blog/news/archives/4167/>

НОВОСТИ

«Академия цифровых технологий», «НПП «Сигнал» и «НИИ «Масштаб» договорились о сотрудничестве в сфере подготовки ИТ-кадров

«Академия цифровых технологий», «НПП «Сигнал» и «НИИ «Масштаб» подписали трехстороннее соглашение о сотрудничестве 21 сентября 2018 года.

Компании договорились о взаимовыгодном сотрудничестве, направленном на развитие Академии, совместной реализации образовательных и просветительских проектов, разработке и внедрению образовательных программ в области информационной безопасности. Это сотрудничество позволит обучать детей на примере

реальных проектов, а предприятия в дальнейшем получат готовых специалистов, обладающих необходимыми компетенциями и навыками.

В новом образовательном пространстве дети от шести до 18 лет смогут изучать инженерные и рабочие профессии в сфере ИТ, робототехники, технологий цифрового производства и экономики. В текущем году здесь пройдут обучение почти две тысячи школьников. В течение нескольких лет эту цифру планируют увеличить втрое.

(По материалам CNews)

А. Л. Королев,

Южно-Уральский государственный гуманитарно-педагогический университет, г. Челябинск

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ПРОГРАММНЫЕ КОМПЛЕКСЫ ПОДДЕРЖКИ ИЗУЧЕНИЯ МОДЕЛИРОВАНИЯ В ШКОЛЕ

Аннотация

В статье рассмотрены особенности изучения моделирования в школьном курсе информатики. Представлены основные теоретические вопросы этой темы: роль моделирования в науке, технике, образовании и практической деятельности, роль компьютерных моделей, актуальность и множественность моделей. Описаны программные комплексы для создания компьютерных моделей, которые применимы в школьном образовании, в частности в проектной деятельности учащихся.

Ключевые слова: компьютерное моделирование, инструментальные программные комплексы моделирования, проектная деятельность учащихся.

DOI: 10.32517/2221-1993-2018-17-7-58-63

Контактная информация

Королев Александр Леонидович, канд. тех. наук, доцент, доцент кафедры информатики, информационных технологий и методики обучения информатике Южно-Уральского государственного гуманитарно-педагогического университета, г. Челябинск; адрес: 454080, г. Челябинск, пр-т Ленина, д. 69; телефон: (351) 216-63-09; e-mail: koroleval@cspu.ru

A. L. Korolev,
South Ural State University of Humanities
and Education, Chelyabinsk

COMPUTER MODELING AND SOFTWARE SYSTEMS TO SUPPORT THE STUDY OF MODELING AT SCHOOL

Abstract

In the article features of studying simulation in a school course of informatics are considered. The main theoretical issues of this theme are presented: the role of simulation in science, engineering, education and practical activities, the role of computer models, the relevance and multiplicity of models. Software complexes for creating computer models that are applicable in school education, in particular, in project activities of students, are described.

Keywords: computer simulation, instrumental program complexes for modeling, project activity of students.

Понятие «модель» всем знакомо на бытовом уровне. Поэтому при изучении темы «Моделирование» необходимо дать четкое **определение модели:**

«Под моделью мы будем понимать самостоятельный объект произвольной природы, отражающий свойства и связи моделируемого объекта (объекта-оригинала), которые считаются существенными для решения поставленной задачи» [5].

При этом отсутствие в модели существенных элементов не менее важно, чем присутствие в ней существенных. Именно поэтому модель проще объекта моделирования и доступнее для изучения.

Выделим главные моменты в определении:

- модель — это тоже объект;
- модель может быть как материальной, так и информационной или мысленной;
- модель замещает моделируемый объект, используется вместо него;
- модель отражает только некоторые черты моделируемого объекта, важные для данного исследования.

Модель всегда связана с изучаемым объектом и решаемой задачей (рис. 1).



Рис. 1. Построение модели

В одном из многочисленных опубликованных тестовых материалов есть вопрос: «Какой программный комплекс выбрать для построения модели рациона?» В качестве вариантов ответа следовало перечисление компьютерных офисных программ. Так как цель моделирования не указана, то нужно считать вопрос некорректно поставленным. Для достижения разных целей, решения разных задач могут потребоваться различные программы.

Процесс построения модели включает следующие этапы (см. рис. 1):

- 1) выявление проблемы;
- 2) постановка задачи (или нескольких задач) для решения проблемы;
- 3) системный анализ объекта моделирования;
- 4) выбор метода моделирования;
- 5) построение модели;
- 6) эксперименты с моделью;
- 7) перенос результатов моделирования на реальный объект.

Модель как самостоятельный объект может иметь свойства, которыми объект моделирования совсем не обладает. Например, такие эффекты могут иметь место при численной реализации математических моделей. Для корректного переноса результатов моделирования на реальный объект создана целая теория — теория подобия [1]. Например, еще в начале

XX века на основе этой теории получена достоверная информация о прочности ферм железнодорожных мостов Транссибирской магистрали на основе испытаний их моделей. По сей день сопротивление движению морских судов исследуется в бассейне на моделях.

Для повышения мотивации учащихся важным моментом при изучении темы «Моделирование» является рассмотрение ее актуальности.

С нашей точки зрения, **моделирование особо актуально в следующих случаях:**

- объект либо еще, либо уже не существует, однако необходимо изучить его свойства;
- масштаб времени реальных процессов или геометрические размеры объекта несоизмеримы с возможностями нашего восприятия;
- реальный объект недоступен для прямого изучения;
- эксперименты с реальным объектом дороги или слишком опасны.

Таким образом, во многих случаях моделирование — это единственно возможный путь изучения объектов и процессов.

Каждому объекту может соответствовать множество моделей, связанных с разными задачами или с разными группами свойств объекта. Кроме того, для одного и того же объекта для отображения одних и тех же свойств можно построить множество моделей разными способами в зависимости от целей моделирования и доступных средств. В зависимости от решаемой задачи для одного и того же объекта одним и тем же способом для отображения одних и тех же свойств можно построить множество моделей с разной степенью детализации описания объекта.

Типичный вопрос при изучении темы «Моделирование» — **классификация моделей**. Классификаций существует достаточно много. Мы предлагаем простую схему, которая опирается на определение модели.

Можно различать модели по следующим признакам:

- принадлежность к конкретному объекту моделирования;
- отражение определенной группы свойств;
- способ создания самой модели;
- принадлежность к определенному классу задач.

В наименовании модели содержатся именно эти признаки. Например: математическая модель, структурная модель, экономико-математическая модель, геометрическая модель, графическая модель и т. п. В этом отношении термин «информационная модель» может трактоваться как модель, которая отражает протекание информационных процессов (прием-передача, сохранение и обработка информации) [6], или как модель, построенная информационными средствами [2].

Основные функции моделей в современной науке и практической деятельности связаны с изучением свойств объектов и процессов, управлением, проектированием, прогнозированием, имитацией. Особую роль играют концептуальные модели, которые являются первичной системой моделей, представляющих интерес для данной науки. Например, «материальная точка», «идеальный газ», «абсолютно черное тело» и т. д. В свою очередь, научные законы формулируются как описание связей и взаимодействий между концептуальными моделями. Примером могут служить законы Ньютона, законы термодинамики и т. п. Таким образом, научные законы также являются в некотором смысле моделями.

На основе соответствующих законов строятся модели целых классов явлений — научные теории. Например, квантовая теория, гидроаэродинамика, термодинамика и т. п.

Особое место принадлежит **математическим моделям**, которые позволяют использовать научные знания для решения конкретных практических задач и проводить сравнение выводов теории с результатами наблюдений или экспериментов. Таким образом, теоретические выводы имеют практически значимый характер, если они выражены в виде количественных математических соотношений, которые позволяют получить численный результат.

Проектирование и эксплуатация современных сложных технических систем требуют «поддержки» со стороны моделирования. Современное проектирование подразумевает применение математических и имитационных моделей, методов оптимизации, систем автоматизированного проектирования (САПР), которые включают в себя 3D-моделирование [8]. Все это позволяет создать проект технической системы в кратчайшие сроки и с высоким качеством. Однако полностью исключить испытания и натурные эксперименты не удастся в силу сложности современных технических систем и протекающих в них процессов. Например, профессия летчика-испытателя в обозримом будущем останется крайне необходимой.

В ходе эксплуатации технических объектов возможны неисправности. Для обеспечения высокой надежности важно распознать приближение аварийной ситуации. Такая задача решается методами диагностики состояния объекта. Для подобных задач необходимо на основе моделирования аварийных ситуаций получить картину развития аварии. В случае распознавания предаварийного состояния технический объект может быть своевременно выведен из эксплуатации.

Рассмотренные выше примеры применения моделей показывают громадную роль моделирования в современной науке и технике. Пройденный путь, который начат с применения макетов и материальных моделей, на сегодня продолжен сложнейшими математическими и имитационными моделями.

Компьютерное моделирование состоит в построении модели, которая в итоге представляет собой некоторый программный комплекс. Компьютерное моделирование привело к существенному увеличению возможностей математического моделирования, численных методов и сокращению времени решения сложных задач, что позволило проводить анализ влияния множества факторов, определять оптимальные параметры. Появились новые направления как в самом моделировании (имитационное моделирование, моделирование знаний, компьютерный эксперимент), так и в различных прикладных науках (вычислительная физика, автоматизированное проектирование и т. д.). Результаты моделирования стало возможно представлять в форме виртуальной реальности. Сам компьютер стал инструментом автоматизированного создания моделей.

Препятствием для широкого использования моделирования в образовательных целях была необходимость создания компьютерных моделей средствами программирования, как, например, делается в работе [7]. Требуется также определенный багаж знаний в области численных методов. В этом случае собственно моделирование отодвигается на второй план.

Решением проблемы является **применение инструментальных программных комплексов визуального моделирования**. Такие программные комплексы предоставляют возможность быстрой разработки компьютерных моделей и проведения модельного эксперимента. Причем программирования (написания кода) при разработке моделей не требуется. Основой изучения процессов и явлений становится компьютерный эксперимент. Это актуально и для информатики, и для дисциплин естественно-научного и технологического циклов, причем от исследователя (учащегося) требуется минимальная подготовка в области информационных технологий, а численные методы выбираются самой системой моделирования. Таким образом, на уроках информатики можно реализовать межпредметные связи с другими дисциплинами.

Какие возможности применения комплексов визуального моделирования существуют в настоящее время?

Идея программного комплекса моделирования **Simulink**, который входит в состав математического пакета MATLAB [11], состоит в том, что модель визуально конструируется из стандартных библиотечных блоков. Так создается функциональная блок-схема объекта, устанавливаются (рисуются) связи между блоками и задаются параметры элементов системы. Комплекс моделирования Simulink реализует *технологии модельно-ориентированного проектирования*. Возможность применения данного пакета в старших классах средней школы (даже на профильном уровне) не просматривается, так как он рассчитан на серьезную проектную деятельность и высокий уровень инженерного образования.

Современный программный комплекс «КОМПАС» [8] позволяет построить трехмерные твердотельные модели объектов и проводить расчеты их массовых характеристик, полей напряжений или температурных полей под действием заданных пользователем внешних воздействий. При этом пользователь строит только трехмерную модель объекта и определяет внешние воздействия. Построенная 3D-модель далее может быть воссоздана на 3D-принтере.

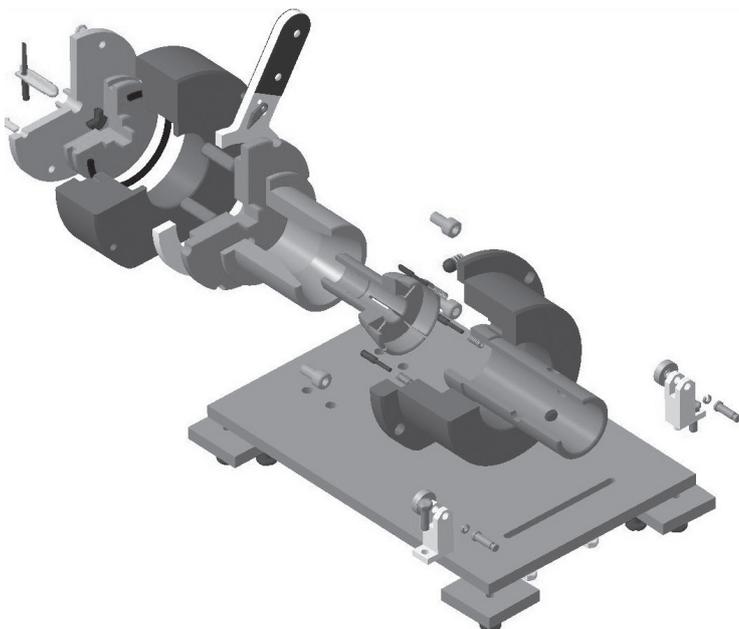


Рис. 3. Пример сборки

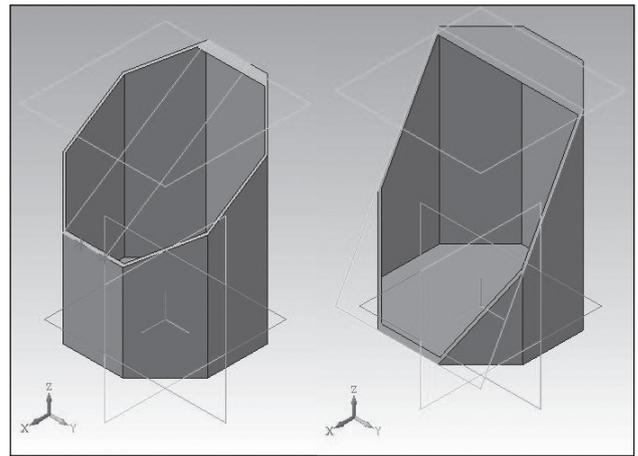


Рис. 2. Сечения призмы плоскостью

Сегодня стоит задача эффективного применения информационно-коммуникационных технологий в педагогической практике. Как применить эту программу в учебном процессе?

Используя учебную версию пакета — «КОМПАС_3D_LT 12», учитель *математики* может создать наглядные трехмерные модели-иллюстрации для курса геометрии и демонстрировать их на уроках. Тогда задачи, требующие трехмерных построений, получают эффективную и наглядную поддержку. Рассмотрение, например, конуса, призмы или пирамиды с разных сторон, различных сечений подобных тел позволяет достаточно легко найти решение задачи (рис. 2). Проведенные уроки информатики с использованием данного пакета при решении задач стереометрии подтвердили это предположение.

Учитель *технологии* имеет возможность создать электронные чертежи и наглядные трехмерные модели-пособия для изучения различных устройств (рис. 3). При этом можно отразить последовательность выполнения всех технологических операций изготовления определенной детали (рис. 4–7). Кроме того, знакомство учащихся с пакетом «КОМПАС» (элемент САПР) актуально в классах технологического (инженерного) профиля.

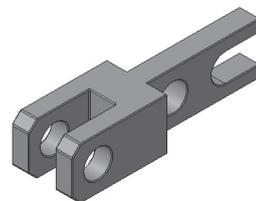


Рис. 4. Готовая деталь

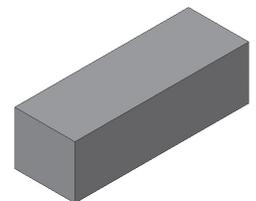


Рис. 5. Заготовка

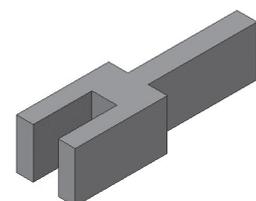


Рис. 6. Промежуточная операция — фрезерование паза

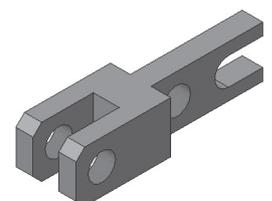


Рис. 7. Операция сверления отверстий

Одной из современных универсальных систем моделирования является система **MVS (Model Vision Studium)** или новая версия этой системы **RMD (Rand Model Designer)** [12]. MVS (RMD) — это интегрированная система для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения с ними вычислительных экспериментов. Ее главные возможности: моделирование дискретно-непрерывных систем; обеспечение численного решения; обеспечение визуализации результатов моделирования без какого-либо программирования.

В общем случае MVS (RMD)-модель системы строится в виде блок-схемы (рис. 8).

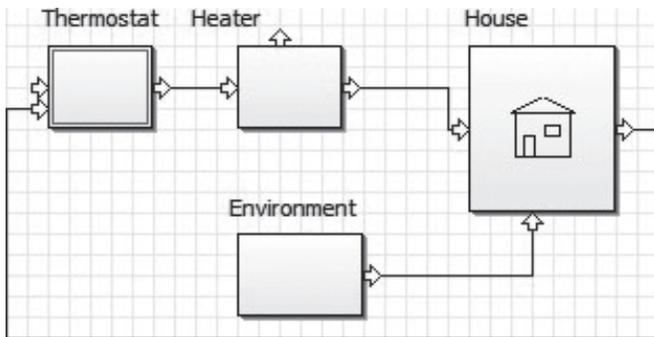


Рис. 8. RMD-модель системы термостатирования, построенная в виде блок-схемы

Любая модель, создаваемая в MVS (RMD), представляет собой «проект», содержащий стандартные компоненты. Система имеет наглядное представление поведения — карту поведения объекта, позволяющую описать логику его поведения (рис. 9).

Освоение пакета MVS (RMD) требует от учителя минимальной начальной подготовки в области информационных технологий и вполне доступно учащимся старших классов, что проверено на практике. В среде данного программного комплекса учитель имеет возможность разработать виртуальные лабораторные стенды

и использовать их в учебном процессе. Пример подобного стенда представлен на рисунке 10. Таким образом, учителя информатики, физики и других предметов могут создать великолепные наглядные модели изучаемых процессов и демонстрировать их на занятиях.

В 1999 году фирмой XJ Technologies (Санкт-Петербург) [10] разработана **система имитационного моделирования AnyLogic**, получившая признание по всему миру. В учебнике [3] построение моделей опирается на этот пакет (его школьную версию), который предоставляет возможность создавать дискретно-событийные модели, агентные модели, модели системной динамики, модели систем массового обслуживания.

Имитационное моделирование позволяет:

- изучать поведение системы, не имея детальных математических зависимостей для закономерностей протекающих процессов;
- проследить поведение системы в динамике;
- исследовать поведение систем, в которых наблюдается действие случайных факторов.

Типичные задачи, решаемые в системе AnyLogic:

- планирование и оптимизация перевозок;
- моделирование социальных и демографических процессов;
- моделирование рынка и поведения потребителей;
- моделирование дорожного движения и движения пассажиров (рис. 11) и т. д.

Учебник информатики для X класса И. А. Калинина, Н. Н. Самылкиной [3] предназначен для углубленного изучения информатики в естественно-научном, технологическом и социально-экономическом профилях. Изучение моделирования с использованием этого учебника и программного комплекса AnyLogic несомненно даст учащимся современный уровень знаний. Собственно, для изучения темы «Имитационное моделирование» пакет AnyLogic (школьная версия), по нашему мнению, является оптимальным средством.

Следует отметить, что выше нами рассмотрены только инструментальные системы моделирования

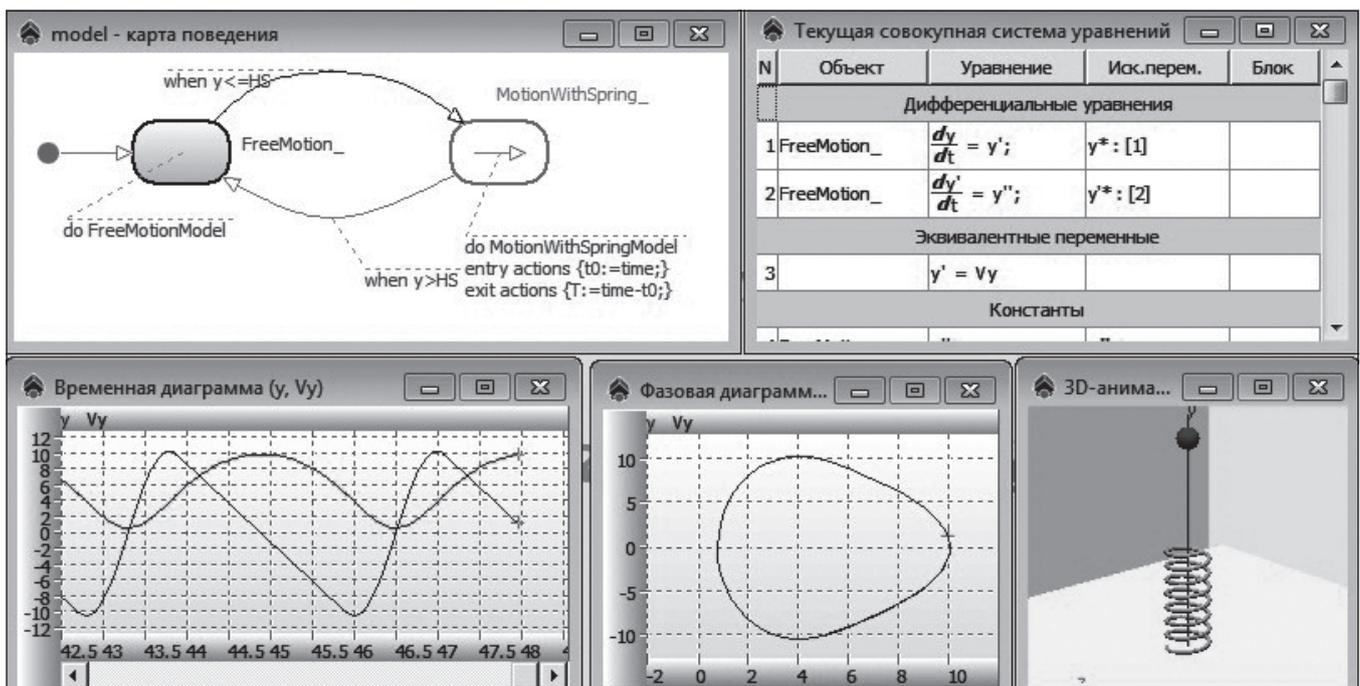


Рис. 9. Карта поведения RMD-модели, система уравнений и отображение результатов моделирования в виде диаграмм и 3D-анимации

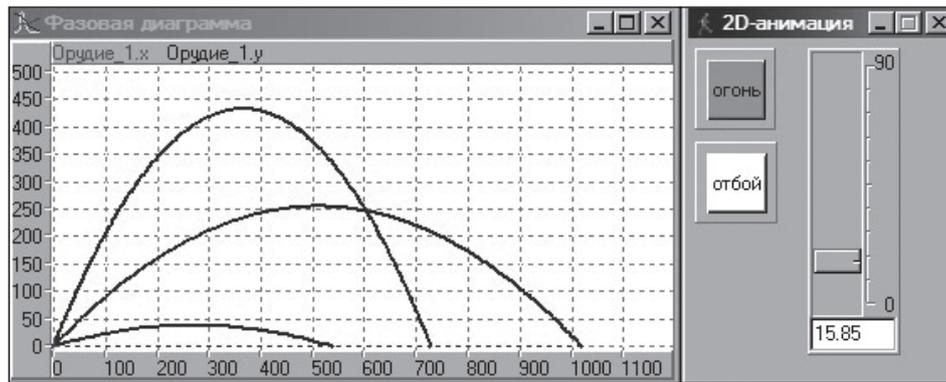


Рис. 10. Виртуальный лабораторный стенд «Орудие» с элементами управления

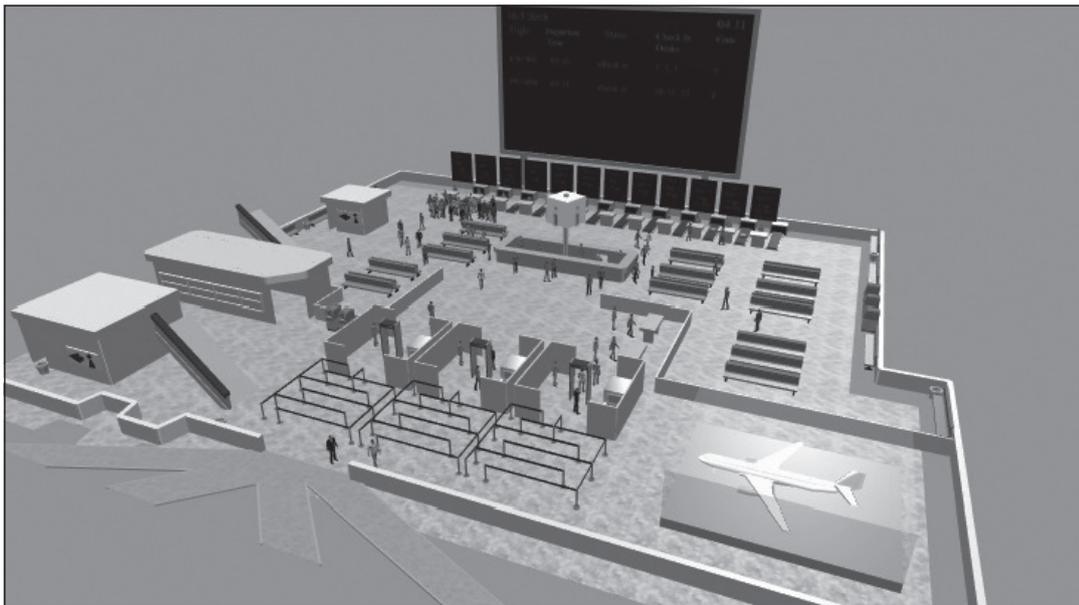


Рис. 11. AnyLogic-модель движения пассажиров в аэропорте Кольцово (Екатеринбург) [10]

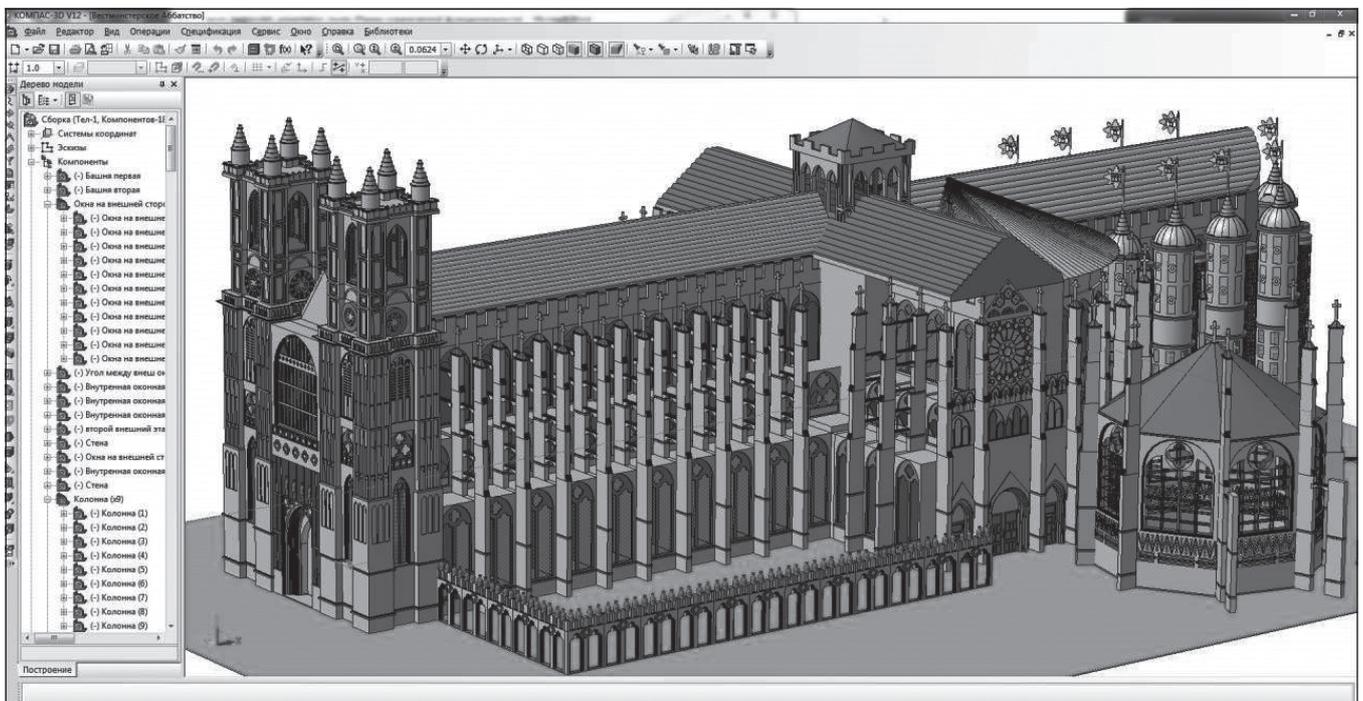


Рис. 12. Собор Парижской богородицы.

Проект выполнен школьником в среде «КОМПАС» и представлен на конкурс «Асы 3D-моделирования» [8]

общего назначения. Кроме них существует множество *специализированных* учебных пакетов моделирования, которые применяются для решения задач конкретной учебной дисциплины. Примером могут служить пакеты для проведения компьютерных экспериментов при изучении физики: «Живая физика» (Interactive Physics, Knowledge Revolution, США) [9], «Физический конструктор» (фирма «1С») [4], «Интерактивная физика» (Институт инновационных технологий) [13]. Сюда же добавим пакеты «Интерактивная химия», «Интерактивная информатика» [13] и множество аналогичных продуктов по другим предметам тех же разработчиков.

Все программные комплексы моделирования («КОМПАС», MVS (RMD), AnyLogic — школьная версия) могут использоваться учащимися как дома, так и в школе при различных формах проведения занятий. Они не требуют особых параметров компьютеров, созданы под операционную систему Windows, однако, позволяют на современном уровне освоить школьные курсы за счет внедрения элементов эксперимента и исследования в учебный процесс. Естественно, залогом успешного применения программных комплексов моделирования является изучение предметной области, т. е. учебной дисциплины.

Представленные программные комплексы демонстрируют эффективное использование современных технологий для моделирования и визуализации объектов и процессов, следовательно, повышают мотивацию учащихся на занятиях.

Эти программные комплексы моделирования являются прекрасной основой для организации разработки учебных проектов и решения творческих задач (рис. 12).

Список использованных источников

1. *Веников В. А., Веников Г. В.* Теория подобия и моделирование. М.: Высшая школа, 1984.
2. Информатика. Задачник-практикум. В 2 т. / под ред. И. Г. Семакина, Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
3. *Калинин И. А., Самылкина Н. Н.* Информатика. Углубленный уровень: учебник для 10 класса. М.: БИНОМ. Лаборатория знаний, 2014.
4. Каталог продуктов. Образовательные программы 1С // Компьютерные программы для образования фирмы 1С. <http://obr.1c.ru/educational/>
5. *Королев А. Л.* Компьютерное моделирование. М.: БИНОМ. Лаборатория знаний, 2010.
6. Математический энциклопедический словарь / под ред. Ю. В. Прохорова. М.: Сов. энциклопедия, 1988.
7. *Никитин А. В., Слободянюк А. И., Шишаков М. Л.* Компьютерное моделирование физических процессов. М.: БИНОМ. Лаборатория знаний, 2011.
8. Образовательный сайт компании ASCON. <http://edu.ascon.ru/main/news/>
9. Фестиваль педагогических идей «Открытый урок» // Издательский дом «Первое сентября». <http://открытыйурок.рф/>
10. AnyLogic: имитационное моделирование для бизнеса: сайт компании AnyLogic. <https://www.anylogic.ru>
11. MathWorks — Makers of MATLAB and Simulink: сайт компании MathWorks. <http://www.mathworks.com>
12. Rand Model Designer — имитационное моделирование сложных динамических систем: сайт компании MvStadium Group. <http://www.mvstadium.com>
13. Stratum: Инновационные разработки в сфере образования и промышленности: сайт ООО «Институт инновационных технологий». <http://stratum.ac.ru>

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/>

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

e-mail: readinfo@infojournal.ru

телефон: (495) 140-19-86

Журнал «Информатика в школе»

Индексы подписки (агентство «Роспечать»)
на 1-е полугодие 2019 года

- 81407 — для индивидуальных подписчиков
- 81408 — для организаций

Периодичность выхода: 5 номеров в полугодие (в январе не выходит)

Редакционная стоимость:
индивидуальная подписка — 150 руб.
подписка для организаций — 300 руб.



Федеральное государственное унитарное предприятие "Почта России" Ф СП - 1											
Бланк заказа периодических изданий											
АБОНЕМЕНТ На газету Информатика в школе журнал <small>(наименование издания)</small>	<input style="width: 100%;" type="text"/> <small>(индекс издания)</small>										
Количество комплектов	<input style="width: 100%;" type="text"/>										
На 2019 год по месяцам											
1	2	3	4	5	6	7	8	9	10	11	12
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Куда <input style="width: 40%;" type="text"/>											
<small>(почтовый индекс)</small> <small>(адрес)</small>											
Кому <input style="width: 100%;" type="text"/>											
----- Линия отреза -----											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> </tr> <tr> <td style="text-align: center;">ПВ</td> <td style="text-align: center;">место</td> <td style="text-align: center;">литер</td> </tr> </table>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	ПВ	место	литер	ДОСТАВОЧНАЯ <input style="width: 100%;" type="text"/> КАРТОЧКА <small>(индекс издания)</small>				
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>									
ПВ	место	литер									
На газету журнал Информатика в школе <small>(наименование издания)</small>											
Стоимость	подписки руб.		Количество комплектов								
	каталожная руб.										
	переадресовки руб.										
На 2019 год по месяцам											
1	2	3	4	5	6	7	8	9	10	11	12
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
почтовый индекс		Город		село		область		Район		улица	
код улицы		дом		корпус		квартира		Фамилия И.О.			

Электронная подписка на журналы ИНФО

Журналы по методике
обучения информатике
и информатизации образования



- ✓ Доступ к журналам не дожидаясь печати типографии
- ✓ С любого устройства, подключенного к Интернет
- ✓ Возможность сохранить файл в формате PDF
- ✓ В два раза дешевле печатной подписки
- ✓ Скидки при оформлении подписки на комплект журналов
- ✓ Оплата на сайте издательства в Интернет-магазине

Информатика и образование

ИЗДАЕТСЯ С 1986 ГОДА • ОТ 64 СТРАНИЦ • ВЫХОДИТ 10 РАЗ В ГОД

Научно-методический журнал по методике преподавания информатики и информатизации образования

Информатика в школе

ИЗДАЕТСЯ С 2002 ГОДА • ОТ 64 СТРАНИЦ • ВЫХОДИТ 10 РАЗ В ГОД

Научно-практический журнал для учителей информатики, методистов, преподавателей вузов и колледжей

Подробную информацию о подписке на наши издания вы можете найти на сайте

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





ХІХ МЕЖДУНАРОДНАЯ НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

Основные направления работы:

- Перспективы развития технологий 1С для создания инфраструктуры цифровой экономики и обновления системы образования.
- Технологическое и методическое обеспечение подготовки граждан к условиям цифровой экономики на основе платформы «1С:Предприятие» и ее прикладных решений.
- Методические, организационные и технологические средства поддержки педагогической деятельности, разработанные на основе решений «1С».
- Создание условий для расширения участия индустрии 1С в системе профессионального образования. Развитие форм сотрудничества образовательных организаций и работодателей.

Мероприятия в рамках конференции:

- Пленарные и секционные заседания
- Мастер-классы по использованию программных продуктов фирмы «1С»
- Вернисаж программных и методических разработок
- Тестирование «1С:Профессионал» по программным продуктам «1С:Предприятие 8»

В 2018 году в конференции приняли участие более 2 300 человек.

Подробнее о тематиках конференции и условиях участия см. сайт 1c.ru/educonf

Участие бесплатное для всех сотрудников образовательных организаций и органов управления образованием (проживание оплачивается отдельно).

Обязательная предварительная регистрация открыта до 28 января 2019 года на сайте 1c.ru/educonf



ФИРМА «1С»
Оргкомитет конференции:
Тел./факс: +7 (495) 688-90-02
Email: npk@1c.ru
www.1c.ru/educonf

29-30 января – 2019 г.
Гостиница «Космос»,
Москва, проспект Мира, д. 150