

ИНФОРМАТИК А

4

А.В. Горячев:

Только давайте
без фанатизма

8

А.Н. Комаровский:

Основной недостаток
«Агата» заключался
в том, что он был
единственным

18

А.Г. Леонов:

Я всегда хотел стать
программистом

электронная версия журнала
дополнительные
материалы
в Личном кабинете
на сайте
www.1september.ru



НА ОБЛОЖКЕ

► Это было только что... Это было целую эпоху назад... Трудно сказать, к какому из утверждений ближе ощущение, которое возникает, когда осознаешь, что Фейсбуку (не самой "старой" из социальных сетей) в феврале исполнилось 10 лет. Можно спорить, говорить, что не самому Фейсбуку, а его прототипу, тому, с чего все началось. Это все не очень существенно — 10 лет все равно уже позади. Невероятно!

На фото – университетский кампус Гарварда. Где-то на этих стульях все и начиналось

В НОМЕРЕ

- 3** ПАРА СЛОВ
 - Шедевры не для всех
- 4** ИНФОРМАТИКА В ЛИЦАХ
 - А.В. Горячев
 - А.Н. Комаровский
 - А.Г. Леонов
- 22** ЯЗЫКИ ПРОГРАММИРОВАНИЯ
 - Новые старые знакомые
 - Графическая среда программирования Blockly (Блокли)
- 42** WEB-ТЕХНОЛОГИИ
 - Создаем скринкаст на платформе HTML5
- 48** ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ
 - "В мир информатики" № 196

В ЛИЧНОМ КАБИНЕТЕ

Облачные технологии от Издательского дома "Первое сентября"

Уважаемые подписчики бумажной версии журнала!

Дополнительные материалы к номеру и электронная версия журнала находятся в вашем Личном кабинете на сайте www.1september.ru.

Для доступа к материалам воспользуйтесь, пожалуйста, кодом доступа, вложенным в январский номер.

Срок действия кода: с 1 января по 30 июня 2014 года.

Для активации кода:

- зайдите на сайт www.1september.ru;
- откройте Личный кабинет (создайте, если у вас его еще нет);
- введите код доступа и выберите свое издание.

Справки: podpiska@1september.ru или через службу поддержки на портале "Первого сентября".



ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ

► Презентации к статьям номера.

ИНФОРМАТИКА

ПОДПИСНЫЕ ИНДЕКСЫ по каталогу "Почта России": 79066 — бумажная версия, 12684 — электронная версия

<http://inf.1september.ru>

Учебно-методический журнал для учителей информатики
Основан в 1995 г.
Выходит один раз в месяц

РЕДАКЦИЯ:
гл. редактор С.Л. Островский
редакторы

Е.В. Андреева,
Д.М. Златопольский
(редактор вкладки
"В мир информатики")

Дизайн макета И.Е. Лукьянов
верстка Н.И. Пронская
корректор Е.Л. Володина
секретарь Н.П. Медведева
Фото: фотобанк Shutterstock
Журнал распространяется по подписке
Цена свободная
Тираж 26 731 экз.
Тел. редакции: (499) 249-48-96
E-mail: inf@1september.ru
<http://inf.1september.ru>

ИЗДАТЕЛЬСКИЙ ДОМ "ПЕРВОЕ СЕНТЯБРЯ"

Главный редактор:
Артем Соловейчик
(генеральный директор)

Коммерческая деятельность:
Константин Шмарковский
(финансовый директор)

Развитие, IT и координация проектов:
Сергей Островский
(исполнительный директор)

Реклама, конференции и техническое обеспечение Издательского дома:
Павел Кузнецов

Производство:
Станислав Савельев

Административно-хозяйственное обеспечение:
Андрей Ушков

Педагогический университет:
Валерия Арсланьян (ректор)

ГАЗЕТА ИЗДАТЕЛЬСКОГО ДОМА
Первое сентября – Е.Бирюкова

ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА
Английский язык – А.Громушкина
Библиотека в школе – О.Громова
Биология – Н.Иванова
География – О.Коротова
Дошкольное образование – Д.Тюттерин
Здоровье детей – Н.Сёмина
Информатика – С.Островский
Искусство – О.Волкова
История – А.Савельев

Классное руководство и воспитание школьников – М.Битанова
Литература – С.Волков
Математика – Л.Рослова
Начальная школа – М.Соловейчик
Немецкий язык – М.Бузоева

ОБЖ – А.Митрофанов
Русский язык – Л.Гончар
Спорт в школе – О.Леонтьева
Технология – А.Митрофанов
Управление школой – Е.Рачевский
Физика – Н.Козлова
Французский язык – Г.Чесновицкая
Химия – О.Блохина
Школа для родителей – Д.Тюттерин
Школьный психолог – М.Чибисова

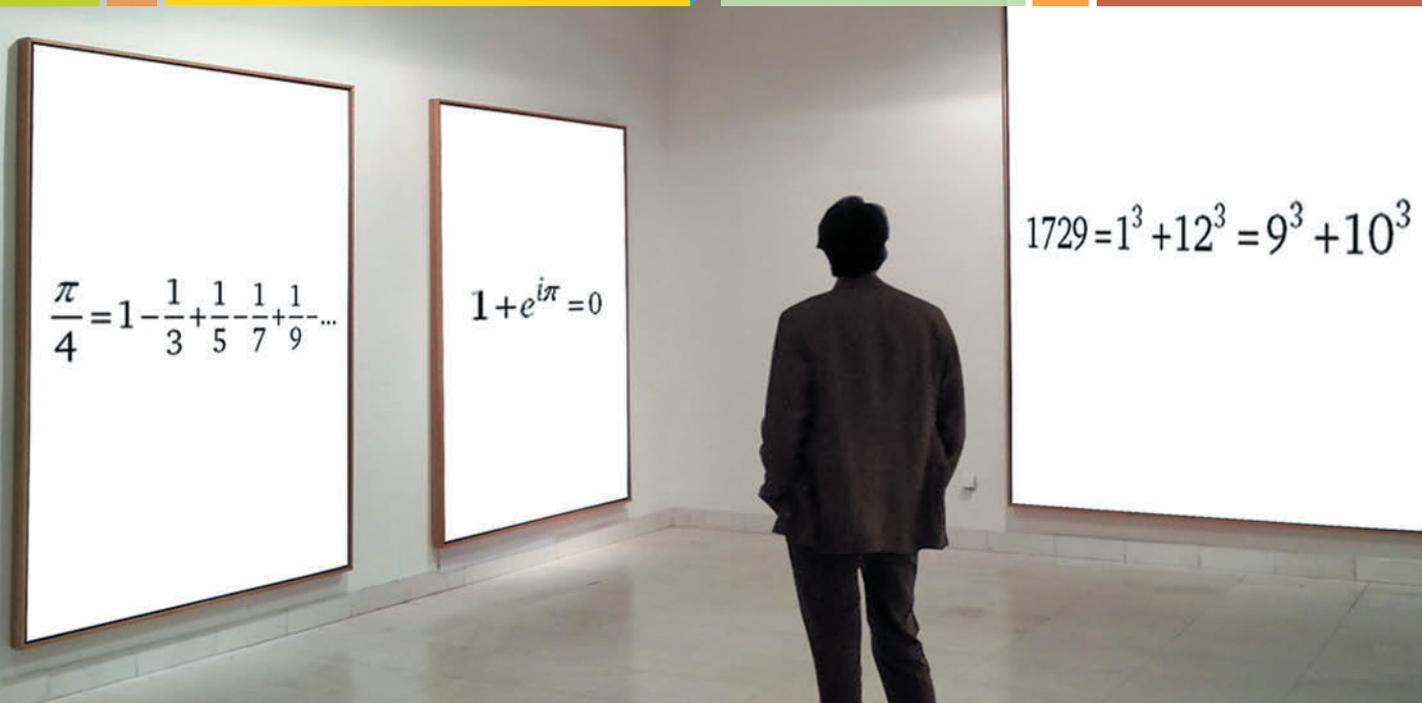
УЧРЕДИТЕЛЬ:
ООО "ЧИСТЫЕ ПРУДЫ"

Зарегистрировано ПИ № ФС77-44341 от 22.03.2011 в Министерстве РФ по делам печати
Подписано в печать: по графику 15.02.2014, фактически 15.02.2014
Заказ № Отпечатано в ОАО "Первая Образцовая типография" Филиал "Чеховский Печатный Двор" ул. Полиграфистов, д. 1, Московская область, г. Чехов, 142300
Сайт: www.chpk.ru
E-mail: sales@chpk.ru
Факс: 8 (495) 988-63-76

АДРЕС ИЗДАТЕЛЯ:
ул. Киевская, д. 24, Москва, 121165
Тел./факс: (499) 249-31-38

Отдел рекламы:
(499) 249-98-70
<http://1september.ru>

ИЗДАТЕЛЬСКАЯ ПОДПИСКА:
Телефон: (499) 249-47-58
E-mail: podpiska@1september.ru



Шедевры не для всех

► Математика подарила миру множество объектов поразительной красоты. Первое, что приходит в голову, — фракталы. Целый ряд издательств выпускает альбомы фрактальных изображений, которые на полках книжных магазинов часто соседствуют со сборниками работ дизайнеров и художников. Но оказывается, что профессиональные математики способны получать эстетическое удовольствие не только (и даже не столько) от действительно завораживающих изображений, порожденных глубинами математики, а от “самых обычных” сухих формул. Математики способны оценить, понять и почувствовать внутреннюю гармонию, красоту, утонченность и изысканность математических выражений, которые большинство людей не просто не поймут, но даже и не заметят.

Выяснили это — вы будете смеяться — британские ученые. Но без шуток — самые настоящие ученые и самые настоящие британцы. Эксперимент был довольно простым. Посредством МРТ измерялась мозговая активность математиков, которым предъявлялись различные формулы. Ученых просили оценить красоту каждой формулы по простой шкале. В ходе экспе-

римента выявилась стойкая корреляция между оценками различных математиков — множество красивых формул определялось с высокой степенью надежности. Но самое интересное, что созерцание красивых формул активировало эмоциональную часть коры головного мозга. То есть математики не “знали” или “думали”, что эти формулы красивы, а чувствовали это!

Кстати, в ходе эксперимента были выявлены не только красивые, но и некрасивые (по “чувственному мнению” участников эксперимента) формулы.

Руководитель исследования, профессор Лондонского университета Semir Zeki в статье, посвященной проведенному исследованию, написал: “...Мы обнаружили, что активность в головном мозге тесно связана с тем, как люди заявляют о своих чувствах о красоте. Это может быть ответом на вопрос, который долгое время мучил эстетов, а именно, могут ли эстетические переживания быть определены количественно... Для многих из нас математические формулы кажутся непонятными и недоступными, но у математиков математические уравнения могут быть эталоном красоты. Платон считал, что абстрактное качество математики является вершиной красоты”.

С.Л. Островский,
главный редактор
(so@1september.ru)



Горячев Александр Владимирович

Кандидат педагогических наук, лауреат премии Правительства РФ в области образования за 2008 год, руководитель авторского коллектива курса “Информатика в играх и задачах” (учебники для 1–4-х классов, учебные пособия для 5–6-х классов), автор пособия для дошкольников “Все по полочкам”, автор учебников для 3–4-х классов “Информатика и ИКТ. Мой инструмент — компьютер”, “Информатика. Логика и алгоритмы”, автор справочников-практикумов для обучения работе с компьютерными программами, руководитель авторского коллектива учебников по информатике для 7–9-х классов. Выпускник факультета прикладной математики МИЭМ (1982), научный сотрудник

Института точной механики и вычислительной техники им. С.А. Лебедева (1982–1998). Координатор направления “Информатика и ИКТ” в Образовательной системе “Школа-2100”.

О себе. Предпочитаю правило “Только давайте без фанатизма”. Дело в том, что я очень высоко ценю здравый смысл, а он несовместим с фанатизмом. Убежден, что даже неуверенное или медленное движение в правильном направлении гораздо лучше уверенного и быстрого движения не туда (на мой взгляд, именно это сейчас происходит с электронными учебниками — движение в тупик). Люблю читать фантастику. Фантасты тоже создают свои миры. Этим они в чем-то похожи на программистов. Я всегда считал фантастику литературой по специальности для программистов.

Как Вы пришли в информатику? Первый компьютер, на котором Вы работали? Первый язык программирования, с которым Вы познакомились? Первые задачи, которые Вы решали?

Первый компьютер — “Урал-1”. Он стоял в школе в отдельной комнате, и у старшекласников были уроки по вычислительной математике. Свои первые программы мы писали в машинных кодах. Мы выполняли работу компиляторов: распределяли память, программировали вычисление выражений. Составленные в машинных кодах программы мы пробивали на черной 35-миллиметровой киноплёнке с помощью специального устройства. При этом, если при отладке программы для внесения исправления было достаточно поменять уже пробитые на плёнке команды, мы иногда либо заклеивали пробитые отверстия, либо прорезали новые. Как правило, первые задачи были на какие-либо вычисления, они предполагали выполнение итераций в цикле до выполнения условия завершения работы. В качестве теории мы изучали курс вычислительной математики. В институте я программировал мало. Когда я начал работать в Институте точной механики и вычислительной техники (ИТМ и ВТ), то работал на БЭСМ-6, программировал на Автокоде. Наш коллектив занимался созданием программного обеспечения для разрабатываемых в институте ЭВМ.

Первые пять фамилий из мира информатики, которые приходят Вам в голову? Если можете, пожалуйста, коротко расскажите, почему именно они.

(Прембула: выносим за скобки фамилии Вирт, Дейкстра, Буч, Кнут и т.д.) Получаем примерно такой список: Ершов. Первин. Бородин. Лебедев. Андреева. 1. Андрей Петрович Ершов как создатель школьной информатики. 2. Юрий Абрамович Первин как основоположник

раннего обучения информатике. 3. Михаил Николаевич Бородин как символ информатики сегодняшнего дня, это человек с очень интересными идеями, поддерживающий многих авторов, к которым я отношусь с большим уважением. 4. Сергей Алексеевич Лебедев как руководитель коллектива разработчиков БЭСМ-6, в этом коллективе работали люди, у которых впоследствии я учился программированию. Сергей Алексеевич для меня одновременно и символ успешных отечественных разработок в сфере ИТ, и символ всех моих учителей. 5. Елена Владимировна Андреева как яркий представитель нашего олимпиадного движения. Я считаю, что наша система подготовки и проведения олимпиад — это очень значимое достижение, которое ни в коем случае не должно быть утеряно ни при каких реформах. Хотя должен честно признаться, что предложение рассказать “о чем вы думаете, когда слышите это слово” может давать непредсказуемые результаты. Как я люблю говорить: “Я не отвечаю за свои ассоциации. Они живут своей жизнью”.

Какая компьютерная техника и IT-сервисы окружают Вас сейчас? Насколько Вы “компьютерный человек” в обычной жизни? Это вопрос про все — про компьютеры и гаджеты, которые Вас окружают, про Ваш личный стиль социальной коммуникации в сети — насколько Вы живете в социальных сетях, насколько не можете жить без электронной почты и других средств коммуникации. Насколько бережете или не бережете свое личное пространство и время.

Гаджетов вокруг меня довольно много. Читаю книги на электронной “читалке”, в качестве навигаторов использую планшеты, в любых поездках не расстаюсь с ноутбуком — всегда держу работу под рукой. Недавно установил домашний сервер и хочу иметь издалека доступ ко всем своим файлам. В социальных сетях я не активен, но с интересом читаю и иногда высказываюсь в профессиональных (педагогических) сетевых сообществах. Со сбережением времени все сложнее. Я человек увлекающийся — это помогает при увлечении очередным проектом, но может и помешать при временном увлечении чем-либо с работой не связанным.

Если бы в Вашем распоряжении была лишь одна лекция (допустим — “пара”, полтора часа) и полная свобода рассказывать о том, что Вам интересно в информатике, чему бы Вы посвятили это время?

Вопрос, для кого бы я читал эту лекцию. Пожалуй, для учителей информатики. И я скорее всего рассказывал бы о том, что меня больше интересует в данный момент. Сейчас это вопросы отражения возрастной периодизации в непрерывном курсе информатики. В информатике есть довольно сложные темы, а часов очень мало. Недостаток ресурсов стимулирует поиск оптимальных решений. Поэтому для нас особо остро стоит вопрос оптимального выбора: в каком возрасте чему учить и какие качества развивать. И, естественно, малое число часов в сочетании с логической сложностью порождают желание использовать различные способы групповой и индивидуальной дифференциации обучения. То есть, так или иначе, речь шла бы о том, как получить максимально возможный результат в условиях острой нехватки учебных часов.

Есть ли какой-то факт из информатики — может быть, какой-то алгоритм, — который произвел на Вас сильное, запоминающееся впечатление?

Когда-то давным-давно я был впечатлен методами быстрой сортировки.

Есть ли у Вас любимая задачка или задачки?

Много лет назад я в процессе знакомства с объектно ориентированным подходом в программировании переписал простую игрушку под MS DOS про космическую перестрелку в объектно ориентированном стиле. Было очень интересно наблюдать, как при этом моделируется пространство и время. Стало очень наглядным, что объектно ориентированное программирование — это (по большому счету) именно моделирование. Это самое близкое воспоминание, подходящее к понятию “любимая задачка”.

Какую тему/темы школьного курса (будем считать, что мы находимся в пространстве ФГОС) Вы не хотели бы рассказывать детям? Ну не нравится она Вам.

Мне не очень нравится философствовать о том, как нас окружает информация, как информатика изучает информационные процессы

разной природы. Видимо, я люблю в первую очередь обращать особое внимание на освоение тех или иных способов действий. В нашем предмете — освоение способов действий человека с помощью компьютерной техники и средств связи. Конечно, эти способы действий включены в какие-то интеллектуальные процессы, и с закономерностями этого контекста необходимо знакомиться, чтобы применение средств ИТ было не самоцелью, а позволяло эффективно решать ту или иную интеллектуальную задачу. Возможно, подобные темы было бы интересно рассматривать как итог, обобщение после освоения средств ИТ в различных действиях.

Надо ли сейчас вообще учить предмету “информатика” в школе? Ведь навыки работы на компьютерах, в сетях, с прикладным ПО дети получают и без этого и гораздо раньше, чем начинают изучать предмет. А специалистов вполне можно готовить в вузах. В школе-то зачем?

Этот вопрос возвращает нас к теме возрастной периодизации. Например, мы считаем, что опоздание с развитием логического мышления — это опоздание навсегда. Кроме того, очень важно еще с детства дать почувствовать это незабываемое ощущение, когда что-то сделанное на компьютере вдруг начинает работать. Это может быть не обязательно компьютерная программа, но и, к примеру, мультфильм. Я полагаю, что отношение к компьютеру не только как к игрушке, а как к инструменту с огромными и разнообразными возможностями (но и с принципиальными ограничениями) может быть заложено именно в школе, и в первую очередь на уроках информатики. Это чем-то напоминает импринтинг, когда цыпленок считает своей мамой то, что увидел сразу вылупившись. Ощути компьютер как полезный инструмент. Ну и, конечно, раннее обучение будущих создателей компьютерной техники и программного обеспечения позволит им получить конкурентное преимущество.

Какой Вы представляете себе информатику в школе через пять лет? А через десять лет?

Наверное, это самый сложный вопрос. Отвечая на него, наверное, стоит оглянуться назад и попытаться увидеть в курсе информатики инварианты, которые отвечали бы каким-то настолько существенным жизненным реалиям, что можно было бы ожидать их существования и в дальнейшем. Одним из таких инвариантов можно считать изначальную двойственность курса, который должен сочетать обучение пользователей и обучение разработчиков. Как мы помним, во времена появления курса информатики единственным способом применения компьютера в своей деятельности было написание компьютерных программ. Поэтому известный лозунг “Программирование — вторая грамотность” можно рассматривать как призыв к массовой подготовке пользователей. Полезно помнить, что во времена начального этапа обучения информатике программирование было и инструментом пользователей, и инструментом профессиональных разработчиков программного обеспечения. Но в дальнейшем усилия разработчиков привели к появлению значительно более широкого спектра инструментов для пользователей. Это привело к тому, что интересы обучающихся пользователей и обучающихся разработчиков стали расходиться. Это не чей-то злой умысел, это объективное свойство учебного предмета, появившегося вслед за активным развитием нового инструмента — средств вычислительной техники. Если на уроках физики, химии, биологии, географии мы видим за партами пользователей и исследователей окружающего нас мира, то на уроках информатики перед нами сидят как пользователи, так и будущие разработчики (при этом хорошо бы, чтобы разработчики сами были бы квалифицированными пользователями). Нашим профильным вузам интересны именно будущие разработчики. Кстати, поскольку поступление в вузы определяется результатами сдачи ЕГЭ и выступлениями на олимпиадах, то нет ничего удивительного в том, что наши олимпиады по информатике — это олимпиады по программированию. Все естественно, ведь олимпиады — это путь в профильные вузы. Здесь необходимо сделать одно пояснение. Я не принимаю критику обучения пользовательским навыкам на том основании, что современные дети и так живут в среде гаджетов и в Сети. Давайте

рассматривать два отдельных вида использования средств ИТ: в быту и развлечениях и в профессиональной деятельности. Жизнь в Сети и среди гаджетов относится к пользовательским умениям в быту и развлечениях. В профессиональной деятельности можно рассматривать применение средств ИТ как в универсальных действиях (познавательных, регулятивных, коммуникативных), так и в специфических действиях какой-либо профессиональной сферы (например, специфические действия, характерные для профессий экономического профиля). Вторым инвариантом для курса информатики мы считаем возрастную периодизацию. Имеются в виду сенситивные периоды — возрастные периоды, наиболее благоприятные для развития тех или иных психологических свойств (нас в первую очередь интересуют познавательные процессы). Исходя из этих двух инвариантов, можно предположить какие-то черты обучения информатике и через пять лет, и через десять. **1.** В 1–6-х классах ориентация на развитие логического и алгоритмического мышления, на развитие фантазии и творческого воображения (очень желательные качества для “создателей миров”). Очевидна польза от этого развития для процесса обучения в целом. Это проявляется и в совпадении интересующих нас мысленных действий с универсальными логическими действиями (составная часть познавательных универсальных учебных действий). Этот этап обучения может завершаться развивающим развлекательным программированием, доступным детям с разным уровнем развития познавательных процессов. Идеальным инструментом для такого развивающего развлекательного программирования были бы конструкторы приложений, позволяющие создавать простые приложения конструированием, а при желании позволяющие дополнять конструирование алгоритмическими конструкциями (что-то наподобие RPGMaker). Кроме того, на этом этапе было бы хорошо учить школьников использовать средства ИТ в своей основной деятельности — учебной, а также применять средства ИТ в созидательной деятельности в рамках различных учебных предметов и во внеурочной деятельности. **2.** В 7–9-х классах уже сейчас мы наблюдаем детей, которые способны и/или имеют желание заниматься профильной информатикой. Я не думаю, что эта тенденция куда-то исчезнет. Сейчас в лучшем случае эти дети выделены в отдельные группы (школы, классы, подгруппы) со своей образовательной программой. Я надеюсь, что в течение и пяти, и десяти лет будут совершенствоваться формы обучения школьников, увлекающихся программированием вообще и программированием приложений в частности. Для школьников, не ориентирующихся на информатику как профиль, целесообразно было бы освоение средств ИТ как инструмента в универсальных действиях (универсальных по отношению к профессиям и к учебным профилям): в регулятивных, коммуникативных, познавательных действиях. Предметные результаты такого обучения мы вполне можем считать и метапредметными результатами, поскольку и ИКТ-компетентность, и универсальные учебные действия сами по себе, согласно ФГОС, относятся к метапредметным результатам обучения. **3.** В 10–11-х классах на этапе профильного обучения логичным было бы ориентировать курсы информатики на освоение средств ИТ в профильно ориентированных действиях, плюс, возможно, еще какие-либо из универсальных надпрофессиональных действий, не рассмотренных в основной школе. Перечень этих универсальных и профильно ориентированных действий может со временем меняться. Эти изменения потребуют дополнения учебников, но не настолько быстро, как смена версий компьютерных программ.

Какой вопрос Вы хотели бы задать себе, чтобы на него ответить?

А что делает ваш коллектив, чтобы приблизить это будущее? Ответ на этот вопрос может оказаться очень объемным. Лучше заходите на сайт Образовательной системы “Школа-2100”, знакомьтесь с предметными программами, выбирайте в каталоге наши учебники (на страницах учебников для основной школы не пропустите ссылку “Подробное описание разделов учебника”), интересуйтесь, какие материалы по информатике предложены в разделе “Скачать”, и скачивайте в свое удовольствие.



Комаровский Анатолий Николаевич

Во время студенческой практики попал в Россошанскую школу-интернат. Прикипел и работает там вот уже 42-й год. “Реликтовый” информатик. Скептик и педант. Головой любит работать больше, чем руками. Обожает молоко и тонкий юмор. Не любит глаголов несовершенного вида.

Обладатель сертификата “Microsoft Office Specialist”. Высшая квалификационная категория с 1995 года. Победитель Конкурса лучших учителей РФ в 2007 году. Лучший внештатный методист кафедры ИТО Воронежского ОИПКиПРО. Заслуженный учитель РФ. Награжден медалью “За доблестный труд”.

Делится своим богатым опытом работы на открытых уроках, мастер-классах, научно-практических конференциях, в многочисленных публикациях, в том

числе и на страницах журнала “Информатика”.

Дважды приглашался Издательским домом “Первое сентября” для проведения мастер-классов на Дне учителя информатики, проводимом в рамках ежегодного Московского педагогического марафона учебных предметов.

Разработчик известного конструктора тестов в MS PowerPoint и менеджера тестирования к нему, а также многочисленных интерактивных динамических моделей в MS Excel.

Он сам и его ученики — победители и призеры нескольких республиканских дистанционных конкурсов по информационным технологиям.

Как Вы пришли в информатику? Первый компьютер, на котором Вы работали? Первый язык программирования, с которым Вы познакомились? Первые задачи, которые Вы решали?

В году примерно 1970-м, в студенческую пору, у нас на физмате был небольшой курс работы с вычислительной техникой. На счётках, правда, не считали. Основной упор был на логарифмическую линейку и арифмометр “Феликс”.

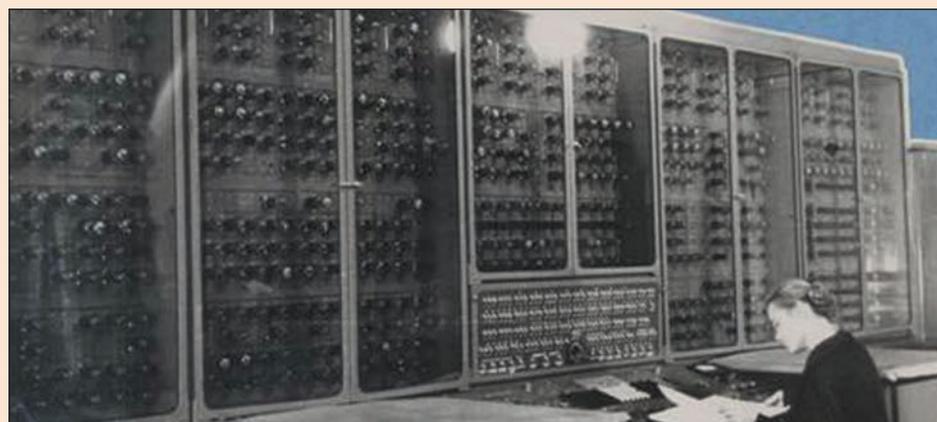
Несколько занятий было отдано знакомству с основами программирования для ЭВМ “Урал-4”. Сама ЭВМ присутствовала в аудитории в виде перефотографированной из какого-то научного



Арифмометр “Феликс”



Логарифмическая линейка



ЭВМ “Урал-4”

журнала черно-белой копии низкого качества. Я нашел исходник в Интернете.

Еще там был самодельный прибор в виде коробка размером с половину буханки хлеба. Над ним возвышались две электронные лампы. Сбоку была приклеена этикетка с загадочной надписью “Триггер”. Для чего он, почему оказался здесь, а не в лаборатории радиотехники и какое отношение имеет к изучаемому предмету — никто не знал.

До сих пор у меня хранится блокнот с системой команд, записанной перьевой авторучкой.

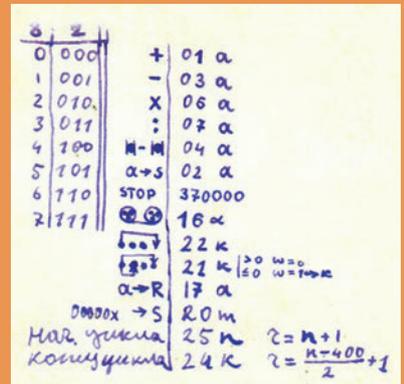
Мы писали в тетрадях примитивные программы на вычисления и возмущались: “Зачем нам, будущим учителям, это нужно? Ведь ЭВМ в школах никогда не будет?” Даже знаменитый в то время своими смелыми прогнозами на столетие американский писатель-фантаст, популяризатор науки Айзек Азимов не предполагал ничего подобного. Тем не менее первую ЭВМ я увидел еще до окончания института, когда в составе делегации от нашей области на Всесоюзный слет студентов оказался на экскурсии в НИИ нефтехимического синтеза. Помню несколько высоких, зеленого цвета металлических шкафов в ряд. Мигали сигнальные лампочки и что-то пиликало. Напротив — стол с бобинами перфоленты. Пытался найти в Интернете фото той ЭВМ. Вероятнее всего, это была “Минск-2”.



ЭВМ “Минск-2”

Мини-ЭВМ с дисплеем довелось увидеть меньше чем через четыре года, когда я был командирован в Москву на Всемирную выставку школьного оборудования. В павильоне Франции стоял компьютер размером с комод. Оператор держал в руках устройство, похожее на маркер с кабелем. Касаясь экрана, он строил отрезки, треугольники, прямоугольники и овалы. (Обладатели интерактивных досок, вам это ничего не напоминает? Шел 1975 год!) “Дорогое удовольствие! Какое же это школьное оборудование? Зажрались, буржуи”, — думалось мне.

Прошло еще 10 лет застоя. Никакие ЭВМ в работе школы не маячили даже на горизонте. В начале 1985 года к нам заглянул начальник управления образования области, завел с директором разговор на тему ЭВМ. Тот вызвал на выручку меня, в надежде на поддержку беседы. Так я узнал о том, что в школьный курс будет вводиться новый предмет — “Основы информатики и вычислительной техники”. Меня как самого “продвинутого” и направили в отпускную летнюю пору на курсы повышения квалификации — изучать алгоритмический язык. К сентябрю появилось тоненькое пробное учебное пособие (часть первая) под



Система команд
ЭВМ “Урал-4”



С этого пособия началась
школьная информатика



Калькулятор "Электроника БЗ-34"



УК-НЦ на базе ПЭВМ "Электроника МС 0511" (рабочее место учителя)



ПЭВМ "Агат"

редакцией А.П. Ершова. До сих пор храню эту брошюрку как реликвию. Об алгоритме Евклида я узнал именно из нее. С тех пор в школьной алгоритмизации "воды" стало больше, а содержания — меньше. Года два "трепали" бумажный вариант, а потом получили программируемые калькуляторы "Электроника БЗ-34". Помню, что на них умудрялись программировать не только вычисления, но даже некоторые простенькие игры типа игры Баше.

Первым реальным компьютером, который пришел к нам в школу, был 8-разрядный "Агат". К нему прилагался пакет программного обеспечения, куда, кроме Бейсика, входили отечественный язык программирования "Рапира", скромные по нынешним временам графический и текстовый редакторы да несколько игр. Принтера не было. Основным недостатком "Агата" заключался в том, что он был единственным. Клавиатуру перерисовывали на двойной лист из тетради и "осваивали". Первые программы, которые пришлось писать на Бейсике: МОД и НОД. МОД — модуль действительного числа — на ветвление, НОД — наибольший общий делитель — на повторение. Более-менее серьезно программированием пришлось заняться с появлением в школе компьютерного класса УК-НЦ на базе компьютеров "Электроника МС 0511".

Первые пять фамилий из мира информатики, которые приходят Вам в голову? Если можете, пожалуйста, коротко расскажите, почему именно они.

А.П. Ершов, Л.З. Шауцукова, Ю.А. Шафрин, С.А. Бешенков, Л.А. Бачурина. Может быть, я не совсем правильно понял ваш вопрос, и вы ожидали услышать от меня что-то из ряда: Бэббидж, фон Нейман, Ада Лавлейс, Лебедев, Шура-Бура? Но уж, извините. Специфика профессии накладывает свой отпечаток, и первое, что пришло в голову, я уже назвал. Это те, кто сыграл заметную роль в формировании меня как учителя информатики. Ведь в пединституте мы этот предмет не изучали, а, по словам Д.И. Писарева: "Настоящее образование есть только самообразование". Об А.П. Ершове как авторе первого учебного пособия по информатике я уже сказал. По учебному пособию для 7–11-х классов по курсу "Информатика и вычислительная техника" и "Практикуму по компьютерной технологии" к нему Ю.А. Шафрина я самостоятельно осваивал информационные технологии. Книга "Информатика: Учебное пособие для 10–11-х классов общеобразовательных учреждений" Л.З. Шауцуковой стала основой моих знаний в области алгоритмизации и устройства компьютера. Учебник С.А. Бешенкова "Информати-



В кабинете информатики и ИКТ (фото из архива автора)

ка. Систематический курс” помог мне углубить и систематизировать свой багаж знаний в области информатики. Книги этих авторов не переключиваются с моего стола на полку.

Что касается Людмилы Алексеевны Бачуриной — заведующей кафедрой информационных технологий в образовании Воронежского ИПКиПРО, то о ней мне хотелось бы сказать отдельно. Это образец влюбленности в свою работу, душевной неуспокоенности, активной жизненной позиции, генератор идей, инициатор и двигатель многих творческих проектов в области информатизации образования и педагогики сотрудничества, человек, который служит для меня, да и для других учителей информатики нашей области, эталоном профессионализма и отношения к гражданскому и служебному долгу.

Какая компьютерная техника и IT-сервисы окружают Вас сейчас? Насколько Вы “компьютерный человек” в обычной жизни? Это вопрос про все — про компьютеры и гаджеты, которые Вас окружают, про Ваш личный стиль социальной коммуникации в сети — насколько Вы живете в социальных сетях, насколько не можете жить без электронной почты и других средств коммуникации. Насколько бережете или не бережете свое личное пространство и время.

В кабинете информатики — комплект компьютеров “Apple iMac 20” с кабельной локальной сетью и Wi-Fi-доступом в Интернет. Работают очень тихо — не мешают думать.

Струйный и лазерный принтеры, фотоаппарат, кинокамера, два сканера, пять графических планшетов, интерактивная доска.

Чаще всего пользуюсь пакетом MS Office Professional 2010: Word, Excel, PowerPoint, Publisher, а также ABBYY FineReader 11. Графическими редакторами: Paint.NET, Adobe Photoshop. Звуковой редактор Audacity. Видеомонтаж: Киностудия Windows Live и Pinnacle Studio 16. На все коммерческое ПО есть лицензии.

Из браузеров предпочитаю Google Chrome. Из антивирусников — Dr.Web. Школьный Интернет фильтруется. Доступ к социальным сетям, YouTube, Instagram, Skype и другим развлечениям заблокирован. Пользуюсь электронной почтой и Агентом Mail.ru для видеосвязи. Дома к ноутбуку прикасаюсь редко (других забот хватает). Чаще его “мучает” жена. Доступа в Интернет пока нет. Мобильный телефон — без особых наворотов. Использую его прежде всего по прямому назначению и как будильник, иногда как калькулятор, редко как фотоаппарат.

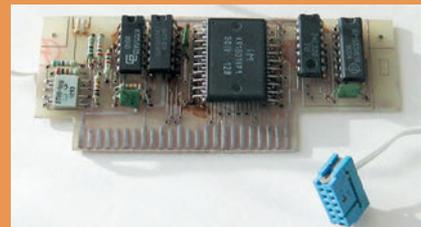
Если бы в Вашем распоряжении была лишь одна лекция (допустим — “пара”, полтора часа) и полная свобода рассказывать о том, что Вам интересно в информатике, чему бы Вы посвятили это время?

Мои интересы и увлечения за время работы учителем информатики менялись. Тем не менее одна страсть остается постоянной в течение более 20 лет. Это локальная сеть.

В КУВТ УК-НЦ (комплект учебной вычислительной техники — учебный комплекс — научный центр) на базе “Электроника МС 0511” был всего один двоянный блок приводов гибких дисков, который размещался на учительском столе. С него загружалась операционная система RT11SJ и другие программы: на учительский компьютер непосредственно, а на компьютеры учащихся — по сети. Это была кольцевая сеть Token Ring с маркерным доступом, основанная на иных стандартах и топологии, чем распространенная сейчас Ethernet. Сама по себе сеть была хоть и медленной, но довольно надежной и неприхотливой. Но ненадежными оказались дисководы с гибкими дисками, да и сами компьютеры (серебра и золота на контакты им не досталось). По этой причине загрузка всех компьютеров отнимала много времени, а частые сбои отнимали заметную часть урока. Но решение проблем нашлось. Начало 90-х — время расцвета кооперативного движения. Я поехал в Зеленоград и, кроме цветного дисплея, привез от продвинутых умельцев мини-винчестер на 500 Мб (тогда это была диковинка), электронный диск такой же емкости (пробораз современных флешек) размером почти с коробку из-под



Мини-винчестер и электронный диск



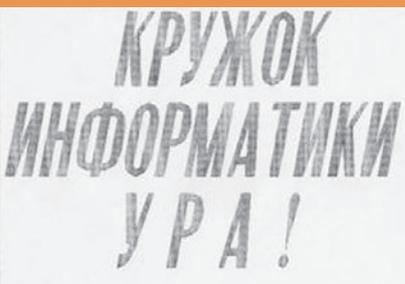
Адаптер “быстрой сети” Token Ring



Музыкальный редактор



Графический редактор



Векторный шрифт



Фрактальная графика

обуви и “быструю сеть”. В музее вычислительной техники Д.М. Златопольского я таких раритетов не видел. Дела, за исключением вечной проблемы с контактами, наладились.

Это было начало. Потом, с появлением в школе компьютеров IBM PC, пришлось самому монтировать и настраивать сначала сеть на основе коаксиала, потом витой пары от Ethernet до Gigabit Ethernet и Wi-Fi. Довелось в местных филиалах воронежского колледжа и московского института вести предмет “Компьютерные сети”.

Но свою лекцию (хотелось бы все же не лекцию, а практическое занятие) я посвятил бы не теории и физическим принципам построения компьютерных сетей. Мне приходилось бывать у многих своих коллег как в плане проведения мероприятий районного и областного методического объединения учителей информатики, так и процедуры аттестации. Везде меня интересовало, насколько эффективно использует учитель в своей деятельности локальную сеть. К сожалению, ситуация оставляет желать лучшего. Во многих случаях сеть не используется вовсе (сеть есть, а доступ к каким-либо сетевым ресурсам полностью закрыт) или сеть используется только для доступа в Интернет. То есть учителя попросту не могут настроить сеть, зачастую не знают, какие возможности она открывает, не имеют представления о существующем программном обеспечении (в основном свободном) для работы в локальной сети компьютерного класса. О своем опыте организации и работы в локальной сети я и хотел бы рассказать.

Есть ли какой-то факт из информатики — может быть, какой-то алгоритм, — который произвел на Вас сильное, запоминающееся впечатление?

Драйвер принтера на Бейсике. Мне довольно долго пришлось работать на КУВТ УК-НЦ. Программное обеспечение для него было скудным. Основной “ресурс” — вильнюсский Бейсик. Чтобы было чем занять детей, я добрался до сигнала ВЕЕР и, путем деления тактовой частоты процессора и привлечения учителя музыки с баяном, написал на безголосом Бейсике музыкальный редактор. Затем был создан графический редактор, который позволял рисовать с помощью графических примитивов (и не только), а также делать надписи в графическом режиме самодельным масштабируемым шрифтом. Сейчас многие уже и не ведают, что основным режимом на тех компьютерах был текстовый. Только позже, с появлением компьютеров IBM PC и Windows, я узнал, что мой музыкальный редактор, оказывается, является аналогом формата MIDI, шрифт (типа Arial) и графический редактор были векторными, что трава, цветы и деревья там рисовались с помощью фракталов. До выхода Windows 95 оставалось еще два года. Вскоре возникла проблема. Учителю музыки потребовалось распечатывать ноты, а дети хотели получить бумажные копии своих рисунков. Кнопки “Print Screen” на той клавиатуре не было. Вот тут мне и взбрела в голову идея написания для интерпретатора Бейсика того, что сейчас называют драйвером. Пришлось разобраться в устройстве, доступе и представлении данных в видеопамяти, системе команд принтера. Выявились проблемы чтения данных и преобразования форматов. В конце концов программа была написана.

Запускаю. Принтер безмолвствует. Тут меня отвлекают дети. Проходит минуты 2–3, и вдруг каретка матричного принтера “оживает” и медленно-медленно распечатывает первую строку шириной 8 пикселей. После продолжительной паузы также медленно появляется вторая строка. А что вы хотите от интерпретатора? У меня смешанное чувство: рад, что заработало, только эффект не тот.

Помню, пришлось помучиться над оптимизацией алгоритма, прежде всего путем отказа от арифметических операций и замены их побитными логическими. Когда копия экрана в графическом режиме стала распечатываться примерно за одну минуту, я испытал настоящий восторг! Вот этот алгоритм для меня самый памятный. Приложенные иллюстрации были распечатаны с помощью этого драйвера.

Есть ли у Вас любимая задача или задачи?

Да. У меня есть своя подборка задач на программирование, которые я прорешиваю с учениками. Большая часть из них отобрана еще из старых публикаций в газете “Информатика”. Может, кому-то это покажется странным, но моими любимыми являются стандартные “Обращение массива” и “Макс-элемент”. Дело в том, что, с одной стороны, они являются ключевыми, с другой стороны, алгоритмы решения этих задач, как никакие другие, подходят в качестве ролевых.

Стулья устанавливаются в ряд, нумеруются — это объявляется массив. Назначаются ответственные за проверку условия и за ведение значений счетчика на доске. В качестве элементов массива выступают сами учащиеся. Разобрали задачу обмена “чай-кофе”, выяснив, что необходим запасной стул. Начали обмен: первый с последним, второй с предпоследним... Кажется, все понятно, разогнались... Проскакиваем середину... И тут начинает нарастать протест, так как часть учеников понимает, что на том месте, где они только что оказались, им уже доводилось быть совсем недавно. Начинаем разбираться, в чем проблема? Оказывается, “очевидное” не всегда есть правильное. Продумывать надо не только начало цикла, но и его завершение!

В задаче на поиск максимального элемента ищем, кто в классе самый высокий. В положении “сидя” не всегда это очевидно. В крайнем случае прошу представить, что ответственный за проверку условия находится в лаборантской и процедуру сравнения роста выполняет там. Потом можно проанализировать таблицу исполнения алгоритма или выполнить трассировку с помощью панели “Отладка”, но чаще всего этой игровой ситуации бывает достаточно, чтобы большинство учащихся уловили основные особенности обработки массивов. Времени потрачено не так много, а эффективность высокая.

Какую тему/темы школьного курса (будем считать, что мы находимся в прострaнстве ФГОС) Вы не хотели бы рассказывать детям? Ну не нравится она Вам.

Основы логики. Причем не могу сказать, что она мне не нравится. Скорее наоборот. В студенческие годы математическая логика была одной из самых любимых моих дисциплин. В начале моей профессиональной деятельности произошел примечательный курьез, связанный с ней. Экзаменационные сочинения и работы по математике претендентов на золотую и серебряную медали “За отличные успехи в учении...” перепроверялись на уровне области. На курсах учителей математики в большой аудитории собрали все группы — около сотни человек. Председатель областной комиссии по математике давала образцы оформления экзаменационных работ на медаль только что закончившегося учебного года, растолковывала требования и тонкости. Вдруг я замечаю, что одно из заданий решено неверно. Я поднимаю руку и заявляю об этом. “Что вы себе позволяете, молодой человек!” — слышу я в ответ. Зал начинает высказывать возмущение в мой адрес. На что я заявляю: “Законы логики голосованием не отменяются”. “При чем здесь логика, мы разбираем задачу по математике”, — заявляет преподавательница. “А при том, что при переходе от системы неравенств к их совокупности вы нарушили правило де Моргана”, — отвечаю я. Поднялся такой шум, что меня, неучтливую балбеса, чуть не выпроводили из аудитории.

На следующий день эта женщина пришла к нам на занятия, отозвала во время перерыва меня в сторону и сказала: “А ведь вы были правы”. Теперь, почему не хотел бы рассказывать. Уточню: не хотел бы рассказывать в настоящее время. Я работаю в школе-интернате для детей-сирот и детей, оставшихся без попечения родителей. В этом году на уроке информатики даю логические задачи, которые решаются табличным методом. В 6-м классе решают почти все. Одна из учениц решает даже устно, без таблиц — методом рассуждений. Почти такие же задачи в 7-м классе решает только половина учащихся, а в 9-м классе — только одна ученица! Большинство наших детей из неблагополучных семей,

но здесь дело даже не в плохой наследственности. Просто 6-й класс еще не вступил в пубертатный период и там никто не курит. Логика и недостаток снабжения мозга кислородом несовместимы.

Надо ли сейчас вообще учить предмету “информатика” в школе? Ведь навыки работы на компьютерах, в сетях, с прикладным ПО дети получают и без этого и гораздо раньше, чем начинают изучать предмет. А специалистов вполне можно готовить в вузах. В школе-то зачем?

У меня на этот счет тоже есть большие сомнения. Отвечу так: “Пока да, но продлится это недолго”. В свое время была мода на трактористов, водителей, инженеров, юристов, экономистов. Ушел из школы предмет “черчение”, почти не стало астрономии. Помню, какой бурный интерес проявляли дети к информатике во времена появления в школе первых компьютеров. Сейчас и за самым навороченным не очень-то удержишь. Все уткнулись в смартфоны. Пройдет мода и на них. В домашней обстановке и обычной деятельности большинство ни текстовым редактором, ни электронными таблицами, ни редактором презентаций не пользуются. К тому же эти приложения стали заметно доступнее для самостоятельного освоения. Про программирование, наверное, не стоит и говорить. Сейчас водителей больше, чем информатиков, но правила дорожного движения и вождение в школе не изучают. Компьютер — это прежде всего инструмент, и относиться к нему надо как к инструменту.

Какой Вы представляете себе информатику в школе через пять лет? А через десять лет?

Фактически это продолжение предыдущего вопроса, и частично я на него уже ответил. К тому же, как показывает мой опыт (а черчение и астрономию мне преподавать доводилось тоже), один час в неделю — это пустая трата времени. За это время в детской голове почти все выветривается, и можно начинать снова. То же можно сказать и об информатике и ИКТ в 3–8-х классах. Мне удавалось уговорить администрацию школы, чтобы ту же астрономию вести по два часа в неделю одно полугодие. Эффективность была заметно выше.

Надеюсь, что это все же будет осознанно. Полагаю, что информатика и ИКТ в старшей школе останется только в профильных классах. Появится интенсивный, может быть только на полугодие, курс информационных технологий в 5-м или 6-м классе. Что касается чистой информатики, то она в основном “растворится” в математике и частично в физике. Правда, есть еще один момент. Мне кажется, что в школе может появиться еще один модуль. Я бы его назвал так: “Основы формирования знаний”. Его также следовало бы определить или в начальную школу, или в начало среднего звена. Но тут я не вполне уверен.

Объясню свою позицию. К понятию системы, точнее, к осознанию этого понятия, наполнению его смыслом, я пришел самостоятельно где-то лет в тридцать, создавая опорные конспекты по своим предметам, заметив, как по-разному на уроках технического труда разбирают и собирают штепсельную вилку девочки и мальчики, как пытается найти пару в куче списанной обуви сторожика тетя Паша. Что такое связь, структура, порядок, иерархия и каково их значение в формировании знаний, почему нам одно понятно, а другое непонятно, я уяснил для себя еще до знакомства с термином “информатика”. Максимум, что мне удалось тогда найти на эту тему в доступных мне библиотеках, — коротенькая статья в “Большой советской энциклопедии”.

С тех пор школьные учебники заметно изменились в плане организации. Теперь и в учебниках информатики начальной школы объясняются понятия: объект, его состав, свойства, действия, отношения. Возможно, что их следовало бы вводить даже раньше, чем существительное, прилагательное, глагол и т.д. Жаль только, что все это дается в контексте предмета информатики и в отрыве от обучения рациональным приемам усвоения знаний и их организации. Чему мы только ни учим, а вот с



Логотип



Эмблема фестиваля



Они создавали школьный сайт (фото из архива автора)

тем, чтобы “учить учиться”, дела обстоят не лучшим образом. Похоже, что это пока еще не выросло в систему, но надеюсь, что вырастет. А сомнения мои оттого, что до меня все это дошло только в тридцать лет. Может, для детского организма, не перегруженного багажом знаний, эта проблема не является существенной. А ведь информация — это то, что помогает нам принимать правильные решения, то, что определяет наше выживание, то есть то, что является существенным (важным для нашего существования). По крайней мере особых успехов в своих попытках внедрить все это в головы учеников начальной школы я пока не достиг. Но надеюсь, что зерно все же заронил.

Какой вопрос Вы хотели бы задать себе, чтобы на него ответить?

“Какое событие в Вашей жизни как учителя информатики запомнилось Вам больше всего?”

В 2003 году Воронежским центром Федерации интернет-образования и Воронежским ИПКиПРО по инициативе и при непосредственном руководстве Бачуриной Л.А. проводился первый межрегиональный дистанционный молодежный фестиваль “Интернет и Мы”.

Участие в конкурсе было добровольным. Заявки принимались от команд школ, лицеев, гимназий и колледжей не только Воронежской области, но и других регионов. В состав команд разрешалось включать даже выпускников школ — студентов 1–2-го курсов. Возглавлял команду учитель информатики. На первом (дистанционном) этапе количество членов команды могло достигать до 20 человек.

Дерзнул и я со своими продвинутыми любителями времяпрепровождения за монитором связаться в эту авантюру. От школ-интернатов наша команда была единственной и в общем обширном списке участников смотрелась белой вороной. В то время у нас в школе не было доступа в Интернет. Чтобы получать материалы, следить за ситуацией на сайте Центра интернет-образования и отправлять или размещать выполненные задания, приходилось бегать в почтовое отделение. Денег на оплату работы в Интернете не было. Приходилось искать спонсоров.

Над первым заданием: “создать школьный сайт” — мы работали все весенние каникулы двумя бригадами. Определили фронт работ, распределили ответственных. Сам сайт размещался на учительском компьютере в папке общего доступа и в перерывах копировался в резерв. Учащиеся, сменяя друг друга, работали по сети каждый над своими страницами. Набирали, редактировали и форматировали тексты, обрабатывали и вставляли графику. Сеть на коаксиале — 10 Мбит/с, оперативная память — 32 Мб. Photoshop еле шевелился. Работа была изнурительной. Сайт занял почти две дискеты. Размещать было негде. Пересылать по электронной почте при скоростях того времени — невысказано.



Первая страница “визитки”



Эмблема нашей команды



Финалисты фестиваля
“Интернет и Мы” 2005 года
(фото из архива автора)

шлось везти электричкой. Узнав, “откуда дровишки”, нашу поделку снисходительно скопировали в папку фестиваля.

Настало время подведения итогов первого конкурса. Выходим на сайт штаба фестиваля и глазам своим не верим: наша работа заняла 1-е место! Это так вдохновило наших ребят, что для поисков ответов на викторину — второй дистанционный конкурс, были брошены все мыслимые и немыслимые ресурсы. Словом, этот конкурс мы тоже выиграли. Финал был очным. В Воронеж были приглашены только 20 команд в ограниченном составе. К первому конкурсу “Визитка команды” мы подготовили и достойно представили ироничную презентацию в виде сайта. Наша визитка оказалась лучшей:

http://www.rosinka.vrn.ru/rosin_wz/index.htm.

Во втором конкурсе “Поиск в Интернете” мы на успех не рассчитывали. Мои финалисты, да и то благодаря спонсорам, имели опыт работы с Internet Explorer и Rambler в пределах 40 минут каждый. Зато в конкурсе на анимирование по выбору одной из десятка предложенных знаменитых картин отечественных художников мы надеялись на успех. Решили остановиться на “Чаепитии в Мытищах” Перова. Мои ребята уверенно владели графическим редактором Photoshop и неплохо работали в Macromedia Flash.

Но тут фортуна от нас отвернулась. Видимо, от волнений и предыдущей бессонной ночи на меня накатил такая мигрень, что у меня стало темно в глазах. По условиям конкурса руководитель имел право оказывать устную помощь своей команде, но я был не в состоянии поддержать своих воспитанников. Тем не менее ребята сами придумали сюжет для анимации и неплохо справились с заданием.

Готовую работу следовало по сети переслать в папку на головной преподавательский компьютер. И тут оказалось, что доступ с нашего рабочего компьютера на головной отсутствует. Конкурс завершился. Нас торопили на экскурсию. Пришлось записать нашу анимацию на дискету, закрыть Photoshop и перенести работу на проверку по назначению. Открыв файл с дискеты, мы вдруг обнаружили вместо анимации... черный прямоугольник!

То ли Photoshop глючил, то ли с дисководом были проблемы, но работу мы потеряли.

В гостиницу мои ребята вернулись в удрученном состоянии, а у меня за время экскурсии прошла мигрень. На следующий день все команды должны были представлять и защищать свои анимации при подведении итогов фестиваля в театре юного зрителя. А мы так успешно лидировали и так неожиданно оконфузились в последнем конкурсе.

Из своего житейского опыта я знаю, что в таких случаях надо не опускать руки и бороться до конца. И вот тут нам взбрело в голову обыграть случившееся под известный шедевр Малевича. Мы с энтузиазмом принялись сочинять легенду.

В списке защитников анимаций художественных полотен наша команда по понятным причинам не значилась, но мне удалось уговорить организаторов фестиваля, чтобы нам также предоставили слово. В составе нашей команды была бойкая и эмоциональная ученица 10-го класса Агibalова Таня. Она сумела с таким юмором преподнести нашу “трагедию”, да и содержание сочиненного нами сюжета тому способствовало, что весь зал встал и восторженно аплодировал. Это был самый памятный момент.

Жюри подвело итоги фестиваля накануне. Призы и дипломы были распределены и подписаны. Наша последняя попытка прорваться в призеры уже не шла в зачет. Мы вошли только в десятку сильнейших команд. Но одну награду мы все же получили и пользуемся ею до сих пор. Это бесплатный хостинг сайта нашей школы на сервере компании “Информсвязь-Черноземье”.

На следующем фестивале “Интернет и Мы” в 2005 году мы снова лидировали в трех дистанционных конкурсах. Наши работы можно посмотреть в Интернете:

Представление команды — <http://www.rosinka.vrn.ru/komanda.htm>.

Никола Тесла (конкурс “Физика”) — <http://www.rosinka.vrn.ru/tesla/index.htm>.

“Феномен пси” (конкурс “Сотрудничество”) — <http://www.rosinka.vrn.ru/psi/index.htm>.

Не без казуса прошли в финал. В конкурсе баннеров от нашей команды участвовал шестиклассник. Он сделал отличный баннер в Macromedia Flash, но в слове “фестиваль” допустил ошибку. Конкурс оценивался по числу набранных голосов в баннерной сети. Естественно, что работа нашего ученика не могла быть выставлена на голосование в пространство Интернета. Тем не менее баллов, набранных в других конкурсах, нам хватило для того, чтобы попасть в число призеров и занять второе место.



Вернулись с победой (фото из архива автора)



Леонов Александр Георгиевич

Кандидат физико-математических наук, доцент, руководитель авторского коллектива ИнфоМир (программное обеспечение КуМир, ПиктоМир, учебники и учебные пособия “Информатика и ИКТ” для школы и вузов), ведущий научный редактор и автор статей тома “Информатика” издательства “Аванта+”. Выпускник механико-математического факультета МГУ (1983), аспирантуры мехмата МГУ (1986), ведущий научный сотрудник кафедры “Теоретическая информатика” мехмата МГУ, заведующий секторами Научно-исследовательского института системных исследований РАН, член Союза журналистов России.

О себе: Люблю зиму и лето, горные лыжи и плавание, спорт и туризм. Несмотря на природную осторожность, готов побороться за жизнь. Считаю, что самое ценное — это время.

Как Вы пришли в информатику? Первый компьютер, на котором Вы работали? Первый язык программирования, с которым Вы познакомились? Первые задачи, которые Вы решали?

Я — счастливый человек!

Каждому хочется стать знаменитым. Еще больше хочется, чтобы о тебе узнали. Но математики — не публичные люди. Страшно далеки они от народа. Если вы бизнесмен, то можно написать книгу воспоминаний (или заказную статью), чтобы потом показывать ее подчиненным и слушать их неприкрытую лесть. Тяжелее заставить прочитать о себе простого, не зависящего от тебя читателя. В 2004 году (по-моему) издательство “Аванта+” выпускало серию книг “Профессия”. Мне посчастливилось быть ведущим научным редактором на томе “Информатика” — то есть профессия IT. В томе должна быть статья про профессию программиста. Написание центральной статьи тома я взял на себя. Я понимал, что статья просто обязана быть интересной.

Нужно было описать профессию программиста так, чтобы вчерашние школьники захотели стать именно программистами. И тут меня осенило! Я напишу про себя, почему я люблю программирование, про своих кумиров, про огромный мир компьютеров и программ и свою жизнь в нем.

Пользуясь “служебным положением”, я и название статьи выбрал очень звонкое: “Я — программист”. (Помнящий да отметит, что книга Норберта Виннера называлась “Я — математик”.)

Не каждому выпадает шанс так широко рассказать про свой жизненный путь, и уж точно не каждый может похвастаться, что его мини-биографию читают добровольно. Но я — счастливый человек, итак:

Я ВСЕГДА хотел стать программистом. Потому что Робот, то, чем можно управлять, учить, программировать, — наверное, самая замечательная игрушка на Земле (для детей и взрослых).

Робота мне подарили родители на Новый год (лет мне тогда было между 8 и 9). Конечно, этот робот ничего не умел. Это была просто заводная игрушка, которая ходила вперед, смешно двигая ногами. Когда завод заканчивался — робот останавливался.

Этому роботу было далеко до тех роботов-станков, которые я увидел спустя несколько лет на выставке в Москве. Это были умные роботы. Они вытаскивали детали, проводили измерения, без усталости перемещали грузы с одной ленты-транспортера на другую и делали еще кучу нужных вещей на производстве. Не все они были похожи на человекоподобных роботов, но

все они работали по Программе. А программы для них составляли люди. И люди использовали компьютеры.

Но и компьютера у меня не было, и мечтать о нем я не мог. Но у меня было много книг. В книге “Математические досуги” Мартина Гарднера была схема “компьютера” для игры в “шашки” на поле 3×3 . Самообучающаяся машина собиралась из 30 спичечных коробков и обычных пуговиц. Я так был очарован возможностью собрать собственный “компьютер”, что выходные посвятил этому занятию. Потом я усовершенствовал “методы обучения”, просто найдя все выигрышные ходы, и после этого мой компьютер не оставил мне шанса на победу. Я занялся сборкой “компьютера” для крестиков-ноликов, на который пришлось потратить все сбережения — 3 рубля и завалить дом спичками из выпотрошенных коробков. Но машина была построена.

Естественно, в будущем первой программой для ЭВМ была программа на Алголе-60 для игры в “крестики-нолики”. Вероятно, все великие люди начинают с этой программы, недаром первой программой Билла Гейтса тоже были “крестики-нолики”.

Первые пять фамилий из мира информатики, которые приходят Вам в голову? Если можете, пожалуйста, кратко расскажите, почему именно они.

Итак, только пять фамилий:

Чарльз Бэббидж, Ада Лавлейс — это люди, которые продолжали идти к цели, зная, что она недостижима. Реальность не позволила Ч.Бэббиджу построить работающий компьютер, но оба этих ученых не сдавались всю свою жизнь. Первые программы первого программиста — дочери великого Байрона меня искренне впечатляют.

Джон Атанасофф — за оригинальное решение и оригинальный метод, который привел к этому решению. Как говорят в таких случаях, Атанасофф “ходил по кругу”. Ему совсем немного не хватало для решения технической проблемы. Он фактически ничего не видел и ничего не замечал несколько дней, он пытался решить проблему. И вот, внезапный нервный срыв, пустое шоссе, машина, скорость, дорога “в никуда”, остановка, кафе, “бурбон”... и мировое признание за изобретение ЭВМ.

Билл Гейтс — за целеустремленность и хорошую игру в карты.

Я действительно восхищаюсь Гейтсом: я бы не смог сказать родной матери, что не буду ей звонить полгода. И еще, когда открываете новое дело, вам не всегда потребуется первоначальный капитал. Достаточно, как Б.Гейтс, иметь друга с деньгами и хорошо сыграть с ним в карты.

Стив Джобс — о нем сейчас сказано слишком много, жаль, что большинство неправда. Мне абсолютно не нравятся фильмы и книги про Джобса. Абсолютно все. Даже если бы он был Махатма Ганди, я уверен, что у него нашлось бы столько грехов... Все это пыль. Стив Джобс просто великий человек, проживший яркую жизнь, человек с большой буквы, навсегда вошедший в историю ИТ и человечества.

Какая компьютерная техника и IT-сервисы окружают Вас сейчас? Насколько Вы “компьютерный человек” в обычной жизни? Это вопрос про все — про компьютеры и гаджеты, которые Вас окружают, про Ваш личный стиль социальной коммуникации в сети — насколько Вы живете в социальных сетях, насколько не можете жить без электронной почты и других средств коммуникации. Насколько бережете или не бережете свое личное пространство и время.

Время меняется, и сейчас компьютером никого не удивишь. Вокруг меня много компьютерной техники, наверное, слишком много. Я даже затрудняюсь подсчитать, какое количество электронных и компьютерных устройств я активно использую в жизни. Я просто передвигаюсь между ними, держа “на борту” с десятков легких устройств: 2 смартфона, 2–3 планшетника, 3–4 ноута... Остальные стоят на столах. Без всего этого просто нельзя работать, а так как работа — это моя жизнь, то и жить.

К счастью, пока я сильно загружен различными интересными проектами, поэтому времени находиться во дворе “на лавочке” у меня просто нет. Со-

циальные сети — это наши электронные “лавочки” для наших молодых “бабушек”. Жаль, что молодое поколение так быстро старится. Очень хочется верить, что работа не позволит мне никогда сесть на лавочку. Очень-очень... Пока у меня 72 часа в сутках. Это тяжело и чертовски здорово.

Если бы в Вашем распоряжении была лишь одна лекция (допустим — “пара”, полтора часа) и полная свобода рассказывать о том, что Вам интересно в информатике, чему бы Вы посвятили это время?

Я пока еще преподаю на мехмате МГУ. Чем это хорошо: тут для преподавателей всегда была свобода. Правда, последнее время она вылилась в свободу от занятий и студентов. Будучи студентами мехмата мы иногда прогуливали пары. У нас была шутка про преподавателя, который “прогулял” занятие. Это было смешно, потому что никто не мог такое даже представить. Теперь на мехмате это случается часто.

Если лекция или семинар абсолютно устарели по материалу, то мы говорим, что это фундаментальное изложение. Если содержание лекции — это просто бред, то это инновационный курс и методика. Так что в МГУ и сейчас полнейшая свобода.

Свою дополнительную лекцию я бы посвятил рассказу про “программирование” самого себя, про умение ставить себе задачу, искать решение, идти к цели. Про умение работать, про желание учиться.

Есть ли какой-то факт из информатики — может быть, какой-то алгоритм, — который произвел на Вас сильное, запоминающееся впечатление?

Вы не поверите, Алгоритм Евклида.

Есть ли у Вас любимая задачка или задачки?

Последнее время мы со студентами любим решать и программировать следующую задачу для младших школьников:

Можно ли найти десятизначное число, в котором:

первая цифра равна количеству единиц в записи этого числа,

вторая цифра равна количеству двоек записи этого числа,

третья цифра равна количеству троек в записи этого числа — и так далее до последней цифры, которая равна количеству нулей записи этого числа.

Примечательно, что программа с простым перебором легко решает эту задачу ☺.

Какую тему/темы школьного курса (будем считать, что мы находимся в стране ФГОС) Вы не хотели бы рассказывать детям? Ну не нравится она Вам.

Все, которые не имеют отношение к алгоритмам и программированию.

Надо ли сейчас вообще учить предмету “информатика” в школе? Ведь навыки работы на компьютерах, в сетях, с прикладным ПО дети получают и без этого и гораздо раньше, чем начинают изучать предмет. А специалистов вполне можно готовить в вузах. В школе-то зачем?

О, да! Если школьнику не привить алгоритмическое мышление, то в вузе это будет уже поздно. Абсолютно. Сейчас уже до 10 первокурсников мехмата в группе из 30 человек не могут записать правильно ни на каком языке программирования (включая русский) алгоритм нахождения числа разных из трех данных чисел. А это уже потерянные люди. Остается удивляться, что они еще умеют находить корни квадратного уравнения.

Какой Вы представляете себе информатику в школе через пять лет? А через десять лет?

Хочу верить, что через пять лет информатика займет в школьном курсе больше часов, чем сейчас, а через десять лет — еще больше.

Какой вопрос Вы хотели бы задать себе, чтобы на него ответить?

Если бы я вдруг стал министром образования, смог бы я сделать наше российское образование СУЩЕСТВЕННО лучше?

Отвечаю — смог бы, АБСОЛЮТНО уверен.



Интерактивный
короткофокусный проектор

MimioProjector

Легкий и доступный способ
внедрения интерактивного обучения
в классах, не оборудованных проектором



Кто сказал, что нельзя получить все сразу? Установите MimioProjector, подключите к компьютеру и оживите ваши уроки, используя все возможности интерактивного обучения на маркерной доске.

В вашем классе уже есть маркерная доска, но нет проектора. А еще вы мечтаете об интерактивной доске, но на все это не хватает средств? Наш новый интерактивный **MimioProjector** даст вам возможность внедрить интерактивное обучение, избежав чрезмерных расходов, — ведь он совмещает в себе функции отличного короткофокусного проектора и полноценной интерактивной доски!

Благодаря функции использования двух интерактивных ручек сразу двое учащихся могут одновременно выполнять манипуляции с объектами на доске, используя все преимущества группового обучения. А ультракраткофокусная модель проектора позволяет снизить количество теней на экране. Проектор легко подключается к компьютеру и позволяет начать использовать имеющуюся у вас маркерную доску или даже светлую стену вашего класса в качестве интерактивной доски. Поставляемое в комплекте с проектором программное обеспечение **MimioStudio** позволяет вам создавать и проводить увлекательные уроки и управляет всем оборудованием семейства **MimioClassroom**.



Проектор: широкоформатный (16:10) с разрешением 1280 x 800 точек (WXGA). Коэффициент контрастности до 3000:1. Размер экрана от 70 до 100 дюймов.

Интерактивные возможности: рабочая область от 75 до 115 дюймов по диагонали. Одна или две интерактивные ручки в зависимости от комплекта поставки. Проектор можно заказать в комплектации с одной или двумя интерактивными ручками либо как традиционный проектор без интерактивных функций.

Продажа оборудования, консультации и обучение:

<http://www.mimioclass.ru>

8 (800) 5555-33-0

Звонок по России бесплатный

ООО «Рене» — генеральный дистрибьютор Mimio в России



mimio
a better way to learn



Новые старые знакомые

Историческая справка

И.А. Сукин,
г. Переславль-
Залесский

► Наверное, каждый близкий к практическому программированию человек так или иначе слышал о PHP — языке, который можно смело назвать главным связующим звеном современного Интернета. PHP занял это место где-то в начале XXI века, заменив собой множество скриптов, написанных на Perl. История этого языка программирования начинается в 1995 году, когда довольно известный датский Perl-хакер¹ Расмус Лердорф написал на Perl, с использованием модуля CGI, набор скриптов для автоматизации учета посещаемости своей домашней страницы. Кроме всего прочего, эти

¹ Здесь слово “хакер” употреблено в своем первоначальном положительном значении. Оно определяет человека, крайне хорошо (до нарочитой “небрежности” и автоматизма) разбирающегося в своей области.

скрипты служили также для шаблонизации — автоматической генерации HTML-страничек на основе некоторых шаблонов, заполняемых программно. Через некоторое время Расмусу перестало хватать возможностей (в том числе и в плане производительности) интерпретатора Perl, и он написал с нуля реализацию собственного языка программирования. Как и изначальный набор скриптов, этот язык программирования получил название “PHP” (*Personal Home Page*)².

Какое-то время PHP развивался исключительно как собственный проект Лердорфа и держался фактически на одном его энтузиазме. В 1997 году два израильских программиста, Энди Гутманс и Зеев Сураски, серьезно заинтересовались языком и полностью переписали его интерпретатор и базовую систему. Это была уже третья версия PHP, в которую вошли инструменты, наличие которых позволило говорить о серьезном практическом применении некогда простого генератора домашних страниц. Главным нововведением стало появление системы модулей для расширения ядра интерпрета-

² С дополнением FI (*Forms Interpreter*).

тора, что привлекло к языку множество сторонних разработчиков. PHP очень быстро обрел поддержку всевозможных API, протоколов, систем управления базами данных, что позволило ему сделать первый шаг в мир профессионального программирования. С этого момента название “PHP” стало рекурсивным бэкронимом от “PHP Hypertext Processor”.

Четвертая версия языка принесла в основном улучшения в производительности и безопасности, но не повлияла слишком сильно на какие-либо базовые концепции. Реальную практическую силу PHP получил с появлением пятой версии интерпретатора, в которой оформилась полноценная поддержка объектно ориентированного программирования, включая пространства имен и интерфейсы, поддержка некоторых “вкусностей” функционального программирования, например, замыканий, XML и Unicode.

Шестая версия PHP, как и шестая версия Perl³, должна была избавить язык от множества устаревших возможностей и добавить новые, при этом в качестве основного улучшения позиционировался полный переход на Unicode внутри интерпретатора. К сожалению, такая идея стала фатальной для PHP6 ввиду некоторых принципиальных сложностей обработки Unicode, и основное внимание разработчиков окончательно сфокусировалось на пятой ветке, о которой и пойдет речь дальше в этой статье.

Стандартный инструмент

Со временем популярность PHP в среде веб-программистов возросла настолько, что язык стал стандартом de facto в современной динамической Всемирной паутине. Названия основных инструментов для создания веб-сайтов обычно сокращают до акронима LAMP — Linux, Apache, MySQL, PHP (ОС, веб-сервер, СУБД, серверный язык программирования). Грубо можно сказать, что каждый из этих инструментов “затачивается” под другие, и именно такая связка является наиболее удобной для применения на практике. У читателей основное затруднение может вызвать необходимость наличия Linux, однако с некоторыми неудобствами работать с PHP можно и под ОС Windows (WAMP, XAMPP).

Пара слов о веб-программировании

Некоторые читатели могут быть не искушены в вопросах разработки веб-приложений, поэтому необходимо внести некоторые разъяснения. Когда клиент через браузер на своем компьютере⁴ обращается к некоторому удаленному серверу с просьбой предоставить определенную веб-страницу, сервер должен прислать какой-то ответ. Оговорим, что нас пока будут интересовать только те случаи, когда веб-сервер не считает запрос ошибочным. В этом случае у него будет два варианта — вернуть

какой-либо HTML-файл, лежащий в заранее заданном каталоге, “как есть”, ничего в нем не меняя, или же вызвать некоторую программу (или подпрограмму), которая каким-нибудь образом сгенерирует требуемую страницу. Первый вариант использовался на заре появления Интернета (и сейчас используется для несложных домашних страниц) и, несмотря на очевидную простоту и легкость использования, обладает множеством недостатков. Самым главным из этих недостатков является невозможность тонкой персонализации возвращаемой страницы для каждого посетителя, что было бы крайне удобным при проектировании и разработке интернет-магазина и совершенно, принципиально важным при написании форума, гостевой книги, социальной сети.

Здесь на первый план выходят языки, подобные PHP, серверные языки, или, иначе говоря, языки “бэкенда”. Весь код, написанный на них, выполняется на стороне сервера, и клиенту возвращается только текст, представляющий собой сгенерированную страницу. При этом выделяют также клиентские языки, языки “фронтенда”, такие, как JavaScript⁵ и набирающие популярность CoffeeScript, ClojureScript, Dart, и другие. Подпрограммы на этих языках встроены в тело веб-страницы и выполняются в браузере клиента, на его компьютере. Эти языки позволяют повысить интерактивность страницы, добавить различные мультимедийные, графические, игровые возможности. Тем не менее полноценный разговор о PHP возможен без какого-либо упоминания о программировании для фронтенда, поэтому мы больше не будем говорить об этих языках.

Начинаем работу

В предположении, что большинство читателей этой статьи используют Windows, мы будем делать основной акцент на набор инструментов XAMPP⁶. В этот набор входят веб-сервер Apache, система управления базами данных MySQL, интерпретатор PHP, интерпретатор Perl, почтовый и FTP-сервера, утилита для визуального управления базами данных phpMyAdmin (написанная, как следует из названия, на PHP). После установки пакета потребуется получить права администратора (или установить все в нестандартную директорию) и разблокировать каждую программу в брандмауэре Windows. Все файлы, содержащие код на PHP, мы будем располагать в подкаталоге htdocs каталога, в который установлен веб-сервер.

Все файлы с исходным кодом будут иметь расширение *.php. Поскольку корни PHP растут из шаблонизатора для HTML, скрипты на этом языке чем-то похожи на HTML-разметку, а точнее, включаются в нее. Код на PHP должен быть заключен между тегов `<?php` и `?>`. В простейшем случае

³ Не так радикально, конечно.

⁴ Это не совсем верная, но хорошая аппроксимация.

⁵ С появлением node.js возникла возможность написания серверной части веб-приложения на JavaScript.

⁶ Который замечательно работает и в других ОС.

страница может не иметь никаких заголовков и дополнительных тегов:

```
<?php
    echo "Hello world";
?>
```

В качестве открывающего тега может быть использован короткий тег `<?`, однако это нежелательно, поскольку слегка ухудшает читабельность сложного кода при совершенно незначительном выигрыше. В случае если файл содержит исключительно код на PHP, некоторые⁷ источники рекомендуют опускать закрывающий тег, но подобная практика не является хорошим тоном программирования⁸, и на протяжении статьи закрывающий тег опускаться не будет. Здесь стоит также отметить такие необязательные, но влияющие на хороший стиль элементы, как возможность опустить точку с запятой после последнего PHP-выражения и автоматическое удаление перевода строки, следующего сразу за закрывающим тегом (все остальные переводы строки будут включены в результирующий файл “как есть”).

PHP и HTML

Как уже говорилось ранее, код на PHP может быть встроен в HTML-верстку страницы. Это, в частности, позволяет использовать интерпретатор этого языка в качестве простейшего шаблонизатора. К примеру⁹:

```
<html>
<head>
<meta charset="windows-1251">
</head>
<body>
<?php if (4 > 5): ?>
    Что-то пошло не так в Вашей Вселенной.
<?php else: ?>
    Есть шанс, что мы с Вами находимся
    в одной и той же Вселенной.
<?php endif; ?>
</body>
</html>
```

Существуют также другие вариации тегов, позволяющих включить код на PHP в верстку страницы, мы рассмотрим лишь один более-менее интересный короткий тег `<?=>`, являющийся сокращением для `<?php echo something; ?>` и служащий для вывода значения любого PHP-выражения:

```
<?="test" ?>
эквивалентно
<?php echo "test"; ?>
```

Лексический анализатор PHP довольно легкомысленно подходит ко всему, что не включено в соответствующие теги `<?php` и `?>`. Это может быть нам на руку, поскольку мы можем, к примеру, встроить код на PHP прямо в тело HTML-тега¹⁰:

⁷ Например, официальная документация.

⁸ Хотя довольно уважительные причины для подобного опускания иногда имеются.

⁹ В этом примере что-то может пойти не так, конечно, в нашей Вселенной, ведь код выполняется на нашем сервере, а не на компьютере пользователя.

¹⁰ Что означает `$condition`, мы рассмотрим чуть позже.

```
<html>
<body>
<p<?php if ($condition): ?>
    class="ourownclass"<?php endif; ?>>
    Some text in a paragraph.
</p>
</body>
</html>
```

В вышеприведенном примере интерпретатор проверяет некоторое условие (заданное в виде переменной `$condition`) и в случае успеха устанавливает атрибут `class` некоторого абзаца равным `"ourownclass"`. То есть в результирующий HTML попадет строка

```
<p class="ourownclass">
```

В случае если условие окажется ложным, будет сгенерирован обычный тег `<p>`.

Напоследок в этом разделе стоит сказать, что комментарии в PHP-коде сходны с комментариями языка C (последних версий стандарта): `//` обозначает однострочный комментарий, `/* ... */` — многострочный, при этом многострочные комментарии не могут быть вложенными. Также допустимы комментарии в стиле Perl и Python, начинающиеся с символа `#`. Здесь нужно обратить особое внимание на практику “закомментированных” нежелательных строчек кода. Лексический анализатор в PHP устроен таким образом, что закрывающий тег `?>` имеет для него настолько большой приоритет, что он действует даже внутри комментария. Это может привести к нежелательным последствиям:

```
<?php
    //echo "some text\n"; ?>
    echo "new text\n"; ?>
```

Вместо ожидаемого `"new text\n"` этот пример выведет строку

```
echo "new text\n"; ?>
```

Такое поведение, вне всякого сомнения, нужно учитывать при написании и отладке собственного кода на PHP.

Основы синтаксиса

Переменные

С некоторыми элементами синтаксиса собственно PHP, а не окружающего его HTML мы уже познакомились чуть выше. Для людей, не знакомых с такими языками, как Perl и Tcl, может быть непривычным то, что имена переменных должны начинаться с сигила `$`. Сторонники такого подхода утверждают, что он помогает отличить имена переменных от имен функций и других сущностей, и с ними сложно не согласиться. Людям, знакомым с Perl, синтаксис работы с переменными в PHP покажется привычным и простым:

```
<?php
    $name = "Ivan";
    echo "My name is $name\n";
?>
```

Имя переменной (идентификатор после сигила `$`) может начинаться с буквы или символа подчеркивания, но не с цифры, и иметь произвольную

длину. Регистр символов учитывается. В коротком примере выше мы также использовали возможность “интерполяции” переменных в строки, позволяющую встраивать значение некоторой переменной непосредственно в строку. Как и в Perl, для ограничения имени интерполируемой переменной в сложных случаях можно использовать фигурные скобки:

```
<?php
    $fruit = "apple";
    echo "We had 8 ${fruit}s for dinner\n";
?>
```

или операцию конкатенации

```
<?php
    $fruit = "apple";
    echo "We had 8 " . $fruit . "s
    for dinner\n";
?>
```

Стоит также упомянуть, что для проверки существования какой-либо переменной можно использовать стандартную функцию `isset`:

```
<?php
    echo isset($test);
?>
```

В данном примере, если переменная `$test` не была инициализирована, не будет выведено ничего, в противном случае будет выведена единица.

Типы данных

Набор поддерживаемых типов данных и расширяемость данного набора являются одними из самых важных характеристик любого языка программирования. PHP поддерживает восемь основных типов данных. К ним относятся четыре скалярных: логический (`bool`), целочисленный (`int`), вещественный (`float`), строковый (`string`); два сложных, смешанных типа: массив (`array`) и объект (`object`); два специальных типа: ресурс (`resource`, некоторый блок “внешних” данных) и `NULL`. Также на уровне языка существует понятие о некоторых директивах, представляющих собой не отдельные типы, а некоторые особые множества типов. Такие директивы называются псевдотипами. Например, псевдотип `mixed` говорит о том, что переменная (аргумент функции) может принимать значения самых разных типов. Также к числу псевдотипов относятся понятия “числа” (в широком смысле) и функции, которая будет передана в качестве фактического параметра. Для получения типа любого выражения используется функция `gettype`:

```
<?php
    $fruit = "apple";
    echo gettype($fruit);
?>
```

В этом примере она выведет в тело HTML-страницы `string`. Для проверки, принадлежит ли значение выражения к определенному типу, используются функции `is_имя_типа`: `is_int`, `is_string`, `is_bool` и т.д. Для явного изменения¹¹ типа переменной можно использовать функцию `settype`, однако на практике в этом редко возникает необходимость, поскольку PHP, как и Perl, использует

¹¹ Однако это не то же самое, что и явное приведение типа.

довольно гибкие правила автоматического преобразования типов переменных. За более подробной информацией по этому вопросу можно обратиться к официальной документации.

Подробнее о переменных

В заключение разговора о переменных нужно рассмотреть несколько особых случаев.

Переменные в PHP, как и в других языках, имеют область видимости. Область видимости может быть глобальной — переменные будут доступны во всем скрипте (не внутри функций или объектов), или локальной — переменные будут доступны только в некотором участке кода или данных (операторе ветвления, цикла, функции, объекта, см. дальше). Локальные переменные обычно перекрывают глобальные переменные (или локальные переменные верхних уровней вложенности), имеющие то же самое имя. Кроме того, существует девять предопределенных переменных, называемых суперглобальными. Эти переменные доступны из совершенно любого места в скрипте, в том числе внутри объектов. Все эти переменные являются массивами: `$_GLOBALS` — массив всех глобальных переменных, `$_GET` — переменные GET-запроса, `$_POST` — переменные POST-запроса, `$_COOKIE` — переменные, переданные в куках, `$_SERVER` — переменные, установленные сервером, и другие. К примеру, можно получить `user-agent` клиента, просматривающего страницу:

```
<?php
    echo $_SERVER["HTTP_USER_AGENT"];
?>
```

Здесь к переменной-массиву `$_SERVER` применяется операция индексации `[]`. Индексами массивов в PHP могут быть произвольные строковые или числовые значения, например, в примере использовалась строка `"HTTP_USER_AGENT"`.

Оператор присваивания в PHP по умолчанию выполняет копирование значения выражения, стоящего справа от него, в некоторую переменную. Однако в ряде случаев программист может захотеть создать не копию, а некое подобие мягкой привязки, такой, чтобы при изменении значения одной переменной изменялось и значение другой. Такая привязка называется “ссылкой” и создается с помощью символа “&”:

```
<?php
    $a = 3;
    $b = &$a;
    $a++;
    echo "a = $a, b = $b\n";
?>
```

В некоторых случаях программист может захотеть объявить константу, объект, значение которого не должно изменяться. В PHP это можно сделать с помощью функции `define`:

```
<?php
    define("CURRENT_TIME", time());
    echo CURRENT_TIME;
?>
```

В приведенном примере в константе `CURRENT_TIME` будет храниться значение времени, получен-

ное именно в момент объявления константы (вызова функции `define`). Здесь стоит обратить внимание, что в отличие от переменных константы не начинаются с сигила “\$”.

Основные операторы PHP

Синтаксис основных управляющих конструкций в PHP заимствован из таких широко распространенных языков, как Perl и C, и не должен вызвать особых затруднений в освоении. Чтобы не увеличивать объем теоретической информации, я не буду подробно останавливаться на особенностях интерпретации каждого отдельного оператора, а лишь продемонстрирую, как работают базовые идеи, на примерах (на случай, если читатель незнаком с C или Perl).

Оператор ветвления¹²:

```
<?php
$flat = 23;
if ($flat > 20 && $flat < 30) {
    echo "Вы проживаете в квартире, которую
мы обслуживаем\n";
} else {
    echo "Извините, Ваша квартира обслужи-
вается не нами\n";
}
?>
```

Многовариантный оператор ветвления:

```
<?php
if ($flat < 10) {
    echo "Вас обслуживает компания №1\n";
} elseif ($flat < 20) {
    echo "Вас обслуживает компания №2\n";
} else {
    echo "Вас обслуживает неизвестная нам
компания\n";
}
?>
```

Оператор выбора¹³:

```
<?php
switch ($flat) {
    case 4: echo "С Вами работает
Иванов А.И.\n"; break;
    case 7: echo "С Вами работает
Петров Б.О.\n"; break;
    default: echo "Уточните фамилию
Вашего оператора по телефону\n";
}
?>
```

PHP поддерживает два общепринятых оператора организации циклов: цикл с предусловием (`while`) и цикл со счетчиком (`for`). Цикл с предусловием:

```
<?php
$age = 0;
while ($age < 18) {
    echo "Ваш возраст: $age\n";
    echo "Вы пока еще не достигли
совершеннолетия\n";
}
```

¹² В примерах я буду использовать символ перевода строки “\n”, но он скорее всего будет отображаться в виде пробела на HTML-странице. Чтобы вставить реальный перевод строки, нужно использовать тег `
`.

¹³ Для тех, кто незнаком с C: если убрать `break` в конце каждого варианта, то интерпретатор “провалится” и выполнит операции для всех вариантов, следующих за верным. Иногда такое поведение может быть желательным.

```
$age++;
}
?>
```

Для тех, кто не программировал на C: оператор “++” — это оператор инкремента, прибавляющий единицу к своему аргументу. На подробностях вычисления выражений “++\$a” и “\$a++” мы здесь останавливаться не будем. Аналогичный код можно записать в виде цикла со счетчиком:

```
<?php
for ($age = 0; $age < 18; $age++) {
    echo "Ваш возраст: $age\n";
    echo "Вы пока еще не достигли
совершеннолетия\n";
}
?>
```

Для тех случаев, когда необходимо, чтобы цикл выполнялся хотя бы один раз, можно использовать цикл с постусловием (как `repeat/until` в Паскале):

```
<?php
$age = 20;
do {
    echo "Я пока не проверил, есть ли
Вам 18 лет\n";
} while ($age < 18);
?>
```

Для мгновенного прерывания цикла можно использовать ключевое слово “`break`” (которое имеет необязательный параметр, обозначающий число уровней вложенности, из которых нужно выйти), а для мгновенного перехода к следующей итерации — ключевое слово “`continue`”.

В PHP реализован базовый набор арифметических и логических операторов: +, -, *, /, . (конкатенация строк), ! (логическое отрицание), && (конъюнкция), || (дизъюнкция), ++ (инкремент), -- (декремент), = (присваивание), == (проверка на равенство), <, >, <=, >= (проверка на абсолютное равенство), <=, >= (проверка на абсолютное равенство означает как равенство значений выражений, с учетом приведения типов, так и равенство самих типов. Для каждого из операторов +, -, *, / и . существуют также версии, объединяющие оператор с оператором присваивания. Следующий код, к примеру

```
$a = $a * 2;
может быть записан так:
$a *= 2;
```

Функции

Как и в подавляющем большинстве других практических языков программирования, в PHP существуют как встроенные, так и определяемые пользователем функции. Каждая функция определяется именем и списком формальных аргументов. Обычно для удобства программиста составляется прототип функции, в котором указываются типы как параметров, так и возвращаемого результата, однако это необязательно.

Программист может создать свою функцию с помощью ключевого слова “`function`”:

```
<?php
function double($number) {
    return $number * 2;
}
?>
```

Результат из функции возвращается с помощью оператора “return”, а список параметров указывается в скобках после названия. Каждый параметр в списке, как и любая переменная, имеет сигил “\$” перед своим именем.

При необходимости передать ссылку в качестве параметра перед именем этого параметра в определении ставить символ “&”:

```
<?php
function double(&$number) {
    $number *= 2;
}
?>
```

В этом примере функция double должна принимать в качестве аргумента не число, а переменную, в которой хранится число. Значение, хранящееся в переменной, удваивается и присваивается той же самой переменной.

Иногда (например, если функция возвращает некий элемент массива) возникает необходимость вернуть ссылку. Это можно обеспечить, указав амперсанд перед именем функции. Следующий пример не слишком подходит для практики, но показывает концепцию:

```
<?php
function &double($number) {
    return $number * 2;
}
$v = & double(3);
?>
```

Параметры функции могут иметь значения по умолчанию:

```
<?php
function hello($object = "World") {
    echo "Hello, $object!\n";
}
?>
```

Также количество параметров может быть переменным. Для получения общего числа параметров служит функция func_num_args, для получения параметра по его порядковому номеру — функция func_get_arg:

```
<?php
function hello_worlds($world) {
    for ($i = 0; $i < func_num_args();
        $i++) {
        $arg = func_get_arg($i);
        echo "Hello, $arg!\n";
    }
}
?>
```

Здесь мы определили функцию, имеющую один обязательный параметр (его имя, \$world, нигде явно не используется, но мы обращаемся к нему по номеру) и произвольное число необязательных.

Массивы

Массивы являются важной частью языка PHP и обязательно должны быть упомянуты как можно раньше. Массивы, как уже было отчасти показано, позволяют хранить несколько значений, каждое из которых имеет определенный индекс или ключ. Создается массив с помощью функции array, по умолчанию все ключи в нем числовые и следуют по порядку друг за другом, начиная с нуля:

```
<?php
$fruits = array("apple", "banana",
               "tangerine", "pear");
$size = count($fruits);
?>
```

Функция count возвращает размер массива. Элементы "apple", "banana", "tangerine" и "pear" имеют индексы, соответственно, 0, 1, 2, 3. Массив с более сложными ключами можно создать, воспользовавшись следующим синтаксисом:

```
<?php
$fruits = array("a" => "apple",
               "b" => "banana", "t" => "tangerine");
?>
```

Теперь элементы будут иметь индексы "a", "b" и "t", соответственно. Основной операцией, применимой к массивам, является операция индексации, обозначаемая квадратными скобками:

```
<?php
$fruits = array("a" => "apple",
               "b" => "banana", "t" => "tangerine");
echo $fruits["t"];
?>
```

Этот пример должен вывести строку "tangerine". Пока мы не будем останавливаться на массивах более подробно.

Стандартные функции

В стандартной библиотеке языка PHP содержится большое количество встроенных функций, практически на все случаи, которые могут возникнуть в жизни веб-программиста. В этом разделе мы рассмотрим некоторые из них.

Работа с переменными

Как уже упоминалось ранее, для проверки того, имеет ли переменная какое-либо значение, то есть, была ли она создана, используется функция isset. Для того чтобы специально удалить переменную, можно применить функцию unset, принимающую имя переменной в качестве аргумента.

Управление выполнением скрипта

До сих пор мы не рассматривали ни одной возможности досрочного прерывания выполнения PHP-кода. Действительно, в отличие от полноценных пользовательских приложений, веб-приложения не так часто требуют экстренного завершения, обычно это происходит в случае какой-либо фатальной ошибки. Функции¹⁴ PHP, предназначенные для мгновенного выхода, совпадают с таковыми в Perl: exit и die. Эти функции выполняют совершенно одинаковые действия (являются синонимами) и принимают один аргумент, значение которого возвращается в качестве кода ошибки.

Для выполнения “подскрипта”, скрипта, заданного в виде некоторой строки в другом скрипте, применяется функция eval. Обобщить все вышесказанное можно в виде следующего примера:

¹⁴ На самом деле в PHP exit и die — это не функции, а операторы языка.

```
<?php
    $s = "exit()";
    eval($s);
?>
```

Время и дата

Достаточно важным семейством функций является набор процедур для работы с датой и временем. В качестве наследства от Perl время в PHP хранится как POSIX-time — в виде числа секунд, прошедшего с 00:00:00 1 января 1970 года. Этот подход имеет свои преимущества и недостатки, обсуждать которые сейчас нет смысла.

С одной из функций описываемой категории мы уже встречались ранее, когда говорили про константы. Это функция `time`, возвращающая текущее время в виде целого числа. Следующий пример

```
<?php
    echo time();
?>
```

выведет это самое число на страницу в виде, не слишком понятном человеку. Функция `time` нужна для простого внутреннего представления времени в программе, а вовсе не для вывода своего результата на экран. Чуть позже мы поговорим о функциях, позволяющих вывести время именно в удобном для нас формате. Для получения более точных значений времени, с точностью до миллисекунд, служит функция `microtime`. `Microtime` возвращает результат либо в виде строки, либо в виде вещественного числа.

Для чтения и вывода человекочитаемых значений времени в PHP есть две очень мощных функции. Функция `strtotime` принимает строку и возвращает число, соответствующее POSIX-времени, представляемому данной строкой. Строка может иметь самую разную структуру:

```
<?php
    echo strtotime("15 Jan. 2014 4:33");
    echo strtotime("1993/05/10");
    echo strtotime("Tomorrow");
    echo strtotime("Last Monday");
?>
```

Можно указать относительное время, при этом второй аргумент функции `strtotime` будет определять точку отсчета:

```
<?php
    echo strtotime("1 month ago", 1212121212);
?>
```

Функция для преобразования числового значения POSIX-времени в строку носит имя `date`¹⁵. Единственным обязательным аргументом `date` является форматная строка, определяющая то, как будет выглядеть результирующее представление времени. Описывать все спецификаторы нет смысла, для этого есть стандартная документация, но вот некоторые из них:

- `h`, `H` — число часов в 12- и 24-часовом формате, соответственно;
- `i` — число минут;

- `u`, `U` — двух- и четырехциферное представление года;
- `M` — короткое название месяца ("Jan", "Sep", ...)
- и т.д.

Второй аргумент `date` — это собственно POSIX-время. Если этот аргумент опущен, то подразумевается, что он равен результату выполнения `time()`, то есть текущему времени.

Строковые функции

Еще один важный класс функций, которые могут нам понадобиться, — это функции для работы со строками. PHP предлагает весьма широкий спектр таких функций.

Первая часто используемая функция — `htmlspecialchars`, преобразующая строку к виду, удобному для непосредственного использования в HTML. В частности, специальные символы, такие, как "&", "'", "<", ">", преобразуются в "&", """, "<", ">". Функция `htmlspecialchars` работает в соответствии с текущей кодировкой. Если необходимо преобразовать все специальные символы, вне зависимости от кодировки, нужно использовать функцию `htmlentities`. Обратное преобразование выполняется функцией `html_entity_decode`.

Функция `substr` возвращает подстроку. Ее первый аргумент — исходная строка, второй — позиция в этой строке, а третий — длина требуемой подстроки:

```
<?php
    echo substr("Hello world", 3, 4); // "lo w"
?>
```

Функция `strpos` ищет первое вхождение подстроки, заданной в качестве второго аргумента в строку, — первый аргумент. Если подстрока не входит в заданную строку, возвращается `false` (логическая ложь). Следующий комплексный пример выведет модель используемого браузера (среди самых распространенных вариаций).

```
<?php
    $browser = strtolower($_SERVER["HTTP_USER_AGENT"]);
    if (strpos($browser, "MSIE")) {
        $browser = "Internet Explorer";
    } elseif (strpos($browser, "FIREFOX")) {
        $browser = "Firefox";
    } elseif (strpos($browser, "CHROME")) {
        $browser = "Chrome";
    } elseif (strpos($browser, "OPERA")) {
        $browser = "Opera";
    } else {
        $browser = "Not so popular browser";
    }
    echo $browser;
?>
```

В этом примере использовалась также функция `strtoupper`, устанавливающая верхний регистр для всех символов в строке. С применением суперглобального массива `$_SERVER` мы уже сталкивались ранее. Аналогичная функция для поиска последнего вхождения подстроки носит название `strrpos`.

Еще две часто используемых функции — это `strpos` и `stristr`. `Strstr` ищет первое вхождение

¹⁵ Кому-то может показаться, что логично было бы назвать ее `timetostr`, однако имя `date` — дань традиции UNIX.

подстроки в строку, но возвращает не число, как `strpos`, а остаток строки, следующий за этим вхождением. К примеру, можно удалить протокол из URL:

```
<?php
echo strstr("http://example.com", "://");
// "://example.com"
?>
```

Функция `strstr` выполняет аналогичные действия, но не чувствительна к регистру. Функция `strchr` является синонимом к `strstr`. Аналогичная функция, выполняющая поиск последнего вхождения, носит имя `strrchr`.

Часто бывает необходимо узнать количество вхождений некоторой подстроки в строку. Для этого можно применить функцию `substr_count`.

```
<?php
// грязная попытка проверки
// сбалансированности скобок в s-выражении
$sexp = "(car (list 1 2 3 (list 4)))";
$ob = substr_count($sexp, "(");
$cb = substr_count($sexp, ")");
if ($ob - $cb == 0) {
    echo "Сбалансировано";
} else {
    echo "Не сбалансировано";
}
?>
```

Регулярные выражения

Функции, представленные в предыдущем разделе, несомненно, нужны, но часто удобно, а иногда и необходимо выполнять более сложные операции со строками, в том числе поиск. Здесь нам на помощь придут регулярные выражения. Описание их синтаксиса займет слишком много места, поэтому я не буду подробно на нем останавливаться, а укажу лишь на способы работы с такими выражениями в PHP.

Для проверки строки на соответствие регулярно-му выражению служит функция `preg_match`:

```
<?php
if (preg_match("/.*ust/i", "DUST")) {
    echo "Совпадает";
}
?>
```

В этом примере строка "DUST" совпадает с регулярным выражением `"/.*ust/i"`, поскольку была отключена проверка регистра, а первый символ может быть любым (метасимвол `"."`, обозначающий любой символ и квантификатор `"+"`, обозначающий одно и более вхождений).

Кроме обычного поиска, можно выполнять замену подстроки, соответствующей регулярному выражению. Функция `preg_replace` принимает три аргумента: регулярное выражение, строку, которая должна быть вставлена, и строку, которая служит для сопоставления и вставки. К примеру, попробуем заменить строку, указывающую возраст на английском, в форме "age XX" на "XX years old":

```
<?php
$s1 = "Ivan Sukin, age 23, address: ...";
$s2 = preg_replace("/age ([0-9]+)/",
    "$1 years old", $s1);
echo $s2;
?>
```

Здесь мы использовали группировку подвыражений с помощью скобок и вставку результата сопоставления подвыражения (под номером один) с помощью `$1`. Интересующиеся использованием регулярных выражений более подробно могут обратиться к списку литературы.

Формы, базы данных

Одной из самых важных задач бэкенда веб-приложения является обработка форм на веб-страницах и работа с базой данных, содержащей какие-либо уникальные данные для каждого пользователя. В этом разделе мы немного коснемся этой темы.

В качестве базы данных будем использовать MySQL из поставки XAMPP. Для начала работы необходимо запустить из панели управления XAMPP сервер MySQL и, если вы хотите создать какую-нибудь тестовую базу, phpMyAdmin (по адресу `http://localhost/phpmyadmin`). База данных с именем "test" скорее всего уже будет создана по умолчанию, если нет, ее можно создать во вкладке "Базы данных". Если щелкнуть на имени базы в дереве слева, появится окно создания таблицы. Создадим таблицу с именем `test` и числом столбцов, равным двум. Пусть столбцами будут `name` (`VARCHAR`, 32) и `age` (`INT`), представляющие, соответственно, имя и возраст некоторого человека. Добавим в базу (через вкладку "Вставить") двух человек: John, 34, и Chris, 20; этого хватит для объяснения основ работы с MySQL в PHP.

Попробуем вывести имя самого молодого человека из тех, что имеются в нашей базе данных. Это Chris:

```
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$people = mysql_query("SELECT * FROM
    test ORDER BY age ASC");
$first = mysql_fetch_assoc($people);
echo $first["name"]; // "Chris"
?>
```

Функция `mysql_connect` подключается к серверу MySQL. В примере этот сервер расположен на нашем локальном компьютере. Подключение производится от имени администратора (`root`), который по умолчанию имеет пустой пароль (в реальной системе так быть не должно!). Функция `mysql_select_db` выбирает базу данных из тех, что имеются на нашем сервере, а функция `mysql_query` осуществляет запрос к этой базе. В нашем запросе мы получаем все строки из таблицы `test` и упорядочиваем их по возрастанию значения в поле `age`. `mysql_fetch_assoc` преобразует текущую строку из запроса в массив PHP и делает текущей следующую строку. После этого мы можем с помощью простой индексации этого массива получить нужное нам значение.

Для простейшей демонстрации работы с формами попробуем написать примитивный интерфейс для добавления людей в нашу базу данных. Так будет выглядеть HTML-документ с кодом на PHP:

```
<!doctype html>
<html>
  <body>
    <?php
      mysql_connect("localhost", "root", "");
      mysql_select_db("test");
      if (isset($_GET["add"])) {
        $name = $_GET["name"];
        $age = $_GET["age"];
        $query = "INSERT INTO test
          VALUES ('$name', '$age')";
        $result = mysql_query($query);
        if (result) {
          echo "Успех!\n";
        } else {
          echo "Произошла ошибка\n";
        }
      }
    ?>
  <div>
    <form method="get">
      Имя: <input type="text"
        name="name"/><br/>
      Возраст: <input type="number"
        name="age"/><br/>
      <input type="submit" name="add"
        value="Добавить"/>
    </form>
  </div>
</body>
</html>
```

В примере мы используем метод GET для передачи параметров запроса. Это означает, что параметры будут переданы прямо в адресной строке браузера, но поскольку у нас здесь нет ничего секретного, это хороший и простой вариант. Если бы нам нужно было передать, например, пароль, нужно было бы использовать метод POST и передавать параметры в теле самого запроса. Для получения значений параметров мы используем суперглобальный массив `$_GET`. Всю работу мы выполняем только в том случае, если была нажата кнопка “Добавить” и, следовательно, установлен параметр “add”, что мы и проверяем знакомой нам функцией `isset`. Результирующий запрос к базе данных — простой INSERT.

Вопросу работы с формами и базами данных на PHP посвящена обширная литература.

Графика

В базовую поставку интерпретатора PHP входит интерфейс к библиотеке GD, позволяющий “на лету” генерировать изображения. Попробуем воспользоваться ею, чтобы нарисовать один из фракталов — дракона Хартера – Хейтуэя. В виде СИФ (системы итерируемых функций) на комплексной плоскости этот фрактал описывается так:

1. $f_1(z) = (1 + i)z / 2$
2. $f_2(z) = (1 - i)z / 2$

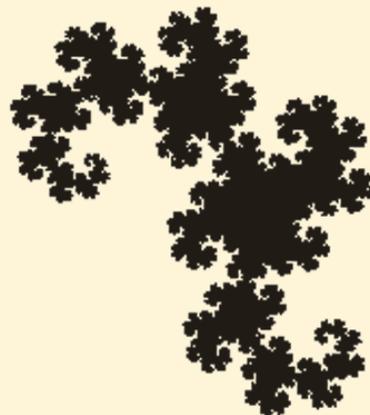
Преобразование этой системы в рекурсивную зависимость от двух начальных точек $a(x_a, y_a)$ и $c(x_c, y_c)$ дает следующие правила:

$$\begin{aligned} \text{Дракон}(a, c) = \\ x_b = (x_a + x_c) / 2 + (y_c - y_a) / 2, \\ y_b = (y_a + y_c) / 2 + (x_c - x_a) / 2 \\ \text{в Дракон}(a, b), \text{ Дракон}(c, b). \end{aligned}$$

Получающаяся при этом точка b образует два перпендикулярных отрезка — ab и bc . На практике нам нужно ограничить количество итераций в этой рекурсивной схеме. Получающаяся программа представлена в следующем листинге¹⁶:

```
<?php
  $order = 20;
  $image = imagecreatetruecolor(800, 600);
  imagefilledrectangle($image, 0, 0,
    imagesx($image) - 1,
    imagesy($image) - 1,
    imagecolorresolve($image, 255, 255, 255));
  $color = imagecolorresolve($image, 0, 0, 0);
  dragon($image, imagesx($image) * 3/8,
    imagesy($image) * 3/8,
    imagesx($image) * 5/8,
    imagesy($image) * 5/8,
    $order, $color);
function dragon($image, $xa, $ya, $xc,
  $yc, $i, $color) {
  if($i == 0) {
    imageline($image, $xa, $ya, $xc, $yc,
      $color);
  }
  else {
    $xb = ($xa + $xc) / 2 + ($yc - $ya) / 2;
    $yb = ($ya + $yc) / 2 - ($xc - $xa) / 2;
    dragon($image, $xa, $ya, $xb, $yb,
      $i - 1, $color);
    dragon($image, $xc, $yc, $xb, $yb,
      $i - 1, $color);
  }
}
header('Content-type: image/png');
imagepng($image);
imagedestroy($image);
?>
```

Результирующая картинка выглядит следующим образом:



Немного разберем пример. Функция `imagecreatetruecolor` создает True Color (32-битный цвет, с альфа-каналом) — изображение заданного размера, а `imagefilledrectangle` рисует заполненный цветом прямоугольник. Переменная `$image` в этом случае как раз име-

¹⁶ Это полноценный файл PHP, данный код не должен быть вставлен в HTML.

ет тип `resource`, о котором мы вскользь упоминали ранее. Функции `imagesx` и `imagesy` возвращают размеры изображения, функция `imagecolorresolve` возвращает номер желаемого цвета в палитре. Функция `dragon` в точности повторяет ход нашей мысли, представленный выше, и выполняет 20 итераций. Параллельная рекурсия — слишком ресурсоемкий тип вычислений, поэтому мы не можем так просто позволить себе большое число вызовов, но даже заданных двадцати уже хватает для получения нормальной картинки.

Плюсы и минусы языка

Все это время мы придерживались нейтральной, сугубо технической позиции относительно PHP и не говорили о его сильных и слабых сторонах. Не углубляясь в запутанные понятия о принципах проектирования языков программирования и “красоте” формальных систем, можно выделить следующие несомненные плюсы PHP:

- Легкость в освоении. Язык прост для начинающих, содержит небольшое число ясных синтаксических конструкций. Люди, знакомые с Perl и C, могут начинать писать на нем практически сразу же после беглого чтения стандартной документации.

- Большое количество удобных встроенных функций практически на все случаи жизни. Их действительно много, они покрывают множество областей: работу с текстом, математику, работу с базами данных, графикой, интерфейсами пользователя.

- Отзывчивое, понимающее и большое сообщество, доставшееся как наследие от сообщества Perl.

К значительным минусам PHP можно отнести следующие факты:

- Тяжелое наследие “инструмента для одного человека”. Несмотря на то что языку скоро будет 20 лет и он пережил уже пять мажорных версий, в нем до сих пор чувствуется легкий привкус проектирования “на скорую руку”¹⁷. Это не представляет большой проблемы, но иногда может сбивать с толку.

- Отчасти следствием предыдущего минуса является туманность перспектив развития. Шестая ветка PHP пока что находится в стагнации, поскольку предложенные изначально идеи оказались слишком сложными и громоздкими для реализации. Примерно то же самое произошло и с Perl.

- Легкость в освоении. Этот факт является как плюсом, так и минусом. Многие заказчики ориентируются на начинающих PHP-программистов, что не всегда оказывается хорошим решением и вносит некоторое влияние в глобальный стиль программирования на этом языке.

Заключение

Стоит ли вам писать что-либо на PHP? Если вы собираетесь серьезно заниматься разработкой бэк-ендов веб-приложений, то ответ, конечно, утвердительный. Несмотря на указанные выше минусы, PHP остается фактически единственным действительно удобным и эффективным языком для этой цели, что бы об этом ни говорили проповедники других технологий. PHP непрерывно развивается, у него появляются новые модули, новые, в том числе компилируемые, реализации, сообщества, инфраструктуры, фреймворки.

В этой статье я попытался дать слишком сжатый обзор всего огромного мира PHP-программирования и обошел стороной слишком сложные или не так часто встречающиеся возможности. К примеру, в веб-программировании не так часто приходится открывать локальные файлы, кроме баз данных, и рисовать самостоятельные интерфейсы пользователя, хотя это тоже можно сделать на PHP. Я надеялся с помощью небольшого набора относительно простых примеров дать читателю возможность почувствовать вкус этого языка и решить для себя вопрос о том, стоит ли его использовать. Даже если кто-то решит, что PHP — не тот инструмент, который ему подходит, знакомство с таким гигантом Всемирной паутины никогда не будет лишним.

Литература

1. <http://www.php.net/manual> — официальная стандартная документация.
2. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS. Питер, 2013.
3. Янк К. PHP и MySQL. От новичка к профессионалу. М.: Эксмо, 2013.
4. Котеров Д.В. PHP 5. BHV-СПб, 2013.
5. Фридл Дж. Регулярные выражения. Символ-Плюс, 2013.
6. Арнольд М. Администрирование Apache. Лори, 2013.

¹⁷ Можно взглянуть, например, на путаницу в именах стандартных функций.

Графическая среда программирования Blockly (Блокли)

Часть 1. Что такое Блокли и почему на него стоит обратить внимание

Лирическое вступление

► Изначально я хотел ограничиться коротким сообщением о графической среде программирования Blockly (Блокли): дело в том, что Блокли предназначен для быстрого и почти самостоятельного старта детей в программировании.

В конце концов, разве не логично дать педагогу возможность тоже побыть ребенком, освежить у него это бесценное детское “Я сам!”?

Но суровый редактор идею не одобрил: “Нет у нас такого формата — короткое сообщение”. Пришлось думать о более серьезной статье. Вероятно, это должна была быть статья с описанием Блокли, методическими идеями, срав-

нительным анализом с другими средствами обучения программированию.

Чем больше я думал, однако, тем меньше мне хотелось оставаться в “объективных” рамках. В конце концов, педагог — это прежде всего личность. Нет второй такой профессии, где личность является столь же большой и неотъемлемой составляющей профессионала, точнее даже, не частью, а стержнем. Если так — я имею право не только на рассказ об инструментах и методиках, но и на слово личности, обращенное к другим личностям.

Почему обучение школьников программированию?

Я — программист, педагогического образования не имею, но я слушал лекции и сдавал экзамены вместе со своей мамой — студенткой педагогического института, еще до своего рождения, по-

М.Н. Семионенков,
переводчик ядра
Блокли, редактор
перевода курсов на
базе Блокли,
к. ф.-м. н.
г. Сан-Франциско

этому, вероятно, тяга к учительско-преподавательской работе у меня в крови. Предметы разные: шахматы, программирование, немного — занимательная математика. Аудитория — детский сад, школа, студенты, аспиранты, курсы повышения квалификации. Я с удовольствием пытался учить тому, что люблю сам. Работа со студентами и взрослыми была приятной подработкой, работа с детьми — занятием для души, где деньги были приятным, но не обязательным дополнением. Потом экономическая неустроенность и растущие семейные обязанности надолго отлучили от занятий для души. Позже, когда быт налажился, я начал писать “в стол” материалы для шахматных занятий. Постепенно, однако, программирование вытеснило шахматы...

...Современное общество затрачивает огромные ресурсы на обучение, воспитание, развитие детей и молодежи. Формы образования, обучения, тренировок очень разнообразны, и во всех есть свой смысл: и в общем школьном образовании, и в специальных школах, секциях, кружках, университетах, дающих как востребованные специальности, так и вполне абстрактные степени, и в балетных школах, где на ребенка сразу смотрят как на будущего профессионала... И все-таки, при всем “равноправии” разных форм образования, ресурсы, которые общество может выделить на обучение, конечны. Значит, задача рационального использования этих ресурсов всегда актуальна. Я — маленький автономный образовательный ресурс, и я должен сам решить вопрос своего рационального использования.

Так почему я выбрал именно обучение программированию? Это выбор между “просто развитием” ребенка или молодого человека и развитием с прицелом на престижную, хорошо оплачиваемую и востребованную профессию — здоровую экономическую базу семьи.

Вероятно, общество еще не вполне осознало феномен программирования как массовой профессии и производительной силы одной из важнейших отраслей экономики. Невосполняемые энергоресурсы рано или поздно кончатся, а информация и потребность в автоматизации будут “производиться” вечно, и к автоматизации обработки информации человечество только приступает.

“Там, у них”, к слову, необходимость пропаганды программирования среди детей и необходимость включения программирования в школьную программу вполне осознаны: в США это движение спонсируется гигантами индустрии программного обеспечения, которые озабочены все возрастающим кадровым голодом. Я — большой противник слепого перенесения чужого опыта на российскую почву, но в данном случае заокеанский пример явно заслуживает внимания: страна, которая имеет богатую практику скупки

готовых специалистов, подготовленных в Индии, Китае, России, озабочилась сохранением своей гегемонии в важнейшей отрасли и подготовкой национальных кадров. Что уж тогда говорить о странах, не столь успешных в привлечении специалистов высокой квалификации на международном рынке труда? Собственно, толчком к этой работе было знакомство с американской кампанией по пропаганде программирования среди школьников. За неделю в общенациональной программе “Час кода” приняли участие десятки тысяч школьников (по “официальным” данным организаторов кампании — миллионы, но кухня формирования общественного мнения в Соединенных Штатах — отдельная история).

Наиболее популярным был “Час кода” на базе графической среды программирования Blokly (Блокли). Собственно, привлек меня не “рейтинг” Блокли, а качества Блокли, которые, очевидно, повлияли на его популярность среди американских школьников и открыли дорогу для его широкого международного признания.

После чуть затянувшегося вступления перейдем к собственно предмету.

Что такое “графическая среда программирования”?

Графическая среда программирования — это среда языка без синтаксиса. С традиционным языком программирования неотъемлемо связано понятие синтаксиса. Пишущий программу должен правильно писать соответствующие ключевые слова и правильно употреблять специальные символы, которые оформляют лексемы и синтаксические конструкции языка. Для профессионала синтаксис нового языка не составляет проблемы: в смысле синтаксиса процедурные языки — весьма близкая родня (то же можно сказать об объектно ориентированных языках). Другое дело — начинающий: для него знакомство с абстракциями программирования усложняется (малосодержательной по сути) проблемой усвоения синтаксиса: увидеть работу программы можно, лишь исправив все синтаксические ошибки. На пути к работе над алгоритмом — рутинная, отдаляющая результат, уменьшающая удовольствие от процесса.

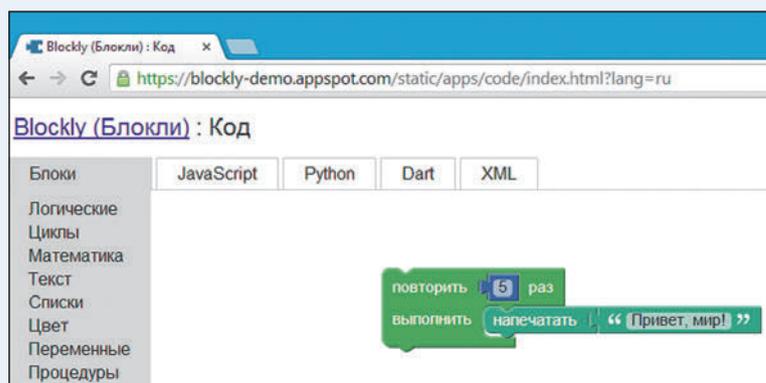


Рис. 1

Графическая среда программирования как раз позволяет уйти от синтаксической рутины: весь синтаксис “запаян” в блоки, из которых составляется программа. Блоки включают в себя ключевые слова и оформляют управляющие конструкции. На рис. 1 на с. 33 классический стартовый пример на Блокли (чуть усложненный циклом).

С клавиатуры здесь введены только текст “Привет, мир!” и число исполнений цикла. Важно: графическая среда программирования не позволяет написать синтаксически неправильную программу.

Немного о родословной Блокли

Блокли — разработка фирмы Google (2012 год). Графических систем ныне существует великое множество. Источник вдохновения для Блокли (и не только Блокли) — система Scratch (<http://scratch.mit.edu/>) (применительно к графическим системам, эту систему, наверное, можно сравнить с Фортраном — прародителем процедурных языков программирования).

Ближайший родственник Блокли — система App Inventor (<http://appinventor.mit.edu/explore/>), предназначенная для разработки приложений для смартфонов на операционной системе Андроид. Blockly и App Inventor выросли из одного проекта и используют общий графический интерфейс.

В какой-то момент проект Блокли был в Google закрыт. Главный разработчик Нэйл Фрэйзер (<https://neil.fraser.name/>) выразил свой протест против перевода на другой проект своеобразно: взял все свои неотгулянные отпуска и два месяца “пахал” на доведение проекта до ума. После выхода из “отпуска” предъявил работу начальству. Проект восстановили, отпуск тоже, и теперь, как видите, о проекте даже в зарубежных журналах пишут 😊.

Доступ по Интернету

Важное достоинство системы Блокли — ее доступность по Интернету. Чтобы начать работу с Блокли, на компьютер пользователя не нужно ничего устанавливать: достаточно иметь выход в Интернет и не слишком старый поисковик (браузер) (<https://blockly-demo.appspot.com/static/apps/index.html?lang=ru>). Это качество трудно переоценить именно для начального обучения программированию, например, в семье. Отсутствие процедуры установки и связанных с ней проблем понижает порог для старта.

Наряду с доступностью по Интернету, предусмотрено поддерживаются также локальные установки Блокли. Эта возможность востребованна в школах, где нет стабильного выхода в Интернет.

Начальный набор приложений Блокли

Google, помимо собственно графической системы, предоставляет пользователям не очень боль-

шой, но достаточно содержательный стартовый набор приложений, реализованных на базе Блокли. Собственно, с точки зрения конечного пользователя именно приложения и интересны: сама система интересует только разработчиков. В этот набор на момент написания статьи входят приложения:

Головоломка — для знакомства с интерфейсом Блокли;

Лабиринт — обучение азам программирования;

Черепашка — известная рисующая рептилия;

Графический калькулятор — позволяет строить графики функций;

Код — ключевое приложение для обучения процедурным языкам программирования;

Калькулятор посадочных мест в самолете — для решения математических задач с одной или двумя переменными.

Предоставляется также приложение для построения новых блоков Блокли.

Во второй части статьи мы уделим большее внимание некоторым приложениям.

Связь Блокли с промышленными языками программирования

Несмотря на заявленную Google на начальном этапе ориентацию на школьные приложения, Блокли не изолирован в чисто учебной среде: система содержит генераторы кода на JavaScript, Dart и Питон. В приложении Код генерируемый код доступен пользователю. Это позволяет рассматривать Блокли как средство изучения промышленных языков и профессиональный инструмент.

Открытые исходные тексты

Блокли — бесплатная система с открытыми текстами. Это — подарок разработчикам во всем мире. Открытые исходные тексты открывают широкие возможности для развития разнообразного учебного обеспечения.

Ориентация на локализацию

Наконец, система Блокли выгодно отличается для неанглоговорящих пользователей изначальной ориентацией на интернационализацию и локализацию.

Результат — мировое признание

Сочетание всех перечисленных качеств привлекло к Блокли огромное внимание. Энтузиасты из разных стран (включая вашего покорного слугу) перевели ядро Блокли на десятки языков и локализовали приложения стартового набора Блокли. Блокли, таким образом, наряду с синтаксическим барьером, снимает для начинающих языковой барьер.

Признание возможностей Блокли и его открытые исходные тексты, в свою очередь, дают им-

пульс развитию учебных средств на базе Блокли независимыми разработчиками. Во-первых, отмечу работу некоммерческой организации code.org, которая координирует в Соединенных Штатах работу по пропаганде программирования среди школьников и продвижению программирования в школьные программы. Эта организация разработала для кампании “Час кода” вводный курс на базе Блокли (<http://learn.code.org/hoc/1>) и, что важно, организовала работу переводчиков. Этот часовой курс является перепевом одного из приложений стартового комплекта Блокли (“Лабиринт”), но содержит больше сцен и учебных инструкций.

Другая разработка code.org — курс введения в информатику из 20 уроков (http://learn.code.org/users/sign_up?user%5Buser_type%5D=teacher), также открытый для локализации (автор имел честь редактировать перевод этого курса на русский, наряду с переводом курса для “Часа кода”). Достоинства и недостатки курса — отдельная тема, но система поддержки учебного процесса определенно заслуживает внимания. Система позволяет учителю организовать виртуальный класс, и в рамках этого класса учитель в любой момент может контролировать по Интернету прогресс каждого из учеников. У учеников, в свою очередь, есть возможность в любой момент возобновить работу с любого задания. Думаю, если исходные тексты системы станут открытыми, она будет подхвачена для развития в разнообразных учебных системах, и не только для обучения программированию.

Другое направление работы сообщества — создание программ на базе приложений из стартового набора Блокли. Выделю работу энтузиаста Пол Йорк (<https://sites.google.com/site/paulyorkeswebsite/code-workshop>) (на английском). Пол создал ряд примеров для приложений Черепашка и Код.

Блокли и программируемые роботы

Успех Блокли привел к бурному росту интереса к нему в области программируемых роботов. Программируемые роботы — и перспективный инструмент для обучения детей робототехнике, и привлекательная коммерческая ниша для производителей игрушек. Интерфейс Блокли пришелся в этой области весьма кстати. На базе Блокли разработан генератор кода для процессоров Ардуино, популярных в игровых роботах.

О перспективах Блокли в контексте дальнейшего обучения программированию

Блокли, как уже отмечалось, не является замкнутой в себе игрушкой. Блокли открывает достаточно ясные пути перехода к программированию, ориентированному на профессию.

Прежде всего Блокли позволяет плавно перейти от локализованной версии на английскую. Хотим

мы этого или нет, английский — латынь современного программирования, и сколько-нибудь серьезное изучение программирования немислимо без освоения базового набора слов, используемых в языках программирования. Перейдя к английской версии, изучающий без проблем может освоить этот базовый набор слов уже после того, как он впитал их семантику на родном языке.

Далее, общность интерфейса с App Inventor упрощает переход на App Inventor, где уже можно писать реальные приложения для смартфонов.

Наконец, Блокли содержит генераторы кода на промышленные языки: JavaScript, Dart и Питон. Приложение Код из стартового набора позволяет видеть созданный код на этих языках. Это — хорошая отправная точка для обучения веб-программированию.

Если внутренние интерфейсы Блокли будут лучше документированы (а со временем, я надеюсь, это произойдет), хорошей задачей для наиболее сильных учеников на факультативах по программированию может быть создание приложений для Блокли.

Часть 2. Как выглядят приложения Блокли

Среда программирования Блокли устроена достаточно просто: пользователь собирает программу из блоков и вставок. Этот процесс напоминает сбор головоломки из фигурных частей. Блоки — аналог управляющих конструкций, процедур и операторов языка программирования. Вставки служат параметрами процедур и аргументами функций, а также представляют значения, возвращаемые функциями. Естественно, детям во время знакомства обо всех этих абстракциях говорить необязательно, достаточно познакомить их с приложением Головоломка.

Приложение Головоломка

Здесь страны — главные блоки, внутрь которых можно помещать другие блоки и вставки, города — блоки, помещаемые внутрь главных блоков, и, наконец, флаги — вставки. Собрав все блоки, пользователь может проверить результат, нажав кнопку “Проверить результат”.

Главный порог для использования этого приложения — знание географии (можно было бы сделать что-то ближе к сбору изображений, чтобы была меньшая зависимость от знаний).

Приложение Лабиринт

Приложение Лабиринт служит для знакомства с азами программирования: последовательным исполнением команд, циклами и условными операторами. Приложение разбито на десять уровней,

по возрастающей сложности. Экран разбит на три части (это — общая черта приложений стартового набора Блокли):

- левая, где программа выполняется;
- средняя, содержащая “кассу” блоков, из которых можно составить программу;
- правая — рабочая область, в которой программа собирается.

Когда программа собрана, остается нажать кнопку “Запустить программу” и посмотреть результат исполнения.

В случае успеха учащемуся предлагается перейти на следующий уровень.

Уровни 1 и 2 демонстрируют последовательное исполнение команд. Уровень 3 вводит цикл до достижения цели.

Уровень 4 несколько сложнее: он требует вычленивать повторяемую группу действий и включить эту группу внутрь цикла до цели.

Уровень 5 закрепляет навыки, полученные на предыдущих уровнях, а уровень 6 вводит условный оператор.

Последующие уровни — возрастающие по сложности лабиринты, которые нужно проходить, используя уже известные средства. Уровень 10 может служить алгоритмическим упражнением даже для профессиональных программистов.

Заметим, что в правом верхнем углу есть кнопка выбора путника. Под полем исполнения программы (левее кнопки “Запустить программу”) есть две кнопки: “Просмотреть созданный код JavaScript” и “Сохранить и показать ссылку на блоки”. Если первой из них на столь ранней стадии изучения пользоваться, пожалуй, рановато, то вторая — очень даже полезна: достигнув успеха, ученик может нажать эту кнопку, чтобы сохранить свое решение. Примеры программ см. на рис. 4–9.

Приложение Черепашка

Приложение Черепашка (как и другие приложения) отличается от стартового приложения Лабиринт гораздо большим числом доступных блоков. По этой причине блоки оформлены в группы, и для извлечения блока из группы необходимо сначала открыть группу щелчком мыши. В остальном интерфейс совпадает с интерфейсом приложения Лабиринт.

Приложение Черепашка содержит гораздо более развитые управляющие конструкции. Так, в группе Циклы присутствуют четыре вида циклов и оператор выхода из цикла. Используя блоки из группы Процедуры, можно создать процедуры и функции (процедуры, возвращающие значения).

В группе Логические представлен конструктор для условного оператора с произвольным числом “иначе если”.

В группе Математика — ряд математических функций.

Пример программы для Черепашки см. на рис. 10.

Приложение Код

Приложение Код — ключевое в стартовом наборе приложений Блокли. Оно не содержит программируемого исполнителя, но содержит наиболее развитый набор встроенных процедур общего назначения. Собственно, главное отличие от, например, Черепашки — наличие средств работы со списками (смотри группу блоков Списки).

В приложении Код подчеркивается нацеленность на генерацию кода в промышленные языки, о чем говорят закладки JavaScript, Python, Dart. Кроме того, программу можно перевести в форму XML.

Пример программы см. на рис. 11.

Ограничения Блокли

В настоящем виде Блокли не приспособлен для написания больших программ. Блокли поддерживает процедуры с параметрами, но при этом полноценных локальных переменных в языке нет. Естественно, это большой барьер на пути использования языка для серьезного обучения программированию.

Вопрос о представлении в языке объектно ориентированного и параллельного программирования на момент написания статьи находится в стадии обсуждения. В целом можно сказать, что Google — разработчик амбициозный, и, хотя четких планов не обозначено, есть основания полагать, что проект будет развиваться.

Резюме

1. Блокли отличается от неграфических систем тем, что в нем от изучающего скрыт синтаксис языка программирования.
2. От других графических систем Блокли отличается тем, что переведен на русский язык.
3. Блокли доступен по Интернету (<https://blockly-demo.appspot.com/static/apps/index.html?lang=ru>).
4. Блокли не замкнут в учебной среде.
5. Блокли — бесплатная и открытая система.

Блокли, безусловно, заслуживает внимания всех интересующихся обучением детей и школьников программированию. Хочется надеяться, что на Блокли обратят внимание и те, кто занимается разработкой программного обеспечения для учебного процесса, и те, от кого может зависеть продвижение перспективных средств в школу.

И, наконец, стартовая точка встречи энтузиастов Блокли (пока не созрело время для более серьезной площадки) — группа “Графическая среда программирования Blockly” ВКонтакте. До встречи!

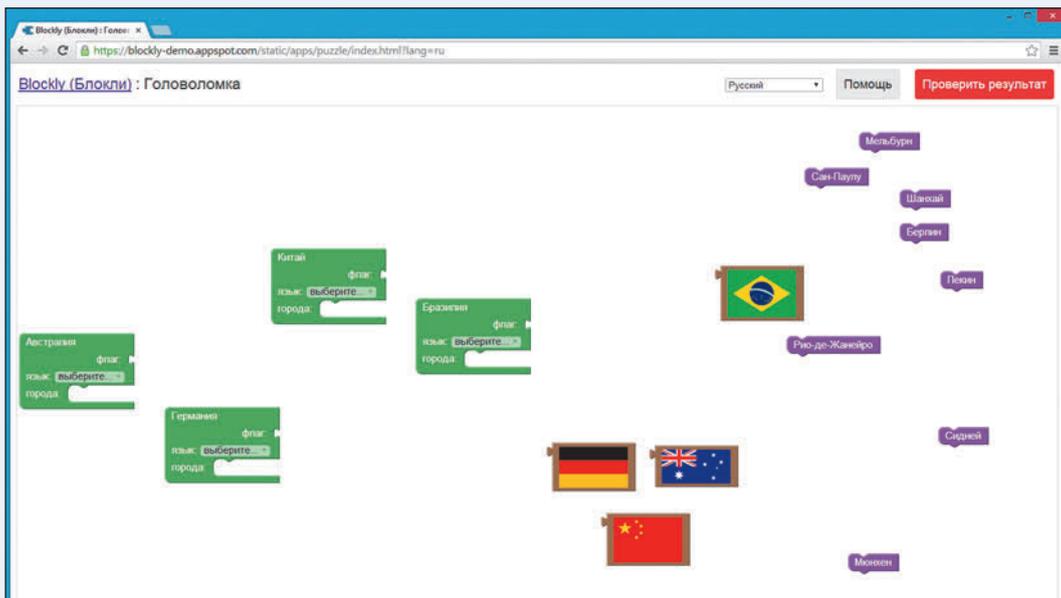


Рис. 2

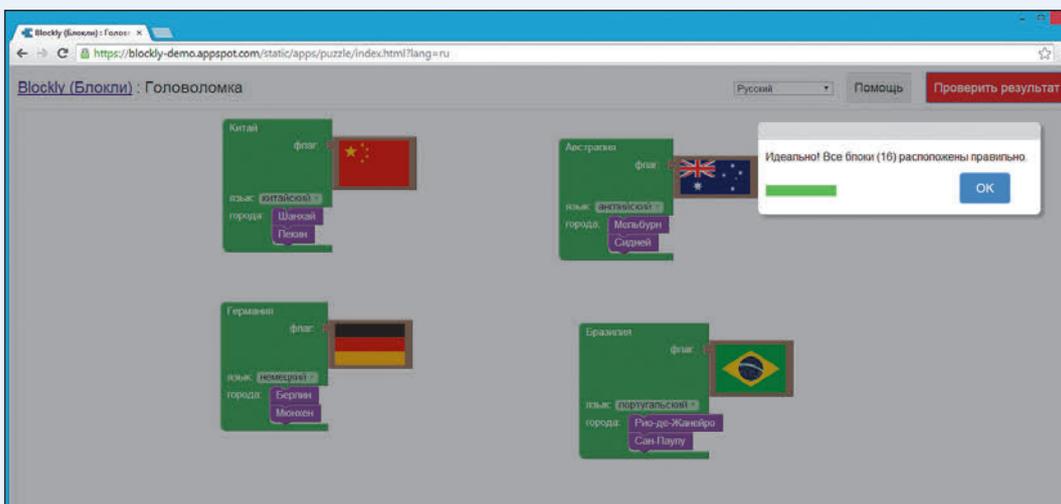


Рис. 3

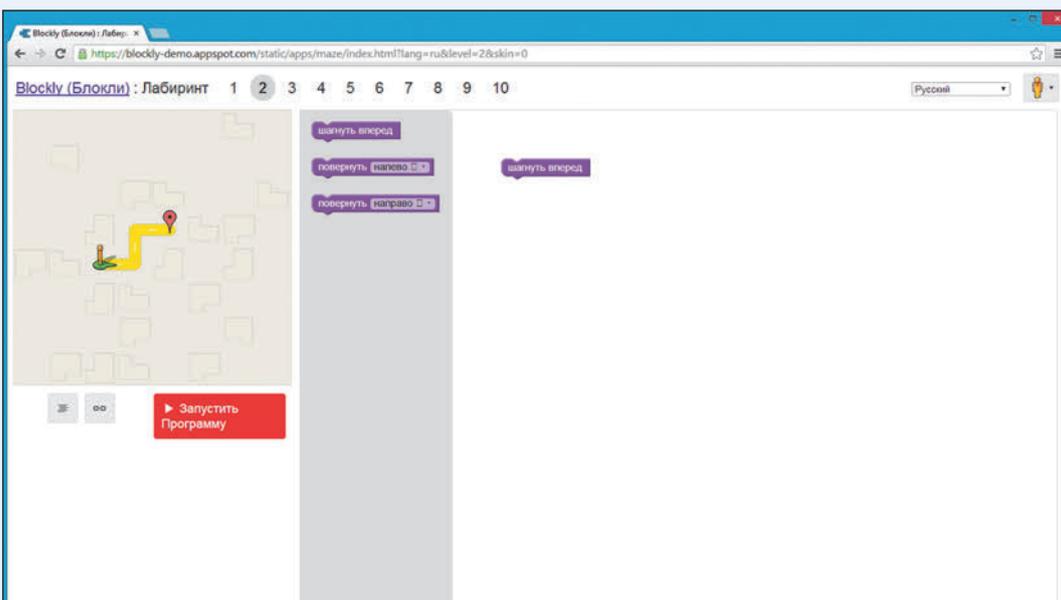


Рис. 4

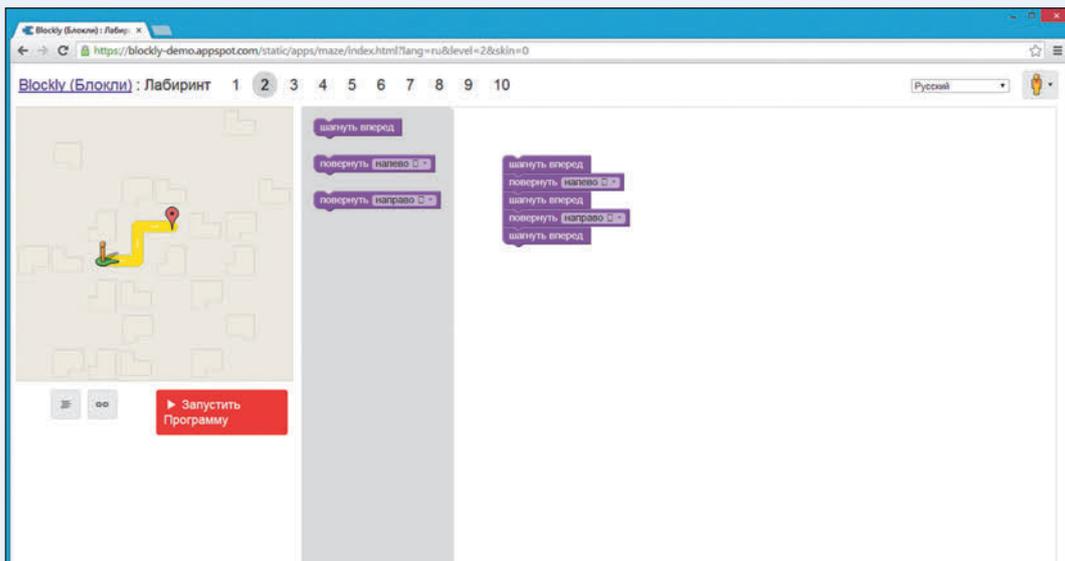


Рис. 5

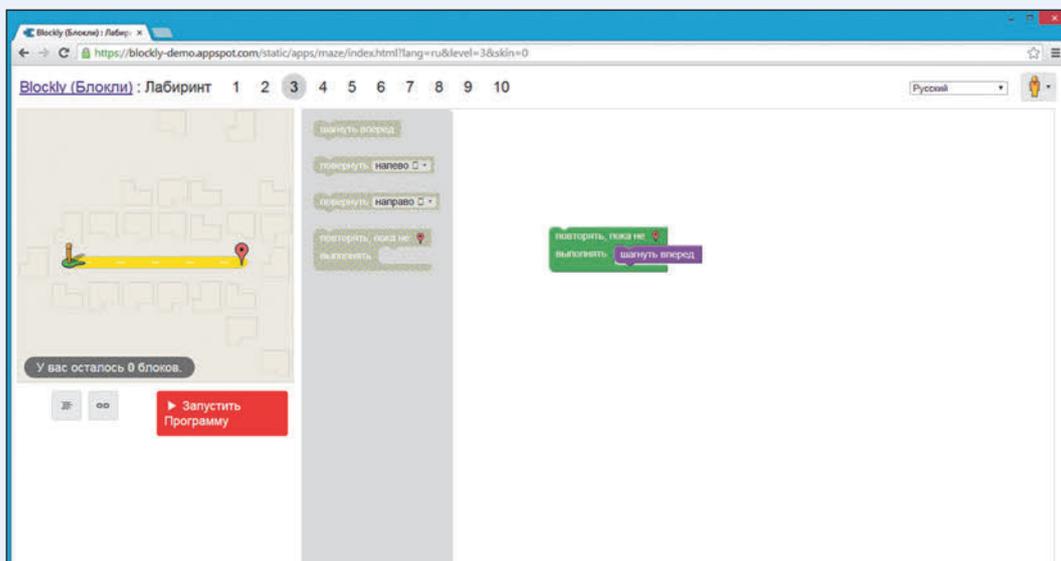


Рис. 6

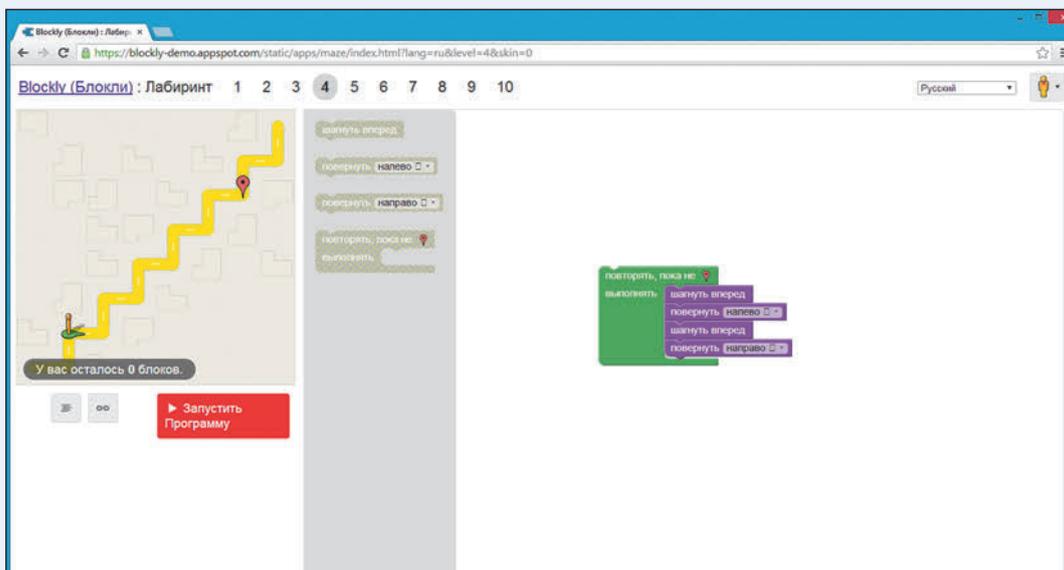


Рис. 7

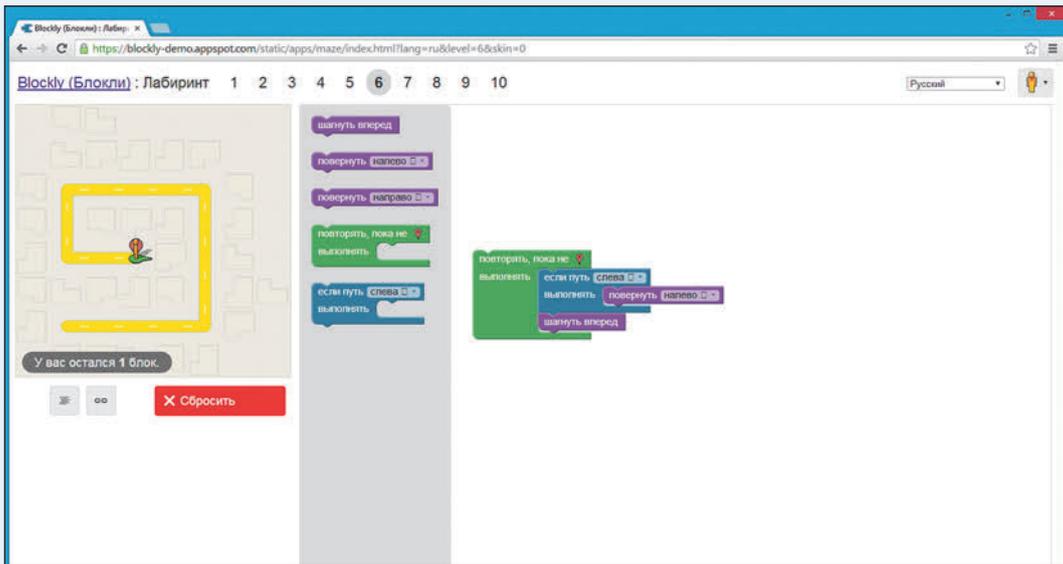


Рис. 8

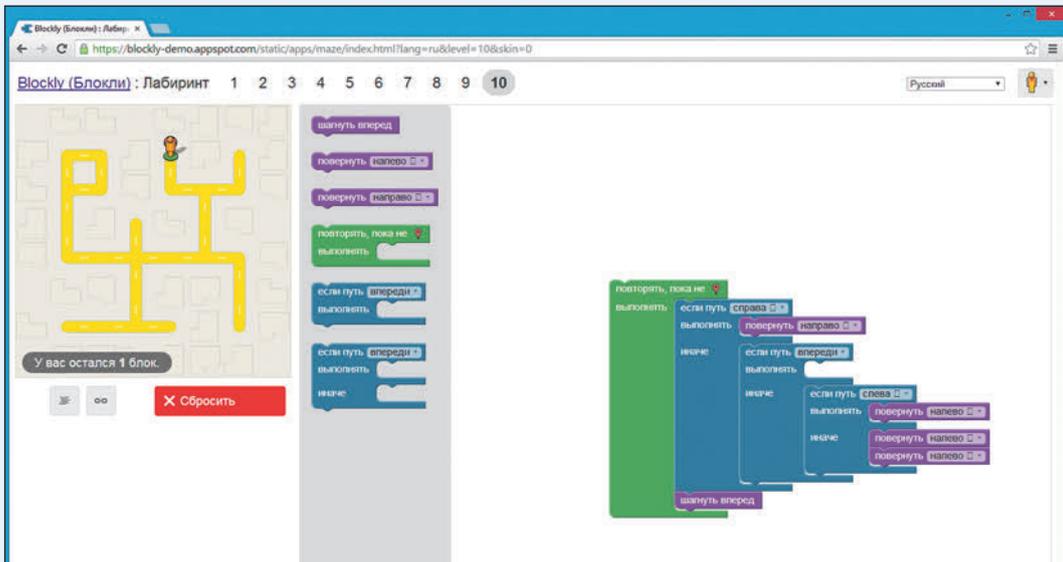


Рис. 9

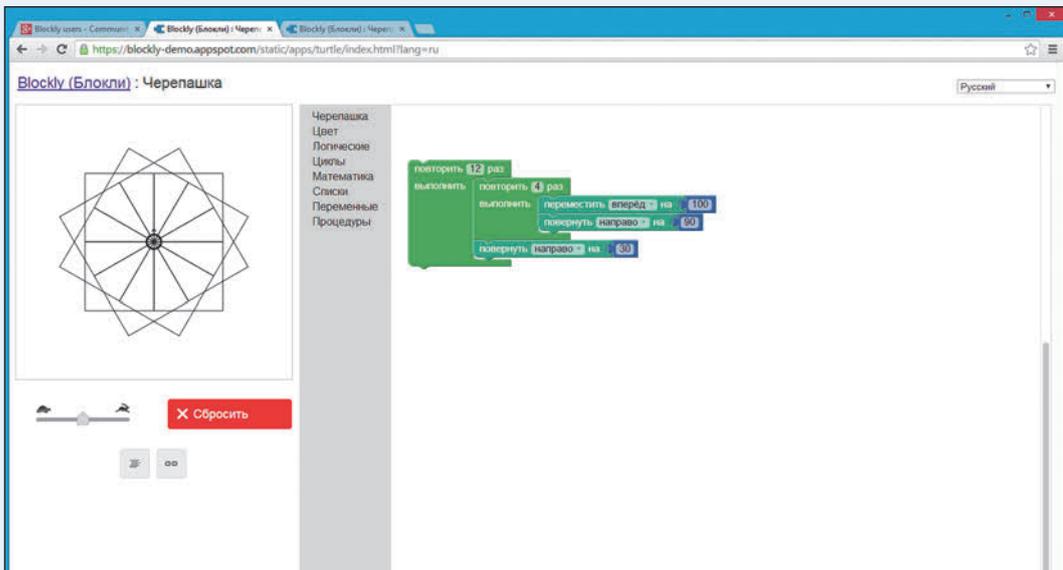


Рис. 10

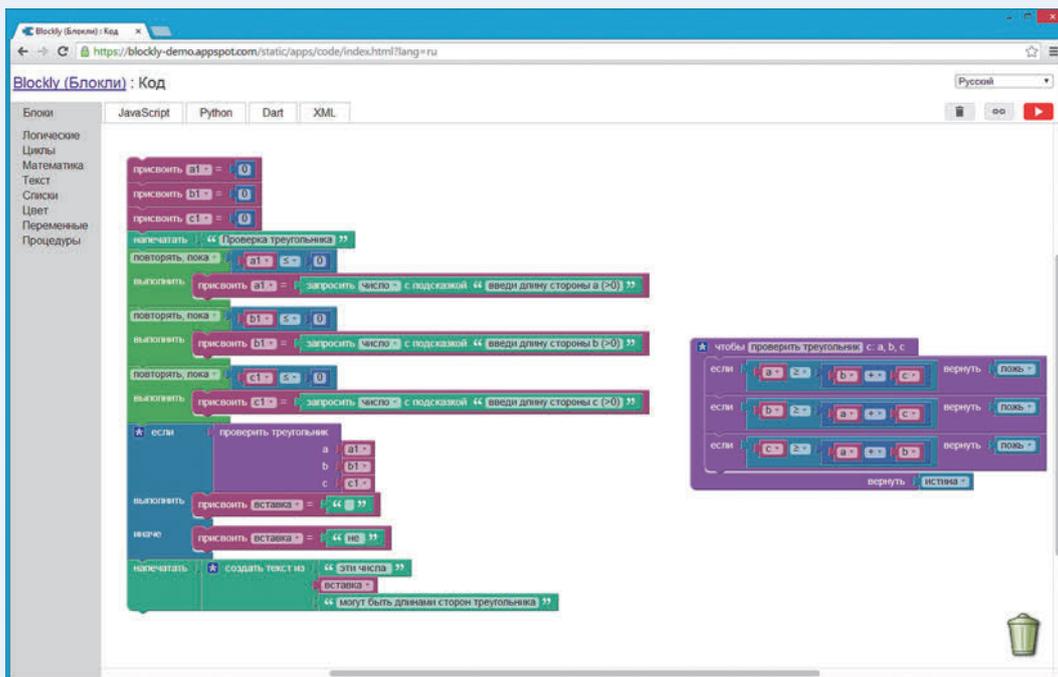


Рис. 11

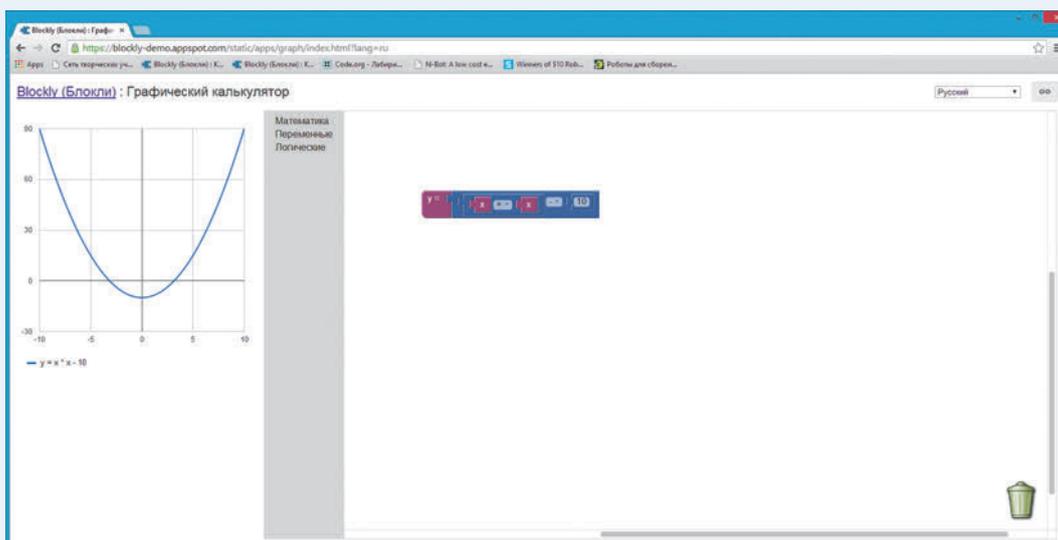


Рис. 12



2014/15 учебный год

ДИСТАНЦИОННЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

(с учетом требований ФГОС)

Прием заявок – с 1 апреля 2014 года

образовательные программы:

- НОРМАТИВНЫЙ СРОК ОСВОЕНИЯ – **108** УЧЕБНЫХ ЧАСОВ

Стоимость – 3990 руб.

- НОРМАТИВНЫЙ СРОК ОСВОЕНИЯ – **72** УЧЕБНЫХ ЧАСА

Стоимость – 3390 руб.

По окончании выдается удостоверение о повышении квалификации
установленного образца

Перечень курсов и подробности – на сайте edu.1september.ru

Пожалуйста, обратите внимание:

заявки на обучение подаются только из Личного кабинета,
который можно открыть на любом сайте портала www.1september.ru



Создаем скринкаст на платформе HTML5

1. Основные сведения

► Скринкаст (от англ. *screen cast* — “трансляция экрана”) — передача или демонстрация происходящего на экране компьютера (рабочем столе или его области). Насколько можно судить, широкое распространение скринкасты получили примерно в то же время, когда стали популярны браузерные приложения и веб-базированные информационные системы с довольно сложным интерфейсом. Например, чтобы объяснить пользователю сайта eBay, как сделать покупку, оказалось проще записать действия с голосовым комментарием, чем описывать их в виде текстовых инструкций. Тем более что веб-интерфейсы имеют склонность часто меняться.

Существует целый ряд способов осуществить такую трансляцию, например:

- Захват изображения и его потоковое вещание в сеть (VLC, Skype);
- Предоставление доступа к рабочему столу для просмотра или управления (VNC, TeamViewer);
- Предварительная запись действий в видеофайл и
 - размещение на видеохостинге (YouTube, Vimeo);
 - размещение на собственном хостинге или в облачном сервисе (Dropbox, Google Drive).

Первые два варианта подразумевают в некотором роде “прямой эфир”, и это, конечно, сильно ограничивает сферу их применения. Поэтому остановимся на последнем варианте. Длительное время главным форматом для хранения медиа (аудио, видео, анимации) служил SWF. Вот пример инструкции по пользованию eBay, упомянутой выше, размещенной в виде флэш-ролика в Dropbox:

`https://dl.dropboxusercontent.com/u/16059006/welcome_to_ebay.swf или j.mp/ebayswf`

Инструкция носит рекламный характер, хотя ее хронометраж больше, чем у стандартного рекламного ролика. Современный посетитель сайта достаточно быстро покидает ресурс, если оказался не заинтересован, поэтому материал

И.Б. Государев,
РГПУ им. А.И. Герцена,
доцент кафедры
информационных и
коммуникационных
технологий, кандидат
педагогических наук,
доцент кафедры
инновационных
образовательных
технологий СПбАПО,
Санкт-Петербург

должен быть довольно динамичным. С другой стороны, скорость изложения не должна быть слишком высокой. Но эти ограничения свойственны коммерческому сектору, в котором нужно привлекать посетителя, заинтересовывать и мотивировать его.

Для проигрывания таких файлов необходим Adobe Flash Player. На момент написания статьи браузер Google Chrome поставляется со встроенным Flash Player, для остальных браузеров нужна установка, которая требует прав администратора. Но в любом случае проприетарный формат SWF (хотя спецификация его и открыта) неуклонно уступает свои позиции новому способу размещения медиа в Интернете — HTML5.

Это в первую очередь связано с лавинообразным увеличением доли просмотров сайтов с мобильных устройств (планшетов и смартфонов), для которых Flash не лицензирован.

До недавнего времени формат Flash был основным для видеохостинга YouTube. В настоящее время часть видеороликов может быть просмотрена без поддержки Flash в связи с переходом на форматы HTML5. Пример такого ролика (скринкаст по работе с интернет-магазином Ozon.ru):

<http://www.youtube.com/watch?v=p8g2Yyq4VUY>

Но размещение на YouTube (и других видеохостингах) связано со следующими проблемами:

1. Масштабирование ролика к стандартным размерам (что влечет потерю качества изображения, которое к тому же перекодируется при выгрузке на видеохостинг);
2. Невозможность легальной загрузки ролика для офлайнного просмотра;
3. Возможная фильтрация контента в образовательном учреждении;
4. Показ непредсказуемого рекламного контента;
5. Непредсказуемость исчезновения роликов или их блокировки самим видеохостингом.

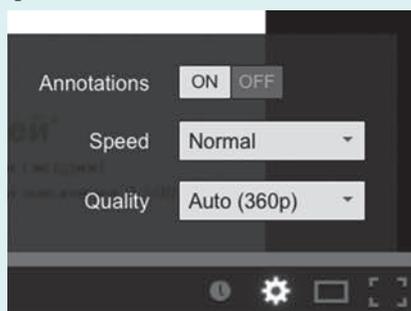


Рис. 1. Настройки плеера YouTube

Поэтому мы будем рассматривать скринкасты, которые создаются для размещения в форматах HTML5 на собственном хостинге или на облачном сервисе. Учитывая, что в России доступность некоторых зарубежных ресурсов может оказаться под вопросом (особенно в образовательных учреждениях), пока компромиссным вариантом можно считать такие “нейтральные” сервисы, как Dropbox.

Всякая инструкция — это компонент некоторого процесса обучения. Само по себе электронное, а также мобильное обучение (рассмотренное нами с позиций обучения информатике в статье [1]) уходит корнями в бизнес, в корпоративное обучение работников фирм. Но в мире веб-ресурсов, которые глубоко проникли в школьное образование, границы между коммерческим и некоммерческим стираются:

1. Интерфейсы веб-ресурсов меняются настолько быстро, что запись видеoinструкций становится для учителя информатики гораздо более дешевым процессом, чем создание текстовых инструкций;
2. Образование рассматривается как услуга, а образовательные учреждения конкурируют между собой за клиентов и нормативно-подушевое финансирование.

Нужно отметить, что использование ЭОР не является прерогативой учителя информатики. Все учителя в настоящее время имеют возможность проходить повышение квалификации, в процессе которого научиться создавать современные ЭОР, включая скринкасты. Более того, создание анимированных роликов — это уровень владения ИКТ учащимися 4-го класса (что следует из современных ФГОС НОО). Но учитель информатики все же пока находится в привилегированном положении, так как в его распоряжении технологии, инструменты и большой опыт. Это, по нашему мнению, может помочь создавать более качественные, конкурентоспособные продукты и в дальнейшем, возможно, получать прибыль.

Еще раз остановимся на общих моментах, а также некоторых противоречиях.

С одной стороны, гораздо эффективнее наглядно продемонстрировать те или иные действия, чем заставлять пользователя долго искать ответ на вопрос в тексте, но с другой стороны, некоторым пользователям может быть проще и быстрее открыть и прочитать текст (особенно небольшой), так как для восприятия аудиовидеоряда нужна продолжительная концентрация внимания, обусловленная его насыщенностью.

С одной стороны, писать текстовые инструкции долго, но с другой — подготовка качественного видеоматериала требует очевидных временных затрат.

Наконец, текстовые инструкции достаточно безличны, а голосовое сопровождение скринкаста явно персонализирует материал (а тем более — изображение автора); это накладывает дополнительные требования к качеству.

Способ подачи материала, как следует из примеров, в эстетическом плане схож с манерой, присущей рекламным роликам: энергично, динамично, достаточно неформально. Чтобы создать привлекательный, мотивирующий продукт, нужно позаботиться об обстановке, хорошем микрофоне, проверить, как звучит голос, если слушать аудио-

дорожку в наушниках и через колонки. Если планируется показывать изображение с веб-камеры, то дополнительно нужна настройка освещения и, возможно, подготовка внешнего вида. Поэтому столь важно подобрать технологию, позволяющую минимизировать затраты. Если выбор удачен, то, отладив процесс единожды, можно воспроизводить технологический цикл многократно.

Для решения каких дидактических задач эффективнее всего использовать скринкасты? Логично предположить, что прежде всего речь идет о репродуктивной деятельности по алгоритму. Вместо того чтобы повторять одно и то же четыре урока подряд, для четырех разных классов или групп, учитель записывает последовательность действий один раз, после чего ее можно продемонстрировать с помощью проектора. Далее — индивидуальный образовательный маршрут, каждый может воспроизвести ролик нужное количество раз (в том числе дома), ставя на паузу в желаемые моменты. Но если учесть, что скринкаст — просто запись происходящего на экране, то тогда диапазон применения можно расширить. Это может быть любое содержание, тезисно изложенное в виде слайдов презентации. Учитель перелистывает слайды, озвучивая их так же, как это происходило бы при демонстрации презентации на уроке.

Понятно, что отдельный скринкаст может быть включен в деятельность на уроке или вне урока разными способами, но для систематического решения дидактических задач он должен входить в некоторый ЭУМК или, как мы предлагали в статье [1], в состав мобильной электронной информационной образовательной среды. К примеру, при изучении темы “Системы счисления” такая ЭОИС включает:

- ЭОР, лекционные или интерактивные, а также электронные версии учебников, например, book.kbsu.ru/theory/chapter4/1_4_6.html;
- Скринкасты, демонстрирующие алгоритмы перевода, например, j.mp/2bin-1 или j.mp/2bin-xl-1, краткие текстовые пояснения и задания к ним;
- Электронные таблицы в Google Drive, содержащие образцы перевода чисел с помощью формул, например, j.mp/2bin-xl-2;
- Онлайн-калькулятор для проверки правильности перевода, например, numsys.ru или baseconvert.com.

Скринкасты могут быть яркими и простыми в исполнении (как на *рис. 2*) и показывать последовательные арифметические действия в текстовом редакторе. Но они в любом случае должны сопровождаться другими дидактическими материалами. Репродуктивным является начальный уровень работы обучающегося со скринкастом. Записанные в скринкасте действия должны быть спланированы так, чтобы допускать некоторое количество степеней свободы. Например, если мы записываем скринкаст по переводу числа в какую-либо систему счисления, то показываем в нем один способ с ого-

воркой о существовании альтернативного способа. Одним из прилагаемых заданий будет найти этот способ. Как можно гарантировать, что он найден? Ведь по конечному результату нельзя точно судить, каким способом он получен. Гарантию, в большей или меньшей степени, можно получить, если еще одним заданием будет запись собственного, учебного скринкаста.

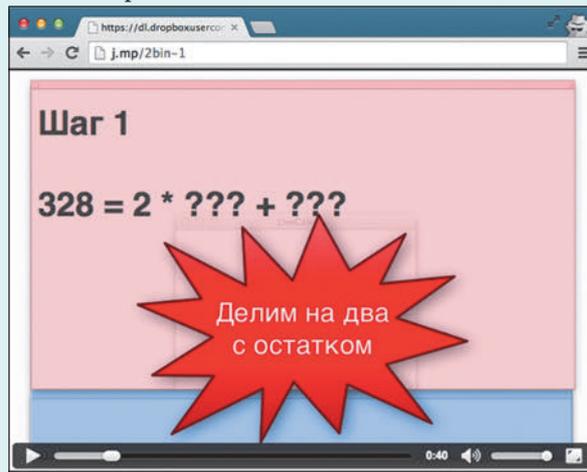


Рис. 2. Скринкаст, обучающий переводу в двоичную систему счисления

Таким образом, сначала обучающиеся знакомятся со скринкастом учителя, затем выполняют задания, а затем создают собственные скринкасты и размещают их в своих портфолио (например, на блоге, сайте, Дневник.Ру или другом ресурсе) (см. *рис. 3* на с. 45).

Можно предложить следующую общую схему разработки скринкаста:

1. Выбор задачи, целеполагание;
2. Отбор содержания и формулирование заданий к нему;
3. Выбор программного средства для создания скринкаста;
4. Схематичное описание демонстрируемых действий (блок-схема, сценарий с включением производимого текста);
5. Репетиция демонстрируемых действий, настройка оборудования;
6. Выбор области записи и осуществление записи;
7. Кодирование в нужные форматы;
8. Пост-продакшн (монтаж, наложение звука, переозвучение и т.п.);
9. Публикация скринкаста в Интернете и/или локальной сети;
10. Переход к учебной деятельности (*рис. 3*).

2. Специальные сведения

В зависимости от выбранного программного средства некоторые из рассмотренных этапов разработки скринкаста автоматизируются. При выборе средства следует учитывать степень его открытости и кроссплатформенности. Например, CamStudio (camstudio.org) является открытым, но не

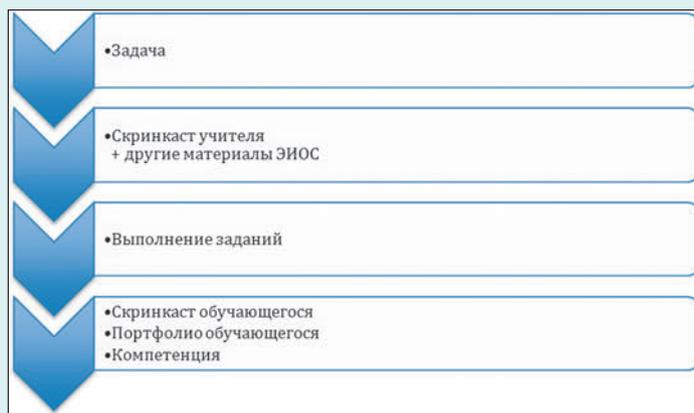


Рис. 3. Этапы деятельности при использовании скринкаста

кроссплатформенным средством. Оно производит ролики в формате AVI, которые нужно затем подвергать процедуре пост-продакшн с помощью какого-либо видеоредактора и самостоятельно кодировать в нужные форматы. Также Camstudio снабжен встроенным SWF-кодером.

Вот пример такого скринкаста:

<https://dl.dropboxusercontent.com/u/16059006/moodle.swf>



Рис. 4. Camstudio

Примером относительно кроссплатформенного (есть версии для Windows и MacOS), но не открытого средства, которое создает ролики в формате SWF, является Jing. В этом случае ролик длительностью до 5 минут является конечным продуктом, никакого постпродакшн с ним быть уже не может.



Рис. 5. Jing

Вот образец такого скринкаста:

https://dl.dropboxusercontent.com/u/16059006/google_form_test.swf

Оба указанных выше средства создают не просто ролик, но еще и некоторый плеер к нему. Однако для обоих характерны те или иные неудобства. Например, в случае CamStudio нужно точно фиксировать ширину и высоту ролика, чтобы указать их верно при встраивании в веб-страницу. Иначе (а также при просмотре прямо в окне браузера) происходит масштабирование с сильной потерей качества. Пример кода для вставки флэш-скринкаста в веб-страницу:

<http://h.dist-learn.spb.ru/?v=4>.

Как это часто бывает, наиболее качественным продуктом является коммерческий. Одним из самых ярких представителей семейства программ, совмещающих запись скринкастов с многофункциональным редактированием, является Camtasia. Это средство позволяет создать и опубликовать скринкаст, который гарантированно сможет быть воспроизведен как в браузерах с поддержкой Flash, так и в браузерах с поддержкой HTML5 video.

Исходя из того, что финансовые возможности учителя, вероятнее всего, ограничены, мы рассматриваем здесь бесплатную программу, которая тем не менее сочетает в себе достоинства Jing и Camstudio. Она называется **Screencast-o-matic**.

<http://www.screencast-o-matic.com/>

Было бы странно, если бы статья, посвященная скринкастам, содержала в себе просто текстовое описание программы, предназначенной для создания скринкастов ☺.

По следующему адресу автор разместил скринкаст, посвященный работе с программой Screencast-o-matic и дальнейшему кодированию видеоролика: <http://dist-learn.spb.ru/screencast/>

Для иллюстрации этой статьи последовательными скриншотами (мгновенными снимками экрана) проще всего было бы открыть этот ролик в каком-либо плеере (VLC Player, MPlayer) и сделать мгновенный снимок из текущего поставленного на паузу кадра (Snapshot, Frame) — см. рис. 6 на с. 46.

На этом кадре виден интерфейс программы в нижней части окна — кнопки записи, микрофона, таймера, полноэкранного режима, веб-камеры, рестарта, завершения работы. В целом интерфейс интуитивно понятен. Программа добавляет в бесплатном режиме едва заметные водяные знаки в углу окна.

После завершения записи можно выбрать нужный формат файла (кодэк) (см. рис. 7).

На момент написания статьи в вебе лидируют два кодека: H.264 и VP8 [2] (хотя уже существуют H.265 и VP9, см. [3]). Кодеку VP8 является (с мая 2010 г.) открытым ПО. Этим кодекам соответствуют форматы mp4 и webm (контейнер Matroska, звук Vorbis). Чтобы гарантировать воспроизведение видеоролика в любом современном браузере, нужно, чтобы видеоролик был представлен в обоих фор-

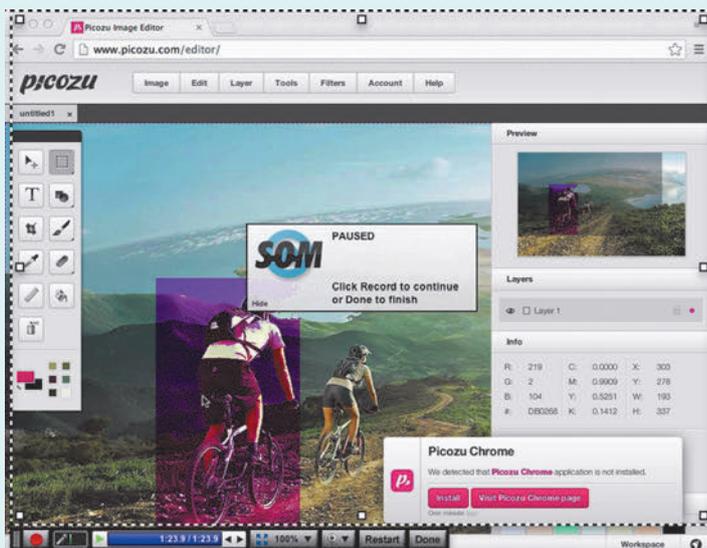


Рис. 6. Запись скринкаста в программе Screencast-o-matic

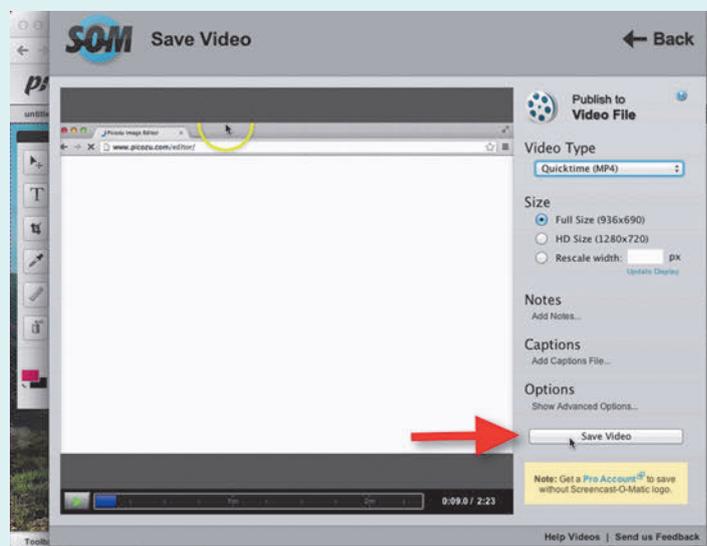


Рис. 7. Выбор формата видеоролика для записанного скринкаста



Рис. 8. Информация о файле mp4 в плеере VLC

матах. Отметим, что пока только браузер **Google Chrome** “понимает” оба формата (см. рис. 8).

Для перекодирования файла mp4 в формат webm можно воспользоваться бесплатным перекодиров-

щиком **Miro Video Converter** или, например, облачным сервисом **online-convert.com** — он позволяет извлекать файл для кодирования из Dropbox и помещать результат кодирования туда же (см. рис. 9 на с. 47).

Еще желательно создать мгновенный снимок из получившегося скринкаста. Это изображение будет использоваться в качестве заставки (заглушки) в процессе загрузки скринкаста. Строго говоря, заставкой может быть любое изображение, но обычно принято использовать именно один из кадров ролика. Таким образом, в итоге получится три файла. Если мы будем исходить из того, что они размещаются в Dropbox, то необходимо получить правильные публичные ссылки. Dropbox создает ссылки вида <https://www.dropbox.com/s/2y35u1xj01m8sjt/h1.mp4>

При попытке перехода по такой ссылке в окне браузера будет открыт интерфейс сайта [dropbox.com](https://www.dropbox.com), в который встроен ролик. Для некоторых примитивных целей этого в принципе достаточно. Но если мы хотим встроить ролик в свой собственный сайт или блог, то необходимо внести простую коррективу — заменить `www` в адресе файла на `dl`.

По адресу <http://h.dist-learn.spb.ru/5> автор разместил готовый к встраиванию пример кода HTML5 для вставки видео в двух форматах. (При ознакомлении с кодом выключите кнопку “Доска”!) Обратите внимание на атрибут `poster` тега `video` — он содержит адрес картинки-заставки (см. рис. 10).

Повторим этот код еще раз:

```
<video controls='controls'
autobuffer='autobuffer'
poster='https://dl.dropbox.com/s/
3uwk5n0gcvruick/h1.png'>
<source src='https://dl.dropbox.com/
s/2y35u1xj01m8sjt/
h1.mp4' type='video/mp4;
codecs="avc1.42E01E, mp4a.40.2" />
<source src='https://dl.dropbox.com/
s/o6ey5jdmuczwn7/h1.webm'
type='video/webm; codecs="vp8,
vorbis" />
</video>
```

Тег `video` отобразит ролик в том разрешении, в котором он записан. Однако при встраивании в блог может понадобиться уменьшить визуальные размеры, используя атрибуты тега `video` `width` и `height`. Но, разумеется, оптимальным решением будет перекодирование ролика с новым разрешением.

Процесс встраивания в блог подробно рассмотрен в еще одном скринкасте:

<http://dist-learn.spb.ru/video/>

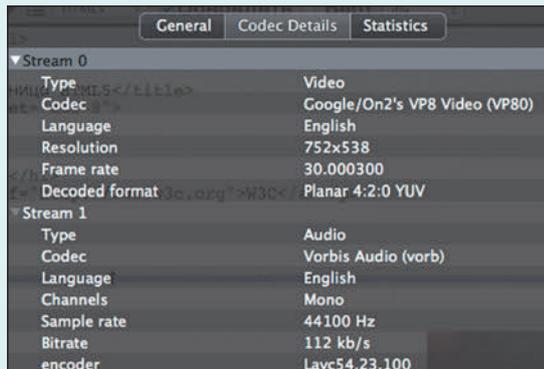


Рис. 9. Информация о файле webm в плеере VLC

Литературные и интернет-источники

1. Государев И.Б. Информатика в облаках // Информатика, № 6, 2013.

2. Покадровое сравнение H.264 и VP8 [Электронный ресурс] URL: <http://habrahabr.ru/post/94394/> (дата обращения: 01.01.2014).

3. Видеокодек VP9 от Google [Электронный ресурс] URL: <http://habrahabr.ru/post/208078/> (дата обращения: 01.01.2014).

АКАДЕМИЯ POLYMEDIA

передовые технологии — продвинутые учителя

Дистанционные курсы и вебинары по созданию интерактивных уроков в ActivInspire и SMART Notebook

очень
давно давно недавно сейчас

POLYMEDIA
8 495 956-85-81 tc@polymedia.ru

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>HTML5 Video</title>
5 <meta charset="utf-8">
6 </head>
7 <body>
8 <article>
9 <h1>HTML5 Video</h1>
10 <video controls="controls" autobuffer="autobuffer" poster="https://dl.dropbox.com/s/0qyn7bocisfibw/h1.png">
11 <source src="https://dl.dropbox.com/s/plzu9b5kdh5muh9/h1.mp4" type="video/mp4; codecs="avc1.42E01E, mp4a.40.2" />
12 <source src="https://dl.dropbox.com/s/5rs4lmqzfxlrx5/h1.webm" type="video/webm; codecs="vp8, vorbis" />
13 </video>
14 </article>
15 </body>
16 </html>

7 <script>
8 $(function(){
9   $("h1").on("click",function(){
10     $.get("time.php", function(result){
11       $("p").html(result);
12     })
13   })
14 })
15 </script>
16 </head>
17 <body>
18 <h1>Нажмите на заголовок страницы!</h1>
19 <p></p>
20 </body>
21 </html>

```

Sun, 05 Jan 2014 17:57:27 +0400

1:56

Рис. 10. Код для вставки HTML5 video в онлайн-редакторе



ШКОЛА ПРОГРАММИРОВАНИЯ

Когда человек хочет передвинуть гору, он начинает с того, что убирает маленькие камни.

Задача “Рыбаки и рыбки”

В одном из предыдущих выпусков “В мир информатики” [1] был опубликован ряд материалов, связанных с использованием в программировании вспомогательных процедур и функций. Рассмотрим еще один пример, иллюстрирующий преимущества применения функций, в частности, логических (возвращающих результат логического типа).

Вот задача, которую придумал известный английский физик Поль Дирак.

“Три рыбака легли спать, не поделив улова. Проснувшийся ночью первый рыбак решил уйти, взяв свою долю. Но число рыб не делилось на три. Тогда он выбросил одну рыбу, а из числа оставшихся забрал треть. Второй и третий рыбаки поступили аналогично (выбросили по одной рыбе и взяли треть из оставшихся). Спрашивается, какое наименьшее количество рыб может удовлетворить условию задачи?”

Поль Дирак был мастер давать различным существительным приставку “анти” — античастица, например. И в этой задаче он, по-видимому, не изменил своей привычке, оригинально решив ее, — ответом является значение “минус две рыбы”. Выбрасываем одну — получаем минус три, забираем треть — останется минус две и т.д.

Попробуем решить задачу, не допуская возможности ловли “антирыб”, то есть найдем наименьшее *положительное* число, удовлетворяющее условию приведенной задачи. Правда, при этом сделаем небольшое уточнение — число рыб, доставшееся каждому рыбаку, не обязательно должно быть одинаковым, как в решении Дирака.

Сначала рассмотрим такую задачу: “Имеется некоторое количество рыб. Определить, возможен ли

дележ рыб между тремя рыбаками в соответствии с условием задачи Дирака” (очевидно, что не при любом числе возможен такой дележ — убедитесь в этом!).

Идея решения такая. Нужно проверять, кратно ли трем число рыб, остающееся после выбрасывания одной рыбы каждым из трех рыбаков. Если для какого-то рыбака ответ отрицательный, то дальнейшие действия, естественно, надо прекратить (используем оператор цикла с условием, в котором учтем, что всего рыбаков — 3).

В программе решения задачи (как принято в нашем издании — на школьном алгоритмическом языке) используем следующие величины:

Коб — общее количество имеющихся (пойманных) рыб;

Кос — количество рыб, оставшееся тому или иному рыбаку;

взял — количество рыб, которые взял тот или иной рыбак;

i — номер рыбака;

можно_делить — величина логического типа, определяющая возможность дележа (взятия каждым рыбаком трети оставшегося количества рыб).

Соответствующая программа имеет вид:

```

алг Проверка
нач цел Коб, Кос, взял, i,
лог можно_делить
вывод нс, "Введите количество рыб"
ввод Коб
Кос := Коб | Сначала так
i := 1
нц | Цикл "действий" каждого рыбака
Кос := Кос - 1 | Осталось после
| выбрасывания одной рыбы
если mod(Кос, 3) = 0
то | i-й рыбак может взять треть
| оставшихся рыб
можно_делить := да
взял := div(Кос, 3) | Берет
| i-й рыбак
Кос := Кос - взял | Оставшееся
| количество рыб

```

```

    иначе
        можно_делить := нет
|При Коб рыбах дележ невозможен
все
    i := i + 1 |Проснулся следующий
                |рыбак (☺)
кц при i > 3 или не можно_делить
|В зависимости от значения величины
|можно_делить
|выводим ответ
вывод нс, "При таком количестве рыб дележ "
если можно_делить
    то
        вывод "возможен"
    иначе
        вывод "невозможен"
все
кон

```

— где *div* — функция, возвращающая результат целочисленного деления своего первого аргумента на второй, а *mod* — функция, рассчитывающая остаток (в других языках программирования для этого используются не функции, а специальные операции).

После этого программа нахождения минимального количества рыб, удовлетворяющего условию задачи Дирака, может быть оформлена так:

```

алг Задача_Рыбаки_и_рыбки
нач цел Коб, Кос, взял, i,
    лог можно_делить
    Коб := 7 |Начальное значение
                |диапазона поиска
|Проверяем, можно ли разделить Коб рыб
| (см. выше)
нц
    Кос := Коб
    i := 1
    нц |Цикл "действий" каждого рыбака
        Кос := Кос - 1 |Осталось после
                        |выбрасывания
        ...
        i := i + 1 |Проснулся следующий рыбак
    кц при i > 3 или не можно_делить
        Коб := Коб + 1 |Очередное значение
    кц при можно_делить
    вывод нс, "Наименьшее количество рыб,"
вывод "удовлетворяющее условию задачи:",
    Коб

```

кон

Читаемость и понятность всей программы может быть существенно улучшена, если создать вспомогательную функцию *Можно_делить(Коб)* логического типа, определяющую возможность дележа *Коб* рыб в соответствии с условием задачи. Ее можно составить на основе программы, приведенной первой:

```

алг лог Можно_делить (арг цел Коб)
нач цел Кос, взял, i, лог можно_делить
    Кос := Коб
    i := 1

```

```

нц
    Кос := Кос - 1
если mod(Кос, 3) = 0
    то
        можно_делить := да
        взял := div(Кос, 3)
        Кос := Кос - взял
    иначе
        можно_делить := нет
все
    i := i + 1
кц при i > 3 или не можно_делить
|Значение функции:
знач := можно_делить
кон

```

Примечание. В школьном алгоритмическом языке регистр букв в именах учитывается, поэтому в функции *Можно_делить* можно использовать переменную величину логического типа *можно_делить*.

С использованием этой функции наша “главная” программа может быть оформлена очень кратко:

```

алг Задача_Рыбаки_и_рыбки
нач цел Коб
    Коб := 7 |Начальное значение
                |диапазона поиска
нц пока не Можно_делить(Коб)
    Коб := Коб + 1 |Очередное значение
кц
вывод нс, "Наименьшее количество рыб,"
вывод "удовлетворяющее условию задачи:",
    Коб

```

кон

Есть и другой способ решения задачи. Можно идти не от общего количества пойманных рыб, а от числа рыб, доставшихся третьему рыбаку (то есть искать, так сказать, “с конца”). Если эту величину обозначить *взял3*, а количество рыб, оставшееся тому или иному рыбаку, — соответственно, *К3*, *К2* и *К1*, то, перебирая значения *взял3*, равные 1, 2, 3, ..., можно найти такое минимальное число, при котором значения величин *К2* и *К1* есть целые числа. Соответствующий вариант программы разработайте самостоятельно.

Другие задания для самостоятельной работы

1. Разработав программы (на языке программирования, который вы изучаете), найдите значение, о котором идет речь в начале статьи, а также следующее большее число.

2. Разработайте аналогичные программы для случая, когда рыбаков — 4, и также найдите соответствующие значения.

Программы и ответы, пожалуйста, присылайте в редакцию.

Литература

1. Златопольский Д.М. Об использовании процедур и функций. / “В мир информатики” № 192 (“Информатика” № 11/2013).

Решаем задачу “Рыбаки и рыбки” с помощью электронной таблицы

Опишем методику решения задачи “Рыбаки и рыбки” средствами электронной таблицы на примере программы Microsoft Excel.

Оформим верхнюю часть листа следующим образом:

	A	B	C	D	E	F	G	H	I	J
1	Число рыб	-1	Взял 1-й	Осталось	-1	Взял 2-й	Осталось	-1	Взял 3-й	Проверка
2	7	6	2	4	3	1	2	1	0,333333	
...										

В столбце A запишем возможные значения общего числа пойманных рыб (допустим, что оно не превышает 300). Эти значения можно получить, используя, например, автозаполнение.

Формулы в ячейках B2:I2 достаточно очевидны. А как проверить тот или иной вариант (столбец J)? Можно рассуждать так. Данный вариант исходного количества рыб подходит, если в столбце I будет записано целое число. Это условие можно формализовать следующим образом:

$$I2=ЦЕЛОЕ(I2)$$

Если оно соблюдается, то выведем в ячейке J2 соответствующее количество рыб (из ячейки A2), в противном случае — “пустое” значение (“”). Вся формула в ячейке J2 имеет вид:

$$=ЕСЛИ(I2=ЦЕЛОЕ(I2);A2;"")$$

Все формулы диапазона B2:J2 можно распространить (скопировать) на другие строки:

	A	B	C	D	E	F	G	H	I	J
1	Число рыб	-1	Взял 1-й	Осталось	-1	Взял 2-й	Осталось	-1	Взял 3-й	Проверка
2	7	6	2	4	3	1	2	1	0,333333	
...										
295	300	299	99,67	199,33	198,33	66,11	132,22	131,22	43,74	
296	Ответ:									

После этого искомое количество в ячейке B296 может быть получено как минимальное числовое значение в диапазоне J2:J295 (с использованием функции МИН):

$$=МИН(J2:J295)$$

Задания для самостоятельной работы

1. Подготовив лист электронной таблицы, найдите значение, о котором идет речь в начале статьи, а также следующее большее число.
2. Подготовьте лист для случая, когда рыбаков — 4, и также найдите соответствующие значения. Ответы (а не листы), пожалуйста, присылайте в редакцию.

ЯПОНСКИЙ УГОЛОК

Два sudoku

Решите, пожалуйста, две японские головоломки “судоку”:

1) простую:

		5			7	8		
				3	2	9		
4			9					
2				1			8	9
			7		6			
1	4			2				3
					1			2
			7	2	5			
		8	6			7		



2) сложную:

				8	2			9
		5						1
					6			
5			1					4
	6			7				1
1		8			4		3	5
	5	3	7	6				
7	2						8	3
							7	6

Ответы (можно не на все головоломки) присылайте в редакцию.

Пять задач на системы счисления

Е.А. Мирончик, учитель информатики
МБНОУ “Лицей № 111”,
г. Новокузнецк Кемеровской обл.

1. Выпишите числа 2^{n-1} , $2^n + 1$, $2^n - 3$, где $n > 10$, в порядке возрастания количества единиц в их двоичной записи.

2. Каждую строку черно-белого рисунка размером $n \times n$ пикселей закодировали следующим образом: пиксель черного цвета обозначили единицей, белого — нулем и соответствующее число представили в восьмеричной системе счисления. Весь рисунок при этом кодируется последовательностью восьмеричных чисел: 234, 242, 242, 202, 204, 210, 0, 210.

Что изображено на рисунке?

Аналогично можно этот рисунок закодировать в шестнадцатеричной системе счисления. Перекодируйте рисунок соответствующим образом.

3. Два некоторых десятичных числа X и Y перевели в системы счисления с основаниями 16 и 8. Часть символов при записи была утеряна. Два из четырех полученных чисел имеют вид (позиции утерянных символов обозначены символом “*”):

$$F_{16}^* \text{ и } 33_8^*$$

Можно ли сделать вывод о том, какое из чисел X и Y больше, или о равенстве этих чисел?

4. Некоторое десятичное число X перевели в системы счисления с основаниями 16, 8, 4. Часть символов при записи была утеряна. Позиции утерянных символов обозначены символом “*”:

$$X = *7_{16}^* = 5*6_8 = ***1_4^*$$

Восстановите все числа и определите число X .

5. Некоторое десятичное число X перевели в системы счисления с основаниями 16 и 8. Часть символов при записи утеряна (позиции утерянных символов обозначены символом “*”):

$$3*9_{16} \text{ и } 1**_8$$

Можно ли однозначно определить значение числа X ? Если нет, то укажите все возможные значения числа.

Куклы

У одной девочки-инопланетянки было 21 разных кукол. Она отдала подруге 12 кукол, после чего у нее осталось 3 куклы. Как такое могло быть?

Очередь в музей

Митя, Сережа, Толя, Юра и Костя пришли в музей до открытия и встали в очередь. Если бы Митя встал посередине очереди, то он стоял бы между

Сережей и Костей, причем Сережа стоял бы впереди Мити, а если бы Митя встал в конец очереди, то рядом с ним стоял бы Юра. В результате Митя встал впереди своих товарищей. В каком порядке ребята встали в очередь?

Дома в Зеленом городе¹

В Зеленом городе 8000 домов, из них 3500 деревянных, остальные — каменные. 2800 домов стоят на левом берегу единственной реки этого города. На карту нанесли 6000 домов (все деревянные дома и все дома на правом берегу реки). Можно ли узнать, сколько деревянных домов на каждом берегу реки и сколько каменных — на правом?

Поход

Группа туристов во время похода частенько мокла под дождем. Пятнадцать дней то утром, то вечером небо заволакивали тучи и полдня лил жесточайший ливень. Но участники похода подметили закономерность: каждый раз, когда дождило с утра, к полудню ветер обязательно разгонял тучи, и наступала чудесная погода. Шестнадцать раз было безоблачное утро. Не могут забыть походники и семнадцать тихих, ясных вечеров. Сколько дней длился поход?

Котлеты

Найдите минимальное время в секундах, за которое можно пожарить 7 котлет, если на сковороде помещается 6 котлет, и с каждой стороны котлету нужно жарить 5 минут. Ответ обоснуйте.



Источник двух последних задач — сайт diafant.ru.

¹ Задачу предложила Е.А. Мирончик, учитель информатики МБНОУ “Лицей № 111”, г. Новокузнецк Кемеровской обл.

Ответы, решения, разъяснения к заданиям, опубликованным в разделе “В мир информатики” ранее

Система числовых ребусов

Напомним, что предлагалось решить систему из двух числовых ребусов:

$$\begin{cases} \text{МА} \cdot \text{МА} = \text{МИР} \\ \text{АМ} \cdot \text{АМ} = \text{РИМ} \end{cases}$$

— в которых, как обычно в таких головоломках, одинаковым буквам соответствуют одинаковые цифры, разным буквам — разные цифры.

Решение

В первом ребусе квадрат двузначного числа **МА** начинается на ту же цифру, что и само число. Это может быть только при $M = 1$, и при этом $A = 2$ или $A = 3$, или $A = 4$. Но из второго ребуса следует, что $A \leq 3$, иначе при возведении в квадрат двузначного числа получится четырехзначное число. Значит, $A = 2$ или $A = 3$.

Если $A = 2$, то в первом ребусе получаем $12 \cdot 12 = 144$, то есть $I = M$, что недопустимо. В случае $A = 3$ получаем ответ:

$$13 \cdot 13 = 169;$$

$$31 \cdot 31 = 961.$$

Правильные ответы представили:

— Аксененко Ирина и Чумаков Илья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Андрищенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Булькин Алексей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Воскресенский Денис, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Гаязова Ильсеяр, Нуретдинов Айдар и Хазиев Линар, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Довгань Алексей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Двойников Елизар, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Ефанов Артем, г. Ярославль, школа № 33, учитель **Цикина Е.Н.**;

— Иванова Виолетта и Левченко Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Истомина Полина, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Копестинская Наталья и Коптева Алена, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Торопов Александр, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Странная таблица умножения

Напомним, что требовалось найти закономерность и поставить вместо знака вопроса нужное число:

$$1 \times 1 = 2$$

$$2 \times 2 = 3$$

$$3 \times 3 = 2$$

$$4 \times 4 = 3$$

$$5 \times 5 = 3$$

$$6 \times 6 = 4$$

$$7 \times 7 = ?$$

Ответ

После знаков равенства стоят числа, равные количеству слогов в названии числа-произведения:

$$1 \times 1 = 1 \text{ (один — 2 слога)}$$

$$2 \times 2 = 3 \text{ (четыре — 3 слога)}$$

...

$$7 \times 7 = 49 \text{ (сорок девять — 4 слога)}$$

Ответы представили:

— Аксененко Ирина и Чумаков Илья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Андреева Елена и Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Андрищенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Березин Василий и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Булькин Алексей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Воскресенский Денис, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Довгань Алексей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Иванова Виолетта и Левченко Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Удалова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

Задача “Четыре подружки”

Напомним, что требовалось определить, в каком классе (5-м “А”, 5-м “Б” и 6-м “А”) учится каждая из

девочек — Маша, Даша, Катя и Оля, если известно, что Маша и Катя учатся в классах с одинаковыми буквами. Катя и Оля — одноклассницы. Маша и Даша — ученицы 5-го класса.

Решение

Так как Катя и Оля — одноклассницы, а Маша и Катя учатся в классах с одинаковыми буквами, то речь может идти о классах с буквой “А” (их в условии — два). О Маше известно, что она — пятиклассница, то есть она учится в 5-м “А”. Тогда Катя и Оля — ученицы 6-го класса (а именно — 6-го “А”). “Оставшаяся”, четвертая, девочка — Даша — учится в классе с буквой “Б” (5-м “Б”).

Правильные ответы прислали:

— Алехина Елена, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Бурмантова Юлия и Епанешникова Полина, Совхозная средняя школа, Московская обл., Серебряно-Прудский р-н, поселок Успенский, учитель **Жарикова Е.Н.**;

— Василенко Степан, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Кротов Олег и Миноцкий Ян, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Мусатов Тимофей, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Назарова Алена, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Непомнящая Анна, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Нуретдинов Айдар, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Озерова Инна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Труханова Яна, Республика Карелия, г. Сегежа, школа № 5, учитель **Меньшиков В.В.** (Яна представила презентацию Microsoft PowerPoint, иллюстрирующую решение задачи);

— Фатова Любовь и Чумакова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Штехерт Виктория, г. Рязань, школа № 44, учитель **Марцинкевич Е.Е.**

Викторина “Рыба, триатлон и динозавры”

Ответы прислали:

— Андреева Елена и Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Баков Анатолий и Макаренченко Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Буравцова Елизавета, Вышегородских Анастасия, Гнездилова Любовь, Карасева Екатерина и Серадская Дарья, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Волчкова Екатерина, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Воскресенский Денис и Кротов Олег, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Гаязова Ильсеяр, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Глотова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Гусева Маргарита, Салимов Владислав и Туртаева Айзада, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Дудка Мария, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Заева Кристина, Республика Башкортостан, г. Уфа, школа № 54 (Центр дистанционного обучения), учитель **Искандарова А.Р.**;

— Иванова Виолетта и Левченко Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Соловьева Марина и Трештау Татьяна, Вадковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

— Телегина Вера, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Задача “Игрушки”

Напомним условие: “У одного мальчика-инопланетянина было 12 разных игрушек. Когда ему подарили еще 5, то общее число игрушек стало равно 21. Как такое могло быть?”

Ответ

Это могло быть, если все значения записаны в шестеричной системе счисления ($12_6 = 8_{10}$, $5_6 = 5_{10}$, $21_6 = 13_{10}$).

Ответы прислали:

— Аксененко Ирина, Волков Сергей и Чумаков Илья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Волчкова Екатерина, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Дудка Мария, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Соловьева Марина, Вадковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

— Телегина Вера, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Удалова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Хазиев Линар, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Хорькова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Задача “Переливание молока”

Напомним, что требовалось разработать алгоритм решения задачи разделения пополам 14 литров молока, находящихся в бидоне емкостью 14 литров, используя также еще два пустых бидона емкостью 9 и 5 литров.

Решение см. в табл. 1.

Ответы прислали:

— Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Волчкова Екатерина, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Воскресенский Денис, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Истомина Полина, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Карасев Даниил и Чумакова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Кузнецов Семен, средняя школа села Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;

— Леоненко Степан, средняя школа поселка Осинковка, Алтайский край, учитель **Евдокимова А.И.**;

— Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Телегина Вера, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Туртаева Айзада, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Удалова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Хазиев Линар, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Хорькова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Ответы на задачу “Ночное ДТП” прислали:

— Андреева Елена и Михайлова Светлана, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Воскресенский Денис и Миноцкий Ян, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Ефанов Артем, г. Ярославль, школа № 33, учитель **Цикина Е.Н.**;

— Леоненко Степан, средняя школа поселка Осинковка, Алтайский край, учитель **Евдокимова А.И.**

Решение задачи “Два возраста” представили:

— Андреева Елена, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Воскресенский Денис, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

Таблица 1

№	Действие	Бидон емкостью 14 л	Бидон емкостью 9 л	Бидон емкостью 5 л
0	Исходное состояние	14	0	0
1	Перелить из 14 л в 9 л	5	9	0
2	Перелить из 9 л в 5 л	5	4	5
3	Перелить из 5 л в 14 л	10	4	0
4	Перелить из 9 л в 5 л	10	0	4
5	Перелить из 14 л в 9 л	1	9	4
6	Перелить из 9 л в 5 л	1	8	5
7	Перелить из 5 л в 14 л	6	8	0
8	Перелить из 9 л в 5 л	6	3	5
9	Перелить из 5 л в 14 л	11	3	0
10	Перелить из 9 л в 5 л	11	0	3
11	Перелить из 14 л в 9 л	2	9	3
12	Перелить из 9 л в 5 л	2	7	5
13	Перелить из 5 л в 14 л	7	7	0

— Зотов Николай, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**
(Ответ: одному человеку 27 лет, другому — 36.)

Программы, предложенные для самостоятельной работы в статье “Задачи о счастливых билетах”, представили:

— Ефанов Артем, г. Ярославль, школа № 33, учитель **Цикина Е.Н.**;

— Яковлев Сергей, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

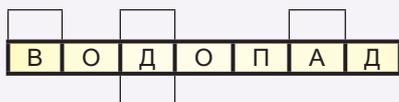
Они же прислали ответ к задаче “27 карт”. Артем и Сергей будут награждены дипломами.

Решение головоломки, связанной с перемещением карточек с буквами таким образом, чтобы из слова “ВОДОПАД” (см. ниже) получилось слово “ПОДВОДА”, прислали:

— Абакумова Лилия, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Андреева Елена, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Удалова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**



Головоломка решается за 42 переключивания.

Елена, Елизавета и Лилия, не испугавшиеся такого числа операций в алгоритме, будут награждены дипломами.

Поздравляем всех награжденных!

Ответы на задания из октябрьского выпуска “В мир информатики” прислали также:

1) учащиеся школы № 11 г. Струнино Владимирской обл. (учитель **Волков Ю.П.**):

— решения ребусов по теме “Программа подготовки презентаций Microsoft PowerPoint” — Кротов Олег, Миноцкий Ян и Тананаева Ксения;

— ответы на пять вопросов (рубрика “Поиск информации”) — Воскресенский Денис и Кротов Олег;

— решение японских головоломок “судоку” — Воскресенский Денис, Миноцкий Ян и Тананаева Ксения;

2) ученицы средней школы поселка Ерофей Павлович, Амурская обл., Сквородинский р-н (учитель **Краснёнкова Л.А.**):

— Абрамова Екатерина (решения японских головоломок);

— Пискунова Полина (решения ребусов);

3) учащиеся школы № 44 г. Рязани (учитель **Марцинкевич Е.Е.**):

— Чухин Павел — решения ребусов;

— Муравьев Иван — решение “судоку”;

4) Бурмантова Юлия, Совхозная средняя школа, Московская обл., Серебряно-Прудский р-н, поселок Успенский (учитель **Жарикова Е.Н.**) — решение кроссворда.

Уже после подготовки списков читателей, пришедших ответы на задания из сентябрьского выпуска “В мир информатики”, были получены ответы учащихся школы № 11 г. Струнино Владимирской обл. (учитель **Волков Ю.П.**):

— на головоломку “Три фигуры” — от Базанова Ильи, Манукяна Григория, Миноцкого Яна и Чурбаевой Регины;

— на задание “Пять вопросов” (рубрика “Для эрудитов”) — от Кротова Олега;

— на задачи о числе 2013 — от Кротова Олега и Чурбаевой Регины;

— на японские головоломки “судоку” — от Кротова Олега и Тананаевой Ксении;

— на задачу “Работники магазина” — от Миноцкого Яна и Чурбаевой Регины.

Ян Миноцкий и Регина Чурбаева, как одни из немногих читателей, правильно решивших непростую задачу “Работники магазина”, будут награждены дипломами. Поздравляем!

Спасибо всем приславшим ответы!

ПОИСК ИНФОРМАЦИИ

Четыре вопроса

Ответы на следующие вопросы найдите в Интернете или по другим источникам информации.

1. Кому принадлежит фраза: “Даже если меня разобьет паралич, я буду критиковать чужую походку”?

2. Каково “подлинное имя” знаменитой Венеры Милосской из Лувра?

3. Какие два архитектора “приложили руку” к созданию одного из семи чудес света?

4. Где перед регистрацией брака молодая пара должна оплатить пошлину крысиными хвостами?

Можно отвечать не на все вопросы.

ПРИЫ

Мечта

Учитель спрашивает ученика: “У тебя есть мечта?”

— Есть.

— Какая?

— Хорошо учиться.

— Так учись!

— А как же жить без мечты?!

Зашифрованный вопрос

Текст шифруется с помощью таблицы: каждой цифре сопоставляется одна из букв, расположенных под ней, а знаку “*” — пробел или одна из букв ю или я. В разных местах зашифрованного текста одна и та же цифра может соответствовать разным буквам.

0	1	2	3	4	5	6	7	8	9	*
А	Г	Ж	Й	М	П	Т	Х	Ш	Ы	Ю
Б	Д	З	К	Н	Р	У	Ц	Щ	Ь	Я
В	Е	И	Л	О	С	Ф	Ч	Ъ	Э	

Расшифруйте следующий текст и ответьте на него:

5343934*150413*6*8156215044414*305041080?

Разлив бензина

В бочке имеется 18 литров бензина. Имеются два ведра объемом по 7 литров, в которые нужно налить по 6 литров бензина. Как это сделать, если есть также черпак объемом 4 литра?

	Действие	Бочка	Ведро 1	Ведро 2
Исходное состояние		18	0	0
1	Налить из бочки черпаком в ведро 2	14	0	4
2	...			

Как получить верное равенство?

На доске написано неверное равенство:



Как провести один прямолинейный отрезок так, чтобы равенство стало верным?

Задание предназначено для учащихся 1–7-х классов.

Источник — журнал “Квант” № 6/2008 (автор задачи — А.Аджанс)

Числовой ребус с ЭКРАНОМ

В приведенном числовом ребусе, как обычно, одинаковыми буквами зашифрованы оди-

наковые цифры, разными буквами — разные цифры.

$$(Э + К + Р + А + Н)^н = ЭКРАН$$

Решите его.

И числовой ребус со звездочками

Решите, пожалуйста, числовой ребус:

$$\begin{array}{r} * * * * * * * * К \\ \times \\ \hline А А А А А А А А А \end{array}$$

в котором А и К — различные числа, а звездочкой (“*”) может быть любая цифра.

Две шкатулки

Имеются две шкатулки, в одной из которых лежит ключ. На первой шкатулке написано: “Ключ находится во второй шкатулке”. На второй написано: “Одно из двух утверждений, записанных на этих шкатулках, истинно, а второе ложно”. В какой из шкатулок находится ключ?

Найти четырехзначное число

В некотором четырехзначном числе переставили цифры, и новое число вычли из первоначального. Получилось число, записанное теми же цифрами. Какое это число?

Все цифры в квадратиках

Впишите в квадратики цифры от 1 до 9 так, чтобы все эти девять цифр были использованы и выполнялись указанные равенства.

$$\square\square\square : \square = \square - \square = \square + \square = \square \times \square$$

Задание предназначено для учащихся 1–7-х классов.

Люстра и переключатель

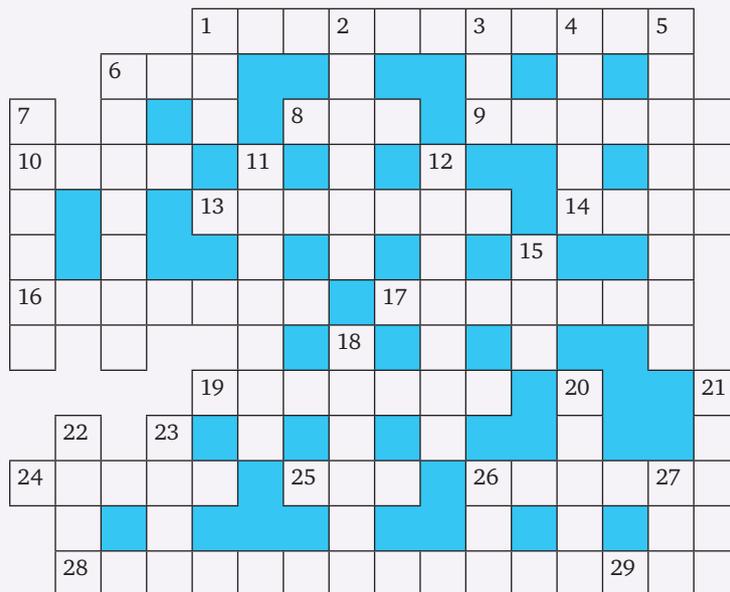
В люстре пять лампочек. Переключатель, управляющий ею, имеет шесть положений, при которых каждый раз должно гореть разное количество лампочек — последовательно от 0 до 5. Однажды часть лампочек перегорела. Какое наименьшее число раз нужно переключить переключатель, чтобы узнать, какие именно?

Кроссворд

Решите, пожалуйста, кроссворд.

По горизонтали

1. В программировании — процесс пошагового выполнения программы, в ходе которого про-



граммист видит последовательность выполнения команд и значения переменных на данном шаге выполнения программы, что позволяет легче обнаруживать ошибки.

6. Цифра восьмеричной системы счисления.

8. Задний план, на котором изображается символ на экране или элементы слайда презентации.

9. Оболочка одного из основных элементов настольного персонального компьютера.

10. Часть света, в которой изготавливаются компьютеры “желтой” сборки.

13. Язык программирования.

14. Несколько соединенных между собой компьютеров.

16. Устройство, отслеживающее адрес выполняемой команды, а также устройство, обеспечивающее учет чего-либо.

17. Результат целочисленного деления.

19. Нестандартное устройство для вывода информации в компьютере.

24. Последовательность цифр.

25. Структура данных — двусторонняя очередь.

26. Древнегреческий математик, автор алгоритма нахождения наибольшего общего делителя двух натуральных чисел.

28. День недели.

29. ...координат.

По вертикали

1. Характеристика величины, определяющая множество ее допустимых значений и применимых к ней операций.

2. Примечание к тексту, размещаемое в нижней части страницы или в конце документа.

3. Направление в популярной музыке.

4. Программа, обладающая способностью к самовоспроизведению.

5. Наука, изучающая звук.

6. Электронная схема, применяемая в регистрах компьютера для запоминания одного бита информации.

7. Комбинированный тип данных.

11. Набор цветов, которые могут использоваться в изображении.

12. Порция из нескольких секторов, выделяемых файлу при необходимости.

15. Величина изменения значения переменной цикла.

18. Номер элемента массива.

20. Алгоритмическая конструкция, обеспечивающая повторение одних и тех же операций.

21. Число в системе условных обозначений символов.

22. Столица государства, в котором программисты получают зарплату в гривнах.

23. Знак арифметической операции.

26. Хвойное дерево.

27. Буква греческого алфавита, которой, как правило, обозначают неизвестную величину.

GAMES.EXE

Сколько лет человеку?

Предложите кому-нибудь умножить число его лет на 2 и к произведению прибавить 4, затем полученную сумму умножить на 5, к этому произведению прибавить 12 и полученную сумму умножить на 10. После объявления результата предложенных арифметических действий вы можете объявить число лет.

Как это сделать?

“Крестики-нолики” — новый вариант

Двое играют в “крестики-нолики” на доске 3×3 по измененным правилам. Каждый на своем ходу может ставить как “крестик”, так и “нолик”. Выигрывает тот, после хода которого образуются три подряд стоящие одинаковых значка (по горизонтали, по вертикали или по диагонали, как в обычных “крестиках-ноликах”). Кто выигрывает в эту игру — начинающий или второй игрок? И как?

ДЛЯ ЭРУДИТОВ

Лучшие изобретения, придуманные женщинами

Этот материал публикуется в честь праздника 8 Марта, с которым редакция поздравляет всех женщин.

1. Астролябия

Прибор для измерения координат небесных тел изобрела женщина-ученый, философ, астроном и математик Гипатия Александрийская в 370 г. до н. э.

2. Розовое шампанское

Николь Барбье Клико в 1808 году разработала технологию “ремюажа”, благодаря которой шампанское за три месяца избавляется от осадка и становится кристально прозрачным. Это возымело настоящий успех, так как значительно улучшилось качество напитка.

3. Циркулярная пила

Табита Бэббитт долго наблюдала за мужчинами, занятыми распилом бревен специальной пилой с двумя ручками, за которые нужно тянуть то вперед, то назад. Хотя нагрузка на обоих мужчин была одинаковой, бревна распиливались только тогда, когда пила двигалась вперед, а при обратном движении с бревном ничего не происходило. Бэббитт подумала, что это пустая трата энергии, и в 1810 году создала прототип циркулярной пилы, которая позднее стала использоваться в лесной промышленности.

4. Перископ для подводных лодок

С помощью этого оптического прибора, который в 1845 году запатентовала Сара Мэтер, можно наблюдать за окрестностями из укрытия (подводной лодки, танка и т.п.).

5. Мясные консервы

В 1873 году на всемирной выставке в Вене наша соотечественница Надежда Кожина продемонстрировала способ приготовления мясных консервов, за что и получила медаль.

6. Посудомоечная машина

Недосчитавшись нескольких тарелок из своего любимого фарфорового сервиза, Джозефина Кокрейн создала машину, которая только моет посуду, а не бьет. Это произошло в 1886 году, но только спустя 40 лет устройство Кокрейн было признано необходимой вещью в хозяйстве.

7. Снегоуборочная машина

Чистота должна быть не только в доме, но и на улицах, решила обычная секретарша Синтия Весто-

вер и собрала прадедушку современных машин для чистки улиц от снега в 1892 году.

8. Стеклоочистители

Первые “дворники” для автомобиля изобрела Мэри Андерсон в 1903 году. Ей стало жалко водителя, который вынужден был во время вьюги по минутно останавливать машину и сгребать снег с ветрового стекла.

9. Фильтр для кофе

Благодаря Мелитте Бенц любители посидеть за чашечкой кофе избавлены от утомительной процедуры по приготовлению порции-другой. Первый фильтр-конус для кофе Мелитта скрутила собственноручно из листка ученической тетради в 1909 году.

10. Глушитель для автомобиля

Этот акустический фильтр изобрела Эль Долорес Джонс в 1917 году, после чего мир вокруг нас стал чуть тише.

11. Подгузники

Усталая мама и домохозяйка Марион Донован в 1917 году села за швейную машинку с душевой занавеской и, после нескольких попыток, создала водонепроницаемые покрытия для подгузника. В отличие от резиновых ползунков, которые уже были на рынке, дизайн Донован не вызывал опрелостей и не жал ребенку.

12. Шоколадное печенье

Во время приготовления печенья по классическому рецепту Рут Уэйкфилд в 1930 году пошла на эксперимент, растопив плитку шоколада и добавив его в тесто. Получилось очень вкусное печенье, которое позже Рут с мужем запустили в серийное производство.

13. Секретная связь

В 1941 году Хэди Ламарр запатентовала секретное средство связи, которое динамически изменяло частоту вещания, чтобы затруднить перехват противником. С 1962 года это устройство использовалось в американских торпедах, а ныне используется в мобильной связи и WiFi.

14. Бронежилет

В 1965 году доктор Стефания Кволек изобрела синтетический материал “кевлар”, который в пять раз прочнее стали. Кевлар стал основой для пуленепробиваемых жилетов. Изобретение Кволек спасло тысячи жизней полицейским, пожарным и военным.

15. Силикон

Скульптор Патрисия Биллингс поставила себе задачу создать такую цементную добавку, которая бы предотвращала ее творения от разрушений.

После нескольких лет экспериментов в 1970 году она наконец достигла своей цели, изобретя нерушимую штукатурку. Вскоре после этого Биллингс обнаружила, что материал был к тому же удивительно устойчив к огню.

Да будет “свет”! — Часть 2²

В перечисленных ниже утверждениях кое-где — чистая правда, кое-где — полная чушь. Распознаете ли, где что?

1. Сериал “Firely” (“Светлячок”) в 2003 году получил премию “Оскар” за лучшие спецэффекты.
2. Первый светофор был установлен в Лондоне возле здания Британского парламента.
3. Актриса Светлана Светличная свою первую роль сыграла в фильме “Им покоряется небо”.

4. Световой год — это расстояние, которое свет проходит в вакууме за один григорианский год.

5. В фильме “Властелин колец” коня Гэндальфа — Святозара играет серый андалузский жеребец Демеро.

6. Первый светодиод был изготовлен в Пенсильванском университете США в 1962 году.

7. Широко использовавшаяся в СССР и пост-СССР синяя лампа имеет научное название “рефлектор Минина”.

8. Название единицы силы света переводится как “свеча”.

Ответы присылайте в редакцию (можно комментировать не все утверждения).

По материалам “Российской газеты”, автор — Татьяна Луговская

КРЕПКИЙ ОРЕШЕК



Ряд задач, опубликованных в августовском и октябрьском выпусках “В мир информатики”, оказались для наших читателей не просто “крепкими орешками”, а “очень крепкими орешками” ☺.

Задача “О полных квадратах”

Напомним, что требовалось ответить на вопрос, существуют ли системы счисления, кроме десятичной, в которых является полным квадратом число:

- | | |
|---------|---------|
| а) 121; | в) 169; |
| б) 144; | г) 196? |

Благодаря Светлану Михайлову, студентку Канашского педагогического колледжа, Чувашская Республика, г. Канаш (преподаватель **Воеводина Р.В.**), ответившую на один из вопросов, приведем решение первой задачи.

Если записать умножение в “столбик”:

$$\begin{array}{r} \times \quad 1 \quad 1 \\ \quad 1 \quad 1 \\ \hline 1 \quad 1 \\ 1 \quad 2 \quad 1 \end{array}$$

то можно увидеть, число 121 является квадратом числа 11 в любой системе счисления с основанием, большим двух (цифра 2 в записи числа-произведения имеется).

Можно также исследовать развернутую запись числа 121:

$$121_p = p^2 + 2p + 1 = (p + 1)^2.$$

Выражение $p + 1$ соответствует свернутой записи числа 11 в p -ичной системе счисления, что по-

зволяет сделать тот же вывод ($121_p = p^2 + 2p + 1 = (p + 1)^2 = (11_p)^2$).

Для решения остальных задач также следует исследовать развернутую запись заданных чисел в p -ичной системе.

Задача “Двоичное число”

Напомним условие: “Про некоторое двоичное число известно, что в нем:

- 1) не более 40 цифр;
- 2) цифры чередуются;
- 3) количество одной из цифр составляет ровно 52% общего числа цифр.

Найдите десятичный эквивалент этого числа”.

Комментарий к решению

Искомое число не может начинаться с нуля, значит, оно имеет вид 1010...0101. По условию, доля единиц в нем — 52%, то есть на 4% больше, чем нулей, а количество единиц больше на 1. Эти количества можно рассчитать, после чего определить соответствующее десятичное значение.

Можно также рассуждать так.

Обозначим общее количество цифр в искомом числе — K , количество единиц — $K1$, нулей — $K0$, причем K — нечетное число. Чтобы доля единиц ($K1/K \times 100$) и доля нулей ($K0/K \times 100$) были целым числом, значение K должно быть равно 5, 15, 25, 35 (10, 20 и 30, как четные числа, не учитываем). Исследовав эти значения, можно получить искомое количество единиц и нулей в числе.

Задача “Найти количество единиц”

Напомним условие: “Дано двухразрядное двоичное число (может быть с начальным нулем или состоящее из двух нулей). Определим количество единиц в нем. Это количество можно записать в виде $r_1 r_2$ (00, 01 или 10). Найдите зависимости значений r_1 и r_2 от цифр заданного числа. Например,

² Часть 1 — см. предыдущий выпуск “В мир информатики”.

если заданное число равно 10, то $r_1 = 0, r_2 = 1$, если 11 — то $r_1 = 1, r_2 = 0$. Операции сравнения (условия) не использовать”.

Приведем начало решения.

Обозначим первый и второй разряд заданного числа, соответственно, b_1 и b_2 и составим таблицу, в первой строке которой запишем все возможные значения заданного числа, а в четвертой — соответствующие искомые значения:

0	0	0	1	1	0	1	1
b_1	b_2	b_1	b_2	b_1	b_2	b_1	b_2
r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2
0	0	0	1	0	1	1	0

Исследуйте возможность использования логических команд для нахождения значений r_1 и r_2 по формулам:

$$r_1 = b_1 \text{ ? } b_2$$

$$r_2 = b_1 \text{ ? } b_2$$

Предлагаем читателям провести анализ всех рассмотренных задач и прислать ответы в редакцию. Фамилии всех приславших будут опубликованы, а лучшие ответы мы поощрим дипломами.

Задача “Сложение двух двоичных цифр”

Напомним, что было предложено на основе анализа таблицы сложения в двоичной системе счисления:

$$\begin{array}{r}
 + \quad 0 \\
 \hline
 0 \quad 0
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 0 \\
 \hline
 0 \quad 1
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 1 \\
 \hline
 0 \quad 1
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 1 \\
 \hline
 1 \quad 0
 \end{array}$$

записать формулы, определяющие значения цифры результата r и значения переноса p в зависимости

от значения слагаемых a и b , в которых операции сравнения (условия) не используются.

Решение

Изменим заданную в условии таблицу сложения, записав в нее нулевые значения переноса для трех первых случаев:

$$\begin{array}{r}
 + \quad 0 \\
 \hline
 0 \quad 0
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 0 \\
 \hline
 0 \quad 1
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 1 \\
 \hline
 0 \quad 1
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 1 \\
 \hline
 1 \quad 0
 \end{array}$$

Ее анализ показывает, что значения r и p могут быть найдены по формулам:

$$r = a \text{ XOR } b \tag{1}$$

$$p = a \text{ AND } b \tag{2}$$

— где **XOR** — логическая операция, которая применяется к числам (см. приложение 1 к статье “Системы счисления и электронные таблицы” в “Информатике” № 3/2013). Эта операция, как и операции **AND**, **OR** и **NOT**, выполняется в процессоре компьютера (поэтому их называют также логическими командами) над числами, представленными в двоичном виде. В отличие от арифметических операций над двумя операндами логические команды являются *поразрядными*. Например, при сложении двух двоичных цифр возможен перенос в старший разряд, а при логических операциях все разряды рассматриваются изолированно друг от друга. Таблица результатов для команды **XOR** аналогична таблице истинности для операции **XOR** (исключающего **ИЛИ**), с той разницей, что вместо значений **ИСТИНА** и **ЛОЖЬ** используются, соответственно, 1 и 0:

a	0	0	1	1
b	0	1	0	1
$a \text{ XOR } b$	0	1	1	0

Убедитесь в справедливости формул (1) и (2).

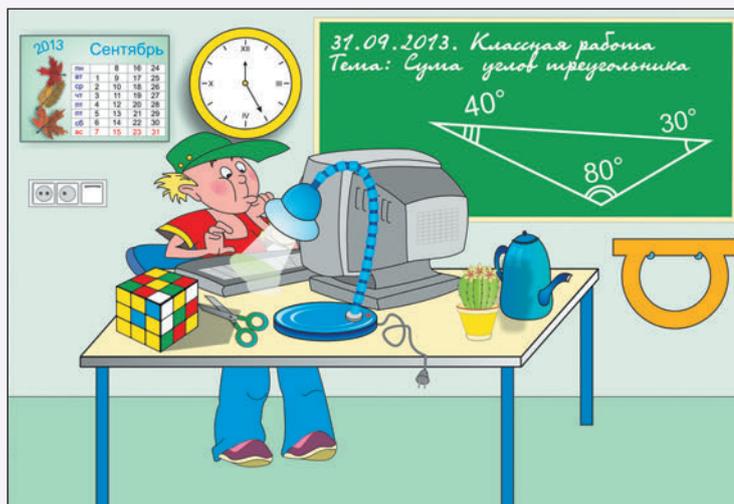
ВНИМАНИЕ! КОНКУРС!

Конкурс № 110³

Ответьте, пожалуйста, на вопрос: “Какие ошибки имеются на картинке, изображенной справа?”.

Ответ отправьте в редакцию до 1 апреля по адресу: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика” или по электронной почте: vmi@1september.ru. Пожалуйста, четко укажите в ответах свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики. Читателей-

выпускников просим учесть, что итоги конкурса будут подведены в осенних выпусках “В мир информатики”.



³Задание для конкурса предложила Е.А. Мирончик, учитель информатики МБНОУ “Лицей № 111”, г. Новокузнецк Кемеровской обл.

Напомним, что предлагалось решить ряд задач, связанных с системами счисления. Перечислим их.

1. Запишите числа 234_7 , 432_7 , 432_{10} , 234_5 в порядке следования (возрастания).

2. В электронных таблицах столбцы нумеруются латинскими буквами A, B, C, ... Z. После столбца Z идут столбцы AA, AB, AC, ... ZZ, AAA, AAB, AAC, ...

а) выпишите номера столбцов ZZZ, AXZ, ZAX, AZX, AAA, XXX в порядке следования (возрастания);

б) между какими двумя столбцами будет находиться столбец KYZ?

Почему данные задачи представлены как задачи на системы счисления?

3. Выпишите числа 2^n , $2^{n+1} + 1$, $2^n - 2$, где $n > 10$, в порядке возрастания количества единиц в их двоичной записи.

4. Каждую строку черно-белого рисунка размером $n \times n$ пикселей закодировали следующим образом: пиксель черного цвета обозначили единицей, белого — нулем и соответствующее число представили в восьмеричной системе счисления. Весь рисунок при этом кодируется последовательностью восьмеричных чисел: 46, 151, 251, 51, 51, 51, 51, 46. Что изображено на рисунке?

Аналогично можно этот рисунок закодировать в шестнадцатеричной системе счисления. Перекодируйте рисунок соответствующим образом.

5. Два некоторых десятичных числа X и Y перевели в системы счисления с основаниями 16 и 8. Часть символов при записи была утеряна. Два из четырех полученных чисел имеют вид (позиции утерянных символов обозначены символом “*”):

$$A^*_{16} \text{ и } 1^*3_8.$$

Можно ли сделать вывод о том, какое из чисел X и Y больше, или о равенстве этих чисел?

6. Некоторое десятичное число X перевели в системы счисления с основаниями 16, 8, 4, 2. Часть символов при записи была утеряна. Позиции утерянных символов обозначены символом “*”:

$$X = E^*_{16} = *5^*_8 = ***1_4 = *****1^*_2.$$

Восстановите все числа и определите число X.

7. Некоторое десятичное число X перевели в системы счисления с основаниями 16 и 8. Часть символов при записи утеряна (позиции утерянных символов обозначены символом “*”):

$$***_{16} \text{ и } 4^*_8.$$

Можно ли однозначно определить значение числа X? Если нет, то укажите все возможные значения числа.

Решение задания 1

Для того чтобы сравнить перечисленные в условии числа, их необязательно представлять в одной и той же системе счисления (запись двух пар чисел одинакова, а два числа представлены в одной и той же системе).

Прежде всего ясно, что $234_7 < 432_7$. Кроме того, $234_5 < 234_7$, а $432_7 < 432_{10}$. Поэтому в порядке возрастания числа будут расположены так:

$$234_5 < 234_7 < 432_7 < 432_{10}.$$

Решение задания 2

Можно все буквы латинского алфавита (их 26) рассматривать как цифры 26-ричной системы счисления. Тогда трехсимвольные обозначения столбцов можно рассматривать как трехзначные числа.

Для сравнения обозначений (чисел) надо сравнить первые буквы (цифры), в случае равенства — сравнить вторые буквы (цифры) и т.д. Учитывая это, заданные номера столбцов в порядке возрастания следует расположить так: AAA, AXZ, AZX, XXX, ZAX, ZZZ.

Так как Z — последняя буква алфавита, то перед столбцом KYZ будет находиться столбец KYU (буква Y — предпоследняя), а после него — столбец KZA (Z — следующая после Y средняя буква, A — буква, используемая после Z).

Решение задания 3

Составим таблицу:

n	1	2	3	...
2ⁿ	2	4	8	
Двоичная запись числа 2ⁿ	10	100	1000	
Количество единиц	1	1	1	

В двоичном виде в числе $2^{n+1} + 1$ будут две единицы (последний 0 заменяется на 1).

Для определения количества единиц в двоичной записи числа $2^n - 2$ проведем вычитание в общем виде:

–	2 ⁿ	1000...000
	2	10
		111...110

Итак, заданные числа в порядке возрастания количества единиц в их двоичной записи:

$$2^n, 2^{n+1} + 1, 2^n - 2.$$

Решение задания 4

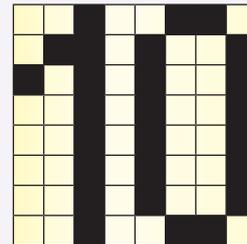
Чтобы получить рисунок, следует знать двоичное представление числа, которым кодируется каждая строка. Поэтому переведем заданные восьмеричные числа в двоичную систему счисления и представим результат как 8-разрядное двоичное число, дописывая в начале соответствующее количество нулей. Это можно сделать без промежуточного перевода чисел восьмеричных чисел в десятичную систему и последующего последовательного деления на 2.

Для перевода целого восьмеричного числа в двоичную систему счисления необходимо каждую цифру заданного числа, рассматривая ее как десятичную, представить в виде трехразрядного двоичного числа (может быть — с начальным нулем, с начальными нулями или состоящим из трех нулей)

$$46_8 = 00100110_2, 151_8 = 01101001_2,$$

$$251_8 = 10101001_2, 51_8 = 00101001_2.$$

Соответствующее изображение (в увеличенном масштабе):



Для кодирования рисунка в шестнадцатеричной системе счисления удобно использовать полученные двоичные числа. Здесь также можно обойтись без промежуточного перевода чисел двоичных чисел в десятичную систему и последующего деления на 16.

Для перевода целого двоичного числа в шестнадцатеричную систему счисления необходимо:

1) начиная с правых разрядов, разбить цифры заданного числа на тетрады — группы из четырех цифр (в крайней слева группе может оказаться одна, две или три цифры);

2) число, образованное цифрами в каждой группе, перевести в десятичную систему. Так как при этом не могут получиться числа, большие 15, то их можно рассматривать как цифры шестнадцатеричной системы счисления (при наличии чисел, больших 9, их следует записать с использованием обозначений для соответствующих цифр — А, В, ..., F);

3) из полученных на предыдущем этапе цифр сформировать единое число, которое и будет являться искомым результатом.

В нашем случае:

$$10\ 0110_2 = 26_{16}; \quad 1010\ 1001_2 = A9_{16};$$

$$110\ 1001_2 = 69_{16}; \quad 10\ 1001_2 = 29_{16};$$

то есть в шестнадцатеричной системе счисления рисунок будет кодироваться так: 26, 69, A9, 29, 29, 29, 26.

Решение задания 5

Запишем каждое из приведенных в условии чисел в двоичной системе счисления, заменяя неизвестные цифры тремя или четырьмя символами “*”:

Основание		A (10 ₁₀)				*			
16									
2	↓	1	0	1	0	*	*	*	*
2			1	*	*	*	0	1	1
8	↑	1	*			3			

Сравнение строк с основанием 2 показывает, что число A*₁₆ больше, чем 1*3₈. Но какое из них соответствует числу X, а какое — Y, определить невозможно.

Решение задания 6

Запишем известную информацию о записи чисел в виде таблицы, выделив, начиная справа, пары, тройки (триады) и группы по 4 (тетрады) цифр:

№ пп	Основание	Запись числа							
1	8	*	5			*			
2	2	*	*	*	*	*	1	*	*
3	4	*	*			1			
4	16	E				*			

Сравнение данных в строках 2 и 3 позволяет получить ряд новых цифр — они оформлены полужирным начертанием и синим цветом, а данных в строках 1 и 2 — новые цифры, выделенные зеленым цветом:

№ пп	Основание	Запись числа							
1	8	*	5			*			
2	2	*	*	1	0	1	1	0	1
3	4	*	*			1			
4	16	E				*			

После этого можно также получить ряд новых цифр:

№ пп	Основание	Запись числа							
1	8	*	5			5			
2	2	*	*	1	0	1	1	0	1
3	4	3	2			3			
4	16	E (14 ₁₀)				C (13₁₀)			

Синим цветом выделены цифры, полученные при сравнении строк 2 и 3, розовым — при сравнении строк 3 и 4, зеленым — строк 1 и 2.

Итак, ответ (новые цифры оформлены полужирным начертанием):

№ пп	Основание	Запись числа							
1	8	3	5			5			
2	2	1	1	1	0	1	1	0	1
3	4	3	2			3			
4	16	E				C			
		X = 237 ₁₀							

Решение задания 7

Рассуждения, аналогичные проведенным при выполнении предыдущего задания, позволяют получить таблицу:

№ пп	Основание	Запись числа										
1	16	*			*				*			
2	2			*	*	*	*	*	*	*	*	*
3	8				4				*			
4	2			1	0	0	*	*	*	0	1	0
		X										

В строке 4 таблицы неизвестны 3 цифры, то есть однозначно определить значение числа X невозможно. Эти 3 цифры могут дать 8 различных значений (от 000 до 111). Соответствующие 8 возможных значений числа X: 258, 266, 274, 282, 290, 298, 306, 314.

Участниками конкурса являлись:

— Арсланов Вадим, Загафуранова Айсылу, Тастиева Эльза и Шарифунова Эльвира, средняя школа села Сейтяково Балтачевского р-на, Республика Башкортостан, учитель **Загафуранова А.Ф.**;

— Билалова Юлия, Республика Татарстан, Актанышский р-н, село Актаныш, средняя школа № 1, учитель **Галиева Р.Ф.**;

— Валеев Нафис, Фатхутдинов Инсаф и Хайрутдинов Инсаф, Адельшинская средняя школа, Чистопольский р-н Республики Татарстан, учитель **Фатхутдинова А.А.**;

— Валуев Иван и Гаязов Рашид, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Власов Юрий, г. Воронеж, школа № 5 им. К.П. Фоктистова, учитель **Чернышева И.А.**;

— Герасимова Наталья и Костина Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Калугин Сергей, средняя школа села Ириновка, Новобурацкий р-н Саратовской обл., учитель **Брунов А.С.**;

— Лагунова Елена, Михолап Роман, Новоселов Алексей и Ошуркова Валерия, г. Киров, школа № 28, учитель **Михолап Э.Л.**;

— Мельников Олег и Романов Егор, г. Кострома, гимназия № 33, учителя **Исакова Е.В.** и **Романова Е.В.**;

— Новиченко Владимир, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Ревков Александр и Шигаев Никита, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Соколова Юлия и Щеголькова Любовь, Алтайский край, средняя школа села Сараса Алтайского р-на, учитель **Барабанова О.В.**;

— Яковлев Сергей, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Все они будут награждены дипломами. Поздравляем!



Общероссийский проект Школа цифрового века

Интернет-обеспечение проекта – Издательский дом «ПЕРВОЕ СЕНТЯБРЯ»

Общероссийский проект «Школа цифрового века» по комплексному обеспечению образовательных учреждений предметно-методическими материалами направлен на вовлечение педагогических работников в цифровое образовательное пространство.

В рамках проекта каждому работнику образовательного учреждения предоставляются **23 журнала** по всем учебным дисциплинам и направлениям школьной жизни с дополнительными материалами для практического использования (презентации, раздаточные материалы, образовательное видео), **методические брошюры, журнал для родителей, модульные 6-часовые курсы** из циклов «Навыки профессиональной и личной эффективности» и «Инклюзивный подход в образовании».

Открыт прием заявок от образовательных учреждений на 2014/15 учебный год

На новом этапе проекта

- Каждому педагогическому работнику образовательного учреждения будет предоставлена возможность выбрать и пройти один из десяти 36-часовых дистанционных курсов по общей педагогике с выдачей удостоверения установленного образца
- Расширится линейка модульных курсов

Стоимость участия в проекте для образовательного учреждения – 6 тысяч рублей за весь учебный год независимо от количества педагогических работников.

Участие образовательного учреждения и педагогических работников в проекте удостоверяется соответствующими документами. Для дошкольных учреждений предусмотрен свой набор удостоверяющих документов.

Срок действия проекта в 2014/15 учебном году: с 1 августа 2014 года по 30 июля 2015 года

Прием заявок и подробности
на сайте

digital.1september.ru

журнал

Информатика – Первое сентября

2-е полугодие 2014 года

ПОДПИСКА

на сайте www.1september.ru и в почтовых отделениях РФ

ИАП МЕЖРЕГИОНАЛЬНОЕ АГЕНТСТВО ПОДПИСКИ

КАТАЛОГ РОССИЙСКОЙ ПРЕССЫ
ПОЧТА РОССИИ

2014
второе полугодие

Индекс	Название издания	Периодичность в полугодие	1 месяц		6 месяцев	
			Каталожная цена (руб.)	Подписная цена (руб.)	Каталожная цена (руб.)	Подписная цена (руб.)
Название блока в разделе «Журналы»	ПЕРВОЕ СЕНТЯБРЯ. ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА (499)249-31-38					
79066	Информатика – Первое сентября. Бумажная версия С электронными приложениями и презентациями. В июне не выходит. Подписка на июнь не принимается. (-) 160 г 64 стр.	5	308.00		1540.00	
12684	Информатика – Первое сентября. Электронная версия на CD (полная копия бумажной версии) В июне не выходит. Подписка на июнь не принимается (-) 75 г	5	118.80		594.00	

При оформлении подписки на сайте www.1september.ru оплата производится по квитанции в отделении банка или электронными платежами on-line





ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ г. МОСКВЫ
ИЗДАТЕЛЬСКИЙ ДОМ «ПЕРВОЕ СЕНТЯБРЯ»
МОСКОВСКИЙ ПЕДАГОГИЧЕСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

2014

24 МАРТА – 18 АПРЕЛЯ

РАСПИСАНИЕ ДНЕЙ ПЕДАГОГИЧЕСКОГО МАРАФОНА

24 марта	День учителя технологии *	4 апреля	День учителя информатики
25 марта	Открытие Марафона День классного руководителя	5 апреля	День учителя физики
26 марта	День школьного психолога День учителя ОБЖ	6 апреля	День учителя математики
27 марта	День здоровья детей, коррекционной педагогики, логопеда, инклюзивного образования и лечебной физической культуры	8 апреля	День учителя истории и обществознания
28 марта	День учителя начальной школы (день первый)	9 апреля	День учителя МХК, музыки и ИЗО
29 марта	День учителя начальной школы (день второй)	10 апреля	День школьного и детского библиотекаря
30 марта	День дошкольного образования	11 апреля	День учителя литературы
1 апреля	День учителя географии	12 апреля	День учителя русского языка
2 апреля	День учителя химии	13 апреля	День учителя английского языка
3 апреля	День учителя биологии	15 апреля	День учителя французского языка
		16 апреля	День школьной администрации
		17 апреля	День учителя физической культуры
		18 апреля	День учителя немецкого языка Закрытие

marathon.1september.ru

-  Обязательная предварительная регистрация на все дни Марафона с 20 февраля 2014 года на сайте marathon.1september.ru
-  Каждый участник Марафона, посетивший три мероприятия одного дня, получает официальный именной сертификат (6 часов)
В дни Марафона ведущие издательства страны представляют книги для учителей
Начало работы каждого дня – 9.00. Завершение работы – 15.00

УЧАСТИЕ БЕСПЛАТНОЕ. ВХОД ПО БИЛЕТАМ

РЕГИСТРИРУЙТЕСЬ, РАСПЕЧАТЫВАЙТЕ СВОЙ БИЛЕТ И ПРИХОДИТЕ!

Место проведения Марафона: МПГУ, ул. Малая Пироговская, дом 1, стр. 1 (в 5 минутах ходьбы от ст. метро «Фрунзенская»)

* Место проведения Дня учителя технологии: ЦО № 293, ул. Касаткина, 1а (ст. метро «ВДНХ»)

По всем вопросам обращайтесь, пожалуйста, по телефону **8-499-249-3138** или по электронной почте marathon@1september.ru