

ИНФОРМАТИК А

4

Ю.А. Первин

Задумавшись над очередной загогулиной блок-схемы, поднимаю глаза, и взор упирается в сидящую напротив симпатичную девушку. Мысли, конечно, не о ней, а о новом повороте блок-схемы.

15

А.А. Дуванов

Какой же была моя радость пополам с удивлением, когда деревяшка заговорила человеческим голосом! Своими руками я сотворил Буратино! Стало понятно, что я волшебник.

38

Д.Ю. Усенков

Словами Огюста Родена... "Нужно взять глыбу мрамора и отсечь от нее все лишнее"





НА ОБЛОЖКЕ

► 3D-принтеры перестали быть экзотикой, увлечением для избранных. Сегодня это вполне промышленный инструмент, используемый в различных областях деятельности. Кстати, среди 3D-принтеров уже имеются и лазерные, и струйные, и даже матричные модели. И стоимость устройств существенно снижается (не резко, но ряд 3D-принтеров уже становится доступен и для домашнего использования). В этом номере мы публикуем подробную статью о текущем состоянии технологий 3D-печати. Может быть, дождемся мы и появления “Информатики” в 3D-версии ©.

В НОМЕРЕ

- 3** ПАРА СЛОВ
 - Не знаю!
- 4** ИНФОРМАТИКА В ЛИЦАХ
 - Ю.А. Первин
 - А.А. Дуванов
- 26** ЕГЭ
 - Люблю ЕГЭ за В15, или Еще раз про метод отображения
- 33** ОЛИМПИАДЫ
 - Школьники играют в СТФ
- 38** ТЕХНОЛОГИИ
 - Что такое 3D-принтер и как он работает
- 48** ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ
 - “В мир информатики” № 199

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 этот номер (№7–8/2014).

Срок действия кода: с 1 июля по 31 декабря 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 “Первое 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 27 622 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)

**ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА
 “ПЕРВОЕ СЕНТЯБРЯ”**

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.Битянова

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 – Л.Печатникова
 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 20.05.2014,
 фактически 20.05.2014
 Заказ №
 Отпечатано в ОАО “Первая
 Образцовая типография”
 Филиал “Чеховский Печатный Двор”
 ул. Полиграфистов, д. 1,
 Московская область,
 г. Чехов, 142300
 Сайт: www.chpd.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



Не знаю!

► Ассоциации — штука загадочная (но не случайная). Недавно я присутствовал на одной профессиональной конференции психологов. Захватывающий мастер-класс, который проводила коллега из лондонского университета, был посвящен роли фразы “не знаю” в работе психолога. Мне было очень интересно, но в какой-то момент я понял, что отключился от происходящего и думаю совсем о другом. Это другое — игра Flappy Birds.

Наверное, многие знают, о чем речь. Об этой игре слышали многие, даже если не успели в нее поиграть. Кто не успел, тот на данный момент опоздал. На самом пике популярности, в начале февраля, автор удалил игру из магазинов приложений, после чего возник отдельный рынок смартфонов, на которых игра уже была установлена. Стоимость таких устройств доходила до заоблачных значений.

Игру написал никому не известный программист из Вьетнама Нгуен Ха Донг. Суть игры — провести птичку через ряд соединяющихся труб. Графика — примитивная (если не знать, что управляешь птичкой, вполне можно принять персонаж игры за свинку, например). Управление — одно действие (подпрыгнуть). В общем — решительно не ясно, чем эта игра могла привлечь кого-то, кроме ав-

тора. Тем не менее за два месяца Flappy стала самым популярным игровым приложением в апп-сторах Apple и Google, и только с рекламы, в ней размещаемой, автор (так считают) имел 50 000 долларов в день.

Ну, почемууу?! Сколько экспертов, аналитиков, жаждущих славы стартапиров задавались с тех пор этим вопросом. Ответ на него один для всех — не знаю! Имеется масса притянутых за уши тонких аналитических обстоятельств. Говорят и про удачный выбор некоего блоггера, который осуществил первое продвижение игры. Но блоггеров этих — как... ну, в общем, много. И про то, что автор специально разместил кнопку оценки игры в апп-сторах так, что на нее легко было случайно нажать. Много чего говорят. Тысячи пытались повторить. Но ни у кого не получилось приблизиться к оригиналу. Результаты — на порядок хуже.

Сам автор называет главным фактором успеха своего продукта удачу. Он считает, что ему просто повезло, что факторы, каждый из которых сам по себе мало что значил, сошлись в одно время и в одном месте — одном продукте, обеспечив особый синергетический эффект. Ну, может быть... Это тоже одно из мнений. Из множества — не дающих ответ на вопрос “Ну, почемууу?!”.

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



Первин Юрий Абрамович

Многие из замышлявшихся дел в своей жизни мне удавались не с первого, а только со второго раза: в комсомол вступил со второго раза (при первой попытке меня обвинили в обмане — приписал к своему возрасту лишние полгода), делал две дипломных работы (см. ниже), кандидатскую диссертацию писал дважды, да и к докторской подошел тоже только со второй попытки... Так складывавшаяся судьба позволила мне почувствовать себя оптимистом и не падать духом в трудных ситуациях: любимой песней не только детства, но и всей жизни стала песня из фильма “Дети капитана Гранта” со словами припева “Кто ищет, тот всегда найдет”.

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

Выбор факультета при поступлении в вуз — физмат Нижегородского (в ту пору — Горьковского) университета имени Н.И. Лобачевского — не был случайным: в жизни я всегда любил и ценил красоту. Поэтому выбор математики как будущей специальности был, по существу, predetermined. Но событие, произошедшее во время учебы на четвертом курсе, оказалось решающим для всей последующей жизни. В нашем университете было впервые решено сформировать группу программистов — трое парней и три девушки. Увлечение идеями программирования пришло сразу. Но электронных вычислительных машин в городе тогда еще не было. И троих из нас — мужскую половину группы — на весь последний, пятый, учебный год отправили в Москву, в Московский государственный университет имени М.В. Ломоносова.

Очень много дали нам лекции и семинары в МГУ: интересные курсы, увлекательные знания, замечательные люди. И почти сразу же наша тройка оказалась на преддипломной практике. Повезло нам сказочно. Как стажер я впервые в жизни получил трудовую книжку. Первой записью в ней стала строчка — старший лаборант Института прикладной математики Академии наук СССР. В то время им руководил академик М.В. Келдыш, Президент Академии наук. Впоследствии институт получил имя М.В. Келдыша.

Моя лаборантская деятельность была связана с отделом, во главе которого стоял С.В. Яблонский, а душой и “мотором” этого отдела — генератором идей — был Алексей Андреевич Ляпунов. Оба они вскоре стали членами-корреспондентами АН СССР. Замечательный Человек и Ученый Алексей Андреевич Ляпунов принял на себя руководство моей преддипломной практикой. Он оказал огромное влияние на всю мою дальнейшую жизнь.

Первой и, как оказалось немного позднее — через три месяца, не последней темой стали программирование, отладка и производственный запуск программы, о физическом содержании параметров которой я не знал тогда и не знаю даже сейчас: все свои записи, промежуточные результаты и саму программу я писал в отдельной комнате, работая в толстой тетради со страницами, перечеркнутыми широкими красными диагоналями. На эту работу я набросился как голодный волк на легкую добычу. Математиком, который был заказчиком этой задачи — автором написанных уравнений, — был тогда еще молодой ученый В.В. Русанов (профессором и членом-корреспондентом он стал позднее). Он работал в соседней комнате и, будучи секретарем партийной организации института, более или

менее регулярно (пожалуй, раз в два месяца) поднимал телефонную трубку и выслушивал указания: подготовить общее собрание коллектива для награждения орденами и медалями, а двоих — профессора побородатее и молодую девушку посимпатичнее — прислать такого-то числа во столько-то часов в Кремль; и, заметив меня, вошедшего без стука, молча поднимал указательный палец вверх. Это была уже ракетная, но пока еще “предспутниковая” эпоха: ракеты запускали довольно часто.

Виктору Владимировичу я понравился, по-видимому, усердием и опытом вычислительной работы по методу сеток. Совсем недавно, на четвертом курсе, мы, громыхая в вычислительной аудитории физмата на механических арифмометрах “Феликс”, вычисляли методом сеток значения дифференциальных уравнений в частных производных.

Работа над уравнениями Русанова шла довольно быстро. Но вдруг в один из обычных дней мои выходы на вычислительную машину “Стрела” пришлось приостановить: первые 8 десятичных знаков машина выдавала верно, а дальше — необъяснимая цепочка цифр... Следовало довести точность по крайней мере до 15 знаков! Та ужасная неделя проходила день за днем, а улучшения не было видно. И когда я с тоской пришел к Виктору Владимировичу, мы оба взялись за внимательный анализ. Мы оба признались в своих грехах: я — в слабом знании древнегреческого алфавита, а он — в небрежности своего почерка. В результате задуманное им имя переменной он обозначал буквой “Ψ” (кси заглавная, с горизонтальной чертой у основания), а я пробивал на перфокартах ее как “ψ” (кси строчную, без горизонтальной черты). Мы оба улыбнулись, пожали друг другу руки. Задача была в срок доведена до конца. И мы расстались, увы, навсегда. Мне не позволяла совесть считать эту мою задачу кодирования переменных в классическом алгоритме дипломной работой, и, воспользовавшись тем, что появился фонд относительно свободного времени, я обратился к Алексею Андреевичу Ляпунову с просьбой разрешить поработать над интересовавшей меня задачей как над предметом дипломной работы.

Сейчас уже трудно вспомнить, что привело меня к этой задаче. Дело в том, что я, хотя и умею играть в домино и даже знаю общеприменяемые “народные” тактики-рекомендации этой игры, никогда не горел ею с увлечением. Эта игра не привлекала меня ни в тенистых городских дворах, ни в плацкартных вагонах железнодорожных поездов. Вероятно, в этом обдуманном мною выборе сказались размышления, родившиеся после интереснейших обсуждений на заседаниях Большого кибернетического семинара А.А. Ляпунова в амфитеатральной аудитории на 16-м этаже главного здания МГУ, где по вечерам четвергов выступали именитые докладчики — И.А. Полетаев, А.И. Китов, Ю.А. Шрейдер, Н.В. Тимофеев-Ресовский, да зачастую и сам Алексей Андреевич Ляпунов...

Но все же наиболее вероятной гипотезой зарождения идеи о программировании игры в домино на ЭВМ стала здоровая, увлекательная, насыщенная, интеллектуальная среда молодых людей, проводящих дни и вечера (а порой и ночи) в машинном зале или на подступах к нему. В ИПМ АН СССР существовал и несколько лет весьма активно работал клуб с выразительным названием “Серкозел”, в котором “сверкали” такие талантливые программисты, как И.Б. Задыхайло, Э.З. Любимский, С.С. Камынин... люди, славными делами вошедшие в историю советского программирования. Все они и каждый из них были моими кумирами и в программировании, и в жизни. У клуба “Серкозел” были свои традиции, свои регулярные (по графику) турниры, свои звания. Корифеи этой игры получали звания действительного члена клуба. Чуть более приземленным было звание — член-корреспондент клуба. Чтобы из членов-корреспондентов пробиться в действительные члены, следовало (по положению о клубе) выиграть несколько партий у команды, составленной из действительных членов.

А может быть, просто душа рвалась к хитросплетениям гибких управляющих структур — ветвлений и циклов — после удручающих серий линейных вычислительных алгоритмов.

Так или иначе, я рассказал о своем замысле Алексею Андреевичу и получил его благословляющее доброе напутствие. С огромным увлечением, если не сказать — страстью, погрузился в конструирование стратегий и тактик алгоритма, который моделировал классическую игру в домино, практически без всяких идеализаций. В сторону отодвинуты кино и книги, спортивные зрелища и развлечения. Даже учебные занятия (кроме, конечно, ляпуновского семинара в МГУ). Даже девушки. Вспоминается меня самого посмешивший позднее эпизод. Поздним вечером возвращаюсь из машинного зала ИПМ в общежитие на метро и продолжаю рисовать нескончаемые блок-схемы (в то время еще не были написаны примечательные антибейсиковые статьи Э. Дейкстры, К. Хоора и А. П. Ершова, громящие блок-схемы, и я жил как многие мои современники — рисованием блок-схем). Задумавшись над очередной загогулиной блок-схемы, поднимаю глаза, и взор упирается в сидящую напротив симпатичную девушку. Мысли, конечно, не о ней, а о новом повороте блок-схемы. Следующая задумка, и еще одна остановка мысли. Снова ненадолго взгляд останавливается на девушке, которую я практически не замечаю. Но она-то воспринимает мои устремленные внутрь самого себя взгляды как мое желание нарисовать ее портрет. Наконец, не видя моих результатов, она демонстративно встает, якобы возмущаясь моей откровенной наглостью, и переходит на мою сторону вагона. Стоит, упираясь спиной в дверь вагона, бросает свой косой взгляд на лист бумаги с блок-схемами, лежащий на моих коленях. Тут она видит, что мои рисунки очень далеки от воспроизведения красивого портрета, и уже в нескрываемом гневе уходит в другой конец вагона. В игре участвуют 28 костей. Программа в начале игры перемешивает их случайным образом. Машина случайно получает свои 7 костей, а остальные она считает с равной вероятностью $1/3$ принадлежащими трем другим игрокам. Они условно пронумерованы так: 1) — машина, 2) — левый противник, 3) — напарник, 4) — правый противник. Получая информацию о ходах игроков 2), 3) и 4), машина пересчитывает вероятностную информацию о распределении костей у участников и на основании этой информации, правил игры и заложенных в программу накопленных практикой рекомендаций делает выбор из своих еще не выставленных костей. Основная гипотеза такой компьютеризованной модели одной игровой партии (одного кона) состоит в том, что с течением игры машина все с большей вероятностью определяет кости трех остальных игроков. Эта гипотеза была подтверждена и показала, что машина оказывается сильным или, во всяком случае, достойным соперником игрокам-людям.

Интерфейс этой модели-игры прост:

Ход противников и напарников — это ввод перфокарты, на которой пробиты два числа, очки выставленной кости (или ввод костей оператором с помощью тумблеров операторского пульта); адекватность игрока и выставленной им кости определяется по очередности игроков 1–2–3–4, которая нарушается только в том случае, если у игрока, которому предстоит делать ход, нет нужной кости ни на один из концов цепочки костей, выставленных на кон; машина узнает об этом, получив сигнал от оператора или перфокарту с недопустимой в игре информацией (например 7:7).

“Размышления” машины реализуются тремя разными тактиками. Ход игры с участием машины представляет собой попеременное включение трех “тактик”, каждая из которых является алгоритмом, моделирующим поведение машины при выборе своего хода:

- “тактика 1” — программа, по которой машина принимает ответственность за развитие игры на себя;
- “тактика 2” — программа, оценивая вероятности распределения костей, строит игру на осложнение игры для противника 2;

- “тактика 3” — программа подыгрывает напарнику (3), поддерживая его инициативу;
 - “тактика 4” — программа, оценивая вероятности распределения костей, строит игру на осложнении игры для противника 4*.
- Несколько иными в организационном аспекте представляются программные реализации, называемые стратегиями. Они предлагают организовать все три возможных способа проведения игры:
- играет одна машина с тремя людьми (машина выступает в качестве одного из участников в одной из пар); это типовая организация игры;
 - играют пара машин против пары игроков-людей;
 - теоретически возможный (но представляющийся малоинтересным) вариант игры: играют четыре машины — одна пара против другой.
- Эта программа отлаживалась на “Стреле” в обычном пакетном режиме, с предварительным заказом машинного времени у диспетчера. Из проверок отдельных блоков и ветвей программы складывалась постепенно структура общего алгоритма. С одной стороны, эта игровая программа выглядела, конечно, “белой вороной” на фоне серьезнейших, подчас государственного значения, производственных программ. Вспоминается один микроскандал, когда один из специалистов института ходил жаловаться самому академику М.В. Келдышу: “Машинного времени катастрофически не хватает, а они тут на машине козла гоняют”. Ситуацию спасло вмешательство профессора А.А. Ляпунова, который убедил директора в кибернетической значимости этого эксперимента. С другой стороны, многие в институте уже знали о машинной отладке игры в домино и относились к этой программе сочувственно и даже с интересом. Однажды, уже на последних этапах отладки, я решил сэкономить время и вместо подготовки новой перфокарты, в которой была обнаружена погрешность, решил “добить” недостающее отверстие на той же перфокарте. Но положение не улучшилось, а, наоборот, стало еще более неприятным и казалось безвыходным: завершающий сеанс машинного времени неминуемо приближался. На помощь пришел Игорь Борисович Задыхайло (уже тогда уважаемый программист, но ставший доктором наук позднее): опаздывая на последнюю электричку, стал копать в колоде перфокарт и обнаружил, что при “добивке” перфокарты выскочил из своего гнезда дополнительно еще один пуансон, и на карте появилось еще одно отверстие. Все эти неприятности произошли поздним вечером 29 апреля 1957 года. А на следующее утро, на утренние часы 30 апреля было заказано машинное время на демонстрационный запуск завершенной программы, который неожиданно для меня превратился, по существу, в публичную защиту первой самостоятельной программы — дипломной работы. Информация об этом событии очень быстро докатилась до всех уголков института, и обычно пустынный машинный зал — “святая святых” — был заполнен людьми. Было решено устроить реальную игру машины против людей. На время этого сеанса я уселся за операторский пульт, а дежурный оператор стал партнером “Стрелы”. После оперативного обсуждения кандидатур в качестве противников были предложены члены клуба “Серкозел” С.С. Камынин и В.С. Штаркман. Было собрано авторитетное жюри. Кости домино были перемешаны. В качестве оператора я ввел в память машины 7 ее перфокарт. Игра началась: первый ход дуплем 1:1 был сделан машиной. Включилась “тактика 1”, в которой у машины и больше возможностей, и больше ответственности. Шла игра, в которой каждый ход высвечивался комбинациями загорающихся неоновых лампочек на пульте. Обстановка накалялась и достигла своего апогея в тот момент, когда (члены клуба “Серкозел” увидели это сразу) у машины появилась возможность сделать “рыбу” — одно из возможных в домино окончаний игры, при котором победитель определяется по минимуму очков, оставшихся на руках у игроков команды.

* Здесь и сейчас нецелесообразно описывать эти алгоритмы-тактики детально, тем более что они позднее были опубликованы в Докладах Академии наук СССР (т. 124, № 1, 1959) и несколько более подробно в журнале “Проблемы кибернетики”, вып. 3, 1960, “Физматгиз” (переведена и опубликована в США (1961), Румынии (1962)).

Зал замер в ожидании. И вдруг — хор, вскричавший одно слово: “Рыба!”. “Стрела” убедительно выиграла эту игру. Правда, тут же нашлись и скептики: мол, причина успеха программы в том, что в команде соперников был один — Сева Штаркман — не действительный член клуба “Серкозел”, а всего-навсего член-корреспондент.

После этого сеанса защита диплома, состоявшегося в тот же день, превратилась в короткое, почти формальное заседание, тем более что почти все члены комиссии только что посмотрели живой результат. А вечером, после предпраздничного собрания с участием академика М.В. Келдыша в Институте прикладной математики АН СССР, я сел на поезд и уехал из Москвы в родной Горький. Там с самого утра поехал не домой, а сразу в праздничную первомайскую колонну своей alma mater — Горьковского университета — в толпу друзей-однокурсников. Но судьба моей первой программы в тот день, 30 апреля 1957 года, еще не завершилась. В следующем же учебном году (1957–1958) А.А. Ляпунов пригласил меня, уже молодого специалиста исследовательского физико-технического института при Горьковском университете имени Н.И. Лобачевского, с докладом на свой Большой кибернетический семинар для рассказа о кибернетическом феномене обучающихся программ. А еще через год благодаря ему же появилась в моем пока еще пустом списке научных работ (сегодня их почти четыре с половиной сотни) первая запись — Доклады Академии наук СССР, т. 124, № 1, 1959. В то время А.А. Ляпунов еще не был членом Академии, а статьи в ДАН — этот самый авторитетный у нас в стране научный журнал — мог представлять только действительный член Академии. По рекомендации А.А. Ляпунова эту статью представил в ДАН выдающийся советский академик математик С.Л. Соболев. А еще год спустя по материалам доклада на семинаре Ляпунова появилась несколько более развернутая публикация с тем же названием в сборнике-журнале “Проблемы кибернетики”, вып. 3, 1960, который организовал и редактировал А.А. Ляпунов. Чуть позже — в 1961 и 1962 гг. — появились переводы этой статьи на английский (в США) и на румынский языки.

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

Алексей Андреевич Ляпунов происходит из старинного русского рода, оставившего след в истории отечественной науки и искусства. В числе его ближайших родственников и российский академик-математик Александр Михайлович Ляпунов, и композитор Сергей Михайлович Ляпунов, и кораблестроитель, механик и математик, академик Алексей Николаевич Крылов, и знаменитый офтальмолог Виктор Петрович Филатов, и филолог академик Борис Михайлович Ляпунов, и знаменитый физик Петр Леонидович Капица... Отсюда и всепоглощающая интеллигентность Алексея Андреевича, и его удивительная энциклопедичность и широта круга интересов.

Его геологическая коллекция была столь фундаментальна, что один из будущих докторов геологических наук признавался в том, что геологией он увлекся, познакомившись с коллекцией Ляпунова. Не будучи лингвистом по специальности, Ляпунов безоговорочно считается патриархом отечественного машинного перевода. Он активно участвовал в проектировании первых отечественных ЭВМ и разработке логических схем программ. Первый курс по программированию в МГУ был разработан и поставлен А.А. Ляпуновым. При этом он сам всегда считал себя математиком, автором заслуживающих внимания результатов по теории множеств и в первую очередь по дескриптивной теории множеств. А конструктивность любой науки он считал возможным оценивать по тому, в какой степени она способна использовать категории и объекты теории множеств.

На Великую Отечественную войну он ушел добровольцем, единственный раз в жизни отодвинув на второй план свои увлечения наукой. Но и тут молодой командир топографического взвода нашел возможность применить свои знания и великолепную память. Характерный пример — история награждения А.А. Ляпунова боевым орденом Красной звезды за участие в боях на Курской дуге — переломном моменте вой-

ны в пользу Советского Союза. Впервые Алексей Андреевич побывал в этих местах еще до поступления в университет, когда приезжал под Курск в составе геологической экспедиции отца, исследовавшей Курскую магнитную аномалию. Помогая отцу в упорядочении записей экспедиционных наблюдений, юноша Ляпунов сохранил их не только в документах экспедиции, но и в собственной памяти. А воспользоваться ими довелось, когда война привела его на Курскую дугу. Вспомнив информацию о необычно сильных магнитных полях в этих местах, он предложил внести изменения в инструкции, официально поставляемые наводчикам-артиллеристам. Но ляпуновские обновления таблиц настолько расходились с теми, к которым боевые командиры-артиллеристы привыкли в учебных классах военных училищ, что у них невольно возникали подозрения по поводу инструкций командира топографического взвода. Военному начальству посыпались “сигналы” об умышленных ошибках офицера А.А. Ляпунова. В те времена в сложной боевой обстановке разбираться было некогда — под трибунал с очевидным исходом. Но успел вмешаться парторг, который предложил провести эксперимент по проверке нестандартных инструкций Ляпунова, тех, что он написал с учетом влияния здешних магнитных полей на траектории полетов тяжелых снарядов. Эксперимент был проведен. Оказалось, что типовые инструкции вызывают не только большой разброс, но и существенные отклонения от цели, а стрельба “по-ляпуновски” все снаряды точно выкладывает в одну и ту же воронку-цель.

Мне довелось общаться с А.А. Ляпуновым в ходе преддипломной практики в МГУ и Институте прикладной математики АН СССР, а потом — в его аспирантуре. Не раз бывал у него дома в Москве и новосибирском Академгородке. Вынес из этих визитов не только новые трактовки элементов теории множеств, но и высокое искусство человеческого общения. Вспоминается одна из московских встреч в квартире на Шаболовке. Пришел, не уточнив предварительно минуты своего прихода, и оказался у Ляпунова тогда, когда он вел беседу с академиком С.Л. Соболевым и членом-корреспондентом АН СССР (тогда) В.Л. Канторовичем. И невольно поймал себя на мысли о том, что посторонний посетитель не мог бы определить по стилю беседы хозяина с каждым из гостей, кто из гостей — всемирно известный академик, а кто — рядовой студент-пятикурсник. Особое очарование в новосибирском коттедже Алексея Андреевича оставили вечера по средам, куда приглашались аспиранты и коллеги Ляпунова с единственным условием — в этот вечер говорить можно было только по-французски (сам Ляпунов французским владел отлично и даже сопровождал президента Франции генерала Шарля де Голля в новосибирском Академгородке в 1966 году). Постоянной гостьей “французских вечеров” была немолодая уже пианистка француженка Вера Августовна Лотар-Шевченко (ее история удивительна, но выходит, конечно, за рамки этих воспоминаний). Она чаще всего исполняла пьесы французских композиторов и любимого ею Шопена. Для нее, для этих вечеров Алексей Андреевич специально приобрел рояль. Когда Алексей Андреевич скончался (в 1973 году), традиция “французских вечеров” была перенесена в Дом ученых Сибирского отделения АН СССР и стала главным содержанием деятельности французского клуба в Доме ученых, самого активного из иностранных клубов Академгородка.

Главным качеством А.А. Ляпунова было умение человеческого общения. Ему удалось совмещать его с высокой требовательностью к коллегам всякий раз, когда это касалось науки. Помню, как мы, группа его аспирантов, собрались вместе, чтобы попросить его о сокращении числа аспирантских экзаменов по специальности. Мол, у других руководителей аспиранты имеют только один экзамен по специальности, а вы с нас требуете два. На эту просьбу он ответил очередным рассказом о своем учителе академике Н.Н. Лузине, которого он очень высоко ценил: “А мне довелось сдавать Лузину семнадцать экзаменов”.

После этого тема была закрыта.

Андрей Петрович Ершов — программист из первого поколения советских программистов, слушавших еще лекции А.А. Ляпунова по програм-

мированию и логическим схемам программ в МГУ им. М.В. Ломоносова. Его научная карьера тесно связана с новосибирским Академгородком — от многочисленных программных систем и книг по теории программирования через педагогические эксперименты в области информатизации образования до концепций национальной программы образования. Сейчас его имя носит крупнейшее научное учреждение Академгородка — Институт систем информатики Сибирского отделения АН. Его философские статьи “О человеческом и эстетическом факторах в программировании” и “Программирование — вторая грамотность” воспитали армию школьных учителей информатики.

Как это ни удивительно, Ершов, признанный во всем мире великим человеком в программировании (на фасаде Станфордского университета в США установлен барельеф А.П. Ершова в память о его пребывании в Кремниевой долине), стал программистом случайно. Заканчивая школу, он мечтал стать физиком-ядерщиком. Но мечте не довелось сбыться, потому что при подаче документов в МГУ им. М.В. Ломоносова выяснилось, что его школьное детство совпало с оккупацией немцами украинского города Луганска, где он тогда жил. В то время такой информации было достаточно для того, чтобы закрыть дорогу в ядерную физику — науку, считавшуюся самой “закрытой” и “секретной” из наук для человека с таким “пятном” в биографии. И только после того, как ему было отказано на физфаке, он подал документы на мехмат. Какого замечательного Человека и государственного масштаба организатора науки потеряла бы советская страна в лице программиста А.П. Ершова, если бы не произошло этого эпизода на приемной комиссии МГУ, быть может, даже ценой появления еще одного крупного физика!

Жить и работать рядом с ним было легко и интересно, он не останавливался и не позволял остановиться. Имея невероятно широкий круг коллег, друзей, учеников, он всегда умел находить нужные слова и нужный тон такими, какими они создавали обстановку доверия и взаимоуважения. Слушать его было приятно и тогда, когда он делал доклад на Ученом совете, или читал лекцию студентам, или на встрече с детьми у костра Летней школы пел песни под собственный гитарный аккомпанемент, или обсуждал с зарубежными коллегами перспективы научного сотрудничества. Казалось, что он никогда не уставал: он работал в кресле авиалайнера, писал статьи на больничной койке, давал интервью, шагая по лесной тропинке от дома до рабочего кабинета в Вычислительном центре.

Это был честный, горячо преданный своему делу Человек. Мне повезло провести с ним вместе полтора десятка интересных лет.

Сергей Федорович Сопрунов — очень интересный человек, в полном смысле слова преданный избранному им научному направлению — языку программирования на Лого. Он заслуженно считается ведущим специалистом по проектированию, разработке и сопровождению Лого-систем в нашей стране, которую он достойно и постоянно представляет в европейском и мировом сообществах Лого. Все отечественные версии Лого — от старого (но до сих пор активно живущего) DOSовского Logo-Writer’a до целого семейства русско-лексических ЛогоМиров и ПервоЛого — все они дело рук Сергея Федоровича и работающих с ним программистов и педагогов. Уже давно ясно, что те, кто работают с Лого-средами, обязаны обладать неистощимой любовью к детям, знать их устремления, хорошо их понимать. Этими качествами С.Ф. Сопрунов, видимо, одарен от природы. В известном смысле можно считать естественным, что хорошее знание английского языка помогает Сергею Федоровичу в проектировании российских локализаций Лого, прототипом которых были англо-лексические программные системы. Однако позднее я нечаянно узнал, что этот иностранный язык является для него не простым средством для перевода технических терминов, а скорее, частью жизни. Правда, он более известен в нашей программистской среде программистом высочайшей квалификации и, в частности, автором книги “Непростое программирование на Лого”, но, не ограничивая свой мир, С.Ф. Сопрунов профессионально пишет и издает свои переводы с английского.

Его интеллигентность и внимательность в общении выделяют Сергея Федоровича даже в той высоконравственной научной среде, где ему приходится постоянно работать. Каждый эпизод общения с ним — это урок, из которого собеседник уносит не только новые знания по предмету, но и самые приятные впечатления. А ведь его собеседники — это прежде всего мастера человеческого общения, педагоги и психологи. Такие уроки общения полны душевной щедрости, бескорыстия и доброты. И если в какой-то момент жизни удастся отметить про себя, что встреча с новым знакомым или новым коллективом сложилась удачно и успешно, то, размышляя о таких встречах, как-то получается, что всегда удастся протянуть ниточку от “уроков” С.Ф. Сопрунова к текущему, даже не обязательно очень значительному событию.

Марсьяль Виве — это известный во Франции ученый, математик, программист, педагог, с которым мне посчастливилось познакомиться в 1969 году, когда во время своей стажировки во Франции я побывал в Ле Мане, областном центре, индустриальном и университетском городе, который в мире, быть может, более известен своей автомобильной промышленностью и особенно всемирно известным автодромом и гонками “24 часа Ле Мана”. Позднее, в 1991 году, мы провели с профессором Виве еще одну неделю в России, в Переславле-Залесском и Москве, где проходил первый (и, увы, последний) советско-французский семинар по информатизации раннего обучения информатике. А самой последней стала встреча незадолго до его кончины на полпути между Парижем и Ле Маном в небольшом французском городке Ферте-Бернар, где он организовал и провел чемпионат мира по робототехнике среди школьников и студентов.

Каждая встреча, еще точнее — каждый день таких встреч, открывал новые черты этого удивительного, многогранно увлеченного, эрудированного и энергичного человека.

Первый раз я пришел в его кабинет в университете Ле Мана, где он был профессором и руководил лабораторией информатики. Это было просторное помещение, настолько заполненное книгами, что зал вполне мог бы считаться и хорошей библиотекой. Я пришел на эту встречу немного раньше назначенного времени, а Мартьяль Виве, еще консультировавший двоих своих аспирантов, заметил мое восхищенное впечатление от библиотеки и пригласил меня посмотреть книги. Огромное количество книг, ярко выражавшее направления интересов профессора Виве. Я обратил внимание сначала на те толстые тома, на которых его имя как автора стояло на обложке: “Дидактика и искусственный интеллект”, “Восприятие структур данных”, “Интеллектуальные обучающие системы”... О том, что Марсьяль Виве — известный программист (и, в частности, специалист по Лого) и организатор крупнейших в Европе конференций и комитетов по Лого), человек, сделавший очень много для подготовки и повышения квалификации школьных учителей в своей стране, о том, что он признанный ученый в Европе и в мире в области искусственного интеллекта, я уже знал. Но еще больше пришлось узнать в предстоящие дни. После разговора со мной он пригласил меня на свою встречу со студентами университета, где он регулярно читает свои лекции. Было интересно взглянуть на Марсьяля Виве в естественной для него среде: увидеть, как увлекательно талантливый ученый рассказывает о фундаментальных основах своей дисциплины, как талантливый педагог увлекает аудиторию своей эрудицией. Это была лекция по структурам данных в языках программирования, иллюстрируемая примерами на Лого.

На следующее утро он рекомендовал мне посетить одно из учебных помещений его университетской лаборатории, в котором я увидел не студентов, а рабочих с автомобильных заводов Ле Мана. Им предстояла интересная лабораторная работа — проектирование на компьютерах и отладка этих программ для использования на станках с числовым программным управлением. Токари и фрезеровщики с большим интересом осваивали для себя новые инструменты своей деятельности, понимая, что компьютеризация машиностроительного производства — это прямой путь к повышению квалификации их труда (и следовательно — их

заработка). Идея связи и эффективного слияния достижений университетской науки с практикой технологий машиностроительных заводов автомобильной индустрии предложена и реализована Марсьялем Виве в университете Ле Мана.

Прошел день, и я побывал в одном из лицеев города, где увидел молодую смену тех рабочих, с которыми виделся накануне, — с лицеистами, занимались школьные учителя и научные сотрудники лаборатории Виве, ведя уроки и практические занятия по программированию. И здесь главным инструментом был Лого.

Но следующий день оказался еще более волнующим — я посетил детский сад, в который университет Ле Мана по соглашению с муниципалитетом города поставил компьютеры. В детском саду я познакомил учителей начальной школы с методикой освоения программирования на Лого. Надо ли говорить, что имя Марсьяля Виве произносилось воспитателями детского сада с благоговейным придыханием и теплой добротой.

Необычайный диапазон направлений деятельности покорял. Поэтому с такой готовностью и признательностью я принял поручение президиума расположенной в Москве Международной академии информатизации передать ему важный и почетный документ об избрании Марсьяля Виве действительным членом Академии. Это поручение для меня совпало с одной из командировок во Францию, а для М.Виве — с его участием в юношеском чемпионате мира по роботостроению во Франции, в Ферте-Бернаре, для которого он подготовил команду из мальчиков и девочек Ле Мана. И хотя его диплом лежал в моем портфеле еще до начала чемпионата, я тайно от него договорился с оргкомитетом, что получу слово для краткого выступления на церемонии награждения после окончания соревнований. И оргкомитет, и я словно предчувствовали, что команда юных учеников Ле Мана взойдет на пьедестал победителей. Так оно и получилось: ученики получили свои призы, а их руководитель — диплом академика одновременно.

В заключение не могу не рассказать о еще одном ранее не известном мне умении Марсьяля Виве. Он в качестве руководителя французской делегации приехал в Переславль-Залесский на семинар по раннему обучению информатике. После обычных забот по расселению участники семинара направились в недалеко от гостиницы расположенную детскую музыкальную школу, в актовом зале которой дети-музыканты подготовили для гостей концерт. Прозвучала речь мэра. Для ответного слова вышел на сцену профессор Марсьяль Виве. Все ожидали, что он скажет какие-нибудь слова на французском языке, и беспокойно оглядывались, разыскивая глазами переводчика. Но Марсьяль Виве, улыбнувшись, засунул руку в карман пиджака, извлек из него маленькую флейту-пикколо, и по залу поплыла красивая, волнующая мелодия. Убедительнее всяких слов.

Михаил Романович Шура-Бура всегда связывал свою деятельность с двумя крупнейшими научно-педагогическими учреждениями Москвы — Институтом прикладной математики имени М.В. Келдыша, где он руководил сектором программирования, и МГУ им. М.В. Ломоносова, где он работал как университетский профессор, а затем — заведующим кафедрой системного программирования на факультете вычислительной математики и кибернетики.

Без громких слов можно сказать, что он внес существенный вклад в становление и развитие программирования в СССР. Правда, мне не довелось быть в числе его учеников, но я был с ним знаком по началу своей лаборантской “карьеры” в ИПМ АН СССР. А имя его всегда было на устах у тех, с кем мне пришлось работать, — с программистами первого поколения — С.С. Камыниным, Э.З. Любимским, В.С. Штаркманом... Все они прошли через Школу Михаила Романовича, вкусили радость общения с компьютером и увлеченность программированием как деятельности, совмещающей науку с высоким ремеслом. Именно они — первое поколение питомцев М.Р. Шуры-Буры — сумели передать нам, людям всего на несколько лет моложе их, беспредельное уважение к Учителю и его ремеслу.

Михаила Романовича я ни разу не видел в “иконостасе” из орденов и медалей за свой программистский несомненно заслуженный труд. И тем

не менее к многим фольклорным легендам о Шура-Буре немало добавляли две больших почетных доски на парадном входе в корпус ИПМ. Первая из них — Доска почета, на которой его фотография постоянно присутствовала, хотя и обновлялась от праздника к празднику. А вот вторая доска представляла собой перечисление спортивных рекордов Института. Половина ее — женская — состояла из разных имен и фамилий: в каждой из физкультурных дисциплин. В каждой дисциплине свой портрет. А во второй половине — мужской — был представлен только один персонаж — Михаил Романович Шура-Бура. Никому из более молодых сотрудников не удавалось соперничать с ним ни в беге, будь то стайерские или спринтерские дистанции, ни в плавании, ни в поднятии тяжестей, ни тем более в списке дисциплин, составлявших комплекс ГТО, — метание гранаты, подтягивания на турнике, стрельба в тире... Не задумываясь об этом, до последних своих дней (он скончался сравнительно недавно — в 2008 году, в возрасте 90 лет) Михаил Романович оставался примером в программистском труде и в повседневной жизни для всех тех, кто его знал.

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.

Вспоминается семинар в Троицке примерно 15 лет тому назад. Выступление американского учителя, завуча средней школы. У него дома — компьютер; жена увлечена самодеятельным театром, пишет сценарии (на своем компьютере), ставит спектакли; у 10-летнего сына — свой компьютер для игр. Помню наше удивленное непонимание: “Зачем столько машин в одной квартире?”.

Сейчас у меня дома стоит стационарный компьютер с системой Windows 7 (максимальная). Рядом стоит ноутбук, с которым приходится путешествовать. Последнее время в поездках приходится все чаще использовать режим удаленного доступа к своему стационарному компьютеру. Дома, в квартире, — локальная сеть. Мой компьютер — это сервер локальной сети. Поэтому к нему подключен принтер, к которому, в свою очередь, подключены все компьютеры домашней сети. Активно используется сканер. Компьютер, оснащенный богатой музыкальной периферией, у жены — композитора и музыкального педагога (на стационарном пишет музыку, аранжирует, на Apple iPad'e — отдыхает и развлекается) и у сыновей (компьютерная графика, дизайн). Дочки разъехались, живут далеко. Поэтому одна из самых используемых программ — электронная почта для получения писем, фотографий и роликов с эпизодами жизни многочисленных внуков... Электронную почту очень люблю (по существу, это мой главный инструмент) и активно использую в рамках дистанционного обучения младших школьников (курсы Роботландского сетевого университета). В этом виде деятельности приходится работать с сетевыми технологиями. Так, с учетом работы с младшим контингентом постоянно использую технологии фотоальбомов web2.0. Но в личном общении сетевые программы не люблю и по возможности избегаю. Регулярно в Интернете: сайт семинара, который надо готовить и проводить; деловые общения; новостные программы...

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?

Да, пожалуй, такая тема есть. Это — этика компьютерных коммуникаций, которая уходит корнями в старые литературные произведения эпистолярного жанра и обеспечивает современный скоростной, содержательный и эффективный стиль общения, о котором мог только мечтать Антуан де Сент-Экзюпери, сказавший: “Самая большая роскошь — это роскошь человеческого общения”. Очень обидно видеть, когда молодые люди в наши дни (включая студентов-информатиков) не понимают этого.

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

Да, есть. Это рекурсия, которая позволяет очень простыми средствами воспроизводить на компьютерном экране красивейшие и далеко не всегда предсказуемые рисунки.

Есть ли у Вас любимая задача или задачи?

Последняя из написанных мной книг (поздняя осень 2013 г.) названа “Алгоритмические и логические задачи начального курса информатики (из опыта дистанционного обучения информатике)”. Тягу (и даже любовь) к таким задачам воспитал во мне Наум Яковлевич Виленкин.

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

Никогда при нынешнем социальном строе не смогу заставить себя произнести перед детьми и учителями кощунственные слова, прописанные в первых статьях ФГОСа и декларирующие равные права и возможности всех учащихся в получении образования. Это откровенная неправда, которую не может в принципе превратить в реальность капиталистическое государство.

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?

Информатика в школе нужна вовсе не потому, что нужно решать определенный класс задач, составлять таблички и рисовать картинки, а потому что информатика учит детей думать, планировать, искать, общаться и учиться, а из всех школьных дисциплин дидактическим инструментарием для решения этих социальных и педагогических задач обладает только информатика. Эта теорема доказана А.П. Ершовым и его Школой.

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

Информатика — это социальная наука в том смысле, что содержание ее формируется требованиями современного развивающегося общества. Следовательно, станет она и через пять, и через десять лет такой, какой она будет нужна человеческому обществу.

Если же говорить терминами информационных технологий, то школьную информатику ждут

- конструктивное использование сенсорных интерфейсов;
- роботизация и 3D-прототипирование пропедевтического курса;
- внедрение сетевых технологий в методику обучения информатике и существенно более широкая информатизация межпредметных связей;
- освоение в школе принципов управления, рожденных суперкомпьютерами и параллельным программированием.

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

Этот вопрос прост, как мир: доживу ли я до этих светлых перспектив? И очень-очень хочется ответить — да, доживу!

Дуванов Александр Александрович

Выпускник механико-математического факультета МГУ (1976), преподаватель Благовещенского педагогического института (1976–1987), руководитель рабочей группы в лаборатории школьной информатики Института программных систем Академии наук СССР (1987–1991), главный разработчик ПМК “Роботландия”, с. н. с. НОУ “Роботландия+”, руководитель Роботландского сетевого университета, автор курсов: “Азы информатики”, “Азы программирования” (совместно с А.В. Рудем и В.П. Семенко), “Web-конструирование” и нового курса “Азбука Роботландии” (совместно с Н.Д. Шумиловой). В работе меня вдохновляют собаки. Я читаю им свои опусы, они с улыбкой внимают и никогда не позволяют себе критических замечаний в мой адрес.



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

Похоже, что в программирование я пришел, играя в волшебство. Первый раз столкнулся с невыдуманным волшебством в 6-м классе. Брат подарил мне книгу Е.Айсберга “Радио?.. Это очень просто!”. Свой первый радиоприемник я спаял на деревянной дощечке, которую выпилил ножовкой из доски, валявшейся в сарае. Пригладил ее рубанком, отполировал шкуркой, приклеил клеем БФ транзисторы, конденсаторы, резисторы... Потом спаял все это согласно схеме. Какой же была моя радость пополам с удивлением, когда эта деревяшка заговорила человеческим голосом! Своими руками я сотворил Буратино! Стало понятно, что я волшебник. Это было удивительное ощущение! Второй раз испытал подобные чувства, когда написал первую программу. Это случилось во время студенчества на механико-математическом факультете МГУ. А потом каждый раз, создавая программные коды, не переставал удивляться (не перестаю и сейчас!), как можно словом единым (на языке программирования) вдохнуть жизнь в железного (теперь) Буратино на моем рабочем столе! Хотя, после того как прочитал Евангелие от Иоанна, многое прояснилось:

Информация к размышлению

- Элементная база — интегральные микросхемы.
- ОЗУ на ферритовых сердечниках — 64–256 Кбайт.
- Накопители на сменных магнитных дисках емкостью 7,25 Мбайта.
- Накопители на магнитных лентах.
- Производительность — 20 тыс. операций в секунду.
- Занимаемая площадь — 100 кв. метров.
- Мощность, потребляемая процессором, — 4,5 кВт.
- Год прекращения производства — 1975-й.

Википедия



ЭВМ “ЕС-1020”

Информация к размышлению

Советская ЭВМ на полупроводниковых элементах “Днепр” выпускалась в течение десяти лет — с 1961 по 1971 год, сначала на киевском заводе “Радиоприбор”, затем на киевском заводе вычислительных и управляющих машин (ВУМ, позже “Электронмаш”), всего было выпущено около 500 машин. ОЗУ — от одного до восьми блоков по 512 слов. Быстродействие: на операциях сложения/вычитания — 20 000 операций в секунду, умножения/деления — 4000 операций в секунду. Занимаемая площадь: 35–40 кв. метров. Потребляемая мощность: 4 кВт.



ЭВМ “Днепр”

1. Вначале было Слово, и Слово было у Бога, и Слово было Бог.
2. Оно было в начале у Бога.
3. Все чрез Него начало быть, и без Него ничто не начало быть, что начало быть.

(Св. Евангелие от Иоанна 1:1-3)

Человек создан по образу и подобию Божию. И программисты, думаю, ближе всего по подобию стоят к Творцу — ведь Господь создал весь тварный мир словом одним. (Потом, как известно, слово это стало Христом, который пришел в мир, воплотившись в человека Иисуса.)

Не помню, как назывался компьютер, на котором студент Дуванов подрабатывал в МГУ оператором. Тогда компьютеры были еще дамами и назывались ЭВМ. Кажется, это была ЭВМ “ЕС-1020”. Помню, что зал, в котором стояли агрегаты, не был большим, но присутствовали все атрибуты славного семейства System/360.

“ЕС-1020” стали выпускать в 1971 году. Именно в этом году я поступил в МГУ, а стал подрабатывать оператором на 3-м курсе.

Сейчас на ПК работают “снаружи”. А раньше на ЭВМ работали “изнутри”. Программист заходил внутрь “системного блока”, устанавливал колоду карт (или бумажную ленту с дырочками) на устройство чтения, ставил “свою” бобину на магнитофон и садился за пишмаш (автоматическая пишущая машинка, подключенная к ЭВМ) общаться. Волшебное было время! Программист отстукивал на клавишах фразу, а ЭВМ печатала ответ на тот же листок бумаги — фантастика!

Впрочем, программистов редко пускали в “системный блок”. Здесь царил Оператор. Оператор брал колоду карт из корзины номер 1, а потом, после запуска, клал ее вместе с распечаткой АЦПУ (алфавитно-цифровое печатающее устройство) в корзину номер 2. Программист жадно хватал “портянку” и находил в ней, как правило, не ожидаемые результаты, а ироничное сообщение о стыдной, по своей простоте, синтаксической ошибке.

Оператор был, конечно, главнее программиста. Программист заискивал перед Оператором, надеясь на внеочередной прогон своей колоды, и мечтал сам оказаться внутри “системного блока”, чтобы насладиться общением с Буратино накоротке и без посредников.

Мне повезло! Будучи студентом мехмата, я был посвящен в Операторы ЭВМ, и мне даже платили за это деньги.

На военной кафедре мы программировали в машинных кодах на ЭВМ “Днепр”.

В моем военном билете была прописана специальность, которая потом долгое время вводила в ступор военных начальников, заведующих курсами переподготовки резервистов. Эта специальность обозначалась так: “инженер алгоритмизации и программирования оператив-

но-тактических задач войск ПВО страны”. В итоге меня направляли “партизанить” в разные ракетные подразделения, цепляясь за спасательный круг “войск ПВО страны”, но один раз почему-то направили в автомобильную часть. Впрочем, “партизаны” валяли дурака везде одинаково, ибо заниматься с нами никому было неохота. Вот что напомнил мне Миша Урюпин, мой одногруппник, о тех временах:

“На 3-м курсе на военной подготовке нас учил программированию полковник Долбышев.

Начинали с ЭВМ “Днепр” (память на ферритовых кольцах!) и с программирования в машинном коде (даже не в автокоде, а в восьмеричных числах).

Писали в машинном коде коротенькие программы типа нахождения максимума/минимума в массиве чисел.

Потом пошел АЛГОЛ и, кажется, была ЭВМ “Минск”.

Тут пошли уже программы посложнее. Я помню последнюю программу, которую тогда писал, — вычислить, под каким углом ракета воткнется в самолет при заданном алгоритме наведения (а вот алгоритм уже не помню, что-то типа “трех точек”). Задача была плоская (в вертикальной плоскости), скорости постоянные, самолет летит прямым курсом. Общение с ЭВМ проходило так: сдавали перфоленты (которые сами набивали) и потом на следующий день получали распечатку.

Должен покаяться перед нашими доблестными военными преподавателями: это я тогда весной 1974 года подложил перфоленту с программой распечатки всей памяти ЭВМ (хорошо, что она была маленькая) с периодическими вкраплениями фразы “ну ни фигя себе!”. Говорят, скандал был, но меня не вычислили и никого не наказали.

А еще нам на мехмате читали вычислительные методы (доц. Дьяченко) и рекомендовали книгу Джермейн “IBM 360”. Великая книга! У меня была, но потом кому-то она потребовалась больше, чем мне. Оттуда немного почерпнул Фортрана.

На сборах в военном лагере мы с тобой (помнишь?!) вдвоем писали на Фортране программу для нашего начвзвода что-то по длительности светового дня (для учета влияния солнечной активности на радиолокацию, кажется). Полковник Долбышев наше творение оценил вполне прилично.

С середины 4-го курса многие из наших одногруппников стали делать курсовые работы в Институте механики на ЭВМ “Одра”. Машина стояла на третьем этаже в холле. Писали на Алголе. Сдавали текст программы, написанный на бланке, — получали перфоленту. Результаты работы программы (распечатки) выдавали по расписанию (через несколько часов).

На 5-м курсе это стало повсеместным. Даже наших девочек заставляли что-то считать. Программы мы им писали, но встраивали разные шалости: у Галочки Кузьминой, например, аварийное сообщение имело вид “Runge is good, but Galka is the best” (автор Борька Круг), “Runge” — Рунге-Кутт — метод решения обыкновенных дифференциальных уравнений. Нечто подобное и я для Людочки написал. Но потом это безобразия пресекли операторы перфолент (а самую запоминающуюся звали Сусанна) и такие комментарии больше не печатали.

Еще я подрабатывал по хоздоговору на кафедре и считал на этой “Одре” полеты ракеты с оптимальным управлением для ЦКБ “Алмаз”. И даже свою дипломную работу тоже для них же сделал».

В это же время я увлекся шагающими аппаратами и дипломную работу писал по этой теме, активно используя ЭВМ “Одра” из Института механики МГУ.

На странице <http://lab301.imec.msu.ru/history.html> описана история Лаборатории общей механики, к которой я был прикреплен. В частности, про “шагалки” сказано так:

Информация к размышлению

ЭВМ “Одра” производства ПНР (МЭРА-ЭЛЬВРО). Это была машина, построенная на принципах архитектуры ICL-1900 английской компьютерной компании International Computers Ltd. В базовой комплектации машина была оснащена памятью объемом 32 Кб 24-разрядных слов. Перфораторы, магнитофоны, магнитные барабаны. В комплект поставки также входило базовое программное обеспечение, включающее язык ассемблера, несколько диалектов Фортрана, Алгол-60.



“Искра-1256”

Информация к размышлению

“Искра-1256” — советская микро-ЭВМ. Разработана в 1979 году. Выпускалась серийно в 1980-х гг. на производственном объединении “Счетмаш”, город Курск.
 Элементная база: интегральные схемы.
 ОЗУ — от 4 до 64 Кб.
 Быстродействие: для коротких операций — 500 тыс. оп./с., для арифметического сложения — 1 млн оп./с.



“Искра-226”

Информация к размышлению

По сравнению с “Искрой-1256” это было чудо техники.
 На этой машине работали Бейсик и ассемблер. ОЗУ 128 Кб.
 Двойной дисковод (НГМД) для 8-дюймовых дисков.
 Монохромный, но графический дисплей, принтер.

“В 70-х годах прошлого века в лаборатории был создан один из первых в мире шестиногий шагающий аппарат. Затем были разработаны шагающий аппарат с двумя телескопическими ногами и антропоморфный. Была построена теория управления двуногой ходьбой. Была разработана теория управления манипуляционными системами с силовой обратной связью”.

Я как раз занимался в рамках своей дипломной работы разработкой походки двуножного аппарата. Не знаю, насколько полезен был мой вклад для лаборатории (скорее всего малополезен), но мне самому поставленная задача очень нравилась, работал с большим интересом. После окончания МГУ распределился преподавателем на кафедру математического анализа Благовещенского пединститута.

Те, кто заражен бациллой программирования, неизлечимы! Сначала на полставки совмещал в местном статистическом управлении программистом на большой ЕС ЭВМ. Причем — о счастье! — мне выделяли время, в которое я лично входил в машинный зал и работал в нем плановый час совсем один натурально подобно Господу!

В рамках своей кафедры в пединституте инициировал создание ЛПМ — лаборатории прикладной математики. Состав творческой группы этой лаборатории: преподаватели А.А. Дуванов и В.В. Немиловитова и студенты О.Г. Какаулин, Ю.В. Прашкович и О.Д. Десятириков. Первый компьютер в ЛПМ — “Искра-1256” с символьным монохромным дисплеем.

На этом аппарате была написана первая версия исполнителя Кукарача — сама среда исполнителя, редактор программ и интерпретатор “тараканьего” языка (Кукарача тогда звался Тараканом).

Сейчас это трудно представить — “Искра-1256” имела оперативную память в 4 килобайта, никаких жестких дисков, единственное внешнее запоминающее устройство — магнитофон с бытовыми компакт-кассетами! Как мы тогда умудрились записать Кукарачу в 4 килобайта ОЗУ — ума не приложу! Программировали на собственном языке “Искра-1256”, больше напоминающем автокод.

Впрочем, может, я и ошибаюсь. Вполне возможно, что на “Искре-1256” мы (я и тогдашний студент Олег Какаулин) создали лишь исполнителя “Тараканчик”, который работал в командном режиме, а программируемого Таракана реализовали на “Искре-226”, которая появилась у нас несколько позже.

В то же примерно время мы активно баловались с программируемыми микрокалькуляторами. Писали для них разные программы, в том числе обучающие и игровые.

В это время стали появляться Ямахи. В Благовещенск поступил один класс (девять ученических и один учительский компьютер, объединенные в локальную сеть). Но этот класс не достался пединституту.

В 1986 году состоялся переезд в город Переславль-Залесский, и работа была продолжена в Институте программных систем, в лабора-



Программируемый микрокалькулятор “МК-54”

Информация к размышлению

“МК-54” — программируемый микрокалькулятор с обратной польской записью для проведения инженерных расчетов.

Память — 98 команд и 14 регистров, быстродействие — около пяти простых операций в секунду. При выключении калькулятора содержимое памяти стирается.

Википедия



“УКНЦ”

Информация к размышлению

Центральный процессор: на тактовой частоте 8 МГц, быстродействие — около 600 тыс. оп./с., ОЗУ — 64 Кбайта.

Исполнение — моноблок, системный блок совмещен с клавиатурой.

Выпускался серийно с конца 1987 года на заводах Минэлектронпрома: “Квант”, г. Зеленоград, Солнечногорский электромеханический завод (СЭМЗ), “Мезон” (Кишинев), “Мион” (Тбилиси) и “Нуклон” (Шяуляй). По некоторым сведениям, всего было выпущено около 310 000 ЭВМ, основная масса машин — в Зеленограде и Солнечногорске. Было сформировано и поставлено около 22 000 классов УКНЦ. В 1991 году выпуск машин был прекращен из-за снижения спроса — начались первые поставки КУВТ на базе IBM PC-совместимых компьютеров.

Википедия



“Ямаха”

Информация к размышлению

Ямаха КУВТ — комплекс учебной вычислительной техники на основе бытовых компьютеров стандарта MSX компании Yamaha.

Применялись на первом этапе информатизации образования в некоторых учебных заведениях СССР и эксплуатировались с середины 1980-х по начало 1990-х годов. Компьютеры, используемые в составе комплексов, представляли собой серийные модели, специально адаптированные для поставок в СССР. Они имели русифицированную клавиатуру и программное обеспечение, а также логотипы КУВТ и КУВТ2 (цифра обозначала версию стандарта MSX).

Википедия

тории школьной информатики, которой руководил Юрий Абрамович Первин.

В рамках лаборатории работало несколько групп, среди которых была группа “Языковые приложения школьной информатики”. Эта группа и стала центром будущей Роботландии.

Первая версия системы была готова уже к лету 1987 года, и на ее базе заработала одна из кафедр детского компьютерного лагеря, который традиционно проводил Институт программных систем. Это было самое начало, первый шаг.

Начинали работать мы на Ямахе.

Это был прекрасный цветной графический компьютер! Первый раз, когда я увидел, как кто-то играл на нем (это случилось в Новосибирске), в графике, динамике и цвете, то просто ошалел от восторга! Вот это да! Вот это техника! Фантастика, воплощенная в реальность!

Изначально на всю группу разработчиков у нас в Переславле был один компьютер. И всего одна дискета емкостью в 1,44 Мб, которую мы делили, как и компьютерное время на Ямахе. Писали на Си.

Состав основной группы разработчиков: А.А. Дуванов (руководитель), Я.Н. Зайдельман, М.А. Гольцман, Н.Б. Дроздов, А.А. Русс (художник). В разработках активное непосредственное участие принимал и руководитель лаборатории Ю.А. Первин. Отдельные исследования и работы выполняли: Ю.Ю. Рупасов, Р.А. Сулейманов, С.В. Дементьев, И.Г. Лилитко, В.Ф. Забавина, А.В. Голубев, Э.Н. Ермаков.

Затем мы дорабатывали и переписывали Роботландию на УКНЦ.

И, наконец, начали работать на первых IBM PC-совместимых компьютерах.

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

Сразу на ум приходят три имени, которые оказали на меня большое влияние: **Кнут** (по причине его замечательного многотомного труда “Искусство программирования для ЭВМ”), **Дейкстра** (впечатлил идеями структурного программирования), **Ершов** (как основоположник школьной информатики). Можно, конечно, продолжить ряд замечательных имен: Пейперт, Джобс, Мейер, Страуструп, Танен-

баум... Но Кнут, Дейкстра и Ершов — самые значимые для меня в этом списке.

Дональд Эрвин Кнут



“Лучший способ в чем-то разобраться до конца — это попробовать научиться этому компьютер”.

Дональд Эрвин Кнут родился 10 января 1938 г. в Милуоки, Висконсин, США. Американский ученый, почетный профессор Стэнфордского университета и нескольких других университетов в разных странах, преподаватель и идеолог программирования, разработчик нескольких известных программных технологий, создатель настольных издательских систем TEX и METAFONT, предназначенных для набора и верстки книг. Автор книг “Искусство программирования” (в четырех томах), “Все про TEX”, “Все про METAFONT” и др.

Большое влияние на юного Дональда Кнута оказали работы Андрея Петровича Ершова, впоследствии его друга.

Википедия

Эдсгер Вибе Дейкстра



“Студентов, ранее изучавших Бейсик, практически невозможно обучить хорошему программированию. Как потенциальные программисты они подверглись необратимой умственной деградации”.

“Программирование на КОБОЛе калечит мозг, поэтому обучение ему должно трактоваться как преступление”.

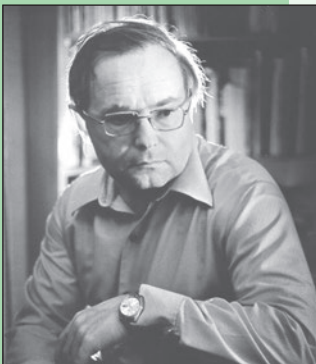
Дейкстра говорил, что самый главный язык, который должен знать программист, — свой родной, на котором он изъясняется в повседневной жизни.

Эдсгер Вибе Дейкстра (11 мая 1930, Роттердам, Нидерланды — 6 августа 2002, Нуенен (англ.), Нидерланды) — нидерландский ученый, идеи которого оказали влияние на развитие компьютерной индустрии.

Известность Дейкстре принесли его работы в области применения математической логики при разработке компьютерных программ. Он активно участвовал в разработке языка программирования Алгол и написал первый компилятор Алгол-60. Будучи одним из авторов концепции структурного программирования, он “проповедовал” отказ от использования инструкции GOTO. Также ему принадлежат идея применения “семафоров” для синхронизации процессов в многозадачных системах и алгоритм нахождения кратчайшего пути на ориентированном графе с неотрицательными весами ребер, известный как Алгоритм Дейкстры. В 1972 году Дейкстра стал лауреатом премии Тьюринга.

Википедия

Андрей Петрович Ершов



Название выступления, которое стало лозунгом начала перестройки школьного образования: “Программирование — вторая грамотность”. Само выступление начиналось такими словами: “Решив так назвать свое выступление, я сознаю, что это — метафора, которая многим покажется рискованной. По одну сторону нашего уравнения — экзотическая, хотя уже и весьма массовая профессия, требующая способности и длинного обучения, а по другую — общее достоинство, фундаментальнейшее свойство современного человека. Тем не менее я постараюсь продемонстрировать поучительность и плодотворность этой метафоры”.

Не могу не привести еще одну замечательную цитату из этой статьи: “Мы привычно понимаем грамотность как способность человека воспринять и выразить знание в текстовой форме. С детства мы слышим простые и емкие слова Максима Горького: “Любите книгу, источник знания”. Однако остается проблема: как перейти от знания к действию.

“Сообразуйте действие со словом, а слово с действием”, — говорит один из героев Шекспира. Вот здесь и возникает программирование”.

Андрей Петрович Ершов (19 апреля 1931, Москва — 8 декабря 1988, Москва) — советский ученый, один из пионеров теоретического и системного программирования, создатель Сибирской школы информатики, академик АН СССР. Его работы оказали огромное влияние на формирование и развитие вычислительной техники не только в СССР, но и во всем мире.

В 1981 году на 3-й Всемирной конференции Международной федерации по обработке информации и ЮНЕСКО по применению ЭВМ в обучении в Лозанне (Швейцария) делает доклад под названием “Программирование — вторая грамотность”. Название доклада быстро становится лозунгом. В Новосибирске начинаются эксперименты по преподаванию программирования, а затем и информатики школьникам. Разрабатываются компьютер “Агат”, обучающая система “Школьница” и язык “Рапира”. В 1985 году Ершовым совместно с группой соавторов был выпущен школьный учебник “Основы информатики и вычислительной техники” (ОИВТ), и началось преподавание информатики как учебного предмета во многих школах Советского Союза. Для записи алгоритмов в этом учебнике применялся Алгоподобный язык, так называемый “Русский алгоритмический язык” (или “Учебный алгоритмический язык”), в шутку называемый “Ершол”. Реализацией этого языка стал Е-практикум, разработанный на механико-математическом факультете МГУ.

Википедия

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.

По натуре я матерый консерватор. Перехожу на что-то новое лишь тогда, когда на старом работать становится совершенно невозможно. Хотя электронные “читалки” стали для меня приятным исключением — не предполагал, что пользоваться ими так удобно. Но запах, шелест, вид и дух бумажных книг не восполняются электронными чернилами...

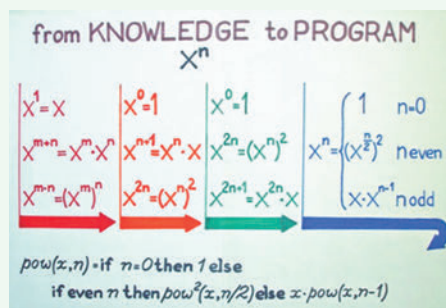
Меня нет ни в одной социальной сети. С самого начала их появления у меня возникло к ним неприятие, которое со временем только усиливается. Мой основной способ общения с людьми — электронная почта.

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?

Пожалуй, такую лекцию я посвятил бы принципам построения синтаксических анализаторов, если предположить, что мне действительно захотелось бы прочитать лекцию!

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

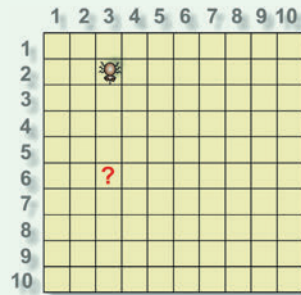
Ну, вот хотя бы алгоритм возведения числа x в целую степень n , приведенный в цитируемой выше статье-выступлении А.П. Ершова:



Есть ли у Вас любимая задачка или задачки?

Задачи для исполнителя Кукарача. В языке исполнителя нет переменных, поэтому любые задания, связанные с числовыми данными и вычислениями, кажутся невероятными. Однако можно показывать ответ положением Кукарачи на поле, конструируя его с помощью номеров строк и столбцов, а также, что особенно интересно, использовать отложенные в рекурсии команды, выполняющиеся столько раз, сколько было рекурсивных вызовов.

Вот простейший пример. Где-то под Кукарачей на клетчатом поле расположен кубик с буквой. Нужно дойти до кубика, толкнуть его и вернуться в исходную клетку.



Если есть переменные, считаем шаги до кубика, потом в цикле возвращаем Кукарачу назад. Но если переменных нет, можно решить задачу и так:

ЭТО Поиск

ВНИЗ

ЕСЛИ ПУСТО ТО Поиск

ВВЕРХ // Сработает столько, сколько работала команда ВНИЗ
КОНЕЦ

А вот еще один пример задания для Кукарачи, на этот раз — построение лексического анализатора!

Задача. Число

Введем определение числа.

Определение

<число> ::= <целое> | <дробное>

<целое> ::= <цифра> | <целое><цифра>

<дробное> ::= .<целое> | <целое>. | <целое>.<целое>

<цифра> ::= 0 | 1

Напишите программу для Кукарачи, которая проверяет, является ли запись на его поле числом в смысле заданного определения. В начальный момент исполнитель расположен перед кубиками с записью во второй строке своего поля.



Если проверяемая запись — число, установить исполнителя в конец записи.



Если запись — не число, поставить исполнителя под первым неверным символом.



Решение

Таблица переходов

Состояние	Следующий символ			
	0, 1	.	пусто	остальное
(вход)	(целое)	(дробное?)	(ошибка)	(ошибка)
(целое)	(целое)	(дробное)	(ответ_число)	(ошибка)
(дробное?)	(дробное)	(ошибка)	(ошибка)	(ошибка)
(дробное)	(дробное)	(ошибка)	(ответ_число)	(ошибка)
(ответ_число)				
(ошибка)				

Программа

```

ЭТО вход // Состояние (вход)
 шаг // Смотрим следующий символ
 ЕСЛИ 0 ТО целое
 ИНАЧЕ ЕСЛИ 1 ТО целое
 ИНАЧЕ ЕСЛИ . ТО дробное?
 ИНАЧЕ ошибка
 КОНЕЦ

ЭТО целое // Состояние (целое)
 шаг // Смотрим следующий символ
 ЕСЛИ ПУСТО ТО ответ_число
 ИНАЧЕ ЕСЛИ 0 ТО целое
 ИНАЧЕ ЕСЛИ 1 ТО целое
 ИНАЧЕ ЕСЛИ . ТО дробное
 ИНАЧЕ ошибка
 КОНЕЦ

ЭТО дробное? // Состояние (дробное?)
 шаг // Смотрим следующий символ
 ЕСЛИ 0 ТО дробное
 ИНАЧЕ ЕСЛИ 1 ТО дробное
 ИНАЧЕ ошибка
 КОНЕЦ

ЭТО дробное // Состояние (дробное)
 шаг // Смотрим следующий символ
 ЕСЛИ ПУСТО ТО ответ_число
 ИНАЧЕ ЕСЛИ 0 ТО дробное
 ИНАЧЕ ЕСЛИ 1 ТО дробное
 ИНАЧЕ ошибка
 КОНЕЦ

ЭТО ошибка // Перемещение остатка записи
 шаг // в первую строку
 ЕСЛИ НЕ ПУСТО ТО ошибка // Возврат к месту ошибки
 ВЛЕВО
 КОНЕЦ

ЭТО ответ_число
 ВВЕРХ
 КОНЕЦ

ЭТО шаг // Толкнуть следующий
 ВНИЗ ВПРАВО ВВЕРХ // символ
 КОНЕЦ

```

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

Не стал бы рекламировать социальные сети, но рассказал бы об онлайн-офисе Google. Рекламировать социальные сети бессмысленно —

и без рекламы в них “зависают” многие, от мала до велика, уходя в трансцендентную реальность, от мира за окном в стене в мир за окном Windows.

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?

Все зависит от того, какие задачи мы возлагаем на предмет “информатика”. Уверен, информатика не должна учить компьютеру, сетям, прикладному ПО. Школьная информатика, как и любой школьный предмет, должна быть фундаментальной. Если на пении дети будут только петь, а учитель — расставлять их в красивый хор, получатся развлекательные сеансы с массовиком-затейником. Если на информатике дети будут только рисовать, писать и что-то еще делать на компьютере или в социальных сетях, то, как бы красиво это ни называлось, деятельность эта будет не обучающей, а развлекательной. На уроках пения должны изучаться основы музыкальной грамоты, а на уроках информатики — **основы алгоритмизации**.

Что это даст детям? Прежде всего научит их **мыслить алгоритмически**. Алгоритмическое мышление (наряду с другими) непременно надо закладывать в младшем возрасте (вот почему мне особенно интересна информатика для младшей школы), оно полезно в любой деятельности, и особенно (как это ни покажется странным) в той, которая не связана с компьютером. Ведь **алгоритмическое мышление — это способность облечь абстрактную идею в последовательность конкретных шагов, необходимых для ее воплощения на практике**.

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

Мне бы хотелось предположить, что компьютер не заменит учителя и что учитель по-прежнему останется главным действующим лицом в обучающем процессе как через 10, так и через 100 лет.

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

Мне интересны вопросы, которые возникают в процессе творческой работы, будь то написание компьютерной обучающей программы или курса информатики. Такие самовольные вопросы рождают порой весьма неординарные ответы, вспоминать о которых — одно удовольствие!

журнал

Информатика – Первое сентября

2-е полугодие 2014 года

ПОДПИСКА

на сайте www.1september.ru и в почтовых отделениях РФ

МЕЖРЕГИОНАЛЬНОЕ АГЕНТСТВО ПОДПИСКИ

КАТАЛОГ РОССИЙСКОЙ ПРЕССЫ

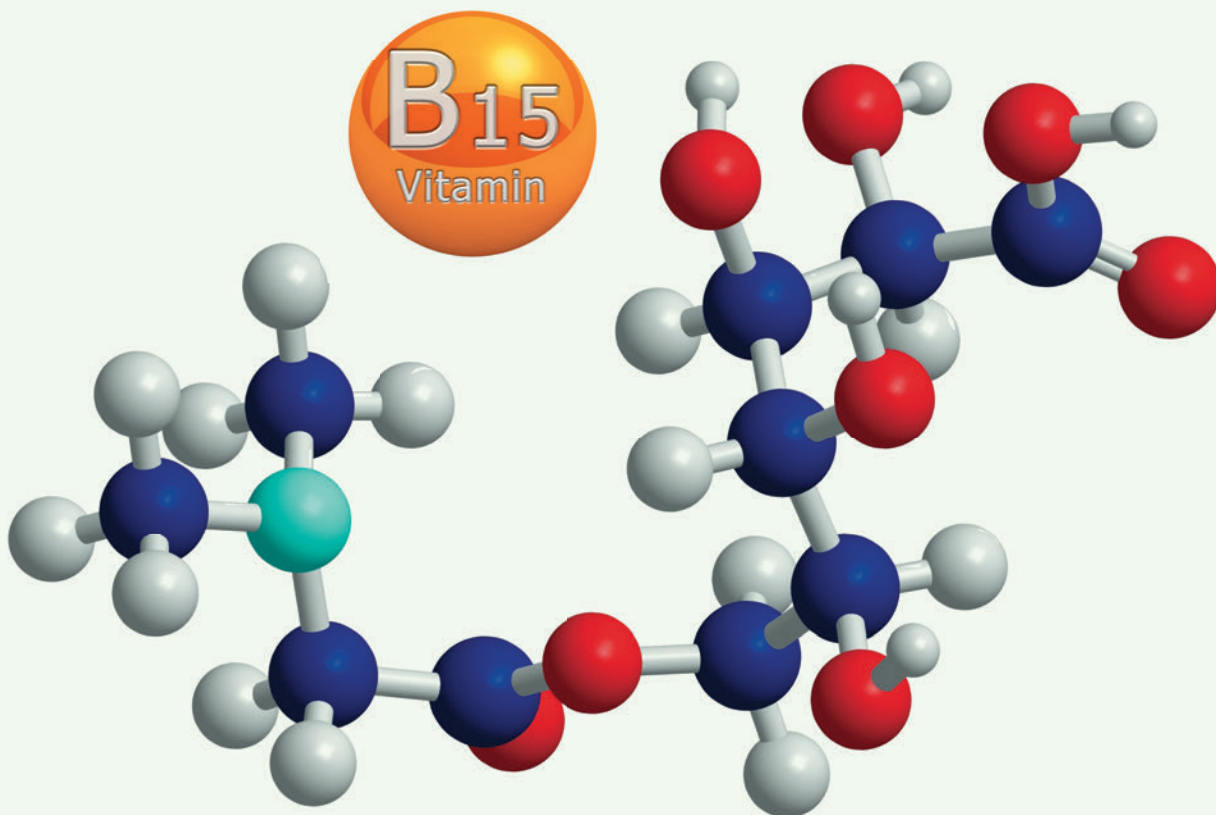
ПОЧТА РОССИИ

2014
второе полугодие

Индекс	Название издания	Периодич. 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. <i>В июле не выходит.</i> <i>Подписка на июль не принимается.</i> (-) 160 г 64 стр.	5	308.00		1540.00	
12684	Информатика – Первое сентября. Электронная версия на CD (полная копия бумажной версии) <i>В июле не выходит.</i> <i>Подписка на июль не принимается.</i> (-) 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





Люблю ЕГЭ за B15, или Еще раз про метод отображения

► Задание на решение системы логических уравнений остается в ЕГЭ одним из самых сложных. Но решение этой системы не только проверяет знание логических операций и умение считать у наших школьников, но и учит рассуждать, строить логические цепочки. Конечно, оно незаслуженно находится в части В. При оценке ответа нет возможности квалифицировать ошибку, так как ответ, как и логическое высказывание, бывает либо истинным, либо ложным. А поводов дать неверный в этом случае ответ много: можно написать наугад, а можно решить все от начала до конца, проделать все логические

преобразования, выстроить верную цепочку рассуждений и в последнем действии допустить арифметическую ошибку. Заметим, что при решении этого задания количество только арифметических действий доходит до 40. Но тут у выпускников и учителей нет выбора. Будем действовать по схеме — сначала купили, потом полюбили. За что можно полюбить это задание? Например, за то, что оно не скучное, что в нем больше разнообразия, чем в задачах на количество информации, где надо просто применить формулу, или в задачах про системы счисления, в которых надо освоить три алгоритма. На примере задания B15 можно еще раз поговорить о сложных понятиях информатики: “деревья”, “графы”, “матрица смежности”. А самое главное, B15 учит думать и рассуждать.

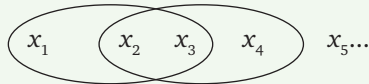
Существует много вариантов решения задания B15, но в этой статье все системы будут решены методом отображений, который разнообразен в своем применении и опубликован в № 10 журнала “Информатика”, 2013 г. Одним из ключе-

вых моментов при разборе систем является определение основного отображения, необходимого для решения системы. Обратите внимание, что в примерах знак “.” обозначает конъюнкцию, а “+” — дизъюнкцию.

Система 1

$$\begin{cases} x_1 \cdot (x_2 + \overline{x_3}) + \overline{x_1} \cdot (x_2 \oplus x_3) = 1 \\ x_2 \cdot (x_3 + \overline{x_4}) + \overline{x_2} \cdot (x_3 \oplus x_4) = 1 \\ \dots \\ x_8 \cdot (x_9 + \overline{x_{10}}) + \overline{x_8} \cdot (x_9 \oplus x_{10}) = 1 \end{cases}$$

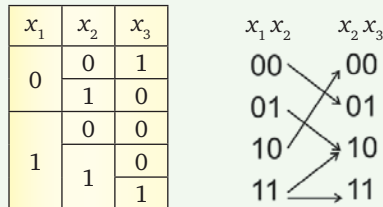
В этой системе в двух соседних уравнениях присутствует пара общих неизвестных



Зная количество пар (x_1, x_2) , можно найти количество пар (x_2, x_3) и найти общее количество решений первого уравнения системы, а продолжая применять правило, построенное для первого уравнения, можно найти количество пар (x_9, x_{10}) и определить, сколько раз дерево решений доведет до x_{10} , что и будет ответом к этому заданию. В цепочке рассуждений будем переходить от пары к паре:

$$(x_1, x_2) \Rightarrow (x_2, x_3) \Rightarrow (x_3, x_4) \Rightarrow (x_4, x_5) \Rightarrow \dots \Rightarrow (x_9, x_{10})$$

Построим дерево решений первого уравнения и отображение, соответствующее первому уравнению:



Пара	Количество пар								
	x_1, x_2	x_2, x_3	x_3, x_4	x_4, x_5	x_5, x_6	x_6, x_7	x_7, x_8	x_8, x_9	x_9, x_{10}
00	1	1	2	2	2	3	3	3	4
01	1	1	1	2	2	2	3	3	3
10	1	2	2	2	3	3	3	4	4
11	1	1	1	1	1	1	1	1	1

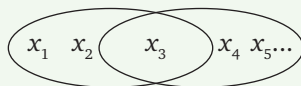
$$4 + 3 + 4 + 1 = 12$$

Ответ: 12

Система 2

$$\begin{cases} x_1 \cdot (x_2 + \overline{x_3}) + \overline{x_1} \cdot (x_2 \oplus x_3) = 1 \\ x_3 \cdot (x_4 + \overline{x_5}) + \overline{x_3} \cdot (x_4 \oplus x_5) = 1 \\ x_5 \cdot (x_6 + \overline{x_7}) + \overline{x_5} \cdot (x_6 \oplus x_7) = 1 \\ x_7 \cdot (x_8 + \overline{x_9}) + \overline{x_7} \cdot (x_8 \oplus x_9) = 1 \end{cases}$$

Эта система похожа на систему 1, но в ней меньше уравнений. В двух соседних уравнениях присутствует одна общая неизвестная.

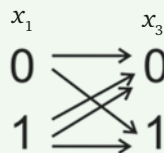


Цепочка рассуждений при построении решения системы будет такой:

$$x_1 \Rightarrow x_3 \Rightarrow x_5 \Rightarrow x_7 \Rightarrow x_9$$

Дерево решений совпадает с деревом первой системы. При решении этой системы можно построить другое отображение. x_1 может принять два различных значения и x_3 также два разных значения.

x_1	x_2	x_3
0	0	1
	1	0
1	0	0
	1	0
		1



Начиная строить дерево решений с $x_1 = 1$, можно построить три ветки, ведущие x_3 . Из них две идут к нулевому значению и одна к единице.

Таблица вычислений будет такой:

	Количество значений				
	x_1	x_3	x_5	x_7	x_9
0	1	3	7	17	41
1	1	2	5	12	29

$41 + 29 = 70$

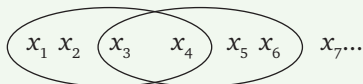
Ответ: 70

Разберем систему, в которой задать отображение с помощью стрелок не удобно.

Система 3

$$\begin{cases} x_1 \cdot (x_2 \rightarrow \overline{x_3}) + \overline{x_1} \cdot x_4 = 1 \\ x_3 \cdot (x_4 \rightarrow \overline{x_5}) + \overline{x_3} \cdot x_6 = 1 \\ x_5 \cdot (x_6 \rightarrow \overline{x_7}) + \overline{x_5} \cdot x_8 = 1 \\ x_7 \cdot (x_8 \rightarrow \overline{x_9}) + \overline{x_7} \cdot x_{10} = 1 \end{cases}$$

В этой системе в двух соседних уравнениях присутствует пара общих неизвестных



Каждое уравнение системы зависит от четырех переменных. Индексы соседних уравнений отличаются на 2. Общими переменными для соседних уравнений является пара переменных. Зная количество пар (x_1, x_2) , можно найти количество пар (x_3, x_4) . Ветки построенного дерева будут вести к элементу x_{10} , что соответствует последней паре (x_9, x_{10}) .

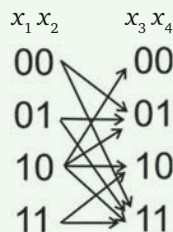
$(x_1, x_2) \Rightarrow (x_3, x_4) \Rightarrow (x_5, x_6) \Rightarrow (x_7, x_8) \Rightarrow (x_9, x_{10})$

Построим дерево решений первого уравнения:

$x_1 \cdot (x_2 \rightarrow \overline{x_3}) + \overline{x_1} \cdot x_4 = 1$

Построим дерево решений первого уравнения и отображение, соответствующее первому уравнению. Из каждой пары может выходить четыре стрелки и в каждую пару может входить до четырех стрелок. Запись отображения с помощью стрелок трудно читается, но стрелки не единственный способ для записи соответствия.

x_1	x_2	x_3	x_4
0	0	0	1
		1	1
	1	0	1
		1	1
1	0	0	0
		0	1
		1	0
		1	1
	1	0	x
		1	0
			0
			1



Можно полученное отображение записать в виде рекуррентных формул:

$$F(00) = F(10);$$

$$F(01) = F(00) + F(01) + F(10);$$

$$F(10) = F(10) + F(11);$$

$$F(11) = F(00) + F(01) + F(10) + F(11).$$

А можно задать отображение с помощью матрицы смежности:

		приемник			
		00	01	10	11
источник	00		+		+
	01		+		+
	10	+	+	+	+
	11			+	+

Пара	Количество пар				
	x_1, x_2	x_3, x_4	x_5, x_6	x_7, x_8	x_9, x_{10}
00	1	1	2	6	16
01	1	3	6	14	36
10	1	2	6	16	40
11	1	4	10	24	60

$$16 + 36 + 40 + 60 = 152$$

Ответ: 152

Система 4

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_3) \cdot (x_3 \rightarrow x_4) \cdot (x_4 \rightarrow x_5) = 1 \\ (y_1 \rightarrow y_2) \cdot (y_2 \rightarrow y_3) \cdot (y_3 \rightarrow y_4) \cdot (y_4 \rightarrow y_5) = 1 \\ x_4 \rightarrow y_4 = 1 \end{cases}$$

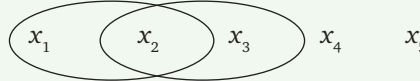
Первый способ

Два первых уравнения отличаются только буквами, значит, решение их будет одинаковым. Рассмотрим первое уравнение:

$$(x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_3) \cdot (x_3 \rightarrow x_4) \cdot (x_4 \rightarrow x_5) = 1$$

Его можно переписать в виде системы:

$$\begin{cases} x_1 \rightarrow x_2 = 1 \\ x_2 \rightarrow x_3 = 1 \\ x_3 \rightarrow x_4 = 1 \\ x_4 \rightarrow x_5 = 1 \end{cases}$$

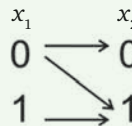


Цепочка рассуждений при построении решения системы будет такой:

$$x_1 \Rightarrow x_2 \Rightarrow x_3 \Rightarrow x_4 \Rightarrow x_5$$

Построим дерево решений и соответствующее ему отображение:

x_1	x_2
0	0
	1
1	1



Для второго уравнения системы рассуждения повторяются. В системе присутствует третье уравнение, связывающее первые два. Значит, необходимо заполнить две таблицы при разных значениях x_4 .

$$x_4 = 0$$

	x_1	x_2	x_3	x_4	x_5
0	1	1	1	1	1
1	1	2	3	0	1

два решения

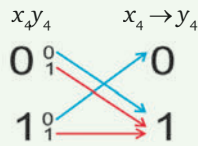
$$x_4 = 1$$

	x_1	x_2	x_3	x_4	x_5
0	1	1	1	0	0
1	1	2	3	4	4

четыре решения

Отображение для третьего уравнения (решение одного уравнения методом отображения также рассмотрено в журнале "Информатика", № 10/2013).

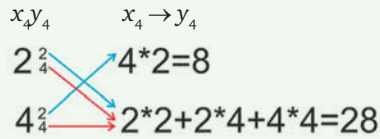
Отображение для импликации:



Ответ: 28

Второй способ

Вычисления:



$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_3) \cdot (x_3 \rightarrow x_4) \cdot (x_4 \rightarrow x_5) = 1 \\ (y_1 \rightarrow y_2) \cdot (y_2 \rightarrow y_3) \cdot (y_3 \rightarrow y_4) \cdot (y_4 \rightarrow y_5) = 1 \\ x_4 \rightarrow y_4 = 1 \end{cases}$$

Можно записать эту систему в виде одного уравнения, применив конъюнкцию к левым частям уравнений. Перегруппируем выражение в левой части уравнения и опять запишем в виде системы:

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (y_1 \rightarrow y_2) = 1 \\ (x_2 \rightarrow x_3) \cdot (y_2 \rightarrow y_3) = 1 \\ (x_3 \rightarrow x_4) \cdot (y_3 \rightarrow y_4) = 1 \\ (x_4 \rightarrow x_5) \cdot (y_4 \rightarrow y_5) = 1 \\ x_4 \rightarrow y_4 = 1 \end{cases}$$

Каждое уравнение системы зависит от четырех переменных. Общими переменными для соседних уравнений является пара переменных.

$$(x_1, y_1) \Rightarrow (x_2, y_2) \Rightarrow (x_3, y_3) \Rightarrow (x_4, y_4) \Rightarrow (x_5, y_5)$$

Построим дерево решений первого уравнения:

x_1	y_1	x_2	y_2	
0	0	0	0	
			1	
		1	0	
			1	
	1	0	0	x
			1	0
1		0	x	
		1	1	

Отображение зададим матрицей смежности и для удобства вычислений совместим в одну таблицу матрицу и вычисления. Так как по последнему уравнению подходят три пары (01), (01), (11), то количество пар (10) в четвертом столбике будет равно 0.

Матрица смежности

Количество решений

		приемник								
		00	01	10	11	x_1, y_1	x_2, y_2	x_3, y_3	x_4, y_4	x_5, y_5
ИСТОЧНИК	00	+	+	+	+	1	1	1	1	1
	01		+		+	1	2	3	4	5
	10			+	+	1	2	3	0	1
	11				+	1	4	9	16	21

$$1 + 5 + 1 + 21 = 28$$

Ответ: 28

Система 5

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_3) \cdot (x_3 \rightarrow x_4) \cdot (x_4 \rightarrow x_5) \cdot (x_5 \rightarrow x_6) = 1 \\ (x_1 \rightarrow y_1) \cdot (x_2 \rightarrow y_2) \cdot (x_3 \rightarrow y_3) \cdot (x_4 \rightarrow y_4) \cdot (x_5 \rightarrow y_5) \cdot (x_6 \rightarrow y_6) = 1 \end{cases}$$

Первый способ

Выражения в левой части первого и второго уравнений равны 1, следовательно, и произведение левых частей равно 1.

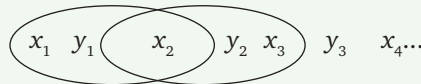
$$(x_1 \rightarrow x_2) \cdot \dots \cdot (x_5 \rightarrow x_6) \cdot (x_1 \rightarrow y_1) \cdot \dots \cdot (x_5 \rightarrow y_6) = 1$$

Перегруппируем множители:

$$(x_1 \rightarrow x_2) \cdot (x_1 \rightarrow y_1) \cdot \dots \cdot (x_5 \rightarrow x_6) \cdot \dots \cdot (x_5 \rightarrow y_5) \cdot (x_6 \rightarrow x_6) = 1$$

И запишем полученное уравнение в виде системы:

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_1 \rightarrow y_1) = 1 \\ (x_2 \rightarrow x_3) \cdot (x_2 \rightarrow y_2) = 1 \\ (x_3 \rightarrow x_4) \cdot (x_3 \rightarrow y_3) = 1 \\ (x_4 \rightarrow x_5) \cdot (x_4 \rightarrow y_4) = 1 \\ (x_5 \rightarrow x_6) \cdot (x_5 \rightarrow y_5) = 1 \\ x_6 \rightarrow y_6 = 1 \end{cases}$$



Общими переменными для соседних уравнений является одна переменная.

$$x_1 \Rightarrow x_2 \Rightarrow x_3 \Rightarrow x_4 \Rightarrow x_5 \Rightarrow x_6 \Rightarrow y_5$$

Для первого уравнения системы (и похожих на него 2, 3, 4, 5-го) дерево решений будет таким:	Первым пяти уравнениям будет соответствовать отображение:													
<table border="1"> <tr> <td>x_1</td> <td>y_1</td> <td>x_2</td> </tr> <tr> <td rowspan="2">0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> <tr> <td rowspan="2">1</td> <td>0</td> <td>X</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </table>	x_1	y_1	x_2	0	0	0	1	1	1	0	X	1	1	
x_1	y_1	x_2												
0	0	0												
	1	1												
1	0	X												
	1	1												
Отображение для последнего уравнения:														

По построенному отображению заполним таблицу для вычисления количества решений:

	Количество решений по первым пяти уравнениям						Количество решений после подключения последнего уравнения
	x_1	x_2	x_3	x_4	x_5	x_6	y_6
0	1	2	4	8	16	32	32
1	1	3	7	15	31	63	95

Ответ: $32 + 95 = 127$ решений

Второй способ

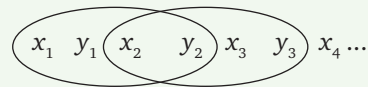
$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_3) \cdot (x_3 \rightarrow x_4) \cdot (x_4 \rightarrow x_5) \cdot (x_5 \rightarrow x_6) = 1 \\ (x_1 \rightarrow y_1) \cdot (x_2 \rightarrow y_2) \cdot (x_3 \rightarrow y_3) \cdot (x_4 \rightarrow y_4) \cdot (x_5 \rightarrow y_5) \cdot (x_6 \rightarrow y_6) = 1 \end{cases}$$

Так же, как и в предыдущем случае, перепишем систему в виде:

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_1 \rightarrow y_1) = 1 \\ (x_2 \rightarrow x_3) \cdot (x_2 \rightarrow y_2) = 1 \\ (x_3 \rightarrow x_4) \cdot (x_3 \rightarrow y_3) = 1 \\ (x_4 \rightarrow x_5) \cdot (x_4 \rightarrow y_4) = 1 \\ (x_5 \rightarrow x_6) \cdot (x_5 \rightarrow y_5) = 1 \\ x_6 \rightarrow y_6 = 1 \end{cases}$$

В первых пяти уравнениях по три переменных. Пара (x_1, y_1) определяет количество возможных сочетаний пары (x_2, y_2) , а пара (x_3, y_3) получается из пары (x_2, y_2) по точно такому же правилу. Но в первом уравнении нет y_2 , а во втором не хватает y_3 и так далее. Значит, первое уравнение не накладывает никаких ограничений на значения y_2 , а второе — на y_3 и т.д. Значит, их значения могут быть любыми. Добавим недостающие переменные в первые пять уравнений, не изменяя систему.

$$\begin{cases} (x_1 \rightarrow x_2) \cdot (x_1 \rightarrow y_1) + y_2 \cdot \overline{y_2} = 1 \\ (x_2 \rightarrow x_3) \cdot (x_2 \rightarrow y_2) + y_3 \cdot \overline{y_3} = 1 \\ (x_3 \rightarrow x_4) \cdot (x_3 \rightarrow y_3) + y_4 \cdot \overline{y_4} = 1 \\ (x_4 \rightarrow x_5) \cdot (x_4 \rightarrow y_4) + y_5 \cdot \overline{y_5} = 1 \\ (x_5 \rightarrow x_6) \cdot (x_5 \rightarrow y_5) + y_6 \cdot \overline{y_6} = 1 \\ x_6 \rightarrow y_6 = 1 \end{cases}$$

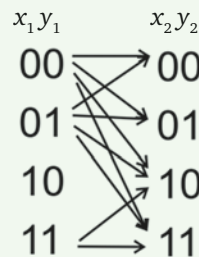


Каждое уравнение системы зависит от четырех переменных. Общими переменными для соседних уравнений является пара переменных.

$$(x_1, y_1) \Rightarrow (x_2, y_2) \Rightarrow (x_3, y_3) \Rightarrow (x_4, y_4) \Rightarrow (x_5, y_5) \Rightarrow (x_6, y_6)$$

Построим дерево решений первого уравнения и отображение множеств ему соответствующее.

x_1	y_1	x_2	y_2
0	0	0	0
		1	1
		0	0
	1	0	1
		1	0
		1	1
1	0	x	
	1	1	1



При заполнении таблицы будем использовать это отображение пять раз и вычислим количество пар (x_6, y_6) . Можно задать отображение матрицей смежности.

		приемник				Количество решений по первым пяти уравнениям					
		00	01	10	11	x_1, y_1	x_2, y_2	x_3, y_3	x_4, y_4	x_5, y_5	x_6, y_6
источник	00	+	+	+	+	1	2	4	8	16	32
	01	+	+	+	+	1	2	4	8	16	32
	10					1	3	7	15	31	63
	11			+	+	1	3	7	15	31	63

Последнему уравнению не удовлетворяет пара (10). Ответ будет складываться из значений последнего столбика, соответствующих парам (00), (01) и (11).

$$32 + 32 + 63 = 127$$

Ответ: 127

Решение задания В15 легко переносится в электронную таблицу или на язык программирования, и выбор можно оставить за учеником. Возможно, когда-нибудь так и будет. На пути к ответу ученик пройдет все этапы: от построения математической модели до ответа.



Школьники играют в CTF

В.В. Ильин, лицей
“Вторая школа”,
В.А. Павленко,
ГБОУ Школа-интернат
“Интеллектуал”

Хотите простую задачу?

МОЕ условие: Ежей 9, а ушей?..

ОТВЕТ:

Восемнадцать? И при чем здесь ежи?

Нет. Задача простая, но не совсем.

Решение — в статье.

На старт!

Известно много задач по алгоритмам и программированию, по этим задачам проводятся олимпиады. А вот задачи и конкурсы по компьютерным технологиям встречаются редко. На самом деле надо лишь знать, где их искать.

CTF (*Capture The Flag*) — это командные соревнования по компьютерной безопасности. Появились они в США уже давно (более 20 лет назад), а в России впервые были проведены на Урале в 2006 году. CTF существует в двух основных вариантах: классический CTF (атака-

защита сервера) и тасковый (task-based) CTF. Второй вариант более доступен школьникам, о нем мы и расскажем.

В тасковом CTF командам дается набор заданий (тасков) из разных категорий и областей знаний. Например, это могут быть задания на криптографию, стеганографию, веб-технологии. Команда пытается решить их любым доступным образом, применяя свои навыки работы с компьютерами или приобретая их по ходу соревнования. Результатом решения является некий ключ (он называется флаг), который и необходимо отправить на проверку.

Для школьников, во всяком случае в России, эти соревнования начали проводиться лишь пару лет назад. А ведь компьютерно-детективные загадки — это очень здорово.

Конечно, немногие школьники хорошо разбираются в современных алгоритмах шифрования, владеют различными способами работы с изображениями или аудиофайлами и знают подробности функционирования Интернета. От участников

требуется не столько владеть знаниями, сколько уметь быстро их получать из Интернета: из статей, документации, литературы. Знания приходится тут же применять на практике.

С начала и до конца марта этого года в Интернете проводилась UFO CTF School — открытая олимпиада Южного федерального университета по информационной безопасности среди школьников. В первых трех частях нашей статьи мы расскажем о некоторых задачах этого конкурса.

На момент написания статьи задачи олимпиады доступны по адресу: <http://ufologists.ictis.sfedu.ru/game>, — но отправлять флаги в проверяющую систему уже нельзя. Обычно CTF-площадки закрываются через некоторое время после проведения соревнования, потому что их поддержание в рабочем состоянии требует сил. Для проведения CTF нужно не только опубликовать условия задач. Некоторые задачи заключаются в анализе сетевых сервисов: сайтов, удаленных компьютеров, и от организаторов конкурса требуется поддерживать их работоспособность.

Некоторые задачи доступны постоянно, например, на сайтах:

<http://5kr.mosuzedu.ru/CTFtasks>

<https://picocft.com/>

<https://ctf.fluxfingers.net/2013/challenges>

Разгон!

Base64

Декодируй это: “ZmxhZ3s2Y2FiOGNlMjRkNTNkYzQ1MGFiNTU0MDliZWZ2ZTQ1Mn0”.

Что такое Base64? Идем в Википедию. “Эта система широко используется в электронной почте для представления бинарных файлов в тексте письма (транспортное кодирование). Все широко известные варианты, известные под названием “Base64”, используют символы A–Z, a–z и 0–9, что составляет 62 знака, для недостающих двух знаков в разных системах используются различные символы”.

Первой ссылкой в поисковиках по запросу “Base64” видим “Online Base64 декодер”. Технически задача оказывается очень простой — скопируем текст в окошко декодера и получим флаг.

Что же, задачи CTF такие простые, и надо лишь искать перекодировщики между разными форматами? Конечно, нет. Но даже такая простая задача могла быть сложнее. Например, названия кодировки могло не быть в условии, и тогда нужно было бы выяснить, какая именно кодировка используется.

Главное, что в процессе решения у неопытного школьника произойдет знакомство с новой технологией из компьютерного мира. А дальше когда-нибудь он может встретить Base64 как одну из технологий в реальном большом проекте. Например, при передаче данных по сети. И тогда уже школьник сможет спра-

виться с более сложной задачей: например, получить доступ к плохо защищенному веб-сервису.

Veni vidi vici

Что же это: “Wkh Txlfn Eurzq Ira Mxpsv Ryhu Wkh Odcb Grj”?

Это уже не просто кодирование, а шифрование. Нужно догадаться, что это за шифр, и расшифровать сообщение. “Пришел, увидел, победил” — фраза Юлия Цезаря. А перед нами шифр Цезаря: алфавит записывается по кругу, и каждая буква меняется на своего соседа справа на расстоянии k букв. Как узнать, чему равно k в нашем случае? Это можно решить подбором (в худшем случае мы попробуем 25 вариантов). Поскольку шифр Цезаря очень известен, то в Интернете есть его дешифровщики: например, <http://online-calculators.appspot.com/caesar/>.

Можно написать и свой дешифровщик на Питоне.

```
text = list(input('Enter message: '))
lowercase = list('abcdefghijklmnopqrstuvwxyz')
uppercase = list('ABCDEFGHIJKLMNOPQRSTUVWXYZ')
key = int(input('Enter key: '))
cipher = []
for c in text:
    if c in lowercase:
        cipher.append(lowercase[(lowercase.index(c) - key) % (len(lowercase))])
    elif c in uppercase:
        cipher.append(uppercase[(uppercase.index(c) - key) % (len(uppercase))])
    else:
        cipher.append(c)
print('Your decrypted message is:',
      ''.join(cipher))
```

А вот простая задача по другой теме.

CookieMonster

КукиМонстру нужны печеньки!

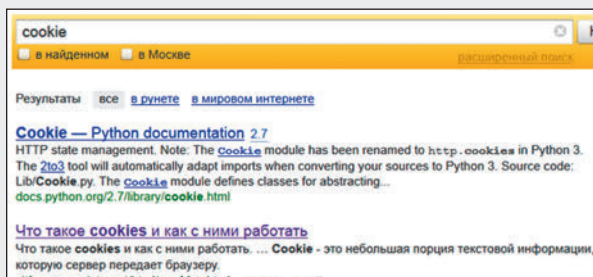
<http://188.226.160.200:802/>

По ссылке открывалась картинка

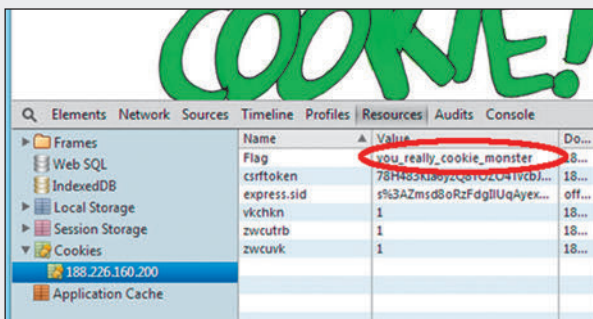


Пытаемся искать скрытый текст на картинке, анализируем количество дырочек в печенье, гуглим имя персонажа из мультфильма — ничего интересного. Как же накормить это забавное существо?

Cookie...



Ага! Может быть, имеются в виду веб-куки? Читаем статью и смотрим куки на страничке с картинкой — это можно сделать стандартными средствами любого браузера. Гуглим так: “как посмотреть cookie chrome”.



AdminPanel

Попробуй зайти в админку и получи флаг!
<http://188.226.160.200:801/>

На сайте по ссылке есть единственная картинка-ссылка, при переходе по которой мы попадаем на страничку с текстом “go away, stranger!”.

Опять просматриваем куки. Оказывается, страница устанавливает куки с ключом admin и значением false. Учимся подменять куки при запросах к сайтам. Это можно сделать из JavaScript-консоли разработчика в браузере одной командой:

```
document.cookie = 'admin=true'
```

Перезагружаем страничку — и флаг наш. Разумеется, куки запросов можно подменять многими другими способами, о которых в Интернете написано немало.

Украденный телефон

Еще в одной задаче дана фотография вместе с таким текстом:

“В Москве у меня украли телефон ☹. Хорошо, что все новые фотографии синхронизируются с моим dropbox. Эта фотография сделана совсем недавно — помоги мне узнать, в каком городе сейчас вор?”

На фотографии русско-английский разговорник. Чем это может помочь?

Оказывается, что разговорник тут ни при чем. Для решения задачи нужно достать из фотографии так называемую “EXIF-информацию”, а вернее, геоданные из нее. Фотокамеры записывают в геоданные внутри файла с картинкой координаты места, в котором был сделан снимок. Посмотреть EXIF можно стандартными средствами Windows (выбираем в Проводнике в контекстном меню пункт Свойства файла). Потом можно вбить полученные

данные широты и долготы в Гугл-карты и попасть в загаданный город.

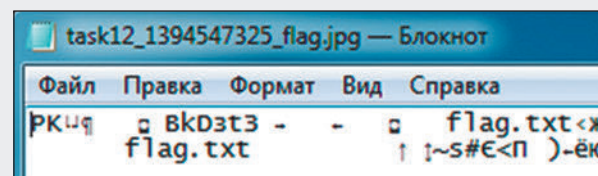
Очень удобен для решения этой задачи онлайн-сервис <http://regex.info/exif.cgi>, он позволяет загрузить картинку и сразу показывает нужный участок географической карты.

Что не так с моей картинкой?

Дана ссылка на картинку, но при переходе по ссылке картинка не открывается, а браузер пишет об ошибке.

Попробуем открыть картинку нестандартными средствами. Лучше всего подходят для исследования внутренней структуры файлов специальные hex-редакторы, например, 010 Editor (<http://www.sweetscape.com/010editor/>). Такие редакторы позволяют анализировать каждый бит содержимого файла.

Для исследования этой задачи подойдет даже обычный Блокнот. Скачиваем файл правой кнопкой мыши (Сохранить объект как) и открываем в Блокноте.



Видим начальные символы PK и строку flag.txt.

По запросу “pk file” Гугл подсказывает, что PK — это сигнатура zip-архивов. Первые два байта в каждом zip-архиве — это символы “PK”. Похоже, нам дали архив под видом картинки: случай, когда “обложка” не соответствует содержимому.

Меняем расширение файла на zip и стандартно открываем. В архиве обнаруживаем файл *flag.txt*. Если у вас Windows и в Свойствах папки стоит галочка “Скрывать расширение у зарегистрированных типов файлов”, то сначала вам нужно будет снять эту галочку. Только после этого вы сможете изменить расширение имени файла из Проводника.

Быстрее!

Не все задачи решаются просто.

Например,

Anonymous

Дается изображение — файл JPEG.

Замечаем, что изображение подозрительно долго грузится. Действительно, файл занимает на диске 7 мегабайт, хотя обычно картинки такого же размера по ширине и высоте занимают сотню килобайт.

Смотрим в hex-редакторе содержимое файла. В конце файла видим что-то странное и непохожее на “хаотичное” содержание jpg:

```
/9j/4AAQSkZJRgABAQEAYABgAAD/  
4RvvRXhpZgAASUkqAAgAAAAMAA4BAg...
```

Это Base64? Тогда раскодируем его и посмотрим на результат.

В начале раскодированных данных видим “JFIF” — сигнатуру формата JPEG. Стало быть, в одной картинке была сокрыта другая.

Формат JPEG — вообще идеальное место для хранения тайной информации от неискушенных пользователей (искушенные сразу откроют картинку в hex-редакторе, и секрет раскроется). Дело в том, что в файле этого формата есть маркер конца изображения, и просмотрщики обычно игнорируют все, что следует после него. Если мы хотим послать скрытые данные внутри картинку, то просто приписываем в конец файла любые символы.

Вернемся к заданию. На второй картинке флага нет. Зато в hex-редакторе снова видим Base64 в конце файла. Снова выцепляем и декодируем, и снова картинка, и снова Base64 в конце файла. 7 мегабайт оказываются спрятанными друг в друга семью картинками. Матрешка! На самой внутренней картинке — флаг.

Для автоматизации процесса можно написать скрипт, хотя и без него задача решается.

Вот shell-скрипт:

```
for i in `seq -w 1 6` ; do
  strings -n 1000 file$i.bin |
  base64 -D > file$((i + 1)).bin
done
```

Кстати, ответ на задачу в начале статьи: AAF.

Сообщение набрано русскими и латинскими буквами. Это можно увидеть по небольшой разнице в изображении символов, а на мысль о необычном наборе наводит слово “МОЕ”, набранное прописными и с нулем вместо “О”. Для того чтобы ее смогли решить и читатели печатной версии, мы набрали текст разными шрифтами. От этого задача стала проще, но неаккуратнее. Обычно подобные задачи оформляют так, что на первый взгляд разница незаметна.

МОЕ условие: Ежей 9, а ушей?.. **ОТВЕТ:**

Если оставить только латинские буквы и цифры, получится строка MOEYeE9yOTBE, и если ее декодировать как Base64, получится 3A2xOt90D, то есть $3A2_{16} \oplus 90D_{16}$, а это AAF_{16} .

Call me

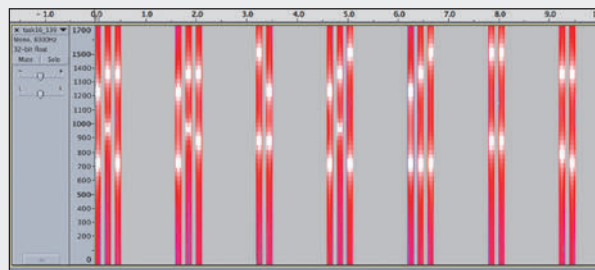
Нам поступил странный звонок. Можешь разобраться, в чем дело?

К заданию приложен wav-файл.

На записи слышен характерный звук набираемых на телефоне клавиш. По запросу “телефон тональный режим код” легко узнать, что при кодировании в тональном режиме используется Dual-Tone Multi-Frequency: система, при которой каждый сигнал передается как сумма двух частот из фиксированного набора:

1	2	3	A	697 Гц
4	5	6	B	770 Гц
7	8	9	C	852 Гц
*	0	#	D	941 Гц
1209 Гц	1336 Гц	1477 Гц	1633 Гц	

Откроем наш аудиофайл в Audacity и выберем для дорожки режим показа Spectrogram:



По оси ординат — частота в герцах. Одна вертикальная полоска на спектрограмме — одно нажатие клавиши. Каждая вертикальная полоска имеет два засвета: нижний задает строку в кодовой таблице, верхний — столбец. Считав коды по засветам, получаем

102-108-97-103-123-99-42-108-108-95-109-51-95-100-107-100-51-125

Переводим эти номера в символы по таблице ASCII и получаем флаг.

Совсем быстро!

Решение некоторых задач требует серьезной подготовки от участников.

Was ist das?

Дана фотография мельницы с огромной надписью “Was ist das?”.

Опыт подсказывает: мельница тут ни при чем. А в правом нижнем углу видна подсказка.



Перед нами изображение формата bmp без сжатия. Красный, зеленый, синий в подсказке — информация спрятана в битах.

Обычно для сокрытия информации в изображении используют младшие биты цветовых каналов. Изменение значения цветового канала на единичку незначительно, поскольку неразлично для глаза. В этой задаче информация скрывается в разных битах, поэтому видны искажения. Если приглядеться, в самой верхней строке видны “артефакты” — цветные точки, выбивающиеся из фона картинки.

Извлечь флаг технически сложно. Тут уже без серьезных навыков программирования не обойтись. Из

красного канала нужно взять младший бит, из зеленого — второй и так собрать текст флага. Решающему эту задачу придется найти какую-нибудь библиотеку, которая достает данные из картинок, и написать с ее помощью программу, собирающую решение задачи.

Блокнотик

Самый надежный шифр — это одноразовый блокнот, но мне лень придумывать много...

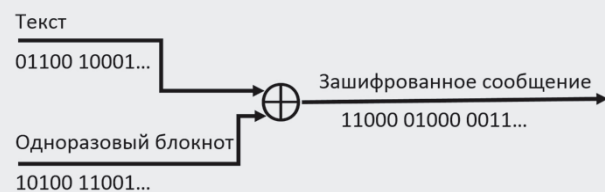
```
2426d6c3ef1a29652be80311a82c031d3ba564992
d2fbf1d3bb3bee6cb523187e64ecb1af636b0a492571
de1ac693ca10483736ee37912ccf544233c5507f14a14
a8da2877a2b0d16a8cb90ce91bc0192fe733b4b254e8
34b943bf41278cf922314c9f8433
```

```
242193b0cf1d236520ff145ce339140b69ec65dc2c
2fbe0f77abf0f08e502885e759db1af62bf1bc834154f0
e97b3ae445897d71f965128df456232e5210f15615be
8e3460b8e5cc65d8a35eff0d94153aec22b0b54aba20
b90ba24d2796f4226559879571
```

... (в задании приводится еще пять строк такого же вида).

Одноразовый блокнот — это шифр, при котором текст зашифровывается и расшифровывается одним и тем же ключом. Исходный текст для шифрования представляется как набор битов. Затем берется ключ — двоичная строка длиной с текст — и побитово ксорится (операция “исключающее или”) с текстом:

$$\text{cipher} = \text{text} \oplus \text{key}$$



Расшифровка проста: ксорим зашифрованный текст с ключом. Поскольку для каждого бита выполняется $\text{key} \oplus \text{key} = 0$ (это свойство операции “исключающее или”), то $\text{cipher} \oplus \text{key} = \text{text} \oplus \text{key} \oplus \text{key} = \text{text}$.

Одноразовый блокнот — идеальный шифр, т.е. его в принципе нельзя взломать. Если, конечно, использовать ключ только один раз в жизни. В нашем же случае на одном ключе зашифровано семь разных текстов. Поэтому можно применить частотный анализ и постепенно расшифровать все сообщения.

Естественно, и при решении этой задачи без программирования не обойтись.

Полностью решение этой задачи можно почитать на странице <http://lights-out-ctf.ghost.io/ufo-ctf-school-2014-quals-bloknodik/>.

Выход в третье измерение

В этом году мы в Москве проводили очный CTF для школьников, который организовала известная уральская CTF-команда Хакердом. Задачи были сложнее, чем “уфошные”, соревнование длилось весь день (восемь часов с перерывом на обед). Мы в Москве решили для желающих устроить допол-

нительные конкурсы на задачах с материальными объектами, чтобы уставшие участники могли хотя бы на время вырваться в реальный мир из двумерности монитора.

Наибольшим успехом, пожалуй, пользовались задачи на станции RealAdmin Corner (админский уголок).

Например, таск **How Long**.

Участникам давался большой моток витой пары и десятисантиметровая линейка. Требовалось измерить длину мотка.

Тем, кто начинал с энтузиазмом наматывать витую пару на локоть, мы замечали, что погрешность измерений должна составить не более двух сантиметров.

Решение: Достаточно было заметить на проводе заводские надписи-маркеры с отсчетами длины: “0002FT”, “0004FT”, “0006FT” и т.д. (это отсчеты в футах). Затем нужно было узнать длину фута, а также померить расстояния от концов провода до ближайших к ним маркеров.

Или таск посложнее:

Your IP

Дается ноутбук и роутер. На роутере отключен WiFi, но есть витая пара, концы и обжиматель сетевого кабеля.

Решение: правильно обжимаем провод с двух сторон, коннектимся к роутеру по полученному кабелю и получаем IP по DHCP.

Естественно, как и при решении задач основного соревнования, участникам разрешалось пользоваться интернет-поисковиками.

Продолжаем движение!

Конечно, CTF-соревнование — лишь “затравка”, после которой школьникам следует более глубоко изучить различные прикладные разделы информатики. Но интерес — это очень важный стимул к обучению.

Мы уверены, что даже после простого чтения нашей статьи уровень компьютерной грамотности каждого читателя в чем-то вырос. А уж тем более он сильно вырастает у школьников, которые не читают решения, а пытаются сами “штурмовать” задачи и получают новые знания в процессе. Кстати, от решения задач часто невозможно оторваться, а задачи достаточно сложны, поэтому взрослые тасковые CTF-соревнования обычно проводят длиной в 48 часов.

CTF — это игра и это начало серьезного образования. Далеко не все участники станут специалистами в области компьютерной безопасности, но вкус к более глубокому изучению компьютерных технологий появится у многих.

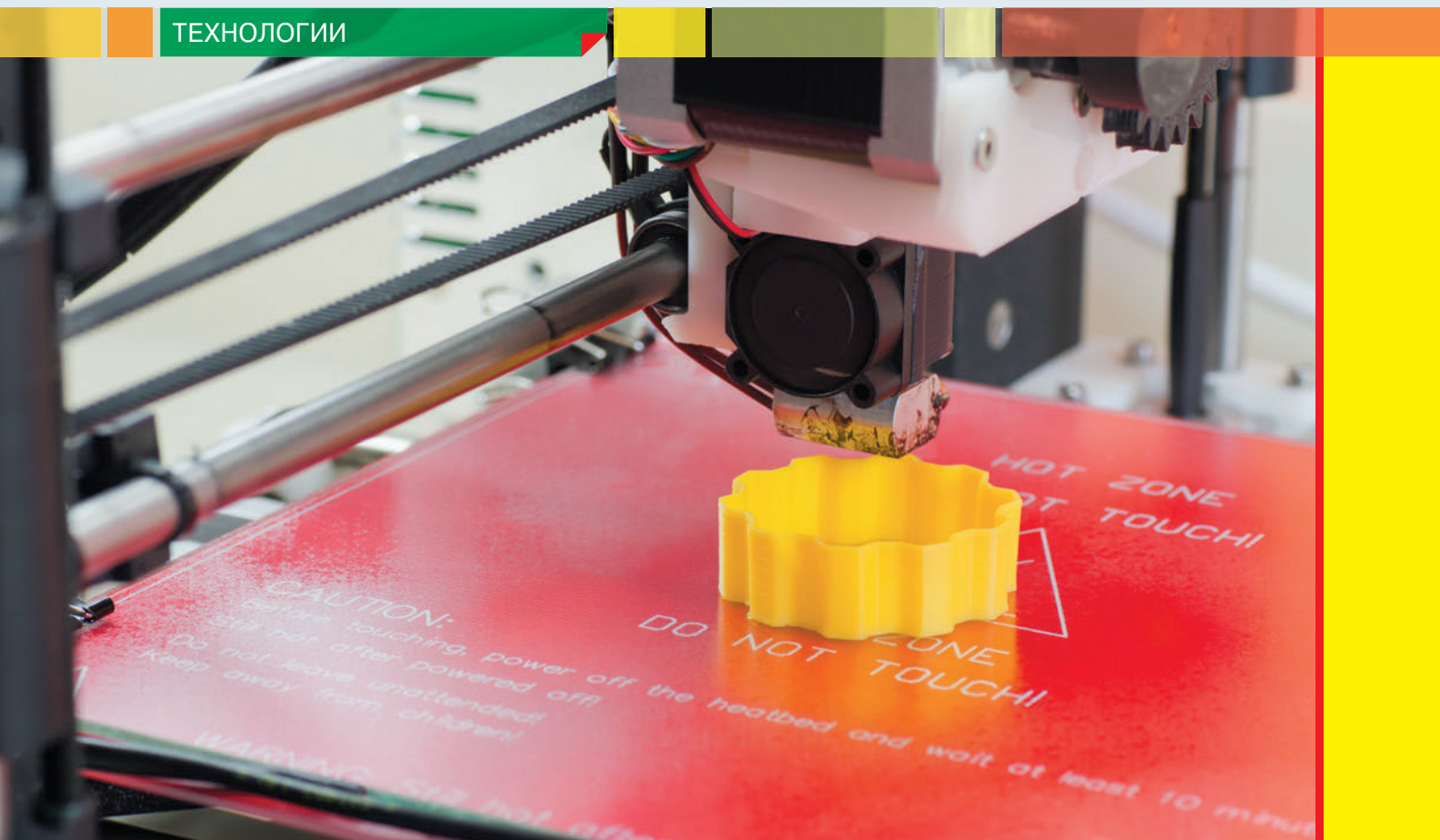
Вместо заключения еще одна задача!

Бывают и задачи-шутки. Попробуйте решить такую:

Ремингтонист

Оо. Что это? “trdcv ytfcvbh iuhbnmk 0okm”

Подсказка. Компьютер сильно помогает решить эту задачу. Особенно выключенный.



Что такое 3D-принтер и как он работает

Д.Ю. Усенков,
Институт
информатизации
образования
Российской академии
образования,
Москва

► Что такое 3D-печать: очередная модная игрушка для богатей — или очередной виток технической эволюции, способный изменить мир? Что 3D-принтеры могут дать обычному пользователю? Как производится 3D-печать и чем она отличается, например, от работы обычных станков с числовым программным управлением? Обо всем этом рассказывается в данной статье.

“Отсечь все лишнее” — или..?

Французский скульптор Огюст Роден когда-то охарактеризовал свое искусство так: “Нужно взять глыбу мрамора и отсечь от нее все лишнее”. Большинство современных производ-

ственных технологий используют ту же идею — берется заготовка и при помощи различных режущих инструментов от нее “отсекается все лишнее”: снимается часть материала для получения требуемой формы изделия. Разве что литье и штамповка работают иначе — исходный материал просто принимает нужную форму и весь, целиком (или почти целиком) идет в дело.

О том, что такое литье, знают, наверное, почти все: кто-то — в связи со своими профессиональными обязанностями, большинство — по книгам, различным публикациям в прессе и телевизионным передачам на производственные темы, а многие еще в детстве плавил олово и свинец и отливали из них своих собственных солдатиков. Если не вдаваться в технологические подробности, то для получения изделий путем литья достаточно в заранее подготовленную форму с полостью, повторяющей контуры будущего изделия, залить расплавленный металл, впрыснуть расплавленную пластмассу либо залить некую жидкую смесь компонентов, ко-

торая затем затвердевает. Материал, находясь в жидком состоянии, заполняет собой полость внутри формы (вытесняя воздух через предусмотренные в форме отверстия), а затем затвердевает и сохраняет полученные очертания, когда изделие будет вынута из формы. В реальности, конечно, все гораздо сложнее: нужно обеспечить заполнение материалом даже самых “укромных” уголков внутри формы и равномерность застывания, а полученную отливку позже придется дообрабатывать (фрезеровать, шлифовать), чтобы удалить различные неровности, “литники” и т.д. Но главная сложность — в том, что литье позволяет получать изделия далеко не любых очертаний, в этой технологии имеется целый ряд ограничений. Например, полости внутри отливаемой детали получить очень сложно, чаще их вырезают уже в готовой отливке сверлением, точением или фрезерованием либо собирают изделие из нескольких частей. Поэтому литье — это технология заводская, а не “домашняя”. А литье из пластмассы — дело еще более хлопотное. Пластмассу (обычно в виде гранул) нужно расплавить, а потом впрыскивать в форму под давлением, для чего в домашних условиях оборудование взять просто негде.

Положение немного улучшилось, когда появились различные “смеси для лепки”, отвердевающие при нагреве или кипячении (та самая знаменитая “Пластика”), но по-настоящему возможность работать с разогретой пластмассой дали “клеевые пистолеты” (рис. 1). “Клеевой пистолет”, несмотря на его название, работает не с обычным клеем (в баночке или тюбике), а именно с пластмассой. “Термоклей” представляет собой легкоплавкий пластик в виде стержня, который вставляется в корпус пистолета. Нажатие на курок продвигает стержень внутрь, нагревательный элемент внутри пистолета расплавляет пластик, и он выдавливается через сопло наружу. После этого главное — быстро нанести его в точке склеивания, сжать склеиваемые части изделия и дождаться, когда пластик остынет и снова затвердеет, соединив эти части.



Рис. 1. Клеевой пистолет

Трудно сказать, была ли изобретена 3D-печать в процессе пользования таким “клеевым пистолетом”, — но эти две технологии очень похожи. Представьте себе: вы поместили на подложку каплю “термокля”, после ее затвердевания добавили

на нее еще одну каплю и т.д. В итоге будет постепенно “расти” своеобразная пластиковая башенка. А это уже не что иное, как трехмерное изделие! Собственно, подобный процесс представляет собой основу чуть ли не всех технологий 3D-печати: требуемое трехмерное изделие создается буквально “по каплям” из исходного материала.

Краткие основы 3D-печати

Типичный 3D-принтер чем-то напоминает *плоттер* (графопостроитель) — устройство для компьютерного рисования различных чертежей. Если обратиться к помощи Интернета, то можно найти многочисленные фотографии плоттеров *рулонного типа*: в них лист бумаги (или другого материала) прокручивается вперед-назад внутри удлиненного корпуса устройства, а расположенное под кожухом перо движется влево-вправо. Перо может быть приподнято или опущено на бумагу; в последнем случае при движении пера и/или листа на бумаге вычерчивается линия. В зависимости от того, насколько компьютер (согласно заложенной в него программе) смещает перо, а насколько прокручивает бумажный лист, можно вычерчивать прямые линии с различным наклоном, окружности, овалы, различные кривые линии и т.д.

Впрочем, нас сейчас больше заинтересует другая конструкция плоттера — *планшетная* (рис. 2).

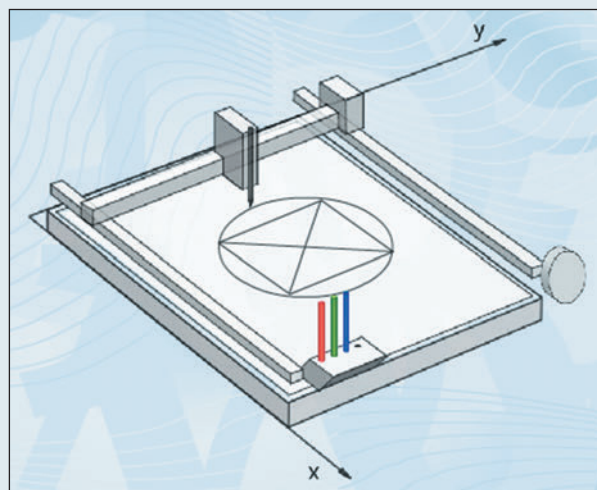


Рис. 2. Планшетный плоттер

Такой плоттер представляет собой плоскую поверхность, на которой неподвижно закрепляется лист бумаги. Над ним по направляющим перемещается каретка — будем считать, что ее движения соответствуют движениям по оси X. По этой каретке как по направляющей перемещается пишущий узел, в котором закреплено специальное перо, фломастер и т.д. Направление движений пишущего узла по каретке перпендикулярно направлению движений самой каретки и соответствует оси Y получаемой координатной плоскости. В результате, перемещая и каретку, и пишущий узел, компьютер

может заставлять перо вычерчивать на листе бумаги любые линии, любые фигуры и даже заштриховывать их многократными проходами пера по поверхности листа внутри контура фигуры с небольшим шагом. А если перо приподнято, то можно перемещать его без следа на бумаге, чтобы, например, перейти в другую точку для начала рисования новой линии.

Типичный 3D-принтер работает, по сути, аналогично. Но если в плоттере пишущий узел перемещается только в плоскости, по двум координатам X и Y , то в 3D-принтере добавляется еще и третья координата Z . А вместо пера пишущий узел 3D-принтера выпускает из себя жидкий или вязкий (например, расплавленный) исходный материал, “закрашивая”, “зарисовывая” им заданный участок. При этом печать производится *послойно*: пишущий узел 3D-принтера вначале движется в одной плоскости и “закрашивает” материалом первый слой будущего объекта. Когда этот слой целиком “зарисован”, пишущий узел 3D-принтера поднимается на один шаг выше по координате Z (либо, наоборот, платформа с заготовкой опускается на один шаг вниз, удаляясь от пишущего узла) и, как только предыдущий слой затвердеет, начинает “закрашивать” следующий слой.

Вот так, слой за слоем, постепенно “выращивается” и все изделие. И чем выше точность (разрешающая способность) 3D-принтера — особенно по координате Z , — тем менее заметны на полученном изделии характерные полосы, зубчатые контуры (рис. 3).



Рис. 3. На макрофотографии распечатанной на 3D-принтере фигурки лягушки хорошо видна полосчатая структура поверхности, полученная за счет послойного нанесения материала (фото автора статьи)

В идеале же должна получаться полностью готовая вещь, которая уже не нуждается ни в чистовой обработке, ни даже в сборке из деталей: например, шарнир или зубчатую передачу можно напечатать прямо в готовом (собранном) виде. Для этого используется еще одно исходное вещество — “поддерживающий материал”, например, водорастворимый. Им “запечатываются” промежутки между деталями, а после завершения печати это вещество вымывается из изделия и освобождает его детали.

Большинство 3D-принтеров, предназначенных для домашнего использования, печатают пластмассой. Исходный материал в этом случае подается в принтер в виде пластиковой нити, а пишущий узел представляет собой *экструдер* — нагреватель с соплом, через которое выдавливается расплавленная пластмасса или “поддерживающий материал”. Экструдер перемещается в плоскости вправо-влево и/или вперед-назад, “замазывая” расплавленной пластмассой нужные участки, потом приподнимается на один шаг выше и по уже застывшему предыдущему слою “замазывает” новые участки — те, которые должны быть в сечении изделия этой плоскостью (рис. 4).

Таким образом, основой для печати изделия на 3D-принтере является *последовательность плоских сечений* созданной на компьютере *трехмерной модели* (для создания таких моделей используются специальные программы, аналогичные всем известной САПР-системе AutoCAD). Очевидно, что такой способ изготовления позволяет получить

практически любую форму изделия, в том числе почти с любыми внутренними полостями, — возможность, недостижимая почти ни в какой другой технологии производства. А можно и создавать изделия с “сетчатой” внутренней структурой вместо сплошного заполнения материалом, обеспечивая при сохранении необходимой прочности меньший вес и меньший расход материала (рис. 5).

В результате комплект из обычного домашнего компьютера (оснащенного соответствующей программой 3D-моделирования) и 3D-принтера превращается в мини-завод, способный буквально в считанные часы развернуть малотиражный (от одного экземпляра) выпуск любых требуемых изделий — от бытовых (различных

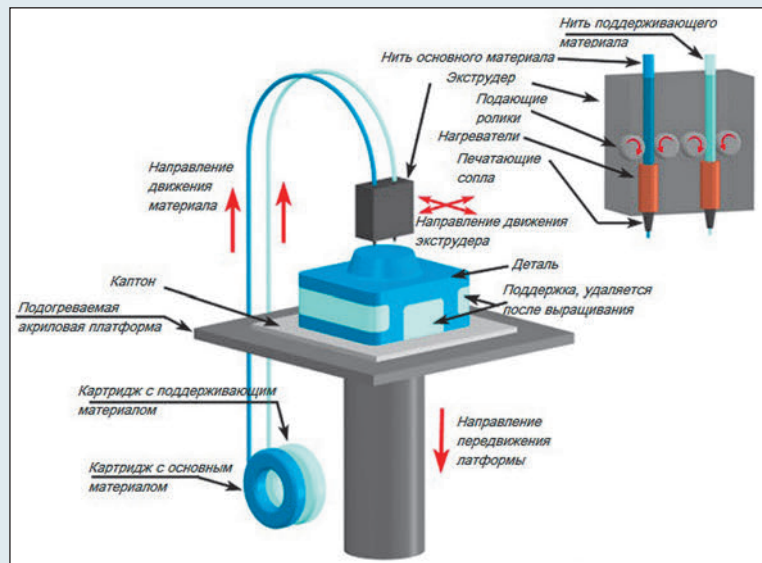


Рис. 4. Принцип работы 3D-принтера (рис. с сайта <http://stxlab.ru/print-3d.html>)



Рис. 5. На этом образце не до конца напечатанной статуэтки на получившемся срезе хорошо видна внутренняя структура изделия с заполнением опорными перегородками для уменьшения расхода материала (фото автора статьи)

вазочек, подставок, чехлов для смартфонов и пр.) до пластиковых запасных частей к любому оборудованию. Причем 3D-чертеж будущего изделия может быть как получен в готовом виде (например, скачан с сайта фирмы-изготовителя ремонтируемого оборудования), так и создан самим владельцем принтера, так что здесь открываются поистине широкие новые горизонты и для творчества, и для индивидуального предпринимательства. Уже сейчас в Интернете открыты специальные сайты для энтузиастов 3D-печати (пока в основном зарубежные), подобные социальным сетям и позволяющие обмениваться не только мнениями и опытом работы с 3D-принтерами, но и исходными файлами своих изделий и даже выставлять их на продажу. А если учесть, что уже существуют и 3D-сканеры — устройства, позволяющие оцифровать почти любой реальный объект и создать в компьютере его трехмерную точную копию, то появляются и еще более интересные возможности: например, можно оцифровать любого человека (ему только потребуется немного постоять неподвижно, пока идет сканирование), получить на компьютере его трехмерный портрет (бюст или в полный рост), а затем после некоторой дообработки распечатать на 3D-принтере статуэтку желаемого размера — точный объемный портрет. Подобные услуги уже оказывает целый ряд различных фирм не только за рубежом, но и у нас в России.

Технологии 3D-печати

Вообще в настоящее время существует несколько технологий 3D-печати, позволяющих работать с различными материалами. При этом интересно, что 3D-принтеры можно классифицировать ана-

логично обычным принтерам. Так, можно условно разделить все многообразие 3D-принтеров на “лазерные” и “струйные”, а также в перспективе (о чем будет сказано позже) — и “матричные”. Однако хронологически эти разновидности 3D-принтеров появлялись в противоположном порядке по сравнению с обычными принтерами: самыми первыми были созданы лазерные 3D-принтеры, затем струйные, а “матричные” пока остаются в стадии теоретической разработки.

Лазерные 3D-принтеры, как это и следует из их названия, основаны на применении лазерного луча в качестве основного “рабочего инструмента”. Так же, как в обычном лазерном принтере луч лазера “зарисовывает” требуемые контуры на светочувствительном барабане и тем самым формирует требуемое (проявляемое при помощи тонера) изображение, в лазерном 3D-принтере луч лазера “зарисовывает” каждый очередной слой выращиваемого трехмерного изделия.

При этом среди лазерных 3D-принтеров можно выделить две основные разновидности.

В системах трехмерной печати на основе **стереолитографии** (рис. 6) используется **фотополимер** — жидкая смесь веществ, затвердевающих под воздействием лазерного излучения (либо ультрафиолета). Рабочая камера 3D-принтера представляет собой емкость (ванну), заполненную жидким фотополимером, дно которой постепенно, по шагам опускается вниз. Лазерный луч сначала “зари-

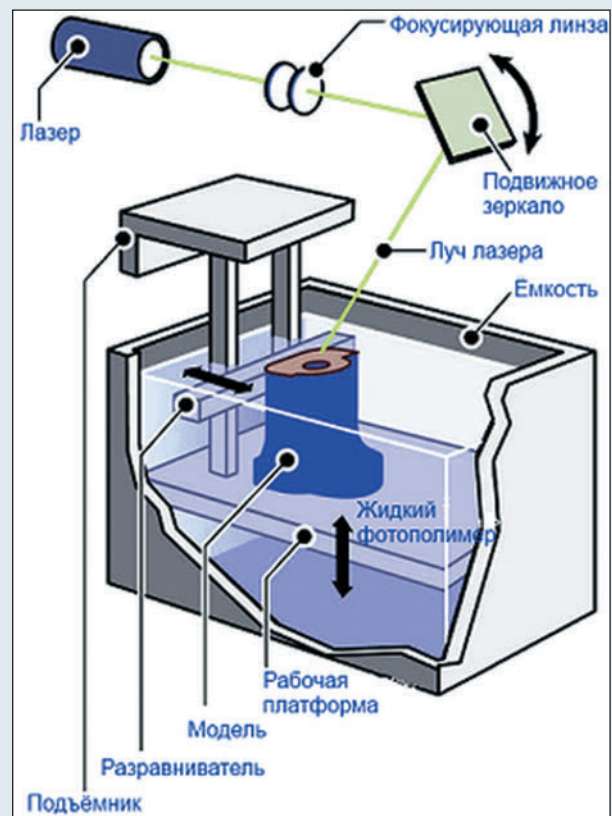


Рис. 6. Лазерный стереолитографический 3D-принтер (рис. с сайта <http://3dsystemtrade.com/index.php/2013-07-17-08-49-53/printsipy-raboty-3d-printerov>)

совывает” контур и заполнение сечения изделия в самой нижней его плоскости и заставляет фотополимер затвердевать в этих участках. Затем дно ванны смещается на один шаг вниз и уже сформированный слой становится “рабочей площадкой” для “зарисовывания” последующего слоя. И так до тех пор, пока не будет “выращено” все требуемое изделие.

Подача исходного вещества при этом происходит самотеком (хотя в некоторых моделях фотополимер подается в зону облучения при помощи специальных сопел и тут же “засвечивается” лазером), а луч лазера может перемещаться как по двум осям, так и только по одной оси, тогда как движения по другой оси отрабатываются за счет перемещения всей рабочей камеры или только ее дна — рабочей платформы (как на рис. 6).

Обычно в качестве фотополимера используется полупрозрачный материал, после затвердевания достаточно хрупкий и не очень влагостойкий, поэтому применение стереолитографической 3D-печати пока довольно ограничено (в основном в исследовательских целях).

Порошковые лазерные 3D-принтеры работают практически так же, но вместо жидкого фотополимера в них используется порошок — пластиковый, керамический или даже металлический. При “зарисовывании” лазерным лучом очередного слоя “выращиваемого” объемного изделия частицы порошка спекаются друг с другом и с предыдущим слоем в более или менее монолитный твердый массив. А после создания очередного слоя на него засыпается новый слой порошка, который будет использован при получении следующего слоя (рис. 7).

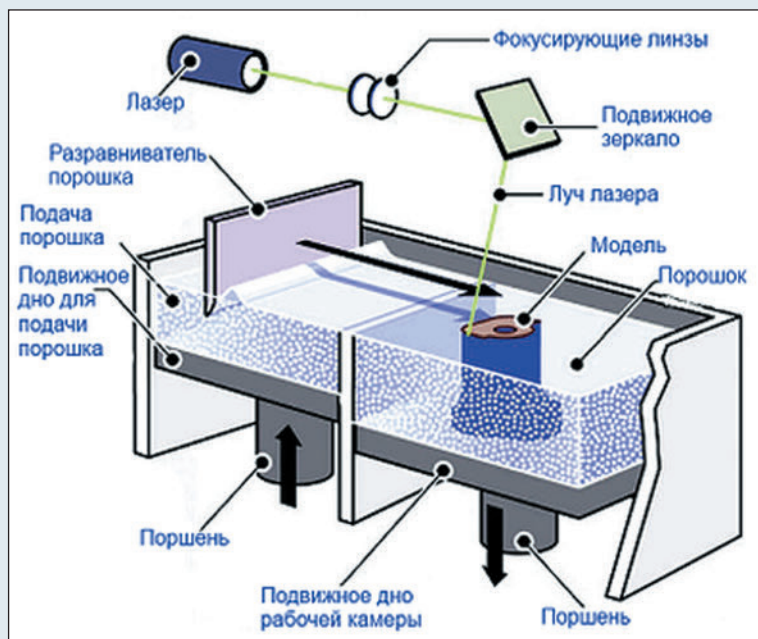


Рис. 7. Лазерный порошковый 3D-принтер (рис. с сайта <http://3dsystemtrade.com/index.php/2013-07-17-08-49-53/printsipy-roboty-3d-printerov>)

Порошковые 3D-принтеры интересны тем, что позволяют получать изделия с достаточно высокой прочностью, которые пригодны не только в качестве бытовых “безделушек”, мелких запчастей или в проектировании для получения “прикидочных” 3D-образцов будущих изделий. Однако мощность лазерного луча, требуемая для спекания порошка, необходима гораздо более высокая, чем в стереолитографии.

Кстати, именно на порошковой технологии работал довольно интересный прототип 3D-принтера для печати из обычного песка. Изготовил такой принтер в качестве своего дипломного проекта немецкий студент Маркус Кайзер в 2011 году. Для спекания песка вместо лазера здесь используются солнечные лучи, сконцентрированные при помощи большой линзы Френеля, а для работы координатной системы принтера и компьютера, управляющего процессом, использовались солнечные батареи.

Существует и еще одна разновидность лазерных 3D-принтеров, работающая за счет нарезки и ламинирования листового материала. В этой конструкции исходным материалом является бумажная лента, которая сматывается с рулона и проходит над рабочей камерой. Сначала рабочая площадка поднята и бумажная лента лежит на ней. Лазерный луч вырезает из бумаги требуемые контуры плоских сечений, а все остальные участки разрезает на мелкие квадратики (которые затем удаляются). Полученный слой ламинируется нагретым валиком, а затем рабочая площадка опускается на шаг, равный толщине бумаги, и поверх уже созданного слоя ложится новая бумага. Далее процесс повторяется, при этом каждый очередной слой при ламинировании спекается с предыдущим, а изделие

получается слоистым, похожим на деревянное, но недостаточно влагостойким. В настоящее время, похоже, существует только одна модель подобных принтеров — Mcor IRIS.

Струйные 3D-принтеры, как и их обычные струйные “собратья”, работают за счет “закрашивания” каждого слоя струей или каплями исходного материала — жидкого (либо расплавленного) и быстро застывающего.

Чаще всего в качестве исходного материала для струйных 3D-принтеров используется пластиковая нить. Как работают такие принтеры, мы уже разбирали выше (см. рис. 4). При этом уже существуют модели подобных принтеров, способные печатать цветные изделия (тогда в качестве исходного материала используются разноцветные пластиковые нити). Обычно при этом используется пластик ABS, пластик PLA (биоразлагаемый, который используют при изготовлении одноразовой по-

суды) либо растворимый пластик PVA (который обычно используется в качестве “поддерживающего материала”). Используются также акрил, хотя он имеет более высокую температуру плавления, нейлон, капрон, полиэтилен и другие пластмассы.

Впрочем, в качестве исходного материала для струйного 3D-принтера может использоваться не только пластик, но и вообще все что угодно, если только оно может быть временно переведено в жидкую фазу с последующим застыванием. Так, уже существуют 3D-принтеры, в которых исходным веществом является разогретый до мягкости шоколад (рис. 8) либо расплавленный сахар (леденечная масса). Исходным материалом для струйного 3D-принтера может служить даже обычная вода, только, конечно, печать в этом случае должна производиться при низких температурах (не выше -20°C), а получаемые изделия должны эксплуатироваться при отрицательных температурах.

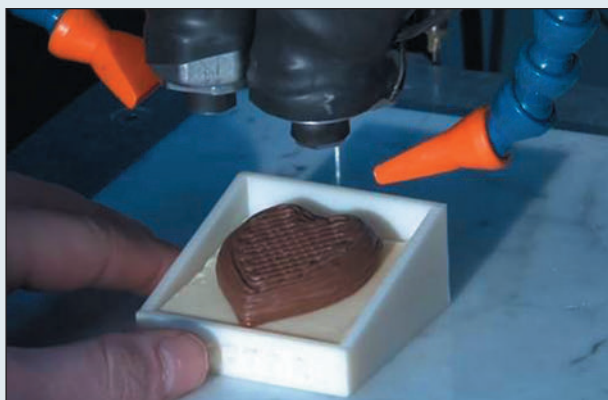


Рис. 8. Шоколадный 3D-принтер (фото с сайта <http://www.orgprint.com/ru/wiki/materialy-dlja-3d-pechati>)

Существует, однако, и еще одна разновидность 3D-принтеров, которую тоже можно отнести к струйным. Такие принтеры можно было бы называть “порошковыми”, но, поскольку в этом случае их легко спутать с лазерными порошковыми 3D-принтерами, мы будем называть их “адгезивными”, или “клеевыми”.

В этом случае в качестве исходных материалов используются сразу два вещества: порошок (как основа) и жидкий, быстро застывающий клей. Принцип работы такого принтера подобен сразу и порошковому лазерному, и только что рассмотренному с печатью пластиком. Для формирования очередного слоя после смещения рабочей площадки на один шаг вниз порошком засыпается вся площадь рабочей камеры, а затем пишущий узел, управляемый компьютером, поливает клеем нужные участки порошка, склеивая его частицы в монолит друг с другом и с предыдущим слоем. После застывания клея рабочая площадка смещается на шаг вниз, в освободившееся пространство подается новая порция порошка и т.д. А в конце работы остается только достать из окружающего неиспользованного

порошка готовое изделие (и по мере возможности высыпать лишний порошок из его внутренних полостей).

Возможен и аналогичный вариант, когда “клеевые” свойства уже заложены в используемый порошок, а вещество, подаваемое через сопло, лишь позволяет этим свойствам проявиться. Так, например, можно печатать бетоном или гипсом, подавая в порошкообразную основу обычную воду. При этом гипсовая 3D-печать открывает новые горизонты не только для скульпторов, но и для медиков. Например, выпускник отделения архитектуры и дизайна Веллингтонского университета Виктории Джейк Эвилл разработал “экзоскелетный гипс”, получивший название Cortex. Суть идеи в том, что после рентгеновского сканирования поломанной конечности и обработки на компьютере полученных данных о месте перелома и форме тела пациента в этом месте создается и распечатывается ажурный гипсовый каркас (рис. 9). Его можно легко надеть на руку или ногу, а затем снять, поскольку у него имеется откидная боковина с прочными креплениями. Конечно же такая гипсовая повязка гораздо удобнее, легче и эстетичнее, чем обычная гипсобинтовая “болванка”.



Рис. 9. Ажурный “экзоскелетный гипс”, напечатанный при помощи 3D-принтера (фото с сайта <http://habrahabr.ru/post/185562>)

Для струйной 3D-печати можно использовать и другие материалы. Например, кроме шоколада, можно использовать сыр, тесто, крем и т.д. Таким способом можно печатать интересные, красиво и аппетитно выглядящие блюда. Не так давно в новостях появилось сообщение, что таким способом из исходной белковой смеси удалось напечатать антрекот, который ни по вкусу, ни по структуре не удастся отличить от натурального (в том числе он имел такую же волокнистую структуру, как настоящее мясо). Конечно, пока речь не идет о повальной замене натуральных продуктов напечатанными (хотя бы потому, что напечатанные существенно дороже), но в будущем, скажем, при освоении других планет или дальнего космоса такие 3D-принтеры, печатающие пищу полученной из хлореллы белковой смесью с различными витаминными и вкусовыми добавками, могли бы оказаться очень полезными. Не зря агентство NASA включи-

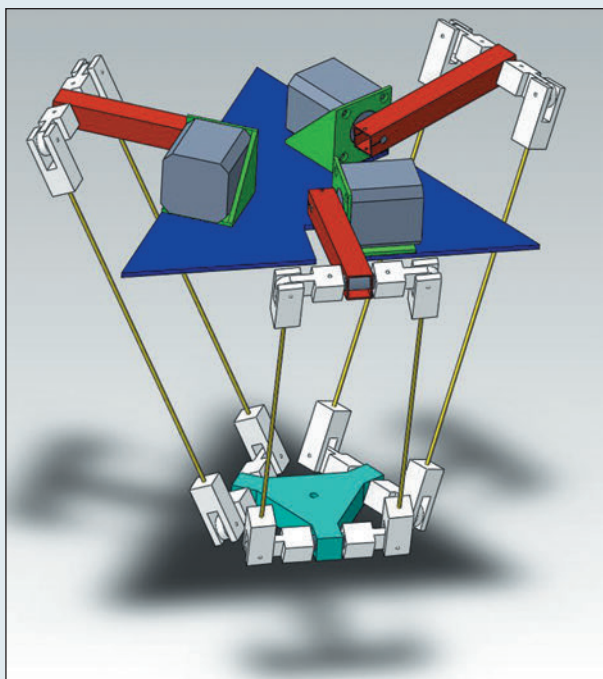


Рис. 10. Конструкция дельта-робота (рис. с сайта <http://www.marginallyclever.com/blog/2012/02/building-a-delta-robot-3d-model-version-2/>)

ло разработку такого пищевого 3D-принтера в план своих исследований и подключилось к финансированию этого проекта.

В настоящее время разрабатываются и вовсе удивительные технологии 3D-печати — живыми клетками (стволовыми или, например, печеночной ткани). Клеточная биомасса при этом выдавливается соплами на основу из стерильного биогеля, формируя контуры требуемого органа. Так что вполне возможно, что через десяток-другой лет донорство органов уйдет в прошлое: гораздо проще будет при необходимости взять клеточный материал у самого пациента, размножить клетки в пробирке и напечатать ему нужный орган заново (соответственно, не будет никаких проблем с отторжением).

Дельта-роботы

Кроме рассмотренных, возможна и еще одна конструкция 3D-принтеров, которая отличается от традиционной, хотя по типу печати обычно относится к струйным. Это 3D-принтеры на основе дельта-роботов.

Дельта-робот — это, по определению Википедии, “вид параллельного робота, состоящего из трех рычагов, прикрепленных посредством карданных шарниров к основанию; при этом использование параллелограммов в конструкции манипулятора позволяет сохранять пространственную ориентацию исполнительного устройства робота”. Такой робот был изобретен в начале 1980-х гг. Реймондом Клавелем (Швейцария).

Выглядит дельта-робот довольно оригинально — он похож на паука и немного на уэлсовский мар-

сианский тренажер. Силовой блок располагается высоко над рабочей площадкой, а вниз спускаются, сходясь к одной точке — к узлу манипулятора, три “руки” из сдвоенных тяг (рис. 10).

Из рисунка видно, что повороты и наклоны “вондил” (красного цвета), на которых закреплены параллелограммы тяг, приводят к тому, что рабочий узел робота может произвольно перемещаться в горизонтальной плоскости в любом направлении по горизонтали. И если теперь в качестве рабочего узла дельта-робота закрепить экструдер, то мы получим 3D-принтер, — остается лишь предусмотреть возможность пошагового подъема всей конструкции по вертикальной координате Z (или, наоборот, опускания рабочей площадки относительно неподвижной рабочей плоскости робота). Подобный 3D-принтер был показан, например, на прошедшей в Москве в феврале 2014 года выставке-конференции 3D Print Expo, посвященной технологиям трехмерного сканирования и печати (рис. 11).

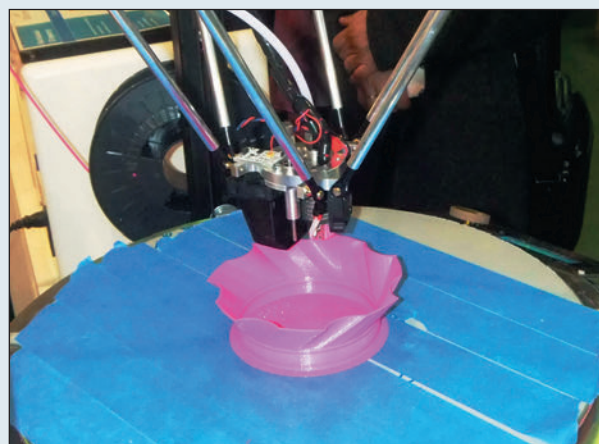
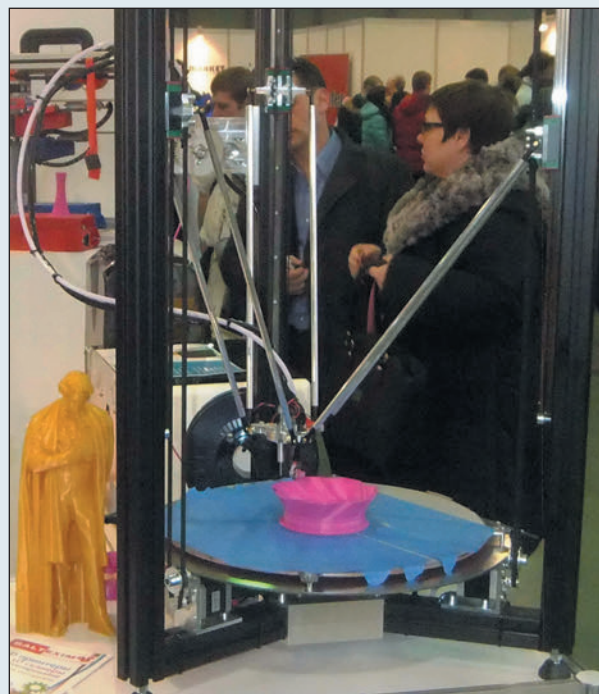


Рис. 11. Швейцарский 3D-принтер Delta Tower (фирма BALTEXIM) (фото автора статьи)

А что же матричные?

Итак, мы разобрались с двумя типами технологий 3D-печати — лазерными и струйными. А что же с упоминавшимися “матричными”?

Чтобы понять, как они могли бы работать, сначала познакомимся с еще одним интересным прибором — *туннельным микроскопом*.

Сканирующий туннельный (или, как его еще называют, зондовый) микроскоп интересен тем, что обеспечивает увеличения, сравнимые с достигаемыми при помощи электронных микроскопов. Но если электронный микроскоп — это крупногабаритное, сложное, дорогое и достаточно “капризное” устройство, то туннельный микроскоп имеет очень простую конструкцию и размеры если не карманные, то, во всяком случае, настольные (размером с крупный электрочайник).

Принцип работы туннельного микроскопа (рис. 12) еще проще, чем его конструкция.

Острая металлическая игла (зонд) подводится вплотную к поверхности исследуемого образца (на расстояние в несколько ангстрем). При подаче на иглу относительно образца небольшого напряжения между концом иглы и образцом начинает течь туннельный ток (за счет туннельного эффекта, в соответствии с законами квантовой механики), величина которого экспоненциально зависит от расстояния между концом иглы и образцом: так, при уменьшении зазора всего на 0,1 нм ток возрастает примерно в 10 раз. Игла в процессе работы движется вдоль поверхности образца (микросмещения управляются подачей электрического напряжения на электроды трубки из пьезокристалла), при этом блок управления за счет обратной связи поддерживает постоянную величину туннельного тока и одновременно фиксирует необходимые для этого изменения высоты подъема иглы. Получаемые данные обрабатываются и на компьютере по ним строится карта высот, от-

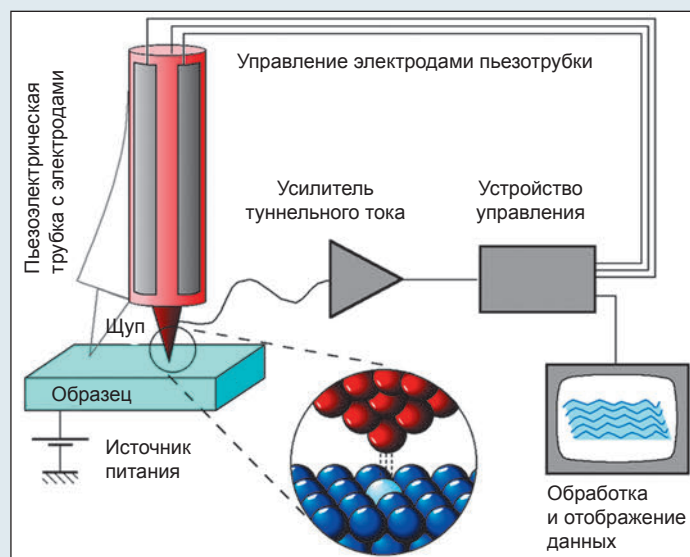


Рис. 12. Принцип работы туннельного микроскопа (рис. с сайта http://ru.wikipedia.org/wiki/Сканирующий_туннельный_микроскоп)

ражающая топографию исследуемой поверхности с точностью буквально до размера атома!

Изобретен сканирующий туннельный микроскоп в 1981 году Гердом Карлом Биннигом и Генрихом Рорером из лаборатории IBM в Цюрихе (хотя общие принципы, заложенные в конструкцию, были известны и ранее). За это изобретение они были удостоены Нобелевской премии по физике за 1986 год.

Но самое интересное, что туннельный микроскоп позволяет не только рассматривать отдельные атомы, но и перемещать их! Дело в том, что при превышении рабочего напряжения в процессе туннелирования отдельные атомы изучаемого вещества могут перескакивать на зондирующую иглу. На ней их можно перенести в другое место и “сбросить”, понизив напряжение. Именно так в 1990 году сотрудники фирмы IBM (Калифорния) под руководством Дональда Эйглера смогли выложить название своей фирмы атомами ксенона на никелевой пластине (рис. 13). При этом букву “I” составили всего из 9 атомов, а буквы “V” и “M” — из 13 атомов каждую.

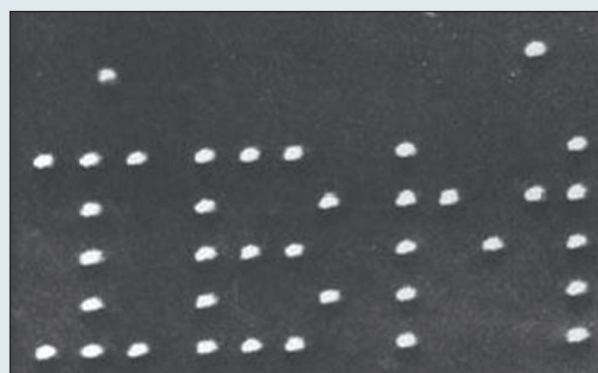


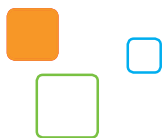
Рис. 13. Логотип фирмы IBM, выложенный из отдельных атомов при помощи туннельного микроскопа

Чем для нас интересен этот любопытный курьез? А тем, что применение таких (или подобных, типа атомно-силового микроскопа) устройств может позволить осуществлять 3D-печать путем сборки объектов из отдельных атомов! Именно такой способ печати мы и могли бы назвать “матричным” — по аналогии с получением обычных изображений на бумаге из отдельных точек в матричном принтере.

Используя такую технологию, возможно, удастся создать уникальные устройства — “синтезаторы”, которые из любой произвольной массы “образцов вещества” (например, из сортированных бытовых отходов) способны воссоздать по заданной программе практически любой предмет. Но это — в будущем. Пока же все надежды на то, что 3D-принтеры (хотя бы самые простые, струйные, печатающие пластиком) станут действительно доступными по цене для любого домашнего пользователя. Ведь только в этом случае можно будет говорить о том, что 3D-печать начинает менять наш мир...

Хорошие цены на отличное оборудование MimioClassroom

Комплекты для интерактивного класса



Интерактивная доска MimioBoard ME78

Размер доски 155x116 см (диагональ 78"). Управление стилусом, инфракрасно-ультразвуковая технология позиционирования. Износостойкая магнитная поверхность ceramicsteel™. Работа в проводном и беспроводном режимах. ПО для Windows, Mac OS и Linux.



Документ-камера MimioView

Одинаково хорошо работает с объемными предметами, документами и объектами микромира, позволяя вывести на большой экран все, что может понадобиться на уроке. Качественный объектив с оптическим 5,5-кратным, цифровым 16-кратным увеличением и автофокусировкой. Светодиодная подсветка. Разрешение UXGA (1600x1200). Адаптер для микроскопа в комплекте.

Цена комплекта

• 44 900 руб.

~~64 900 руб.~~

Mimio Classroom — весь спектр интерактивного оборудования для современной школы



MimioBoard

Стационарная интерактивная доска



MimioTeach

Интерактивная приставка к обычной классной доске



MimioVote

Создание и проведение тестов и контрольных



MimioView

Документ-камера с переходником для микроскопа



MimioProjector

Интерактивный проектор

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



ДИСТАНЦИОННЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

(с учетом требований ФГОС)

До 30 сентября производится прием заявок на 2014/15 учебный год

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



ШКОЛА ПРОГРАММИРОВАНИЯ

Рекурсия — эффектно, но не всегда эффективно

Д.М. Златопольский,
Москва

Схватив себя за эту косичку (своего парика. — Авт.), я изо всех сил дернул вверх и без большого труда вытащил из болота и себя, и своего коня... Если вы думаете, что это легко, попробуйте проделать это сами.

Барон Мюнхгаузен¹

В статье [1] рассказывалось о так называемых “процедурах” и “функциях” — фрагментах программы, в которых решается какая-то часть всей задачи. Эти фрагменты оформляются особым образом и снабжаются именем. По этому имени процедура или функция вызывается и выполняется как вспомогательная в основной части программы. Было показано, что использование процедур и функций сокращает размер программы, улучшает ее понятность, что помогает найти нужное место в программе, возможную ошибку и т.п.

Часто в программировании используют прием, который называется “рекурсия”², — когда некоторая процедура или функция вызывает (использует) саму себя как вспомогательную. Этот прием наглядно иллюстрировало шуточное стихотворение поэта Андрея Усачева о жучке в пиджачке (оно было опубликовано в декабрьском выпуске “В мир информатики” в 2013 году):

*Шел по улице жучок
в модном пиджачке.
На груди висел значок,
а на том значке
нарисован был жучок
тоже в пиджачке.
И на нем висел значок,
а на том значке
был еще один жучок...*

¹ Надеемся, вам известен этот литературный персонаж.

² От лат. *recursio* — возвращение.

Вспомните также: “У попа была собака...”.

Рассмотрим пример из программирования. Но сначала вспомним задачу, рассмотренную в [1]. Пусть в программе надо рассчитывать значения x :

$$x = \frac{2 + \sqrt{2}}{5 + \sqrt{5}} + \frac{5 + \sqrt{5}}{13 + \sqrt{13}} + \frac{11 + \sqrt{11}}{8 + \sqrt{8}}.$$

Видно, что в выражении имеются похожие фрагменты. Желательно для расчета дробей оформить функцию, которая вычисляла бы значения отдельных дробей вида $\frac{a + \sqrt{a}}{b + \sqrt{b}}$ при любых значениях a и b , а затем использовать ее для расчетов.

На школьном алгоритмическом языке соответствующая функция оформляется так:

```
алг вещ Дробь (арг цел a, b)
нач цел числитель, знаменатель
числитель := a + sqrt(a)
знаменатель := b + sqrt(b)
|Значение функции:
знач := числитель/знаменатель
кон
```

Как и стандартные функции, функция, созданная программистом, используется в правой части оператора присваивания. Для этого надо указать ее имя, а в скобках — значения фактических параметров:

```
x := Дробь(2, 5) + Дробь(5, 13) +
      + Дробь(13, 8)
```

Так вот, при оформлении собственной функции (или процедуры) в качестве вспомогательной может быть применена и сама эта функция (или процедура), т.е. можно использовать рекурсию. Для иллюстрации создадим функцию с рекурсией (такие функции называются *рекурсивными*) для расчета факториала натурального числа n (факториал числа n , обозначаемый $n!$, равен $1 \times 2 \times 3 \times \dots \times n$). Такую функцию можно создать, имея в виду, что $n! = (n - 1)! \times n$:

```
алг цел Факториал (арг цел n)
нач
знач := Факториал(n - 1) * n
|Рекурсивный вызов функции Факториал
кон
```

На самом деле эта функция оформлена не по правилам (и при выполнении программы появит-

ся сообщение об ошибке), но об этом чуть позже. Здесь же мы обсудим закономерный вопрос, который уже, наверное, появился у вас: а зачем нужно было “городить огород”, когда можно было создать такую, естественную и понятную, функцию:

```

алг цел Факториал (арг цел n)
нач цел произведение, i
    произведение := 1
нц для i от 2 до n
    произведение := произведение * i
кц
знач := произведение
кон

```

На этот ваш вопрос в данном случае трудно ответить убедительно. Но давайте рассмотрим еще одну задачу: “Разработать функцию для расчета k-го члена последовательности Фибоначчи (последовательность Фибоначчи образуют числа 1, 1, 2, 3, 5, 8, 13, ...)”.

Если вы хотите попробовать создать такую функцию самостоятельно (не используя рекурсию), то не смотрите на приведенный далее вариант:

```

алг цел Фиб (арг цел k)
нач
знач := Фиб(k - 2) + Фиб(k - 1)
кон

```

Быстро ли вы получили требуемую функцию? А теперь посмотрите, как просто и логично выглядит рекурсивный вариант функции³:

```

алг цел Фиб (арг цел k)
нач
знач := Фиб(k - 2) + Фиб(k - 1)
кон

```

Такое оформление полностью соответствует закону построения последовательности Фибоначчи — очередной элемент последовательности равен сумме двух предыдущих. При нем не требуется применять оператор цикла и думать над последовательностью расчета значений *пред* и *предпред*.

Если этот пример не убедил вас в преимуществе использования рекурсии (конечно, в определенных случаях), то рассмотрим следующую задачу: “Получить на экране изображение, как на рис. 1”.

³ Он также сознательно оформлен неправильно.

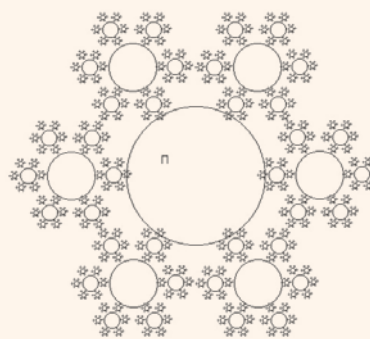


Рис. 1

Ну как, беретесь ли вы решить задачу без рекурсии? Если нет, то обсудим рекурсивный вариант.

Можно считать, что на рис. 1 изображена центральная планета П с шестью спутниками, у каждого из которых есть свои спутники, у тех — свои и т.д. Очевидно, что каждый спутник может рассматриваться как планета с соответствующими спутниками. Поэтому если составить процедуру, с помощью которой можно изобразить на экране некоторую окружность-планету с рядом окружностей-спутников, а для рисования спутников использовать эту же процедуру, но с другими параметрами — координатами, радиусами и т.п. (то есть применить рекурсию), то можно получить требуемое изображение. Соответствующая рекурсивная процедура имеет вид:

```

алг Планета (арг цел x, y, рад, Nсп,
вещ k_орб, k_спут)
|x, y - координаты центра планеты,
|рад - ее радиус
|Nсп - число спутников у каждой планеты
|k_спут - отношение радиуса спутника
|к радиусу “своей” планеты
|k_орб - то же для радиуса орбиты
|спутников
нач цел x1, y1, i, вещ угол, рад_орб
    поз(x, y) | Центр планеты
    окружность(рад) | Рисуем планету
    | Радиус орбиты
    | спутников
    рад_орб := рад * k_орб
    угол := 6.28/Nсп | Угол между спутниками
нц для i от 1 до Nсп | Для каждого спутника
    | Координаты центра
    x1 := x + рад_орб * cos(угол * i)
    | i-го спутника
    y1 := x + рад_орб * sin(угол * i)
    | Вызываем процедуру Планета
    | с новыми параметрами
    Планета(x1, y1, int(рад * k_спут),
    Nсп, k_орб, k_спут)
кц
кон

```

Однако и эта процедура пока оформлена неправильно. Теперь настало время сказать, почему. Вы, возможно, знаете, что при каждом вызове вспомогательной процедуры (функции) для нее

отводится место в оперативной памяти, которое “освобождается” после завершения ее (процедуры) работы. Но если во вспомогательной процедуре имеется рекурсия, то вызовы вспомогательных процедур будут продолжаться до тех пор, пока место в памяти не будет исчерпано (именно с этим связано сообщение об ошибке, о котором говорилось применительно к рекурсивной функции Факториал). Чтобы устранить этот недостаток, необходимо так оформлять процедуры (функции), чтобы рекурсивные вызовы осуществлялись по условию, которое когда-то станет ложным. В приведенной процедуре для этого можно использовать в качестве аргумента некоторую величину n , которая при каждом новом вызове будет уменьшаться на 1, а в тело процедуры надо включить условие, что ее команды должны выполняться только при $n > 0$:

```

алг Планета (арг цел  $x, y, \text{рад}, n, N_{\text{сп}},$ 
               вещ  $k_{\text{орб}}, k_{\text{спут}}$ )
нач
  если  $n > 0$ 
  то
    поз( $x, y$ )
    ...
    |Рекурсивный вызов процедуры Планета
    |с уменьшенным на 1 значением
    |параметра  $n$ 
    Планета( $x1, y1, \text{int}(\text{рад} * k_{\text{спут}}),$ 
             $n - 1, N_{\text{сп}}, k_{\text{орб}}, k_{\text{спут}}$ )
  все
кон

```

Приведем также “правильные” варианты функций Факториал и Фиб:

```

алг цел Факториал (арг цел  $n$ )
нач
  если  $n > 1$ 
  то
    |Рекурсивный вызов функции Факториал
    знач := Факториал( $n - 1$ ) *  $n$ 
  иначе
    знач := 1
  все
кон
алг цел Фиб (арг цел  $k$ )
нач
  если  $k > 2$ 
  то
    |Рекурсивный вызов функции Фиб
    знач := Фиб( $k - 2$ ) + Фиб( $k - 1$ )
  иначе
    знач := 1
  все
кон

```

Вернемся к задаче рисования изображения на рис. 1 и представим основную часть программы решения этой задачи:

```

алг Система_планет
нач цел  $n, N_{\text{сп}}, \text{рад}, x, y,$ 

```

```

вещ  $k_{\text{орб}}, k_{\text{спут}}$ 
 $n := 7; N_{\text{сп}} := 6$ 
 $x := \text{макс}X/2$  |Координаты центра планеты  $P$ 
 $y := \text{макс}Y/2$  |(в центре экрана)
 $\text{рад} := 120; k_{\text{орб}} := 1.6; k_{\text{спут}} := 0.5$ 
Планета( $x, y, \text{рад}, n, N_{\text{сп}}, k_{\text{орб}}, k_{\text{спут}}$ )

```

кон

Кстати говоря, с помощью этой программы, задавая различные значения исходных величин (радиуса, числа спутников и др.), можно получать красивые картинки.

Удивить своих товарищей вы можете также, получив на экране следующие изображения:

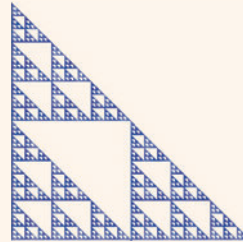


Рис. 2

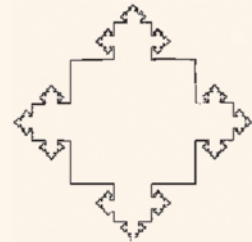


Рис. 3

Картинка на рис. 2 получена так. В прямоугольном треугольнике проводятся все средние линии. Тем самым он разбивается на 4 треугольника. К трем из них, примыкающим к вершинам первоначального треугольника, применяются те же действия. Рекурсивная процедура Треугольник, в которой это реализуется, имеет вид:

```

алг Треугольник (арг цел  $x_a, y_a,$ 
                   $x_b, y_b,$ 
                   $x_c, y_c, n$ )
нач цел  $x_r, y_r, x_q, y_q, x_r, y_r$ 
  если  $n > 0$ 
  то
    |Координаты середин сторон
    |треугольника
     $x_r := (x_b + x_c)/2; y_r := (y_b + y_c)/2$ 
     $x_q := (x_a + x_c)/2; y_q := (y_a + y_c)/2$ 
     $x_r := (x_a + x_b)/2; y_r := (y_a + y_b)/2$ 
    поз( $x_r, y_r$ )
    |Рисуем средние линии
    линия( $x_q, y_q$ )
    линия( $x_r, y_r$ )
    линия( $x_r, y_r$ )
    |Рекурсивно вызываем процедуру
    |Треугольник
    Треугольник( $x_a, y_a, x_r, y_r,$ 
                 $x_q, y_q, n - 1$ )
    |для каждой
    Треугольник( $x_b, y_b, x_r, y_r,$ 
                 $x_r, y_r, n - 1$ )
    |из трех
    Треугольник( $x_c, y_c, x_q, y_q,$ 
                 $x_r, y_r, n - 1$ )
    |сторон

```

все

кон

Основная часть программы:

```
алг Множество_треугольников
нач цел ха, уа, хb, уb, хс, ус
|Координаты вершин самого большого
|треугольника
хс := 0; ус := 0
хb := максX; уb := максY
ха := 0; уа := максY
поз(ха, уа)
|Рисуем самый большой треугольник
линия(хb, уb)
линия(хс, ус)
линия(ха, уа)
|Начинаем рисовать внутренние
|треугольники
Треугольник(ха, уа, хb, уb, хс, ус, б)
```

кон

Изображение на рис. 3 получено следующим образом. На каждой из сторон внутреннего (самого большого) квадрата нарисованы 3 стороны малого квадрата, на каждой из сторон которого также изображены 3 стороны еще меньшего квадрата и т.д. Процедура, которая выполняет соответствующие действия на некотором отрезке с координатами концов x_a, y_a, x_b, y_b , может быть оформлена следующим образом:

```
алг Сторона(арг цел ха, уа, хb, уb, n,
вещ k)
нач цел хр, ур, хq, уq, хr, уr,
xs, ys, dx, dy
|хр, ур, хq, уq, хr, уr, xs, ys -
|координаты вершин малого квадрата
если n = 0
то
линия(хb, уb)
иначе
|k - коэф. уменьшения
dx := 0.5 * (1 - k) * (хb - ха)
dy := 0.5 * (1 - k) * (yb - уа)
|размера квадратов
|Координаты фигуры, изображаемой
|на отрезке ab
хр := ха + dx; ур := уа + dy
xs := хb - dx; ys := yb - dy
хq := хр + (ys - ур);
уq := ур - (xs - хр);
хr := хq + (xs - хр);
ур := уq + (ys - ур)
линия(хр, ур)
|К началу малого квадрата
Сторона(хр, ур, хq, уq, n - 1, k)
Сторона(хq, уq, хr, уr, n - 1, k)
Сторона(хr, уr, xs, ys, n - 1, k)
линия(хb, уb) |к концу отрезка
```

все

кон

Основная часть программы:

```
алг Картинка
нач цел n, хс, ус, b, вещ k
```

```
хс := максX/2; ус := максY/2
|Половина длины стороны базового
|квадрата
b := 100
n := 5; k := 0.4
|Точка, с которой начинается рисунок
поз(хс - b, ус - b)
Сторона(хс - b, ус - b, хс + b,
уc - b, n, k)
Сторона(хс + b, ус - b, хс + b,
уc + b, n, k)
Сторона(хс + b, ус + b, хс - b,
уc + b, n, k)
Сторона(хс - b, ус + b, хс - b,
уc - b, n, k)
```

кон

Когда вы подготовите аналогичные программы на известном вам языке программирования, то получите, мы уверены, истинное удовольствие. Красиво получается, не правда ли? Без рекурсии так, наверное, не сделаешь.

Сложно (но можно — см., например, [2]) без рекурсии решить и известную задачу “Ханойские башни”, в которой речь идет о перекладывании дисков со стержня на стержень по определенным правилам.

Одним из самых ярких примеров использования рекурсии является метод сортировки числовых массивов, разработанный в 1962 г. в Англии профессором Оксфордского университета Ч.Хоаром (С.Хоаре). Этот метод, считающийся самым быстрым из всех известных, основан на рекурсии. О нем мы планируем рассказать в одном из следующих выпусков “В мир информатики”. А сейчас рассмотрим различные формы рекурсивных процедур.

В общем случае любая рекурсивная процедура P включает в себя некоторое множество операторов D и один или несколько операторов рекурсивного вызова P . Как уже отмечалось, главное требование к рекурсивным процедурам и функциям заключается в том, что рекурсивный вызов должен осуществляться по условию, которое в какой-то момент станет ложным. Структура рекурсивных процедур может принимать три разных формы.

1. Форма с выполнением действий после рекурсивного вызова (или с выполнением действий на рекурсивном возврате).

Ее схема:

алг P	или	алг P
нач		нач
если <условие>		если <условие>
то		то
P		P
все		D
D		все
кон		кон

Пример такой процедуры:

<pre> алг ВыводЧисла (арг цел n) нач если n > 1 то ВыводЧисла (n - 1) все вывод n кон </pre>	или	<pre> алг ВыводЧисла (арг цел n) нач если n > 0 то ВыводЧисла (n - 1) вывод n все кон </pre>
---	-----	---

Нетрудно предсказать результат ее выполнения, например, при $n = 5$: 1 2 3 4 5.

2. Форма с выполнением действий до рекурсивного вызова (или с выполнением действий на рекурсивном спуске). Ее схема:

<pre> алг P нач Д если <условие> то P все кон </pre>	или	<pre> алг P нач если <условие> то Д P все кон </pre>
--	-----	--

Пример:

<pre> алг ВыводЧисла (арг цел n) нач вывод n если n > 1 то ВыводЧисла (n - 1) все кон </pre>	или	<pre> алг ВыводЧисла (арг цел n) нач если n > 0 то вывод n ВыводЧисла (n - 1) все кон </pre>
---	-----	---

Результат (при $n = 5$): 5 4 3 2 1.

3. Форма с выполнением действий как до, так и после рекурсивного вызова (или с выполнением действий как на рекурсивном спуске, так и на рекурсивном возврате). Ее схема:

<pre> алг P нач Д если <условие> то P все Д кон </pre>	или	<pre> алг P нач если <условие> то Д P Д все кон </pre>
--	-----	--

Пример:

<pre> алг ВыводЧисла (арг цел n) нач вывод n если n > 1 то ВыводЧисла (n - 1) все вывод n кон </pre>	или	<pre> алг ВыводЧисла (арг цел n) нач если n > 0 то вывод n ВыводЧисла (n - 1) вывод n все кон </pre>
--	-----	--

Ее результат: 5 4 3 2 1 1 2 3 4 5.

Многие задачи безразличны к тому, какая используется форма рекурсивной процедуры. Однако есть классы задач, при решении которых программисту требуется сознательно управлять ходом работы рекурсивных процедур и функций. Поэтому глубокое понимание рекурсивного механизма и умение управлять им по собственному желанию является необходимым качеством квалифицированного программиста.

Существует также разновидность рекурсии, которую называют *косвенной*, или *непрямой*. Такой рекурсией является ситуация, когда процедура А вызывает себя в качестве вспомогательной не непосредственно, а через другую вспомогательную процедуру Б.

Образно косвенную рекурсию можно описать так. Перед зеркалом 1 стоит зеркало 2. Что видно в зеркале 1? Зеркало 2, в котором отражается само зеркало 1. В последнем видно зеркало 2 и т.д.



Рис. 4

В качестве примера косвенной рекурсии можно привести несколько измененную программу рисования изображения на рис. 1:

```

алг Планета (арг цел x, y, рад, n, Nсп,
                вещ к_орб, к_спут)
нач
  если n > 0
  то
    поз(x, y) | Центр планеты
    окружность(рад) | Рисуем планету
    Спутники(x, y, рад, n, Nсп,
            к_орб, к_спут)
    | Спутники - вспомогательный алгоритм
  все
кон
алг Спутники(арг цел x, y, рад, n, Nсп,
             вещ к_орб, к_спут)
нач цел x1, y1, i, вещ угол, рад_орб
  рад_орб := рад * к_орб
  | Радиус орбиты спутников
  угол := 6.28 / Nсп
  | Угол между спутниками
  нц для i от 1 до Nсп
  | Для каждого спутника
    x1 := x + рад_орб * cos(угол * i)
    | Координаты центра
    y1 := x + рад_орб * sin(угол * i)
    | i-го спутника
    | Рекурсивный вызов процедуры Планета
    Планета(x, y, int(рад * к_спут),
            Nсп - 1, к_орб, к_спут)
  кц
кон

  Основная часть программы:
алг Система планет
нач цел n, Nсп, рад, x, y,
     вещ к_орб, к_спут
  n := 7; Nсп := 6
  x := максX / 2 | Координаты центра
  x := максX / 2 | планеты П
  рад := 120; к_орб := 1.6; к_спут := 0.5
  Планета(x, y, рад, n, Nсп, к_орб, к_спут)
кон

```

Процедура Планета выводит на экран центральную окружность-планету, а для рисования ее спутников используется вспомогательная процедура Спутники. Последняя рисует спутники с помощью рекурсивного вызова процедуры Планета. Не кажется ли вам, читатель, что в таком виде программа даже более логична и понятна, чем с “обычной” рекурсией?

Примечание. При написании аналогичной программы на языке Паскаль возникнет проблема, связанная с тем, что процедура Планета использует в качестве вспомогательной еще не описанную процедуру Спутники, что в этом языке недопустимо. Последнюю процедуру также нельзя описать первой, поскольку в ней имеется вызов процедуры Планета. Выходом здесь является применение так называемого “опережающего” описания процедур.

Справедливость требует, чтобы мы добавили, как говорится, “в бочку меда немного дегтя” ©. Вспомним задачу разработки функции для расчета k-го члена последовательности Фибоначчи. Схема работы созданной рекурсивной функции Фиб при расчете 6-го члена последовательности приведена на рис. 5.

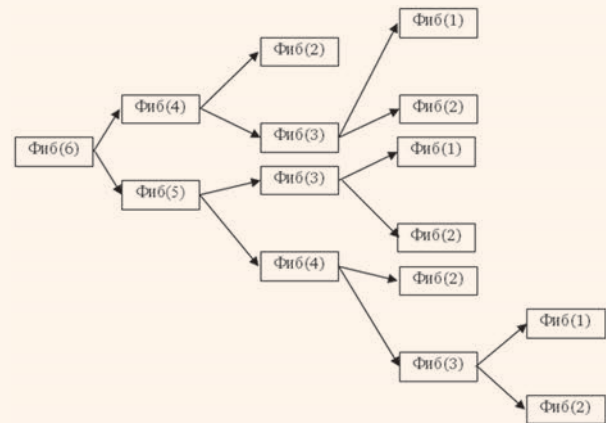


Рис. 5

Из рис. 5 видно, что одна и та же функция с некоторым конкретным (фактическим) значением параметра вызывается и выполняется неоднократно.

В процессе вычислений, например Фиб(17), функция для расчета Фиб(15) будет вызываться 2 раза, Фиб(14) — 3 раза, Фиб(13) — 5 раз, Фиб(12) — 8 раз и т.д.

Всего при вычислении Фиб(17) компьютер выполнит более 1000 операций сложения, Фиб(31) — более 1 млн., Фиб(45) — более 1 миллиарда операций сложения! В то же время в нерекурсивном варианте определение 45-го члена последовательности потребует всего 43 операции сложения, то есть рекурсивные процедуры и функции — эффективны, но не всегда эффективны.

Если о рекурсии вы все поняли и “думаете, что это легко, попробуйте проделать это сами” (см. эпиграф к статье).

Итак, задания для самостоятельной работы.

1. Составить программу, в результате выполнения которой на экран 10 раз выводится текст истории о попе и его собаке (“У попа была собака...”).

Указания по выполнению. Для того чтобы наблюдать вывод каждой новой истории, предусмотрите в процедуре небольшую паузу в выполнении программы.

2. Составить программы для расчета:

- 1) значения a^n для заданных значений a и n (a — вещественное число, n — натуральное);
- 2) n -го члена арифметической прогрессии;
- 3) то же, геометрической;
- 4) суммы n первых членов арифметической прогрессии;
- 5) то же, геометрической;
- 6) минимального элемента массива;
- 7) индекса минимального элемента массива.

3. Составить программу, в которой вводятся 10 заданных чисел, а затем они выводятся на экран в обратном порядке. Каждое из заданных чисел должно вводиться один раз. Массивы не применять.

4. Дано натуральное число n . Вывести все его делители в порядке их возрастания. Проверку воз-

можных делителей числа n проводить только до значения, не превышающего \sqrt{n} (все остальные делители получаются в результате деления числа n на ранее найденные делители).

Все задачи решите с использованием рекурсивных функций и процедур.

Разработайте также программы для построения изображений на рис. 1–3.

Программы (можно не все), пожалуйста, присылайте в редакцию. Срок представления — 1 ноября 2014 г. Авторы лучших работ будут награждены дипломами.

Литература

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

2. Вьюкова Н.И., Галатенко В.А., Ходулев А.Б. Систематический подход к программированию. М.: Наука, 1988.

ЗАДАЧНИК

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

Задача “Три учителя”

Напомним, что требовалось установить, какие предметы — математику, физику, химию, историю, литературу или английский язык преподает каждый из трех учителей — Иванов, Петров и Сидоров, если каждый из них преподает по два предмета. Известно, что:

- 1) Сидоров — самый молодой из них;
- 2) учитель химии старше учителя истории;
- 3) все трое — учитель химии, учитель физики и Петров — любят играть в шахматы;
- 4) когда между собой играют учитель литературы и учитель английского языка, то Сидоров болеет за последнего;
- 5) Петров не знает английского языка и не преподает математику.

Решение

На основании известных из условия фактов можно составить таблицу (в ней символом “–” отмечен невозможный вариант, в скобках — номер факта, на основании которого такой вывод сделан).

Из таблицы и ее последующего заполнения следует, что Петров преподает историю и литературу, Сидоров — математику и физику, Иванов — химию и английский язык.

Учитель	Предмет					
	Математика	Физика	Химия	История	Литература	Английский язык
Иванов						
Петров	– (5)	– (3)	– (3)			– (5)
Сидоров			– (1, 2)		– (4)	– (4)

Правильный ответ прислали:

— Абдуллина Лина, Габбасов Ильяс, Ельницкая Анна, Зыков Иван, Истомина Полина, Морозов Иван, Мусатов Максим, Сатыбалдина Ксения, Сыч Андрей, Тонт Анастасия и Шигаев Никита, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Абдувахидова Алина, Абдувахидова Софья, Головченко Тихон, Попов Никита, Серов Георгий и Телегин Дмитрий, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

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

— Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Васина Светлана и Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

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

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Курышова Елизавета, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнева М.В.**;

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

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

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

Кроссворд (январский выпуск)

Ответы

По горизонтали: 2. Текст. 5. Поле. 7. Фон. 8. Поколение. 11. Исполнитель. 12. Тиристор. 13. Ять. 15. Право. 16. Курсор. 18. Диск. 19. Арбитр. 22. График. 26. Калькулятор.

По вертикали: 1. Лого. 2. Тело. 3. Кибернетика. 4. Тори. 6. Компьютер. 8. Принтер. 9. Копирование. 10. Елец. 14. Истина. 15. Подача. 17. Порт. 20. Угол. 21. Икс. 24. Шаг. 25. Лье.

Ответы представили:

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

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

— Вяткин Игорь, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Герасименко Богдан, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Гололобов Дмитрий, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Другова Анастасия и Киришина Татьяна, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

— Курышова Елизавета, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Мищенко Маргарита, Краснодарский край, г. Приморско-Ахтарск, школа №22, учитель **Корнеева М.В.**;

— Новикова Анна и Потапова Алевтина, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Кроссворд (февральский выпуск)

Ответы

По горизонтали: 2. Алгоритм. 5. Ток. 7. Инверсия. 8. Зет. 9. Файл. 10. Игра. 12. Код. 13. Арбитр. 16. Грамм. 17. Отладка. 18. Нажатие. 20. База. 24. Ре. 25. Программист. 26. Логика.

По вертикали: 1. Домен. 3. Линейка. 4. Триггер. 6. Катод. 8. Закладка. 9. Фортран. 11. Килобит. 14. Рамка. 15. Параметр. 19. Автор. 20. Бейсик. 21. Тип. 22. Три. 23. Омега.

Ответы прислали:

— Ахметова Динара, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

— Мищенко Маргарита, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

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

— Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

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

— Якушов Александр, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина, учитель **Чапкевич И.М.**

Задача “Равенства”

Напомним, что требовалось установить, в какой системе счисления справедливы равенства:

а) $3 \times 4 = 10$;

б) $5 \times 5 = 31$;

в) $2 \times 2 = 100$.

Ответы

а) в двенадцатеричной;

б) в восьмеричной;

в) в двоичной.

Правильные ответы представили:

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

— Богданова Алина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Вяткин Игорь и Колесников Антон, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

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

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Крысанов Виктор, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Лазаренко Нина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Рябых Анна и Цугуй Елена, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Ситникова Елена, средняя школа поселка Осинновка, Алтайский край, учитель **Евдокимова А.И.**;

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

Задание “Файлы и папки с масками”

Напомним, что предлагались задачи, связанные со структурой “дерева” папок и файлов и с именами последних.

Задание 2

Среди файлов, имена которых указаны на рис. 1, отберите файлы, соответствующие маске:

- | | |
|-------------|-------------|
| 1) *.txt; | 3) ?о*. *; |
| 2) 800*. *; | 4) *н*. ??? |

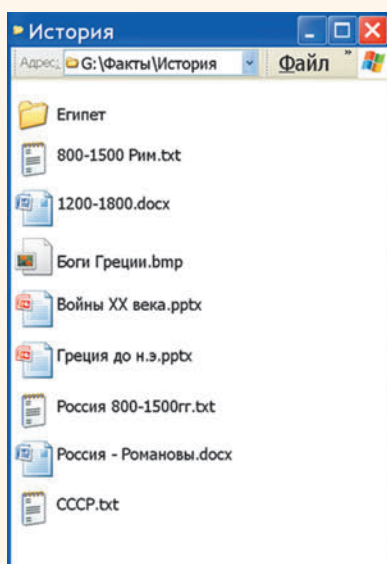


Рис. 1

Решение

Прежде всего скажем, что символ “?” в маске соответствует одному любому символу в имени файла (в его основной части и в расширении), символ “*” — любой последовательности символов, в том числе “пустой”.

Маска *.txt объединяет файлы с любым именем и расширением “txt”, то есть: 800–1500 Рим.txt; Россия 800–1500гг.txt; СССР.txt.

Маске 800*. * соответствуют файлы с любым расширением, основная часть имени которых начинается с трех символов “800”, в нашем случае — файл 800–1500 Рим.txt.

Маска ?о*. * объединяет файлы с любым расширением, в основной части имени второй символ — буква “о”, то есть файлы Боги Греции.bmp; Войны XX века.pptx; Россия 800–1500гг.txt; Россия – Романовы.docx.

Маске *н*. ??? соответствуют файлы, расширение имени которых состоит из трех символов, а в основной части имени имеется буква н. Файлы с именами Войны XX века.pptx; Россия – Романовы.docx и Греция до н.э.pptx под эту маску не подходят, так как их расширение состоит из четырех символов.

Задание 3

Укажите, какая маска объединяет файлы 1200–1800.docx, Россия 800–1500гг.txt, 800–1500 Рим.txt, Войны 20 века.txt в списке на рис. 1.

Решение

У всех перечисленных файлов в расширении имеется символ “x”. До него в расширении один или три символа, после — один символ или “пустая” последовательность символов. Значит, расширение имен файлов соответствует маске “*x*”. Аналогично можно установить, что для основной части имен маска такая: “*0*”, то есть полная маска имен файлов: “*0*. *x*”.

Ответы прислали:

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

— Антипов Анатолий, средняя школа поселка Осинновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Дмитриенко Максим, г. Благовещенск, гимназия № 25, учитель **Максимова Е.П.**;

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Лазаренко Нина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Махонина Елизавета, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Новикова Анна и Потапова Алевтина, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Попов Никита, Серов Георгий и Телегин Дмитрий, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Стороженко Степан, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Четыре вопроса (рубрика “Поиск информации”)

Ответы

1–2. Песня “Падает снег” бельгийского эстрадно-го певца Сальваторе Адамо стала японской национальной песней о сакуре в хлопьях снега.

3. Согласно скандинавским легендам, боги сотворили первую женщину из ольхи (в других вариантах перевода — из ивы и из вяза).

4. Пряность, которую больше всего не любят клопы и тараканы, — анис.

Правильные ответы представили:

— Аксененко Сергей, Перова Валентина, Иванов Иван и Яковлева Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Бородюк Анна и Василенко Татьяна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Дмитриенко Максим, г. Благовещенск, гимназия № 25, учитель **Максимова Е.П.**;

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

— Евграфова Ксения, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Есакова Анна, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Лежнева Александра, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Торопова Александра, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Головоломка “Последняя табличка”

Напомним, что нужно было определить, как должна выглядеть последняя табличка, чтобы закон изменения положения символов во всех табличках сохранился:

7	X								
6									
5									
4									
3									
2		o	#						
1									

Решение

Кружок “o” перемещается на одну клетку вверх за каждый шаг. Символ “#” (его называют “диз”) скачет — “прыгает” сначала на две клетки вверх, потом на одну клетку вниз. Символ “X” двигается по часовой стрелке вдоль периметра на количество клеток, равное сумме номеров уровней кружка “o” и дизза “#”, то есть $2 + 2 = 4$; $3 + 4 = 7$; $4 + 3 = 7$; $5 + 5 = 10$; $6 + 4 = 10$. Следовательно, последняя табличка будет выглядеть так:

Ответы представили:

— Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимов А.И.**;

— Волков Владимир и Глушаков Андрей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Вяткин Игорь, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

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

— Живило Андрей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

— Лежнева Александра, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Манукян Григорий и Телегин Дмитрий, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Михайлов Иван, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Задача “Кто есть кто?”

Напомним, что требовалось определить, кто из двух людей — *A* и *B* — рыцарь (всегда говорит правду), а кто лжец (всегда врет), если *A* сказал: “По крайней мере один из нас лжец”.

Решение

Допустим, что *A* — лжец. Тогда его ложное утверждение “По крайней мере один из нас лжец” означает, что лжецов на острове нет. Но это противоречит сделанному допущению (к тому же известно, что один из встреченных людей является лжецом). Значит, *A* — рыцарь, а *B* — лжец.

Ответ: *A* — рыцарь, *B* — лжец.

Правильный ответ прислали:

— Волков Владимир и Глушаков Андрей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Григоренко Василий и Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Живило Андрей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Зубов Владислав, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

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

— Михайлов Иван, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Семенов Андрей и Турков Андрей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

Головоломка “На шахматной доске”

Напомним, что предлагалось перечислить все маршруты, ведущие из клетки с3 шахматной доски в клетку a1, соблюдая следующее правило: каждый ход делается либо на одну клетку влево, либо на одну клетку вниз, либо на одну клетку вниз и одну клетку влево.

Ответ. Возможны 13 вариантов маршрута: 1. ЛЛНН. 2. ЛДН. 3. ЛНЛН. 4. ЛНД. 5. ЛННЛ. 6. ДЛН. 7. ДД. 8. ДНЛ. 9. НЛЛН. 10. НЛД. 11. НЛНЛ. 12. НДЛ. 13. ННЛЛ, где буквами обозначены ходы:

- Л — ход влево;
- Н — ход вниз;
- Д — ход по диагонали (вниз–влево).

Ответы прислали:

— Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Вяткин Игорь, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Григоренко Василий, Григоренко Дмитрий, Круглякова Мария и Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Зубов Владислав, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

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

— Семенов Андрей и Турков Андрей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

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

Квадрат из костей домино

Напомним условие: “Кости домино выложены в виде кружевной салфетки, причем сумма очков в каждом вертикальном и горизонтальном рядах одна и та же. Значения очков на костях, кроме “пустышек”, зашифрованы буквами (см. рисунок). Расшифруйте их.

	W	W	X	X	X	Z	W
		Z	Z	Z			V
	X	W	V	X	U	V	X
	Z	Y			U	Y	V
Y	V	X			Z		V
Y	U	W	Y	U		X	U
Y		W	W	V	Z		Y
Z	Y	U	U	U	W	V	

Заметим, что методом проб и подбором найти решение сложно. Попробуйте решить задачу, так сказать, “с помощью математики”.

Решение

Прежде всего можно обнаружить, что выложены 28 костей домино, причем в “салфетке” нет двух одинаковых костей. Следовательно, использован полный набор костей домино.

Общее число очков на всех костях равно 168 (проверьте!), поэтому сумма в каждом вертикальном и горизонтальном рядах составляет 21.

Из 1-го столбца следует: $3Y + Z = 21$.

Из 2-й строки — $3Z + V = 21$.

Из первого уравнения получается, что Z кратно трем, из второго — что V кратно трем, значит, одно из них 3, другое — 6.

Если $Z = 3$, а $V = 6$, то $3Z + V = 15$, а не 21. Если же $Z = 6$, а $V = 3$, то второе уравнение удовлетворяется, а из первого находим, что $Y = 5$.

Из 4-й строки: $U = 2$, из 5-й — $X = 4$, из 1-й — $W = 1$. Окончательное решение — на рисунке.

0	1	1	4	4	4	6	1
0		6	6	6	0		3
0	4	1	3	4	2	3	4
0	6	5			2	5	3
5	3	4			6	0	3
5	2	1	5	2	0	4	2
5		1	1	3	6		5
6	5	2	2	2	1	3	0

Ответы прислали:

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Живилов Андрей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Зубов Владислав, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Колесников Антон, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

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

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.** (Тарас решил задачу, разработав компьютерную программу);

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

— Семенов Андрей и Турков Андрей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

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

Задача “Серебряная цепочка”

Напомним условие: “В гостиницу приехал путешественник. Выяснилось, что деньги он забыл дома. У него была лишь серебряная цепочка из семи звеньев. Он договорился с хозяином гостиницы о том, что за каждый день пребывания в гостинице будет расплачиваться одним звеном цепочки.

Какое звено цепочки надо аккуратно расцепить, чтобы прожить в гостинице семь дней и ежедневно расплачиваться с хозяином? (Хозяин может давать сдачу звеньями, полученными им ранее.)”

Решение

Десятичные числа 1, 2, ..., 7 можно “получить” как сумму одного или нескольких чисел 1, 2 и 4. Значит, надо иметь 1, 2 и 4 звена. Их можно получить, если расцепить третье или пятое звенья — цепочка распадется на три части: в одной два звена, в другой — четыре звена, а в третьей части — одно (разъединенное) звено. С их помощью удастся расплатиться в течение семи дней так, как показано в таблице:

Номер дня проживания, d	Количество звеньев цепочки, отдаваемых в d -й день	Количество звеньев, возвращаемых хозяином
1	1	–
2	2	1
3	1	
4	4	1 + 2
5	1	–
6	2	1
7	1	–

Правильные ответы представили:

— Волков Владимир, Демьянова Елена и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

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

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Колесников Антон, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Коробов Сергей, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

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

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

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

— Стороженко Степан, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Вопрос о карандаше (“Сколько граней у шестигранного карандаша?”)

Правильный ответ (“У незаточенного шестигранного карандаша 8 граней”) представили:

— Бальцеров Семен, средняя школа поселка Ерофей Павлович, Амурская обл., Сквородинский р-н, учитель **Краснёнкова Л.А.**;

— Волченков Алексей и Жарова Мария, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Габбасов Ильяс, Ельницкая Анна, Зыков Иван, Истомина Полина, Морозов Иван, Мусатов Максим, Мусатов Тимофей (ученик 2-го класса), Сатыбалдина Ксения, Сыч Андрей и Тонт Анастасия, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Дмитриенко Максим, г. Благовещенск, гимназия № 25, учитель **Максимова Е.П.**;

— Казаков Сергей и Чуйкова Яна, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Колесников Владимир (ученик 2-го класса), Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

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

— Мищенко Маргарита, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

— Федосеева Анастасия, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Харламова Виктория, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Напомним, что вопрос был адресован ученикам 1–7-х классов.

Японские головоломки “судоку”, опубликованные в январском выпуске “В мир информатики”, правильно решили:

— Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Волков Владимир и Глушаков Андрей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Вяткин Игорь, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

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

— Коростелев Иннокентий и Марун Виталий, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

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

— Окаемов Станислав, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Машонина Ирина, г. Фрязино Московской обл., школа № 4, учитель **Сенюта Е.И.**;

— Новиков Сергей и Хромченкова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Новиков Филипп и Цыплаков Евгений, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Пахомов Владислав, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Головоломку “судоку” без цифр, в которой нужно было расставить все заданные блоки с цифрами так, чтобы форма блоков совпадала и при этом соблюдались правила “классического” судоку (в каждом столбце, в каждой строке и в каждом квадратике 3×3 цифры не должны повторяться), правильно решили:

— Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Волков Владимир и Глушаков Андрей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Колесников Антон и Колесников Владимир, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Новиков Сергей и Хромченкова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Новиков Филипп и Цыплаков Евгений, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Окаемов Станислав, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Пахомов Владислав, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

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

— Шукин Василий, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Решение головоломки “Деревья и скамейки” прислали:

— Вяткин Игорь и Колесников Антон, Удмуртская Республика, г. Можга, школа № 1, учитель **Колесникова С.В.**;

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Коробов Сергей и Марков Алексей, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**

Программы для построения графиков функций в полярных координатах разработали:

— Кисельников Сергей, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

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

— Филимонова Галина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Антон, Галина, Елена и Сергей будут награждены дипломами. Поздравляем!

Разбор задачи “Необычный маршрут” и числового ребуса “Трижды ШЕСТЬ” будет проведен в рубрике “Крепкий орешек” в осенних выпусках журнала.

Итоги конкурса № 107

Напомним условие: “Для перевода десятичного числа 241 762 в двоичную систему методом последовательного деления числа на основание понадобится выполнить 17 операций деления. Предложите методу перевода, согласно которой можно решить задачу за гораздо меньшее число операций деления”.

Участниками конкурса являлись:

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

— Рябых Анна и Цугуй Елена, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

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

— Пахомов Владислав, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Все перечисленные читатели будут награждены дипломами. Поздравляем!

Решение

Необходимо запомнить, например, двоичные представления восьмеричных цифр:

- 0 — 000
- 1 — 001
- 2 — 010
- 3 — 011
- 4 — 100
- 5 — 101
- 6 — 110
- 7 — 111,

а затем методом последовательного деления на основание перевести заданное число в восьмеричную систему счисления (всего понадобится пять операций деления):

$$\begin{array}{r}
 2\ 4\ 1\ 7\ 6\ 2\ |8 \\
 \underline{2\ 4\ 1\ 7\ 6\ 0\ 3\ 0\ 2\ 2\ 0\ |8} \\
 2\ 3\ 0\ 2\ 1\ 6\ 3\ 7\ 7\ 7\ |8 \\
 4\ 3\ 7\ 7\ 6\ 4\ 7\ 2\ |8 \\
 1\ 4\ 7\ 2\ 5\ 9\ |8 \\
 0\ 5\ 6\ 7 \\
 3
 \end{array}$$

после чего в полученном числе 730142 заменить каждую цифру ее двоичным представлением:
111011000001100010.

При использовании в качестве “промежуточной” не восьмеричной, а шестнадцатеричной системы счисления количество операций деления будет еще меньше.

ЛИЧНОСТИ

Фредерик Брукс

Фредерик Брукс (родился в 1932 году) — американский ученый в области теории вычислительных систем. Руководил разработкой группой операционных систем для семейства компьютеров OS/360 в фирме IBM, начиная с 1964 года.

В 1999 году награжден самой престижной премией в информатике — Премией Тьюринга — “за исторически значимый вклад в архитектуру компьютеров, операционные системы и инженерию программного обеспечения”.

В 1975 году Ф.Брукс написал книгу “Мифический человек-месяц, или Как создаются программные системы” об управлении проектами в области разработки программного обеспечения. Отрывками из этой книги мы хотим ознакомить наших читателей.



Эпилог. Пятьдесят лет удивления, восхищения и радости

В моей памяти все еще живы удивление и восторг, с которым я — мне тогда было 13 лет — читал отчет от 7 августа 1944 года об освящении компьютера Mark I, архитектором которого был Говард Айкен (Howard Aiken), а проектировщиками — инженеры Клер Лейк (Clair D. Lake), Бенджамин Дурфи (B. M. Durfee) и Фрэнсис Гамильтон (F. E. Hamilton). Такой же вызывающей ощущение чуда была статья Ванневары Буша (Vannevar Bush) “That We May Think” в апрельском номере “Atlantic Monthly” 1945 года, в которой он предложил организовать знания в виде огромной гипертекстовой паутины и обеспечить пользователей машинами для переходов по существующим ссылкам и создания новых ассоциативных следов.

Новый толчок моя страсть к компьютерам получила в 1952 году, когда, работая летом на IBM в Эдинкоте, штат Нью-Йорк, я получил практический опыт программирования для IBM 604 и формальное обучение программированию для IBM 701, их первой машины с хранимой программой. Аспи-

Тарас Леженников из школы № 22 г. Приморско-Ахтарска предложил решать задачу методом выделения максимальных степеней числа 2. При описанном методе перевода деление вообще не используется (что, конечно, “гораздо меньше”, чем 17 операций деления ☺).

рантура у Айкена и Иверсона в Гарварде сделала реальностью мои мечты о профессии, и я связал с ней всю свою жизнь. Не многим Бог дает право зарабатывать на жизнь тем, чем они с радостью занимались бы по собственной воле, по увлечению. Я благодарен судьбе.

Для человека, влюбленного в компьютеры, трудно было бы придумать иное время, когда так радостно было жить. От механических устройств до вакуумных ламп, транзисторов и интегральных схем шло бурное развитие технологии. Первый компьютер, на котором я работал сразу после выпуска из Гарварда, был суперкомпьютер IBM Stretch. Этот компьютер царствовал над миром как самый быстрый с 1961-го по 1964 годы; было изготовлено 9 экземпляров. Мой сегодняшний Macintosh Powerbook не только быстрее, с большей памятью и большим диском, но и в тысячу раз дешевле (в пять тысяч раз дешевле с учетом инфляции). Мы были свидетелями того, как поочередно произошли компьютерная революция, революция электронных компьютеров, революция мини-компьютеров и революция микрокомпьютеров, в результате каждой из которых компьютеров становилось на порядки больше.

Радости профессии

Почему заниматься программированием интересно? Какими радостями вознаграждаются те, кто им занимается?

Во-первых, это просто радость, получаемая при создании чего-либо своими руками. Как ребенок радуется, делая куличики из песка, так и взрослый получает удовольствие, создавая какие-либо вещи, особенно если сам их и придумал. Я думаю, что этот восторг — отражение восторга Господа, творящего мир, восторга, проявляющегося в индивидуальности и новизне каждого листочка и каждой снежинки.

Во-вторых, это удовольствие создавать вещи, которые могут быть полезны другим людям. Глубоко в душе мы испытываем потребность в том, чтобы другие использовали результаты нашего труда и считали их полезными. В этом отношении программная система по своей сути — то же, что и изготовленная ребенком подставка для карандашей “папе в подарок”.

В-третьих, это очарование создания сложных головоломных объектов, состоящих из взаимодействующих движущихся частей и наблюдения за их

работой, круг за кругом демонстрирующей результаты изначально заложенных принципов. Компьютер с работающей на нем программой обладает доведенным до высшего предела очарованием игрового или музыкального автомата.

В-четвертых, это радость, получаемая от неизменного узнавания нового, проистекающего из неповторимой природы задачи. В том или ином отношении задача всегда ставится по-новому, и тот, кто ее решает, получает новые знания — либо практические, либо теоретические, либо те и другие вместе.

Наконец, наслаждение доставляет работа со столь податливым материалом. Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки в воздухе и из воздуха, творя силой воображения. Трудно найти другой материал, используемый в творчестве который столь же гибок, прост для шлифовки или переработки и доступен для воплощения грандиозных замыслов. Однако программная конструкция в отличие от поэтических творений реальна, в том смысле, что она движется и работает, производя видимые результаты, которые отделимы от самой конструкции. Она печатает результаты, рисует картинки, производит звуки, приводит в движение рычаги. В наше время осуществилось волшебство мифа и легенды. С клавиатуры вводится верное заклинание, и экран монитора оживает, показывая то, чего никогда не было и не могло быть.

Таким образом, программирование доставляет удовольствие, поскольку отвечает глубокой внутренней потребности в творчестве и удовлетворяет чувственные потребности, которые есть у всех нас.

Печали профессии

Не все, однако, в радость, и если предвидеть прищипывание этому ремеслу огорчения, то они легче переносятся.

Во-первых, необходима безошибочная точность действий. В этом отношении компьютер также напоминает волшебство. Один неверный знак, одна пауза в заклинании, и чудо не состоялось. Человеку не свойственно совершенство, и оно является необходимым лишь в немногих сферах его деятельности. Мне кажется, что при освоении программирования труднее всего привыкнуть к требованию совершенства.

Кроме того, постановка задач, обеспечение ресурсами и предоставление информации осуществляются другими людьми. Редко удается контролировать условия работы и даже ее цели. На языке администрирования это означает, что полномочия ниже ответственности. Впрочем, похоже, что в любой работе, где должен быть получен результат, формальная власть никогда не соизмерима с ответственностью. На практике фактическая (в противоположность формальной) власть приобретается в результате успешного выполнения задач.

Зависимость от других имеет особенно неприятную системную сторону программиста. Он находится в зависимости от программ, написанных другими людьми, и эти программы зачастую плохо спроектированы, слабо написаны, получены в неполном виде (без исходного текста и контрольных примеров) и плохо документированы. Поэтому программисту приходится тратить многие часы на изучение и исправление вещей, которые в идеале должны быть полными, доступными и годными к использованию.

Следующий “минус” связан с тем, что разработка грандиозных идей — это удовольствие, а поиск паршивых маленьких “жучков”⁴ — это всего лишь работа. В каждом творческом деле бывают ужасные периоды однообразного и кропотливого труда, и программирование не является исключением.

Далее оказывается, что при отладке программы сходимость является линейной, если не хуже, хотя можно было предполагать некое квадратичное приближение к окончанию. В итоге отладка продолжается долго, причем на поиск последних, более сложных, ошибок уходит больше времени, чем на отыскание первых.

Последняя горесть, а часто и последняя капля, — то, что продукт, на который было положено столько труда, оказывается устаревшим в момент его завершения (или даже раньше). Коллеги и конкуренты уже с пылом работают над новыми и лучшими идеями. И уничтожение плода вашей мысли уже не только задумано, но и запланировано.

На самом деле положение обычно лучше, чем кажется. В то время как ваш продукт уже завершен, этот новый и лучший продукт, как правило, отсутствует на рынке, о нем лишь много разговоров, и для его разработки потребуются месяцы. Настоящий тигр не пара бумажному, если требуется реальное использование. Реальное существование имеет преимущества.

Конечно, технологическая основа разработки всегда развивается. Как только разработка проекта закончена, он становится устаревшим в смысле заложенных в нем концепций. Но для осуществления реального проекта необходимо разбиение на стадии и уровни. Судить о том, является ли некая реализация устаревшей, можно лишь сравнивая ее с другими существующими реализациями, а не с нереализованными идеями. Трудность и цель состоят в том, чтобы найти реальные решения для реальных задач в установленные сроки, используя имеющиеся ресурсы.

Таково программирование — и смоляная яма, в которой увязли многие проекты, и творчество со своими радостями и печалью. Для многих радости значат гораздо больше, чем печали. Для них и написана эта книга в попытке проложить какие-то мостки через это болото.

⁴ Автор имеет в виду процесс поиска и устранения ошибок в программе — ее отладку (по-английски *debugging* — от слова *bug* — жук). О происхождении термина *bug* (*bug*) — см. статью В.В. Шиловой “Этот зловерный «bug»” в “Информатике” № 11 за 2012 год. — Прим. ред.

Редакция сообщает, что по итогам 2013–2014 учебного года будут отмечены дипломами читатели, принявшие наиболее активное участие в выполнении заданий рубрики “Поиск информации” и в решении кроссвордов (итоги будут подведены после получения ответов на задания, опубликованных в апрельском выпуске “В мир информатики”, а награждение — в новом учебном году).

Спасибо вам, уважаемые коллеги!

За активную работу с учащимися в 2013–2014 учебному году по материалам раздела “В мир информатики” руководство издательского дома “Первое сентября” и редакция журнала “Информатика” выражает благодарность следующим учителям:

- **Абизяевой Валентине Николаевне**, Республика Татарстан, г. Нижнекамск, школа № 27;
- **Алейниковой Галине Николаевне**, Вадыковская средняя школа, Брянская обл., Погарский р-н;
- **Артамоновой Виктории Васильевне**, средняя школа поселка Новопетровский Московской обл.;
- **Багаутдиновой Р.Р.**, Республика Татарстан, г. Казань, МБОУ “Татарская гимназия № 1”;
- **Багровой Ольге Алексеевне**, средняя школа города Пионерский, Калининградская обл.;
- **Барабановой Ольге Владимировне**, Алтайский край, средняя школа села Сараса Алтайского р-на;
- **Беловой И.Л.**, г. Санкт-Петербург, школа № 639;
- **Богдановой Людмиле Михайловне**, Республика Карелия, поселок Надвоицы, школа № 1;
- **Болдыревой Светлане Владимировне**, Республика Башкортостан, г. Уфа, гимназия № 3 им. А.М. Горького;
- **Брунову Александру Сергеевичу**, средняя школа села Ириновка, Новобурасский р-н Саратовской обл.;
- **Валиевой Дание Ильясовне**, средняя школа села Сулево им. Р.Г. Галева, Республика Татарстан, Альметьевский р-н;
- **Валиевой Раушании Нурлимановне**, Чистопольско-Выселская средняя школа, Республика Татарстан, Чистопольский р-н;
- **Васниной Ольге Васильевне**, Костромская обл., Буйский р-н, г.п.п. Чистые Боры, школа № 1;
- **Воеводиной Розе Витальевне**, Чувашская Республика, г. Канаш, Канашский педагогический колледж;
- **Волкову Юрию Павловичу**, Владимирская обл., г. Струнино, школа № 11;
- **Гавриловой Маргарите Игоревне**, г. Пенза, школа № 512;
- **Галиевой Рузиле Ферролевне**, Республика Татарстан, Актанышский р-н, село Актаныш, средняя школа № 1;
- **Гильзер Н.В.**, Республика Башкортостан, г. Уфа, лицей № 60;
- **Гранаткиной О.М.**, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”;
- **Долговой Галине Александровне**, средняя школа села Восточное Нижегородской обл.;
- **Евдокимовой Александре Игоревне**, средняя школа поселка Осиновка, Алтайский край;
- **Евдокимовой Наталье Леонидовне**, средняя школа села Новое Барятино, Республика Башкортостан, Стерлитамакский р-н;
- **Жариковой Елене Николаевне**, Совхозная средняя школа, Московская обл., Серебряно-Прудский р-н, поселок Успенский;
- **Загафурановой Альбине Фингатовне**, средняя школа села Сейтяково Балтачевского р-на, Республика Башкортостан;
- **Ивановой Ларисе Николаевне**, средняя школа деревни Павлищево, Московская обл., Можайский р-н;
- **Игошевой Анне Аркадьевне**, Свердловская обл., г. Ревда, школа № 10;
- **Исаковой Екатерине Валерьевне**, г. Кострома, гимназия № 33;
- **Искандаровой Айгуль Римовне**, Республика Башкортостан, г. Уфа, школа № 18; школа № 54 (Центр дистанционного обучения);
- **Кирилловой Л.Н.**, Чувашская Республика, г. Канаш, Канашский педагогический колледж;
- **Клоповой Елене Владимировне**, средняя школа села Дохновичи Брянской обл.;
- **Колесниковой Светлане Владимировне**, Удмуртская Республика, г. Можга, школа № 1;
- **Колодюку Василию Николаевичу**, средняя школа села Черемушка, Красноярский край, Каратузский р-н;
- **Комбаров Светлане Ивановне**, г. Воронеж, лицей № 2;
- **Кординой Нине Евгеньевне**, Смоленская обл., г. Демидов, школа № 1;
- **Корнеевой Марине Викторовне**, Краснодарский край, г. Приморско-Ахтарск, школа № 22;
- **Краснёнковой Лидии Александровне**, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н;
- **Куликовой Елене Анатольевне**, г. Киров, школа № 28;
- **Куликовой Т.Н.**, Московская обл., г. Мытищи, школа “Логос М”;
- **Липиной Вере Мурадовне**, г. Москва, школа № 827;
- **Максимовой Екатерине Петровне**, г. Благовещенск, гимназия № 25;
- **Марцинкевич Елене Евгеньевне**, г. Рязань, школа № 44;
- **Меньшикову Виталию Владимировичу**, Республика Карелия, г. Сегежа, школа № 5;
- **Михолап Эльвире Леонидовне**, г. Киров, школа № 28;
- **Мусатовой Ирине Борисовне**, Челябинская обл., г. Златоуст, школа № 9;
- **Муравьевой Ольге Викторовне**, средняя школа деревни Муравьево, Вологодская обл.;
- **Пуйшо Николаю Валерияновичу**, г. Саратов, школа № 52;
- **Пышной Елене Александровне**, Кабардино-Балкарская Республика, г. Нальчик, школа № 32;
- **Романовой Елене Викторовне**, г. Кострома, гимназия № 33;
- **Рябенко Наталье Родионовне**, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6;
- **Сенюта Елене Ивановне**, Московская обл., г. Фрязино, школа № 4;
- **Серовой Ларисе Александровне**, Свердловская обл., г. Качканар, средняя школа им. К.Н. Новикова;
- **Уливановой Розалии Минировне**, Республика Татарстан, г. Альметьевск, гимназия № 5;
- **Фатхутдиновой Альбине Анваровне**, Адельшинская средняя школа, Республика Татарстан, Чистопольский р-н;
- **Цикиной Елене Николаевне**, г. Ярославль, школа № 33;
- **Цыганковой Ие Юрьевне**, Вадыковская средняя школа, Брянская обл., Погарский р-н;
- **Чапкевич Ирине Михайловне**, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина;
- **Черновой Ларисе Ивановне**, средняя школа села Сердар, Республика Марий Эл;
- **Чернышевой Инне Анатольевне**, г. Воронеж, школа № 5 им. К.П. Феоктистовой;
- **Шитовой Любови Александровне**, средняя школа села Горелово Тамбовской обл.;
- **Юртаевой Галине Юрьевне**, г. Челябинск, школа № 124;
- **Ярцеву Виктору Александровичу**, Свердловская обл., Красноуфимский р-н, Тавринская средняя школа;
- **Ярцевой Ольге Владимировне**, г. Ярославль, школа № 33.

2014/15 учебный год

«Школа цифрового века» открыта!

Участие образовательной организации в проекте обеспечивает каждого педагога современными предметно-методическими материалами и возможностью подтвердить или повысить квалификацию без отрыва от работы в школе

Стоимость участия образовательной организации в проекте – 6 тысяч рублей за весь учебный год. Стоимость участия не зависит от количества педагогических работников в образовательной организации

Каждый педагог получает доступ ко всем материалам проекта по принципу «все включено» (без дополнительной платы)

МАТЕРИАЛЫ ПРОЕКТА

- **23 предметно-методических журнала** по всем предметам и направлениям школьной жизни плюс журнал для родителей
- **Модульные дистанционные курсы*** из циклов «Навыки профессиональной и личной эффективности педагога» и «Инклюзивный подход в образовании»
- **Дистанционные 36-часовые курсы**** повышения квалификации с выдачей удостоверения установленного образца
- **Методические брошюры** по всем школьным предметам

digital.1september.ru

Продление участия в проекте
и прием новых заявок продолжается

СТРАТЕГИЧЕСКИЕ ПАРТНЕРЫ ПРОЕКТА

1september.ru
издательский дом «Первое сентября»


ПРОСВЕЩЕНИЕ
ИЗДАТЕЛЬСТВО

* в течение указанного срока предоставляются без ограничения количества

** предоставляются по одному курсу для одного педагогического работника в течение одного учебного года (выбор конкретного курса – на усмотрение педагога)