

ИНФОРМАТИК А

4

Массовое обслуживание
Не значит "плохое"

20

Облака собираются
к хорошей работе

48

За полярным кругом
и прямые – кривые





НА ОБЛОЖКЕ

► Биткоины — пожалуй, самая известная электронная виртуальная валюта. Ее еще называют криптовалютой. Идея подобных виртуальных денег была описана в 2008 году, а уже в 2009-м появилось приложение, которое позволяет всем желающим эти деньги “добывать”. Для кого-то из добытчиков биткоинов это — бизнес, кто-то руководствуется исключительно спортивным интересом, но, как бы то ни было, за один биткоин сейчас “дают” (как получить — отдельный вопрос) почти 1000 американских долларов. Читайте заметку “Деньги из воздуха” на с. 3.

В НОМЕРЕ

- 3** ПАРА СЛОВ
 - Деньги из воздуха
- 4** УГЛУБЛЕНКА
 - Имитационное моделирование: практикум
- 16** ЕГЭ
 - “Около ЕГЭ”, или Несколько слов об А11
- 20** ТЕХНОЛОГИИ
 - Российская школа и ИКТ: облачные вычисления
- 40** МЕТОДИКА
 - Схема Горнера
- 44** ВНЕКЛАСНАЯ РАБОТА
 - Вопросы по информатике для проведения школьных конкурсов “Что? Где? Когда?” и “Брейн-ринг”
- 48** ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ
 - “В мир информатики” № 194

В ЛИЧНОМ КАБИНЕТЕ

Облачные технологии от Издательского дома “Первое сентября”

Уважаемые подписчики бумажной версии журнала!

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

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

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

Для активации кода:

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

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



ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ

- | Исходные файлы к статье “Имитационное моделирование”.
- | Программа для построения графиков в полярной системе координат к статье “Полярные координаты”.
- | 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 23 138 экз.
Тел. редакции: (499) 249-48-96
E-mail: inf@1september.ru
<http://inf.1september.ru>

ИЗДАТЕЛЬСКИЙ ДОМ “ПЕРВОЕ СЕНТЯБРЯ”

Главный редактор:
Артем Соловейчик
(генеральный директор)

Коммерческая деятельность:
Константин Шмарковский
(финансовый директор)

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

Реклама, конференции и техническое обеспечение Издательского дома:
Павел Кузнецов

Производство:
Станислав Савельев

Административно-хозяйственное обеспечение:
Андрей Ушков

Педагогический университет:
Валерия Арсланьян (ректор)

ГАЗЕТА ИЗДАТЕЛЬСКОГО ДОМА
Первое сентября – Е.Бирюкова

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

Классное руководство и воспитание школьников – М.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 – Д.Тюттерин
Школьный психолог – М.Чибисова

УЧРЕДИТЕЛЬ:
ООО “ЧИСТЫЕ ПРУДЫ”

Зарегистрировано
ПИ № ФС77-44341
от 22.03.2011
в Министерстве РФ по делам печати
Подписано в печать:
по графику 13.11.2013,
фактически 13.11.2013
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



Деньги из воздуха

► В конце 2013 года, который все еще непривычно называют “прошлым”, весьма высокопоставленные банкиры высказывали свое мнение о возможности эмиссии электронных валют. Мнения были разными — от “пока и говорить не о чем” до “почему бы и не попробовать прямо сейчас”.

Отдельный интерес представляет то, что упомянутые высказывания сильно возбудили тех, к кому они не относились вовсе (причем вне зависимости от того, были банкиры “за” или “против”). Так, например, сильно воодушевились фанаты биткоинов. Интересно, как они себе представляют банковскую эмиссию своей валюты? Видимо, только одним способом — банк закупает крутейшие компьютеры и генерит биткоины пачками.

Давайте кратко разберемся, о чем вообще речь.

Традиционные электронные деньги представляют собой денежные обязательства эмитента в электронном виде — на электронном носителе. По сути, электронные деньги проделывают тот же путь, который когда-то прошли деньги обычные — бумажные и “железные”. Ведь было время, когда каждая “бумажка” рассматривалась как обязательство, которое обеспечено “реальной ценностью” — драгоценным металлом. Просто слитки носить с собой было не так удобно. Сейчас история повторяется. Теперь уже электронные деньги обеспечены “реальной ценностью”, которую представляют собой деньги бумажные. Технология работы с такими электронными деньгами (на языке специалистов — фиатными) проста. И это не пустые слова — практически каждый из нас имел

дело с электронными деньгами описанного вида и не сильно задумывался о том, как все это работает. Действительно — раз нет проблем и все прозрачно, о чем задумываться?

Любители задумываться добывают биткоины. Тут все сложнее. Но множество людей готовы и задумываться, и платить немалые и очень реальные деньги за возможность извлекать деньги из воздуха. То есть буквально из воздуха. Ну... или, скорее, из электричества, так как “жрут” мощные компьютеры, которые используются для генерации биткоинов, очень прилично.

В двух словах работает это так. Для генерации денежных единиц — биткоинов — используется свободно распространяемое ПО. Взять его может любой. И запустить у себя на компьютере. ПО будет генерить для вас деньги совершенно бесплатно. Проблема заключается в том, что для генерации пачки биткоинов (а эмитируются они именно пачками, на данный момент — по 25 биткоинов) требуется решить весьма сложную криптографическую задачу. Программа, конечно, будет решать ее сама — участие человека не требуется. Зато требуется время, много времени. И много компьютерных мощностей. Поэтому добытчики биткоинов (процесс называется “майнинг”) объединяют свои компьютеры для совместной работы. Свалившееся счастье в виде пачки биткоинов при этом делится между участниками консорциума.

Биткоины можно обменять на вполне реальные деньги, причем курс биткоина пока стабильно и очень круто растет.

Таким образом, эмитирует биткоины фактически компьютерный алгоритм. Эмиссия, кстати, ограничена, что и обеспечивает (по задумке создателей) твердость валюты.

В общем, жутко увлекательно!

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



Имитационное моделирование: практикум

Введение

К.Ю. Поляков,
д. т. н., Санкт-Петербург

► Слова “имитационное моделирование” воспринимаются с благоговейным трепетом всеми, кто не очень хорошо понимает, что это такое. Между тем это одно из важных направлений в современной науке, чрезвычайно полезное прежде всего при решении *практических* задач управления сложными системами.

Задачи, которые приходится решать в жизни, обладают, как правило, одной важной особенностью: они не решаются (или очень трудно решаются) с помощью существующих теоретических методов. Достаточно вспомнить, что не существует (и в ближайшее время не предвидится) общих способов решения нелинейных дифференциальных уравнений. Это значит, что реальные системы очень часто настолько сложны, что мы не можем теоретически предсказать их поведение в интересующих нас условиях и тем более не можем сразу установить, как следует изменить систему для

того, чтобы она стала выполнять нужные нам функции.

В такой ситуации появляется два возможных направления движения:

1) упростить математическую модель до такой степени, чтобы ее можно было исследовать теоретически (например, “убрать” все нелинейности в модели системы, линеаризовать ее);

2) применить методы имитационного моделирования [1].

Первый подход (упрощение модели) очень важен, и им не стоит пренебрегать, поскольку только он позволяет получать хотя бы приближенные теоретические результаты и сделать качественные выводы. В то же время эти выводы (полученные на упрощенной, то есть на *другой* модели) обязательно нуждаются в проверке либо на реальном объекте, либо путем компьютерного моделирования с использованием полной модели.

Во многих случаях мы можем достаточно подробно описать поведение

Автор благодарит
к. ф.-м. н.
Е.А. Еремина
за обсуждение
статьи и полезные
замечания.

сложной системы, то есть алгоритм ее работы при известных входных данных. Это позволяет *сымитировать* работу системы в интересующих нас условиях и посмотреть, что же с ней будет происходить. Такой подход и получил название *имитационного моделирования*. Особенно широко он стал применяться с внедрением компьютерных технологий, которые позволили выполнять огромный объем вычислений за приемлемое время. Нередко цель имитационного моделирования — определить оптимальные значения параметров объекта путем многократных экспериментов с моделью, методом проб и ошибок.

Имитационное моделирование имеет свои сильные стороны:

- применимо даже в тех случаях, когда не для всех элементов системы найдено строгое математическое описание;
- при наличии достаточных вычислительных ресурсов может быть выполнено практически при любой степени детализации модели.

В то же время нужно понимать, что имитационное моделирование

- возникло не “от хорошей жизни”, оно появилось потому, что иначе (проще и быстрее) решить задачу не удается;
- дает только количественные результаты для тех случаев, которые фактически моделировались; при этом ничего нельзя сказать о поведении системы при других условиях.

Примером может служить отладка программы для встроенного контроллера какого-нибудь устройства. Очень часто специальных средств отладки для нужного контроллера не существует. Однако, зная систему команд этого контроллера, несложно построить его компьютерную имитационную модель и отлаживать программу на компьютере общего назначения. Такой прием, называемый *кросс-программированием*, тоже можно считать вариантом имитационного моделирования.

Как правило, реальные сложные системы подвержены влиянию случайных факторов, поэтому используемые в этих случаях имитационные модели относятся к классу стохастических, или *вероятностных*, моделей. Одна из таких моделей — модель работы банка — рассматривается в учебнике информатики углубленного уровня [2–3], который вышел в 2013 году в издательстве “Бином”. По мнению авторов этого учебника, очень полезно начинать изучение темы “Имитационное моделирование” не с использования готовых программных пакетов, а с “ручного” программирования, которое значительно лучше позволяет понять “что же там происходит”.

В настоящей статье вероятностная модель из учебника [3] исследуется подробно с несколькими стадиями усложнения. Изложенный в статье материал далеко выходит за рамки учебника и

может быть использован для стимулирования исследовательской работы заинтересованных школьников, например, в рамках элективных курсов.

Все работы можно выполнять в любой среде программирования на языке Паскаль, например, в АЛГО [4], *PascalABC.NET* [5], *FreePascal* [6] или *Delphi*. Все полные программы, рассмотренные в статье, можно найти в личном кабинете в электронных приложениях к этому номеру.

Модель работы банка

Предприятия сферы обслуживания (магазины, банки, узлы телефонной связи, станции “скорой помощи” и т.п.) — очень интересные, но очень сложные объекты. На вход таких систем поступает поток заявок на обслуживание — это могут быть, например, покупатели, входящие в магазин. Все заявки нужно обработать, для этого предприятие имеет несколько точек обслуживания (например, касс в магазине). При этом большую роль играет случайность:

- заявки поступают через случайные промежутки времени;
- время обслуживания — случайная величина.

Такие системы называют *системами массового обслуживания* [7]. Их теоретическое описание, как правило, достаточно сложно, поэтому часто применяют имитационные модели, которые позволяют найти решение задач с помощью многократных компьютерных экспериментов.

Рассмотрим простую модель работы банка. Клиенты входят через случайные промежутки времени, их обслуживают несколько кассиров, причем время обслуживания — также случайная величина. Будем считать, что все клиенты становятся в общую очередь, и при освобождении кассы начинается обслуживание того клиента, который стоит в этой очереди первым. Требуется определить, сколько кассиров нужно для того, чтобы клиент затратил на визит в банк не более M минут (с учетом обслуживания).

Сначала построим *детерминированную модель*, в которой случайность не учитывается. Будем считать, что за одну минуту входит ровно P клиентов, причем каждый клиент обслуживается ровно T минут.

Если количество касс равно K , то за T минут будет обслужено K клиентов. За это же время в банк войдут $P \cdot T$ новых клиентов. Несложно понять, что если $K < P \cdot T$, клиенты входят быстрее, чем их успевают обслуживать, поэтому очередь будет постоянно расти. Если же $K > P \cdot T$, очереди вообще не будет, из этого условия и нужно выбирать количество касс. Например, при $P = 2$ и $T = 2$ для обслуживания нужно не менее 10 касс.

Теперь усложним модель: введем в нее случайные события. Будем считать, что

- за одну минуту в банк входит случайное число клиентов, от 0 до P_{max} ;
- на обслуживание каждого клиента требуется от T_{min} до T_{max} минут.

Остается определить, как распределены случайные данные (количество входящих за одну минуту и время обслуживания) внутри заданных интервалов. Сначала для простоты мы будем считать, что в обоих случаях распределение *равномерное*. Это значит, что можно применить стандартные генераторы случайных чисел, которые дают именно равномерно распределенные значения.

Теперь построим имитационную модель и выполним моделирование ситуации для достаточно большого интервала времени L (например, для 8-часового рабочего дня $L = 8 \cdot 60 = 480$ минут).

Моделирование выполняется в дискретном времени с постоянным шагом, единица времени — одна минута. Число клиентов, находящихся в помещении банка, изменяется по закону

$$N_{i+1} = N_i + P_i - R_i$$

где N_i — количество клиентов в банке в i -ю минуту; P_i — количество клиентов, вошедших за i -ю минуту, а R_i — количество клиентов, обслуженных за эту минуту. Если считать, что N клиентов равномерно распределяются по K кассам, получаем, что средняя длина очереди в одну кассу равна $q = \frac{N}{K}$, а среднее

$$\text{время ожидания равно } q \cdot T = \frac{N \cdot T}{K}.$$

Количество вошедших P_i — это случайное целое число в интервале от 0 до P_{max} . На языке Паскаль его можно получить так:

```
P := random(PMax + 1);
```

Определить число обслуженных R_i оказывается не так просто. Если кассир обслуживает клиента за T минут, то можно считать, что за одну минуту он сделает часть работы, равную $\frac{1}{T}$. Если предположить, что скорость работы кассиров одинакова, то K касс за одну минуту обслужат $\frac{K}{T}$ клиентов.

Чтобы учесть случайное время обслуживания, величину T будем определять заново для каждой минуты случайным образом. Это случайное вещественное число в интервале от T_{min} до T_{max} :

```
T := Tmin + random * (Tmax - Tmin);
```

Тогда среднее время ожидания (с учетом времени обслуживания)

$$\Delta t = q \cdot T = \frac{N}{K} \cdot T$$

будет меняться, поскольку N и T — случайные величины. Именно поэтому невозможно *гарантировать*, что клиент *точно* не будет ждать больше

положенного времени M . Вместо этого мы можем (с помощью имитационного моделирования) подсчитать, какую *долю времени* период ожидания Δt будет больше, чем допустимое значение M . Для определения этой доли нужно подсчитать количество “плохих” минут и разделить его на общее время моделирования L .

Если доля “плохого” времени получилась, например, 0,95 (или 95%), то клиент практически всегда вынужден будет ждать слишком долго, и количество касс нужно увеличивать. Если эта доля равна 0,05, то время ожидания будет больше допустимого только в 5% случаев, и такой результат можно считать приемлемым.

Таким образом, нужно составить программу, которая запрашивает количество касс, выполняет моделирование работы банка в течение рабочего дня и выводит долю “плохих” минут. Задаем начальные значения переменных:

```
{ максимальное число входящих за 1 мин. }
Pmax := 4;
{ минимальное время обслуживания }
Tmin := 1;
{ максимальное время обслуживания }
Tmax := 9;
{ период моделирования в минутах (8 часов) }
L := 480;
M := 15;      { допустимое время ожидания }
N := 0;       { сначала в банке никого нет }
Count := 0;   { счетчик "плохих" минут }
```

Затем нужно ввести с клавиатуры количество касс K :

```
write('Введите количество касс: ');
readln(K);
```

Основной цикл моделирует работу банка в течение L минут:

```
for i := 1 to L do begin
  P := random(PMax + 1);
  { равномерное распределение }
  T := Tmin + random * (Tmax - Tmin);
  { равномерное распределение }
  R := round(K / T);
  { число обслуженных }
  N := N + P - R;
  { число клиентов в банке }
  if N < 0 then N := 0;
  dT := N / K * T;
  { среднее затраченное время }
  if dT > M then Inc(count)
end;
```

На каждом шаге последовательно вычисляются

- случайное число входящих клиентов P ;
- случайное время обслуживания T ;
- число клиентов R , обслуженных за эту минуту;
- число клиентов в помещении банка N ; если оно получилось отрицательным, то используется нулевое значение;

• время ожидания; если оно больше допустимого времени M , увеличивается счетчик “плохих” минут.

После окончания работы цикла остается вывести результат — отношение count/L , которое представляет собой долю “плохих” минут. Нужно провести серию экспериментов с моделью и выбрать минимальное значение K , при котором доля

“плохих” минут будет менее 5%. Помните, что в модели используются случайные числа, поэтому при каждом новом запуске программы результаты расчетов будут немного меняться. Для того чтобы результаты были более стабильными, значение L лучше увеличить, скажем, до 480 000 (если при этом время моделирования не становится недопустимо большим).

Выполнив моделирование в среде PascalABC.NET при различных значениях K для $L = 480\,000$, автор получил следующие результаты:

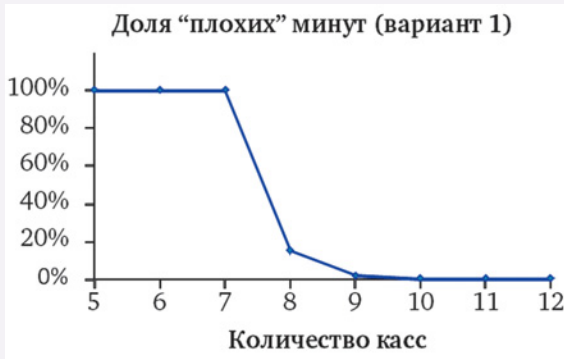


Рис. 1

— из которых следует, что для обеспечения допустимого времени ожидания достаточно 9 касс.

Обратите внимание, что мы выбрали исходные данные так, чтобы среднее число входящих клиентов и среднее время обслуживания совпадали с теми же данными для детерминированной модели. Поэтому интересно сравнить результаты, полученные с помощью моделей двух типов: детерминированной и вероятностной. Как вы помните, с помощью детерминированной модели мы получили минимальное значение $K = 10$, что близко к результатам проведенного вероятностного моделирования.

Распределение Пуассона

В рассмотренной вероятностной модели мы предполагали, что количество входящих за одну минуту распределено равномерно в диапазоне $[0, P_{\max}]$. На самом деле, согласно теории массового обслуживания [7], количество заявок, поступающих за минуту, описывается *распределением Пуассона*. Это распределение характеризует дискретную случайную величину, принимающую целые неотрицательные значения. Вероятность того, что эта случайная величина равна заданному числу k , вычисляется по формуле

$$p(k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda}. \quad (*)$$

Здесь λ — среднее значение случайной величины, и $k! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot k$ обозначает факториал числа k . Гистограмма, иллюстрирующая распределение Пуассона для $\lambda = 1$, показана на рис. 2.

Как же получить на компьютере последовательность псевдослучайных чисел с таким распределением? Как правило, в современных системах программирования есть встроенный генератор

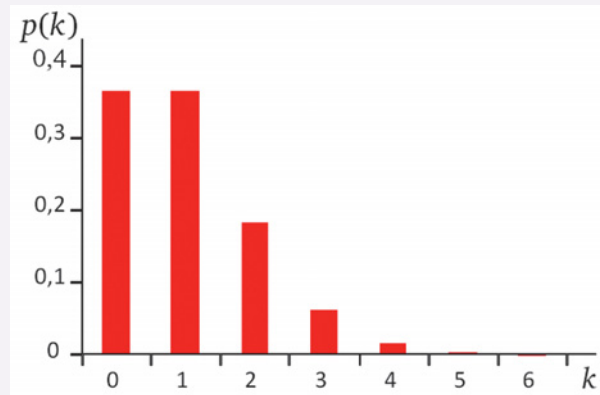


Рис. 2

псевдослучайных чисел с равномерным распределением. Оказывается, с помощью математических методов из такой последовательности чисел можно (теоретически) получить последовательность чисел с *любым* желаемым распределением. Для этого существует несколько различных методов, наиболее популярный из них — метод *обратной функции*. Ниже мы кратко изложим основные идеи этого метода и покажем, как применить его для получения случайных чисел, “распределенных по Пуассону”.

Одна из универсальных характеристик любого распределения случайных чисел — функция распределения $F(x)$, которая определяется для любого вещественного x как вероятность того, что случайная величина будет меньше или равна x . Функция $F(x)$ непрерывна, она равна нулю при $x \rightarrow -\infty$ и стремится к единице при $x \rightarrow \infty$.

Вспомним, что распределение Пуассона описывает целые неотрицательные значения, то есть $F(x) = 0$ для $x < 0$. Вероятность того, что $x = 0$ — ненулевая, она определяется по формуле, которая следует из (*):

$$p(0) = \frac{\lambda^0}{0!} \cdot e^{-\lambda} = e^{-\lambda} \quad (**)$$

Это означает, что при $x = 0$ функция распределения $F(x)$ изменяется скачком с нуля до значения $e^{-\lambda}$. Аналогичные скачки будут происходить при всех целых значениях x , тогда как при всех нецелых x функция $F(x)$ сохраняет постоянное значение. Таким образом, для любого дискретного распределения функция распределения получается ступенчатой. На рис. 3 показана функция распределения Пуассона для среднего значения $\lambda = 1$.

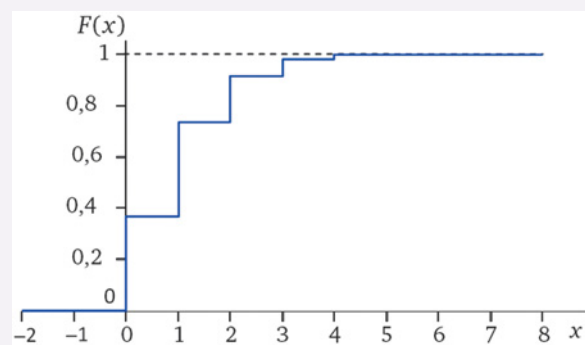


Рис. 3

Обратите внимание, что $F(x) \neq 0$ при $x = 0$, как и следует из формулы (**).

Вспомним, что мы хотим получить последовательность псевдослучайных чисел, соответствующую этой функции распределения. Предположим, что мы имеем генератор псевдослучайных вещественных чисел с равномерным распределением на интервале (0;1). Получив с помощью этого датчика очередное (псевдо-)случайное число z , нужно отложить это значение на вертикальной оси, провести горизонтальный отрезок до пересечения с графиком функции $F(x)$ и затем опустить вертикально вниз перпендикуляр к оси абсцисс. Значение x , полученное таким образом, и есть нужное нам случайное число. Очевидно, что в нашем случае все получаемые таким образом значения $k = x$ будут целыми неотрицательными числами. На рис. 4 графически показана эта процедура для $z = 0,5$.

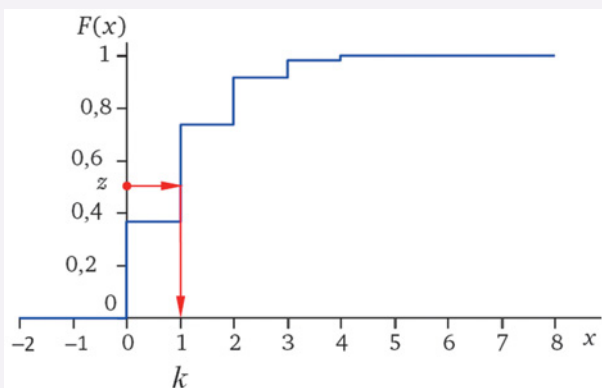


Рис. 4

Как же доказать, что новая последовательность будет иметь распределение Пуассона? Для дискретной случайной величины это сделать очень просто. По графику на рис. 4 видно, что вероятность получить $k = 0$ равна (с учетом равномерного распределения величины z) величине скачка функции $F(x)$ при $x = 0$, то есть $p(0)$. Аналогично вероятность получить $k = 1$ равна величине скачка $F(x)$ при $x = 1$, то есть $p(1)$, и т.д.

Теперь остается запрограммировать процедуру для перехода от случайной величины z к случайному целому числу k . Начинаем накапливать значение $F(x)$, начиная с $x = 0$. Если выполняется условие¹

$$z \leq F(0 + \varepsilon) = p(0),$$

то сразу получаем $k = 0$. Если это условие не выполняется, проверяем второе условие

$$z \leq F(1 + \varepsilon) = p(0) + p(1)$$

при выполнении которого $k = 1$, и т.д. Процедура на языке Паскаль, использующая такой подход и формулу (*) для вычисления $p(k)$, приводится ниже. Здесь параметр функции **Lam** — это среднее количество клиентов, входящих за одну минуту:

¹ Здесь $F(0 + \varepsilon)$ означает предел значения функции $F(x)$ при стремлении x к нулю справа.

```

{-----}
POISSON
Генератор псевдослучайных целых чисел
с распределением Пуассона.
Вход: Lam — среднее число заявок за единицу времени
Выход: случайное число заявок
{-----}
function Poisson(Lam: integer): integer;
var s, r, z: real;
    k: integer;
begin
    r := exp(-Lam); s := r; k := 0;
    z := random;
    while s < z do begin
        k := k + 1;
        r := r * Lam / k;
        s := s + r
    end;
    Poisson := k
end;

```

Выполним моделирование, немного изменив основную программу: теперь в основном цикле количество входящих клиентов за одну минуту определяется с помощью функции **Poisson**, параметром которой служит среднее значение, то есть $P_{max}/2$:

```

for i := 1 to L do begin
    P := Poisson(PMax div 2);
    ...
end;

```

Результат моделирования может показаться неожиданным: мы получим практически ту же самую кривую, что и на рис. 1. Это означает, что допущение о равномерности распределения количества входящих в данной задаче может быть использовано, поскольку не искажает результат в сравнении с более точной моделью, использующей распределение Пуассона. Тем не менее нужно понимать, что при других исходных данных этот вывод может оказаться неверным, на то это и имитационная модель. Поэтому лучше всегда использовать наиболее точную из имеющихся моделей.

Новая модель обслуживания

Еще одно слабое звено нашей модели — определение числа клиентов, обслуженных за текущую минуту. Ранее предполагалось, что производительность всех касс меняется одинаково каждую минуту, при этом случайное время обслуживания равномерно распределено в интервале от T_{min} до T_{max} . Скорее всего эта модель далека от реальности.

Используем другую модель обслуживания: все клиенты, находящиеся в банке, делятся на две группы: тех, кто в данный момент обслуживается, и тех, кто ждет своей очереди. Будем хранить общую длину очереди в переменной **NWait**. Для каждой кассы запоем время, оставшееся до окончания обслуживания текущего клиента, в массиве

```

var ServiceTime: array[1..100] of integer;

```

Предполагается, что время обслуживания — это целое число минут, и количество касс K не превышает 100.

Когда P клиентов в течение очередной минуты входят в банк, они сразу становятся в очередь, то есть значение переменной $NWait$ увеличивается на P :

```
NWait := NWait + P;
```

Каждую минуту все кассы выполняют очередную часть обслуживания своих клиентов, в результате некоторые кассы могут освободиться. В программе эту работу будет выполнять процедура `DoService`:

```
-----
DO SERVICE
Очередной цикл обслуживания за единицу
времени. Некоторые кассы могут освободиться.
-----}
```

```
procedure DoService;
var i: integer;
begin
  for i := 1 to K do
    if ServiceTime[i] > 0 then
      Dec(ServiceTime[i])
end;
```

В этой процедуре, которая вызывается на каждом шаге моделирования, оставшееся время для всех K касс уменьшается на единицу.

После того как кассы выполнили очередной шаг обслуживания, нужно попытаться распределить клиентов, стоящих в очереди, по свободным кассам. Для этой цели используем процедуру `StartService`:

```
-----
START SERVICE
Распределение очереди по свободным кассам.
-----}
```

```
procedure StartService;
var i: integer;
begin
  if NWait > 0 then
    for i := 1 to K do
      if ServiceTime[i] = 0 then begin
        ServiceTime[i] :=
          Tmin + random(Tmax - Tmin + 1);
        Dec(NWait);
        if NWait = 0 then break
      end
end;
```

Если очередь не пуста, мы просматриваем все кассы и пытаемся найти освободившуюся, у которой оставшееся время обслуживания равно нулю. Затем определяем случайное время обслуживания и записываем его в массив `ServiceTime`. В этом варианте процедуры предполагается, что время обслуживания — это случайная целая величина с равномерным распределением на отрезке $[T_{min}; T_{max}]$.

Теперь надо определить среднее время ожидания для только что вошедших клиентов. С учетом времени обслуживания, оно складывается из двух составляющих:

- времени обслуживания тех, кто уже находится у касс (массив `ServiceTime`);
- ожидаемого времени обслуживания для всех клиентов, стоящих в очереди, которое можно оценить как $NWait * Tcp$, где Tcp — среднее время обслуживания.

Полученное общее время нужно разделить на количество касс K .

С учетом всех приведенных выше рассуждений, основной цикл программы будет выглядеть так:

```
Count := 0;
NWait := 0;
Tcp := (Tmax + Tmin) div 2;
for i := 1 to L do begin
  P := Poisson(Pmax div 2);
  { распределение Пуассона }
  NWait := NWait + P;
  DoService;
  { работа касс }
  StartService;
  { распределение очереди }
  sumT := 0;
  for j := 1 to K do
  { оставшееся время обслуживания работающих касс }
    sumT := sumT + ServiceTime[j];
  dT := (NWait * Tcp + sumT) / K;
  if dT > M then Inc(count)
end;
```

Напомним, что результатом работы программы при заданном количестве касс K будет доля “плохих” минут — отношение $count/L$, которое для нормальной работы должно быть не более 5%, или 0,05.

Моделирование с помощью нового варианта программы (обозначим его как “вариант 3”) приводит к следующим результатам:

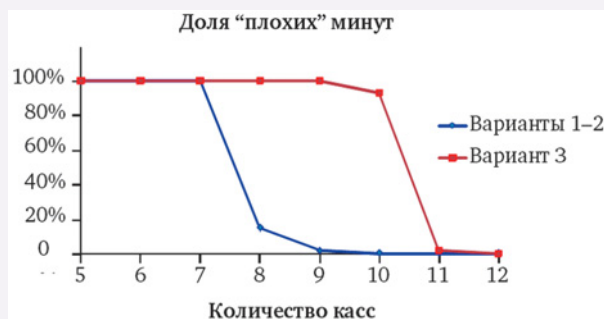


Рис. 5

Видим, что ситуация несколько изменилась: по вновь полученному графику (красная линия на рис. 5) можно сделать вывод, что количество касс должно быть не менее 11.

Экспоненциальное распределение

Несмотря на то что последняя модель обслуживания уже намного более реалистичная, чем первая, в ней осталась сильное допущение о том, что случайное время обслуживания имеет равномерное распределение. В теории массового обслуживания [7] обычно предполагают, что распределение этой величины экспоненциальное² (или показательное). Экспоненциальное распределение описывается плотностью вероятности:

² Нужно отметить, что это предположение связано в значительной степени с тем, что именно для этого случая удается получить аналитические решения многих задач (см. [1,7]).

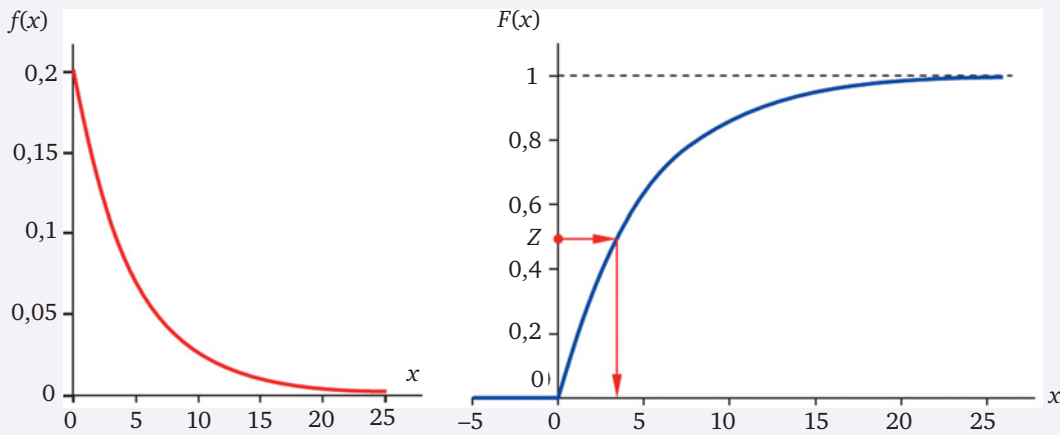


Рис. 6

$$f(x) = \begin{cases} \mu e^{-\mu x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

и соответствующей функцией распределения

$$F(x) = \begin{cases} 1 - e^{-\mu x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

В этих формулах параметр μ — это величина, обратная к среднему времени обслуживания T_{cp} , то есть $\mu = \frac{1}{T_{cp}}$. На рис. 6 показаны графики плотности

вероятности и функции распределения для $T_{cp} = 5$.

Для того чтобы получить последовательность случайных чисел с экспоненциальным распределением, можно использовать метод обратной функции, который мы уже обсуждали выше (см. рис. 3 на с. 7). Фактически нам нужно найти обратную функцию для $F(x)$. В данном случае это легко сделать аналитически из равенства:

$$z = 1 - e^{-\mu x},$$

откуда сразу следует, что

$$x = -\frac{1}{\mu} \ln(1 - z) = -T_{cp} \ln(1 - z).$$

Таким образом, получив случайную величину z с равномерным распределением на интервале $(0;1)$, мы можем найти соответствующее значение x , причём последовательность этих чисел будет иметь экспоненциальное распределение.

Учитывая, что последовательность значений $1 - z$ также имеет равномерное распределение на $(0;1)$, для преобразования обычно используют формулу $x = -T_{cp} \ln z$. Поскольку мы договорились, что время обслуживания составляет целое число минут, полученное таким образом значение нужно округлить вверх (до ближайшего большего целого числа). Функция, которая выдаёт очередное случайное время обслуживания, может выглядеть так:

```

-----
FULL SERVICE TIME
Случайное время обслуживания
(экспоненциальное распределение)
-----
    
```

```

function FullServiceTime(Tcp: integer):
integer;
var z: real;
begin
    z := random;
    FullServiceTime := Ceil(-ln(z) * Tcp)
end;
    
```

Параметр функции — это среднее время обслуживания. Функция `Ceil` выполняет округление вверх, в среде *FreePascal* для использования этой функции нужно подключить модуль *Math*.

Теперь изменим в процедуре `StartService` одну строку, где определяется случайное время обслуживания для очередного клиента:

```

...
ServiceTime[i] := FullServiceTime(Tcp);
...
    
```

Выполнив моделирование с учетом этих изменений, получаем новый график (он обозначен на рис. 7 как “вариант 4”):

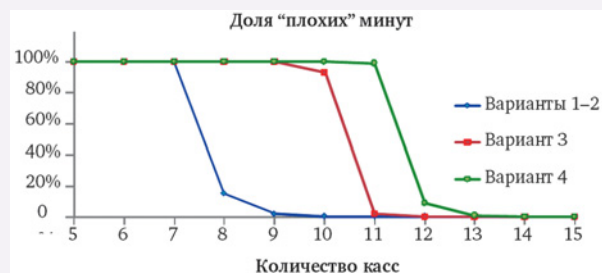


Рис. 7

По графику (см. зеленую линию на рис. 7) определяем, что количество касс должно быть не менее 13.

Распределение Эрланга

Как следует из рис. 6, используя экспоненциальное распределение для оценки времени обслуживания, мы считаем, что чаще всего клиенты обслуживаются очень быстро — плотность вероятности максимальна при $x = 0$. Как уже отмечалось [1,7], это допущение в теории массового обслуживания принимается главным образом из-за (относительной) простоты аналитического исследования. В реальной ситуации существует

некоторое “наиболее ожидаемое” время обслуживания, не равное нулю. Это справедливо, в частности, когда процедура обслуживания складывается из нескольких операций со случайным временем выполнения. Поэтому для уточнения модели часто используют распределение Эрланга, которое имеет плотность вероятности

$$f(x) = \begin{cases} \frac{\mu^k x^{k-1} e^{-\mu x}}{(k-1)!}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Параметр k можно рассматривать как количество операций при обслуживании клиента. Для того чтобы обеспечить заданное среднее время обслуживания T_{cp} , значение μ вычисляется по формуле

$$\mu = \frac{k}{T_{cp}}.$$

Легко доказать, что экспоненциальное распределение — это частный случай распределения Эрланга³ при $k = 1$.

На рис. 8 показана плотность вероятности для распределения Эрланга при $k = 2$ и среднем времени обслуживания $T_{cp} = 5$.

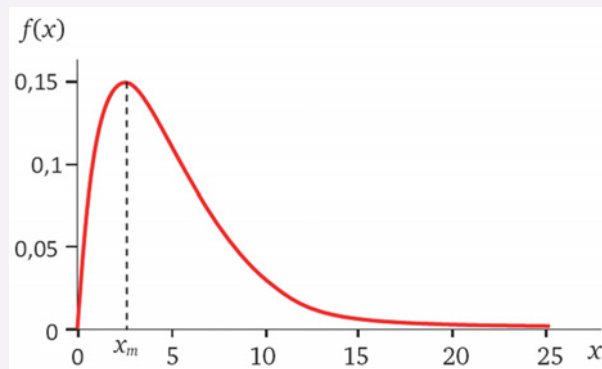


Рис. 8

Наиболее вероятное время обслуживания (точка максимума этой функции) вычисляется по формуле

$$x_m = \frac{k-1}{\mu} = \frac{k-1}{k} \cdot T_{cp}.$$

В данном случае получаем $x_m = \frac{2-1}{2} \cdot 5 = 2,5$, а при

увеличении k это значение стремится к T_{cp} .

Для того чтобы генерировать последовательность случайных чисел с распределением Эрланга, достаточно на каждом шаге вычислять среднее арифметическое k независимых значений, имеющих экспоненциальное распределение с тем же параметром μ . Таким образом, функция, определяющая случайное время обслуживания, приобретает следующий вид:

```
-----
FULL SERVICE TIME
Случайное время обслуживания
(распределение Эрланга с параметром k)
-----
```

³ Напомним, что $0! = 1$.

```
function FullServiceTime(Tcp, k: integer):
integer;
var z: real;
    i: integer;
begin
    z := 0;
    for i := 1 to k do
        z := z - ln(random);
    FullServiceTime := Ceil(z * Tcp / k)
end;
```

Остальные части программы не изменяются.

Выполнив моделирование с помощью нового варианта программы, обнаружим, что результаты в нашем случае мало изменились в сравнении с вариантом 4.

Дискретно-событийное моделирование

До этого момента мы выполняли моделирование с постоянным шагом по времени и предполагали, что время обслуживания составляет целое число минут. Однако в профессиональных средствах имитационного моделирования систем такой подход практически не применяется. Вместо этого чаще всего используют *дискретно-событийное моделирование* [1], при котором предполагается, что

- система меняет свое состояние только в отдельные моменты времени (дискретно);
- эти моменты времени совпадают с *событиями*, происходящими в системе.

Вернемся снова к нашей модели работы банка. С точки зрения рассмотренной задачи состояние системы определяется двумя значениями:

- количеством клиентов, стоящих в очереди (его мы храним в переменной **NWait**), и
- количеством занятых касс (или, что то же самое, количеством клиентов, которые в данный момент обслуживаются), — для хранения этого значения введем новую целую переменную **NBusy**.

В системе происходят два типа событий: вход очередного клиента и окончание обслуживания клиента на одной из касс, они могут произойти в любой момент времени. Модель меняет состояние только тогда, когда в системе произошло какое-то событие, при этом модельное время “продвигается” вперед до момента этого события.

Для работы с событиями в программе введем следующие типы данных:

```
type
    TEventType = (evIn, evOut);
    TEvent = record
        time: real;
        evType: TEventType;
    end;
```

Тип **TEventType** — это тип события: **evIn** означает вход нового клиента, а **evOut** — окончание обслуживания и выход клиента. Объект типа **TEvent** — это структура (запись), содержащая два поля: планируемое время события **time** и тип этого события **evType**.

Организуем очередь событий, в которой предстоящие события расставлены в порядке возрастания времени, так что ближайшее событие стоит в голове очереди. Соответствующая структура данных называется очередь с приоритетом [8], ее элементами будут записи типа `TEvent`, а приоритетом — время (поле `time`). Рамки статьи не позволяют углубляться в детали реализации такой очереди, нам важно только то, что она поддерживает операции “добавить событие” и “получить событие с наивысшим приоритетом” (с наименьшим значением поля `time`). В программе эти операции будут выполнять подпрограммы, имеющие такие заголовки⁴:

```
{ добавить событие }
procedure AddToQueue(E: TEvent);
{ получить событие из очереди }
function GetFromQueue: TEvent;
```

Основной цикл программы моделирования будет выглядеть так:

```
{ количество клиентов в очереди }
NWait := 0;
NBusy := 0; { количество занятых касс }
symTime := 0; { модельное время }
NewClient; { вход первого клиента }
while symTime < L do begin
  { выбрать событие из очереди }
  E := GetFromQueue;
  HandleEvent(E); { обработать событие }
end;
```

Здесь вещественная переменная `symTime` — это модельное время; процедура `NewClient` записывает в очередь событие прихода очередного (в данном случае — первого) клиента, а процедура `HandleEvent` “продвигает” модельное время и обрабатывает очередное событие, взятое из очереди. Цикл заканчивается, когда модельное время станет не меньше, чем заданное общее время моделирования `L`.

Теперь займемся реализацией новых процедур. Процедура `NewClient` создает новое событие типа `evIn` (вход нового клиента), определяет случайное время этого события и записывает событие в очередь:

```
{-----}
NEW CLIENT
Планирование времени входа очередного
клиента.
{-----}
procedure NewClient;
var E: TEvent;
begin
  E.evType := evIn;
  E.time := symTime + ArriveTime;
  AddToQueue(E);
end;
```

Время события (поле `E.time`) — это сумма текущего модельного времени и случайного интервала между входом двух клиентов. В теории массового обслуживания [1,7] обычно предпола-

гают, что этот интервал имеет экспоненциальное распределение, поэтому функцию `ArriveTime` можно записать так:

```
{-----}
ARRIVE TIME
Случайное время между входом клиентов.
{-----}
```

```
function ArriveTime: real;
begin
  ArriveTime := -ln(random) / Pcp;
end;
```

В глобальной переменной `Pcp` хранится среднее количество клиентов, входящих в банк за одну минуту.

Нам будет нужна еще одна процедура, которая выбирает из очереди очередного клиента и направляет его в освободившуюся кассу:

```
{-----}
START SERVICE
Начало обслуживания очередного клиента.
{-----}
```

```
procedure StartService;
var E: TEvent;
begin
  if (NWait > 0) and (NBusy < K) then begin
    NBusy := NBusy + 1;
    NWait := NWait - 1;
    E.evType := evOut;
    E.time := symTime + FullServiceTime(Tcp, 2);
    AddToQueue(E);
  end;
end;
```

Эта процедура срабатывает, когда число ожидающих в очереди `NWait` больше нуля и количество занятых касс `NBusy` меньше их общего количества `K` (есть свободные кассы). В этом случае число занятых касс увеличивается, число стоящих в очереди уменьшается, а в очередь записывается новое (планируемое) событие типа `evOut` (клиент освобожден). Время обслуживания вычисляется с помощью функции `FullServiceTime`, которая генерирует случайную последовательность, имеющую распределение Эрланга с заданными средним значением `Tcp` и параметром `k`:

```
{-----}
FULL SERVICE TIME
Случайное время обслуживания
(распределение Эрланга порядка k)
{-----}
```

```
function FullServiceTime(Tcp, k: integer): real;
var
  z: real;
  i: integer;
begin
  z := 0;
  for i := 1 to k do
    z := z - ln(random);
  FullServiceTime := z * Tcp / k;
end;
```

Новая версия этой функции в отличие от предыдущей возвращает вещественное число (без округления).

Теперь мы готовы написать процедуру `HandleEvent`:

⁴ Полный текст программы моделирования, где используется один из простейших вариантов реализации такой очереди, приведен на диске в приложении к этому номеру.

```

-----
HANDLE EVENT
Обработка очередного события
-----
procedure HandleEvent(E: TEvent);
begin
{ продвинуть модельное время }
symTime := E.time;
case E.evType of { выбор типа события }
  evIn:          { вход нового клиента }
    begin
      NewClient;
      NWait := NWait + 1;
    end;
  evOut:        { завершение обслуживания }
    NBusy := NBusy - 1;
end;
StartService;
end;

```

Эта процедура продвигает модельное время до времени переданного ее события. Затем, в зависимости от типа события, либо добавляем в очередь нового клиента, либо уменьшаем количество занятых касс. После этого вызывается процедура **StartService**, которая при наличии свободных касс пытается начать обслуживание клиентов из очереди.

Как же нам оценить работу этой модели? Наиболее простой вариант — использовать для оценки среднюю длину очереди в одну кассу \bar{q} , которая вычисляется по формуле:

$$\bar{q} = \frac{1}{K} \cdot \frac{1}{T} \int_0^T Q(t) dt,$$

где $Q(t)$ — текущая длина общей очереди в момент времени t , K — количество касс, а T — общее время моделирования. Поскольку состояние системы между моментами событий не меняется, функция $Q(t)$ — кусочно-постоянная:

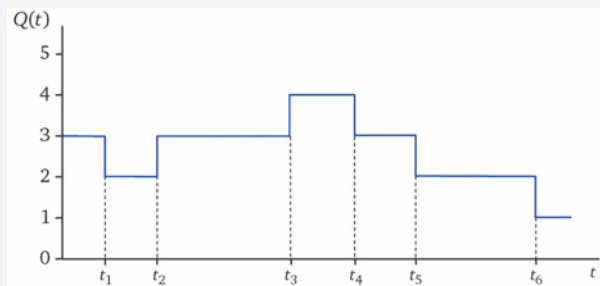


Рис. 9

На рис. 9 приведена диаграмма, показывающая изменение длины общей очереди $Q(t)$. В моменты времени t_2 и t_3 очередь увеличивается на единицу при входе клиентов. В моменты t_1 , t_4 , t_5 и t_6 заканчивается обслуживание очередного клиента, и длина очереди уменьшается. Как следует из графика, расчет \bar{q} может быть выполнен достаточно просто:

- перед началом основного цикла переменная Q_{cp} обнуляется;
- сразу после выборки очередного события из очереди добавляем очередную “порцию интеграла” (площадь прямоугольника) к значению этой переменной: $Q_{cp} := Q_{cp} + (E.Time - symTime) * NWait$;

Выражение в скобках представляет собой промежуток времени от предыдущего события до текущего (только что выбранного), а значение переменной $NWait$ равно длине очереди в этот период;

- после окончания цикла разделим значение переменной Q_{cp} на общее время моделирования, которое хранится в переменной $symTime$, и количество касс K :

$$Q_{cp} := Q_{cp} / symTime / K;$$

Это и будет средняя длина очереди в одну кассу.

Цикл в основной программе приобретает такой вид:

```

while symTime < L do begin
  E := GetFromQueue;
{ выбрать событие из очереди }
  Qcp := Qcp + (E.Time - symTime)*NWait;
  HandleEvent(E);
end;
Qcp := Qcp / symTime / K;

```

Полную программу для моделирования читатели могут найти на диске в приложении к этому номеру. Для исходных данных, использованных при моделировании в предыдущих вариантах программы, мы получаем, что при $K < 11$ длина очереди недопустимо велика (она может превышать 2000 человек, что говорит о явной перегрузке), а при $K = 11$ банк успевает обслуживать всех входящих клиентов (средняя длина очереди меньше 1). Таким образом, согласно этой модели, нужно 11 касс.

Дополнительные критерии

Кроме средней длины очереди, для оценки качества работы системы массового обслуживания часто используют еще две характеристики: среднее время ожидания \bar{W} (до начала обслуживания) и средний коэффициент использования (или эффективность) касс \bar{b} . Добавим в программу вычисление этих величин.

Время ожидания для клиента — это разница во времени между моментом начала обслуживания и моментом входа в банк. Для вычисления этой величины нужно где-то хранить время входа для каждого клиента, стоящего в очереди. Выделим для этой цели массив **timeIn**, предполагая, что длина очереди не будет больше, чем **MAXWAIT**:

```

const MAXWAIT = 10000;
var timeIn: array[1..MAXWAIT] of real;

```

Как вы помните, в любой момент количество клиентов в очереди хранится в переменной **NWait**, при этом первые **NWait** элементов массива **timeIn** будут содержать время входа для всех клиентов, ожидающих обслуживания.

Если прибыл очередной клиент, он ставится в очередь. В процедуре **HandleEvent** обработка события **evIn** теперь запишется так:

```

NewClient;
NWait := NWait + 1;
timeIn[NWait] := symTime;

```

В последней строке в первый свободный элемент массива **timeIn** записывается время прибытия нового клиента, которое совпадает с текущим модель-

ным временем. Конечно, сюда нужно добавить проверку на переполнение очереди, но мы оставим это читателю в качестве упражнения.

В переменной **Wcp** будем накапливать суммарное время ожидания для всех клиентов, которое после окончания моделирования разделим на число обслуженных клиентов, — это и будет среднее время ожидания.

Когда очередной клиент из очереди направляется в освободившуюся кассу, мы добавляем к переменной **Wcp** разницу между текущим временем и временем его прибытия (время ожидания), увеличиваем счетчик обслуженных клиентов **count** и сдвигаем все элементы массива **timeIn** на одну позицию к началу массива⁵:

```
Wcp := Wcp + symTime - timeIn[1];
```

```
Count := count + 1;
```

```
for i := 1 to NWait - 1 do
```

```
  timeIn[i] := timeIn[i + 1];
```

В конце программы вычисляется среднее время ожидания:

```
Wcp := Wcp / count;
```

Остается найти коэффициент использования касс \bar{b} . Обозначим через $B(t)$ количество занятых касс в момент времени t . Тогда \bar{b} вычисляется как среднее значение аналогично \bar{q} :

$$\bar{b} = \frac{1}{K} \cdot \frac{1}{T} \int_0^T B(t) dt.$$

Поскольку состояние системы между моментами событий не меняется, функция $B(t)$ — кусочно-постоянная, и расчет \bar{b} может быть выполнен следующим образом:

- перед началом основного цикла переменная **Wcp** обнуляется;
- сразу после выборки очередного события из очереди добавляем “порцию интеграла” (площадь прямоугольника) к значению этой переменной:

```
Wcp := Wcp + (E.time - symTime) * NBusy;
```

- после окончания цикла разделим значение переменной **Wcp** на общее время моделирования, которое хранится в переменной **symTime**, и количество касс **K**:

```
Wcp := Wcp / symTime / K;
```

Это и будет средняя эффективность касс.

В таблице приводятся результаты моделирования при различных значениях K (для $L = 100\,000$):

Количество касс, K	9	10	11	12
Средняя длина очереди, \bar{q}	107	7	0,5	0,15
Среднее время ожидания, \bar{W}	492	37	2,5	0,88
Средняя эффективность касс, \bar{b}	1,00	0,99	0,91	0,84

Эти данные показывают, что для обеспечения допустимого времени ожидания клиентов нужно установить 11 касс. При этом эффективность работы касс составляет 0,91, что вполне приемлемо.

⁵ Это самый простой, но неэффективный способ организации очереди. Лучше использовать массив, замкнутый в кольцо (см. [3]), чтобы не перемещать элементы.

Модель с отказами

Во всех предыдущих моделях мы предполагали, что клиенты банка — люди настойчивые и целеустремленные до такой степени, что они всегда встают в очередь и терпеливо ждут начала обслуживания, независимо от длины этой очереди. На практике это, конечно, не так. Действительно, увидев очередь в кассу из 10 человек, большая часть реальных клиентов развернется и уйдет, выразив про себя или вслух недовольство работой банка. Таким образом, часть заявок останутся необслуженными — мы получили так называемую *модель с отказами* (или модель с потерями).

Пусть в момент входа клиента средняя длина очереди в одной кассе равна q . Для моделирования ситуации отказа вычислим вероятность отказа $p(q)$, используя следующие идеи:

- если очереди нет, клиент никогда не отказывается от обслуживания, то есть $p(0) = 0$;
- при возрастании длины очереди вероятность отказа повышается и асимптотически стремится к 1 при $q \rightarrow \infty$;
- при $q = 2$ около 60% клиентов отказываются от стояния в очереди и уходят.

Этим условиям удовлетворяет, например, функция $p(q) = 1 - e^{-0,5q}$, график которой показан на рис. 10.

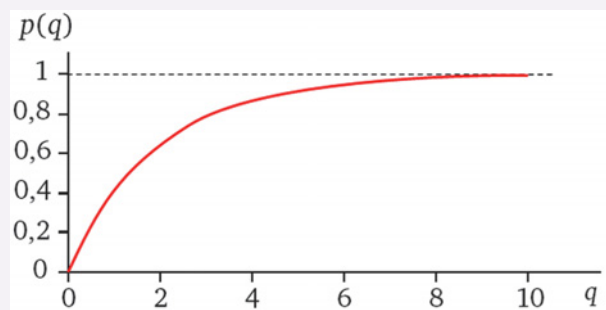


Рис. 10

Заметим, что средняя длина очереди q в одну кассу может быть вычислена как отношение общей длины очереди (переменная **NWait**) к количеству касс **K**. Перепишем соответствующим образом процедуру **HandleEvent**:

```
{-----
HANDLE EVENT
Обработка события. Модель с отказами.
-----}
procedure HandleEvent(E: TEvent);
begin
  symTime := E.time;
  case E.evType of
    evIn:
      begin
        NewClient;
        { отказ }
        if random <
          1-exp(-0.5 * NWait / K) then
          count0 := count0 + 1
        else begin
          NWait := NWait + 1;
          timeIn[NWait] := symTime;
        end;
      end;
  end;
```

```

evOut:
  NBusy := NBusy - 1;
end;
StartService;
end;

```

Здесь переменная `count0` — это счетчик отказов. Отказ происходит тогда, когда случайное число, полученное при вызове функции `random`, меньше, чем вероятность отказа $p(q) = 1 - e^{-0,5q}$.

Для того чтобы после окончания моделирования вычислить долю необслуженных заявок (долю отказов, долю потерянных клиентов), нужно разделить `count0` на общее количество вошедших клиентов, которое равно `count + count0` (напомним, что в переменной `count` хранится число обслуженных клиентов).

Выполнив моделирование после этих изменений, получаем следующие результаты (рис. 11).

В данной ситуации длина очереди \bar{q} и связанное с ней среднее время ожидания \bar{w} ничего не говорят об эффективности работы банка. На первый план выходят два других показателя — эффективность работы касс и доля отказов. Они противоречат друг другу в том смысле, что при увеличении количества касс доля отказов уменьшается (это хорошо!), но зато падает и эффективность работы касс, что на практике означает лишние расходы. Поэтому приходится искать компромиссное решение. Пусть, например, наиболее важна доля отказов, и она должна быть не более 10%. По последнему графику определяем, что нужно выбрать $K = 10$ (как в детерминированной модели!), при этом эффективность работы касс оказывается достаточно высокой — около 90%.

Построенную имитационную модель можно уточнять и дальше. Например, полезно предусмотреть возможность отказа оборудования касс, перерывы на обед (когда некоторые кассы закрываются) и т.д. Эти изменения читатели уже могут сделать самостоятельно.

Заключение

Подведем некоторые итоги. Для решения задачи мы использовали одну детерминированную модель (согласно которой требуется не менее 10 касс) и не-

сколько вероятностных, которые привели к различным ответам. Все результаты достаточно близки, то есть в данной задаче при выбранных параметрах можно было использовать даже простейшую детерминированную модель. Еще раз отметим, что при других исходных данных это может быть совсем не так.

В результате проведенных вычислительных экспериментов учащиеся должны понять, что

- имитационные модели используются тогда, когда задачу не удастся (невозможно или очень сложно) решить аналитически;
- любая имитационная модель дает только численное решение для тех входных данных, которые использовались в проведенных сеансах моделирования (“прогонах” модели); что будет при других исходных данных — неизвестно;
- результаты имитационного моделирования зависят от принятых допущений;
- проверить результаты имитационного моделирования и таким образом установить адекватность (или неадекватность) модели можно только с помощью эксперимента, проведенного на объекте-оригинале.

Лишь после этого имеет смысл переходить к использованию готовых пакетов для имитационного моделирования типа GPSS [9] или AnyLogic [10]. Такой подход позволяет избежать очень коварных ловушек нашего времени — “нажимания на кнопки” без понимания сути происходящих процессов и абсолютного доверия к тем числам, которые получаются в результате работы какой-либо программы.

Литература

1. Лоу А.М., Кельтон В.Д. Имитационное моделирование. СПб.: Питер, 2004.
2. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бинном, 2013.
3. Поляков К.Ю., Еремин Е.А. Информатика. 11-й класс. Углубленный уровень. В двух частях. М.: Бинном, 2013.
4. Программа АЛГО [Электронный ресурс] URL: <http://petriv.ho.com.ua/algo/rus>.
5. PascalABC.NET. Современное программирование на языке Паскаль [Электронный ресурс] URL: <http://pascalabc.net>.
6. FreePascal [Электронный ресурс] URL: <http://www.freepascal.org>.
7. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. М.: URSS: КомКнига, 2013.
8. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И.В. Красикова. М.: Вильямс, 2005.
9. GPSS: имитационное моделирование систем [Электронный ресурс] URL: <http://gpss.ru>.
10. AnyLogic: Многоподходное имитационное моделирование [Электронный ресурс] URL: <http://www.anylogic.ru>.

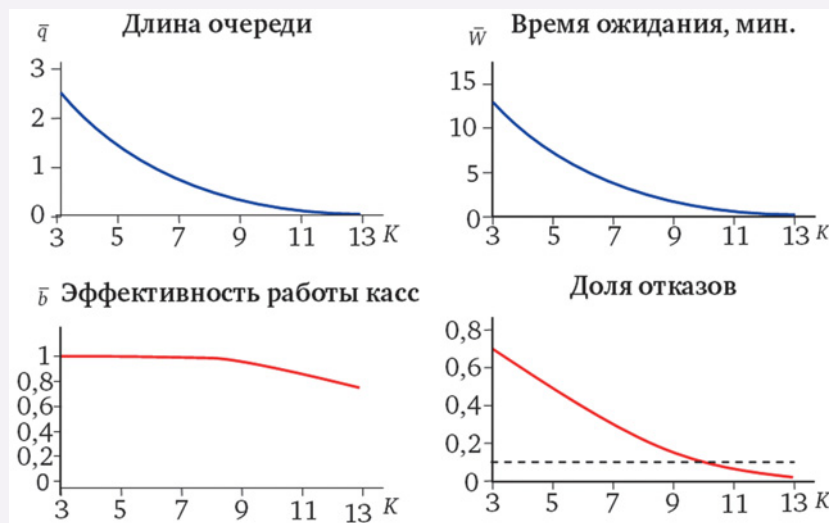
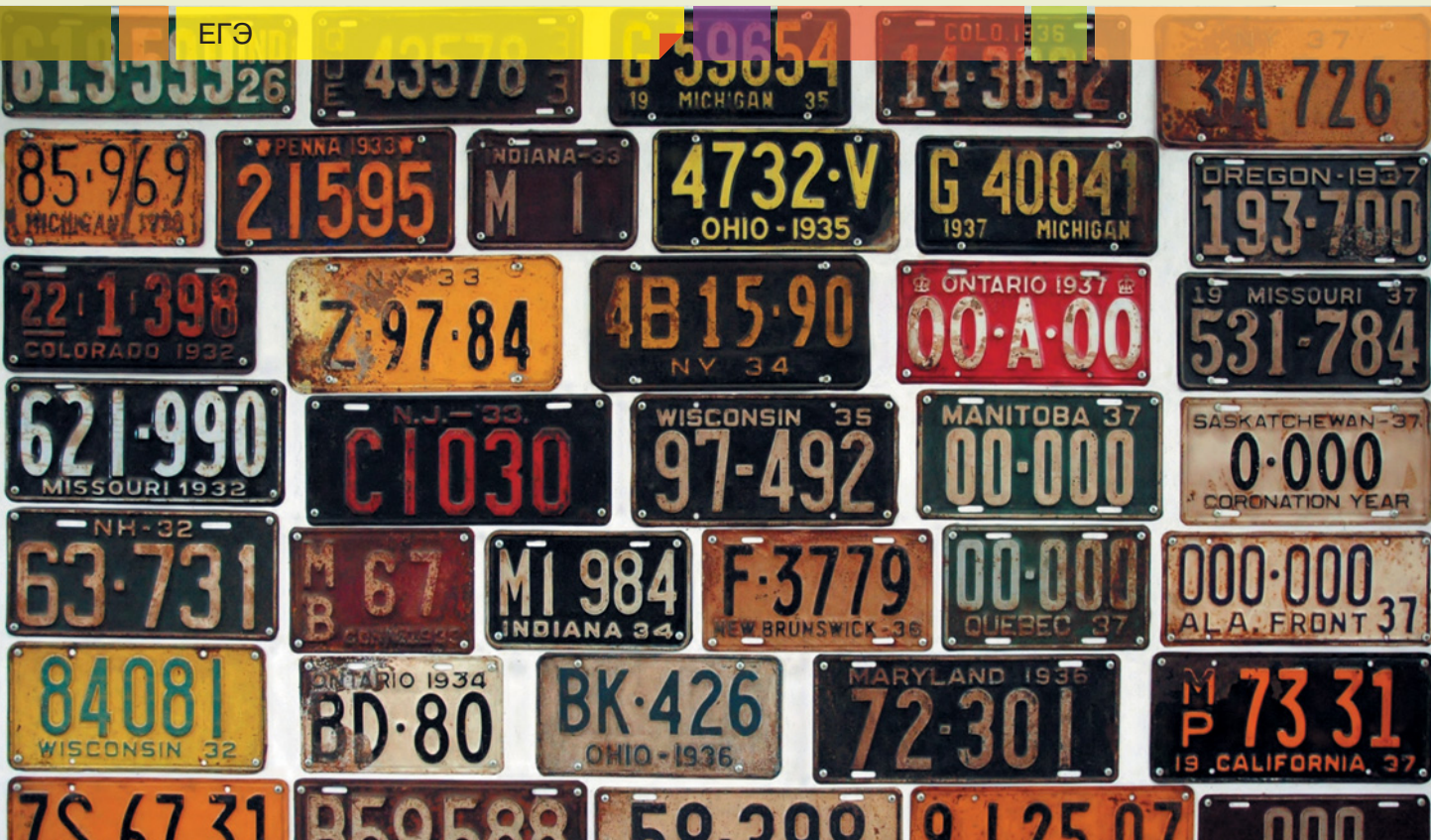


Рис. 11

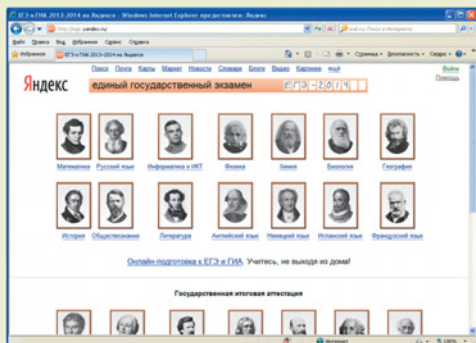


“Около ЕГЭ”, или Несколько слов об А11

О.Б. Богомолова,
д. п. н., преподаватель
Московского
государственного
гуманитарного
университета (МГГУ)
им. М.А. Шолохова,
Д.Ю. Усенков,
ст. н. с. Института
информатизации
образования Российской
академии образования,
Москва

► Сегодня в Интернете есть уже большое количество сайтов, посвященных подготовке к Единому государственному экзамену. Есть сайты — подборки демоверсий ЕГЭ по различным предметам, есть банки заданий по типу ЕГЭ, есть даже онлайн-тесты по заданиям, подобным ЕГЭ.

Одним из них является сайт “Яндекс. Единый государственный экзамен” (<http://ege.yandex.ru>) — один из разделов портала Яндекс, содержащий онлайн-тесты для подготовки к ЕГЭ и ГИА.



Чтобы начать тренировку, достаточно нажать на название предмета или на сопровождающий его портрет (заодно оценив шутку разработчиков: при наведении курсора мыши портреты “дорисовываются” в духе детских дорисовок синей ручкой в школьных учебниках ☺). А далее предлагается или выбрать один из вариантов теста, или в режиме тренировки — один из подразделов, чтобы прорешать задания по определенной теме, взятые из всех вариантов теста. Эти задания разработаны экспертами специально для портала Яндекс и в чем-то подобны задачам реального ЕГЭ и ГИА.

Среди заданий, посвященных умению подсчитывать информационный объем сообщения (группа А11), в вариантах Яндекс-ЕГЭ попала пара интересных заданий, несколько выделяющихся из ряда аналогичных задач, к которым школьники привыкли в предыдущих демовариантах ЕГЭ, сборниках задач ЕГЭ и тренировочных работах. Эти “нестандартные” задачи оказались для ребят достаточно трудными. Однако на сайте Яндекс-ЕГЭ пояснения к решению этих задач даны очень коротко. Да и на сайте К.Полякова (<http://kpolyakov.narod.ru/school/ege.htm>), где эти задачи повторены в разделе “Задачи для тренировки”, разбор их решений отсутствует.

В данной статье мы предлагаем свой вариант разбора решения указанных за-

дач. Надеемся, что этот материал окажется полезным для учителей и школьников, тем более что нет никаких гарантий, что такие задачи не появятся на реальном ЕГЭ следующим летом.

Начнем с “типовой” задачи такого рода, чтобы вспомнить, как они решаются.

Задача 1. Автомобильные номера имеют длину 8 символов и составлены из заглавных английских букв (всего используется 26 букв) и десятичных цифр в любом порядке. Каждый символ кодируется одинаковым и минимально возможным количеством битов, а каждый номер — одинаковым и минимально возможным целым количеством байтов. Требуется определить объем памяти, необходимый для хранения 100 таких номеров.

Решение

Решать такие задачи большинство школьников уже умеют.

Если в номере “на равных” можно использовать буквы и цифры, то общее количество допустимых символов в алфавите равно $26 + 10 = 36$ (26 букв и 10 цифр).

Тогда для хранения одного символа такого алфавита требуется 6 битов (5 битов слишком мало, так как они позволяют хранить не более $2^5 = 32$ символов, а 6 битов достаточно, так как они позволяют хранить до $2^6 = 64$ символов).

Если в номере используется 8 символов, то для их хранения потребуется $6 \times 8 = 48$ битов. Это составляет как раз ровно 6 байтов (если же количество битов не делится нацело на 8, то надо округлять результат деления в большую сторону).

Тогда для хранения 100 таких номеров потребуется $6 \times 100 = 600$ байтов.

*Ответ*¹: 600.

Как видим, задача действительно несложная. Но давайте ее немного модифицируем.

Задача 2. Автомобильные номера имеют длину 7 символов и составлены из заглавных русских букв (всего используется 30 букв) и десятичных цифр. При этом в номер входит 4 цифры и 3 буквы. Каждая буква и каждая цифра кодируется одинаковым и минимально возможным количеством битов, а каждый номер — одинаковым и минимально возможным целым количеством байтов. Требуется определить объем памяти, необходимый для хранения 200 таких номеров.

Решение

Усложнение здесь в том, что буквы и цифры в номере “неравноправны”. Нельзя “смешать” их все в одном алфавите и производить вычисления исходя из суммарного количества букв и цифр. Следовательно, надо вычислять количество битов отдельно для букв и отдельно для цифр.

Имеем в “буквенном” алфавите 30 букв — значит, для хранения одной буквы требуется 5 битов ($2^5 = 32$).

Имеем в “цифровом” алфавите 10 цифр — значит, для хранения одной цифры нужно 4 бита ($2^4 = 16$, а $2^3 = 8$ — слишком мало).

¹ Нужно не забывать, что в отличие от общепринятых правил записи ответа с указанием размерности на ЕГЭ ответ представляет собой только само число. На это нужно особо обращать внимание учащихся при их подготовке к ЕГЭ.

Тогда если в номер входит 4 цифры и 3 буквы, то на хранение этого номера требуется $4 \times 4 + 3 \times 5 = 31$ бит. Сколько потребуется для хранения такого номера целых байтов? Очевидно, что с учетом округления результата деления 31 на 8 в большую сторону это значение будет равно 4 байтам.

Остается только подсчитать объем информации, соответствующий 200 таким номерам: $200 \times 4 = 800$ байтов.

Ответ: 800.

Можно усложнить задачу и в другом “направлении”.

Задача 3. Автомобильные номера имеют длину 7 символов и составлены из заглавных русских букв (всего используется 30 букв) и десятичных цифр. При этом в номер входит 4 цифры и 3 буквы. Дополнительно к собственно номеру в учетную запись в базе данных ГИБДД входит информация о владельце автомобиля, занимающая 50 битов. Каждая буква и каждая цифра кодируется одинаковым и минимально возможным количеством битов, а каждая учетная запись — одинаковым и минимально возможным целым количеством байтов. Требуется определить объем памяти, необходимый для хранения 50 таких учетных записей.

Решение

Начало решения — такое же, как в предыдущем случае: получаем, что для хранения одной буквы требуется 5 битов, для каждой цифры — 4 бита, а для всего номера — 31 бит.

Но дальше надо вспомнить, что к номеру надо добавить еще 50 битов дополнительной информации. Тогда вся учетная запись будет иметь объем $31 + 50 = 81$ бит. Это будет соответствовать (с учетом округления в большую сторону) 11 байтам.

А теперь остается только подсчитать, сколько байтов требуется для хранения 50 таких записей: $50 \times 11 = 550$ байтов.

Ответ: 550.

А вот теперь — самое сложное и самое интересное ☺.

Задача 4 (Яндекс-ЕГЭ). При регистрации в компьютерной системе, используемой при проведении командной олимпиады, каждому ученику выдается уникальный идентификатор — целое число от 1 до 1000. Для хранения каждого идентификатора используется одинаковое и минимально возможное количество битов. Идентификатор команды состоит из последовательно записанных идентификаторов учеников и 8 дополнительных битов. Для записи каждого идентификатора команды система использует одинаковое и минимально возможное количество байтов. Во всех командах равное количество участников. Сколько участников в каждой команде, если для хранения идентификаторов 20 команд-участниц потребовалось 180 байтов?

Решение

Итак, идентификатор ученика — это число от 1 до 1000. Сколько битов потребуется для представления такого десятичного числа? Очевидно, надо найти, в какое количество разрядов можно уместить такое число в двоичном представлении. В 9 двоичных разрядах можно записать десятичные числа от 0 до 512, этого нам недостаточно. А вот 10 разрядов хватит для записи чисел от 0 до 1024, и

это нам подходит. Значит, для хранения идентификатора ученика требуется 10 битов.

Сколько учеников входит в каждую команду (а их количество, по условию, одинаково для всех команд), мы не знаем. Поэтому обозначим его как x .

Тогда идентификатор команды занимает $x \times 10 + 8$ битов.

Мы также знаем, что для хранения 20 идентификаторов таких команд требуется 180 байтов. Значит, для хранения одного идентификатора команды нужно $180/20 = 9$ байтов, или $9 \times 8 = 72$ бита.

А теперь — важное замечание: **эти 9 байтов не обязательно заполнены информацией целиком!** Вспомним, что, когда в предыдущих задачах мы пересчитывали количество битов в количество целых байтов, мы округляли результат деления количества битов на 8 в большую сторону, т.е. фактически добавляли некоторое число “пустых” битов до кратности 8.

Поэтому результат решения вполне очевидно уравнения: $x \times 10 + 8 = 72$ — не обязательно окажется целым. Но нам-то нужно целое число! Ведь x — это количество учеников, а “полтора землекопа”, как известно, бывает только в анекдотах.

А в какую сторону округлять — в меньшую или в большую? Если мы округлим дробное значение x в большую сторону, то тем самым мы попытаемся вписать в запись *больше* битов, чем получается для “точного” дробного значения x . Но столько информации в нашу запись точно не уместится. Поэтому в данном случае **округлять x надо в меньшую сторону** — пусть лучше сколько-то битов останутся в записи команды неиспользованными.

В нашем случае (уравнение $x \times 10 + 8 = 72$) $x = 6,4$, и, округляя его до целого в меньшую сторону, получаем, что $x = 6$. То есть максимально возможное количество участников в командах равно 6.

Ответ: 6.

Задача 5 (Яндекс-ЕГЭ). Автомобильный номер состоит из нескольких букв (количество букв одинаково во всех номерах), за которыми следуют три цифры. При этом используются 10 цифр и только 5 букв: Н, О, М, Е и Р. Нужно иметь не менее 100 тысяч различных номеров. Какое наименьшее количество букв должно быть в автомобильном номере?

Решение

Начнем с того, что, как мы знаем, в номере используются три цифры. Всего цифр в “алфавите” — 10, и мы разными способами выбираем из них три какие-то цифры, которые в том числе могут быть и одинаковыми. Сколько может быть таких “выборок”?

Кажется, что вычислить их количество будет сложно. Но давайте немного отвлечемся и посмотрим на... ряд натуральных чисел. А точнее — трехзначных натуральных чисел: от 000 до 999. Эти числа, если присмотреться повнимательнее, вполне удовлетворяют условиям нашей задачи:

- они все различны (среди них нет одинаковых чисел);
- каждое из них содержит три цифры, выбранные из возможных 10 цифр;
- они представляют собой все возможные комбинации десятичных цифр, выбранных по 3 из 10 возможных.

Следовательно, эти трехзначные десятичные числа от 000 до 999 и есть нужные нам варианты

номеров, если бы они состояли только из цифр. И таких вариантов номеров будет 1000 штук (оно равно значению максимального числа плюс 1, для учета числа 000).

А теперь **расширим найденную аналогию**. Пусть речь идет о выборке трех букв из некоторого алфавита, включающего 10 букв. Сами получаемые варианты таких трехбуквенных сочетаний, конечно, будут другими по сравнению с ранее полученными нами числами. Но если мы выполним простую замену букв цифрами, то получим, что такая “буквенная” задача полностью эквивалентна предыдущей, так как дает **то же самое количество вариантов**.

А если сделать еще один шаг, то нетрудно понять, что **количество вариантов выборки n символов из алфавита, включающего m символов, можно определить как количество всех возможных n -разрядных чисел в системе счисления с основанием m** .

Запомним эту важную эвристику, она нам еще пригодится!

Продолжим решение задачи. Итак, количество всех возможных вариантов выбора 3 цифр из 10 (или, что то же самое, количество всех трехзначных десятичных чисел) равно 1000. Но нам нужно 100 000 различных номеров! За счет чего можно их получить? Очевидно, дополнительные варианты нам даст выбор x букв из 5 возможных, где число x нам и нужно определить. При этом для каждого варианта выбора 3 цифр нам надо “сгенерировать” $100\,000/1000 = 100$ вариантов буквенных записей.

Вспомним нашу эвристику. В нашем “буквенном” алфавите 5 букв. Из них надо выбирать x букв. Следовательно, возможное количество вариантов буквенных сочетаний будет равно количеству всех возможных x -разрядных чисел в пятеричной системе счисления.

Чтобы упростить решение, будем составлять таблицу и определять количество таких чисел для разных значений x (не забывая, что имеем дело с десятичными числами, следовательно, получаемые диапазоны чисел надо переводить в десятичные).

x	Числа	Количество чисел (количество вариантов выбора x символов из 5)	Комментарий
1	0 ... 4	5	Мало (надо не менее 100)
2	00 ... 44	$44_5 + 1 = 25$	Мало
3	000 ... 444	$444_5 + 1 = 125$	Уже достаточно

Итак, если мы будем включать в номер, кроме 3 цифр, еще 3 буквы, выбираемые из 5 возможных, то мы получим количество возможных вариантов таких номеров, равное:

1000 вариантов за счет различного выбора цифр \times 125 вариантов за счет выбора 3 букв из возможных 5 = 125 000 разных вариантов номеров.

Это с запасом покрывает заданное требуемое количество различных номеров (100 000) и допустимо, так как по условию задачи надо обеспечить **не менее** 100 000 различных номеров.

Ответ: 3.



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ г. МОСКВЫ
ИЗДАТЕЛЬСКИЙ ДОМ «ПЕРВОЕ СЕНТЯБРЯ»
МОСКОВСКИЙ ПЕДАГОГИЧЕСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

2014

24 МАРТА – 18 АПРЕЛЯ

РАСПИСАНИЕ ДНЕЙ ПЕДАГОГИЧЕСКОГО МАРАФОНА

24 марта	День учителя технологии *	4 апреля	День учителя информатики
25 марта	Открытие Марафона День классного руководителя	5 апреля	День учителя физики
26 марта	День школьного психолога День учителя ОБЖ	6 апреля	День учителя математики
27 марта	День здоровья детей, коррекционной педагогики, логопеда, инклюзивного образования и лечебной физической культуры	8 апреля	День учителя истории и обществознания
28 марта	День учителя начальной школы (день первый)	9 апреля	День учителя МХК, музыки и ИЗО
29 марта	День учителя начальной школы (день второй)	10 апреля	День школьного и детского библиотекаря
30 марта	День дошкольного образования	11 апреля	День учителя литературы
1 апреля	День учителя географии	12 апреля	День учителя русского языка
2 апреля	День учителя химии	13 апреля	День учителя английского языка
3 апреля	День учителя биологии	15 апреля	День учителя французского языка
		16 апреля	День школьной администрации
		17 апреля	День учителя физической культуры
		18 апреля	День учителя немецкого языка Заккрытие

marathon.1september.ru



Обязательная предварительная регистрация на все дни Марафона 20 февраля 2014 года на сайте marathon.1september.ru



Каждый участник Марафона, посетивший три мероприятия одного дня, получает официальный именной сертификат (6 часов)

В дни Марафона ведущие издательства страны представляют книги для учителей
Начало работы каждого дня – 9.00. Завершение работы – 15.00

УЧАСТИЕ БЕСПЛАТНОЕ. ВХОД ПО БИЛЕТАМ

РЕГИСТРИРУЙТЕСЬ, РАСПЕЧАТЫВАЙТЕ СВОЙ БИЛЕТ И ПРИХОДИТЕ!

Место проведения Марафона: МПГУ, ул. Малая Пироговская, дом 1, стр. 1

(в 5 минутах ходьбы от станции метро «Фрунзенская»)

* Место проведения Дня учителя технологии: ЦО № 293, ул. Касаткина, 1а (ст. метро «ВДНХ»)

По всем вопросам обращайтесь, пожалуйста, по телефону **8-499-249-3138** или по электронной почте marathon@1september.ru



Российская школа и ИКТ: облачные вычисления

А.Ю. Уваров,
Вычислительный центр
Российской академии наук

► Сегодня педагоги и школьники уже не понаслышке знают о цифровой революции и формировании цифровой техносферы, которая открывает новые направления развития школы и обновления образовательного процесса. Для многих повседневной реальностью становятся мобильные вычислительные устройства (смартфоны, планшеты, электронные книги, ультрабуки, трансформеры и т.п.), постоянный широкополосный доступ в Интернет. Однако носимые устройства — лишь верхушка айсберга. В его основании лежит информационная инфраструктура или киберпространство.

Авторы книги «Российская школа и ИКТ: взгляд в следующее десятилетие» [2] прогнозировали, что распространение широкополосного доступа в Интернет и облачных вычислений

приведет к появлению в школе облачных сервисов уже в этом десятилетии. «Образовательные материалы и сервисы станут доступны каждому участнику учебного процесса там, где они требуются, и тогда, когда они нужны. Это поможет обеспечить каждому равный доступ к качественному образованию» [2]. Прогноз, сделанный четыре года назад, начинает сбываться.

В США президент Обама летом 2013 года выступил с инициативой ConnectedED¹. Ее цель — в течение пяти лет обеспечить всех американских школьников высокоскоростным широкополосным и беспроводным доступом к Интернету. К 2015/2016 учебному году каждая школа должна получить доступ в Интернет со скоростью не менее 100 Мб/сек., а к концу десятилетия — со скоростью не менее 1 Гб/сек. [16].

В нашей стране тоже быстро растет количество школьников и педагогов, получивших широкополосный доступ к Интернету. Министерство образования и науки РФ обсуждает новую Концепцию

¹ <http://www.whitehouse.gov/the-press-office/2013/06/06/president-obama-unveils-connected-initiative-bring-america-s-students-di>.

развития единой информационной образовательной среды в Российской Федерации², одним из ключевых направлений которой является развитие Интернета.

Облачные сервисы стали реальностью, быстро растет количество провайдеров и потребителей облачных услуг. Интернет-провайдеры и разработчики программного обеспечения широко рекламируют свои “облачные услуги для образования”, а словосочетание “облачные сервисы” стало маркетинговым инструментом. Сегодня практически каждый интернет-провайдер уверенно заявляет, что все сервисы, которые он предлагает своим потребителям, — это “облачные сервисы”. В результате у работников школы складывается ошибочное представление, будто бы облачные вычисления — дело интернет-провайдеров; будто бы разбираться в том, что эти вычисления собой представляют и как работают, необязательно; будто бы “переход в облако” сам по себе решит все болезненные технические и организационные проблемы информатизации школы.

Новая составляющая компьютерной грамотности

Лишь небольшая часть работников образования осознает сегодня значение перемен, которые несут с собой облачные вычисления. В многочисленных выступлениях и публикациях поставщики информационных технологий (ИТ) используют облако как маркетинговый слоган, как место для размещения “электронных учебников”, “дистанционных образовательных технологий”, “электронных журналов” и т.п. Немногие знают, что появление и распространение облачных сервисов непосредственно связано с массовым внедрением параллельных вычислений, с изменением традиционного представления о работе компьютеров и возможностями их повседневного использования. Вычислительные мощности, доступные прежде только профессионалам, ныне уже доступны каждому. Примером может служить машинный перевод.

Несмотря на фундаментальную роль облачных вычислений для информатизации школы, их использование пока не стало значимым фактором педагогических исследований и разработок. Облачные вычисления не обсуждаются в курсах информатики и вычислительной техники ни в школе, ни в вузах, где готовят будущих учителей. Ощущается острый дефицит в материалах, из которых педагоги могут узнать о сути облачных вычислений и о том, почему развитие облачных сервисов (вместе с широким распространением персональных вычислений) создает качественно новые условия для информационной работы и жизни людей, для решения задач информатизации образования.

Автор убежден, что эти вопросы должны найти отражение в программах по информатике и ком-

пьютерной грамотности для школьников, педагогов и работников управления школой на всех уровнях. В свое время курсы компьютерной грамотности помогли сформировать у работников школы представление о компьютере как о рабочем инструменте, который заменяет калькулятор, проектор и т.п. Знакомство с Интернетом позволило увидеть его как средство коммуникации, поиска информации и доступа к удаленным хранилищам данных. Сегодня каждому требуется понимать основные идеи облачных вычислений, чтобы осознанно приобретать новые технические устройства и программные средства, использовать поистине фантастические возможности, которые открывает складывающаяся в настоящее время единая цифровая информационная среда, или киберпространство.

Знакомство с этими идеями особенно важно для тех, кто принимает (или консультирует) решения о приобретении технических средств, проведении новых разработок и использовании услуг, предлагаемых многочисленными поставщиками ИТ. Представление об облачных вычислениях необходимо и тем, кто ведет собственные (в том числе педагогические) разработки, планирует развитие инновационных процессов в отдельной школе, в муниципальном образовании или в сети школ региона.

И учащиеся, и работники школы должны понимать, как развитие интернет-сервисов, виртуализация вычислений и программируемый web привели к появлению облачных вычислений и сервисов. Каждый житель информационного века должен представлять себе принципы их построения, основы виртуализации вычислительной инфраструктуры и т.п.

Облачные вычисления — молодая и быстро развивающаяся область ИТ. Сам термин “облачные вычисления” появился немногим более пяти лет назад [3]. На рынке уже предлагаются комплексные решения, которые позволяют предоставлять облачные сервисы самым различным категориям потребителей. Здесь и финансовый сектор, и сектор телекоммуникаций, и промышленность и сфера услуг, торговля, государственный сектор, наука и образование. Современный потребитель ИТ-услуг должен не только знать о существовании облачных сервисов, но и различать их с точки зрения пользователя, понимать их возможности и ограничения.

Работники образования должны быть информированы о путях влияния облачных вычислений на процессы информатизации школы, понимать возможности и потенциал, который можно использовать для развития инновационных процессов в ее рамках. Без знаний о современном состоянии и перспективах облачных вычислений невозможно осознанно разрабатывать программы развития школы на всех уровнях, выбирать стратегии ее оснащения средствами ИТ, закупать технические средства и программное обеспечение с учетом ближайшей перспективы, оценивать, приобретать и осваивать цифровые учебные и методические материалы и пр.

² http://minobr.gov-murman.ru/opencms/export/sites/minobr/content/docs/GLAS/koncepciya_eios.pdf.

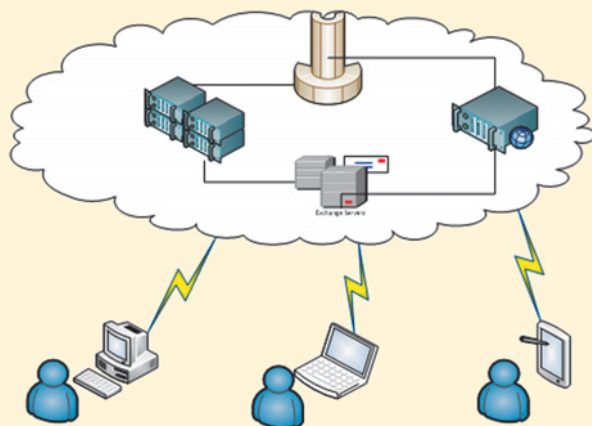


Рис. 1. Метафора “вычисления в облаке”

Работникам образования желательно знать о перспективных облачных решениях, дающих реальную (по людским и материальным ресурсам) возможность их школе, муниципальному образованию или сети школ региона:

- использовать облачные сервисы для комплексного решения задачи своей информатизации,
- расширить образовательное пространство и потенциал учебной среды учащихся,
- перейти к лично ориентированной (индивидуализированной) системе учебной работы.

Чтобы достаточно полно осветить все эти вопросы, требуется написать не одну книгу. Цель настоящей публикации — представить читателям круг идей, которые привели к появлению облачных вычислений. Мы рассмотрим их истоки, состав основного пакета облачных сервисов, принципиальную схему их работы. Если читатель данного материала увидит, почему эта тема должна найти место в курсе информатики, почему распространение облачных сервисов наряду с мобильными средствами ИТ качественно меняет представление об информационной инфраструктуре школы и всей сферы образования, автор будет считать свою задачу выполненной.

Сервисы из облака

Рис. 1 иллюстрирует истоки возникновения метафоры “вычисления в облаке”. Традиционно разработчики программных систем и архитекторы ИТ-решений рисовали на своих схемах изображение облака. Оно обозначало удаленный ИТ-ресурс, который доступен через web. Постепенно изображение облака стало распространенной логической связкой между локальными ресурсами и удаленными ресурсами, которые доступны через Интернет. Эта примитивная трактовка облачных вычислений сегодня широко эксплуатируется поставщиками интернет-услуг. Они пытаются убедить недостаточно осведомленных клиентов, что, поскольку предоставляемые им ИТ-услуги используют хост-компьютеры (хост-узлы) поставщиков интернет-услуг, эти услуги можно с полным правом называть “облачными

сервисами”. Хотя эта логика имеет право на существование, такое понимание облачных вычислений слишком упрощено и сильно расходится с принятым сегодня пониманием облачных вычислений в среде специалистов³. Чтобы разобраться в этом, следует рассмотреть три составляющих развития ИТ-услуг, которые привели к облачным вычислениям. К ним относятся:

- развитие услуг, которые предоставляют клиентам поставщики интернет-сервисов,
- основные интернет-сервисы,
- виртуализация вычислений.

Развитие интернет-услуг

Одна из ключевых причин появления облачных вычислений — становление крупных и достаточно зрелых поставщиков интернет-услуг, которые предоставляют потребителям доступ к вычислительным ресурсам и сервисам сети. В литературе поставщиков интернет-услуг обычно называют ISP (*Internet Service Providers*). ISP появились более четверти века назад, и за это время их услуги несколько раз качественно обогащались. Можно сказать, что ISP прошли пять ступеней развития, расширяя спектр и содержание своих услуг. Переход на очередную ступень можно назвать появлением очередного поколения ISP.

ISP первого поколения (ISP-1) предоставляли своим клиентам точки доступа к Интернету. Потребители услуг могли подключиться к этим точкам по коммутируемым и выделенным телефонным каналам, а также по специально проложенным линиям цифровой связи.

ISP второго поколения (ISP-2) стали предоставлять своим клиентам услуги на компьютерном оборудовании (хост-компьютеры, хост-узлы), которое сами ISP устанавливали в точках доступа к Интернету. Клиенты могли воспользоваться почтовыми системами, хранить свои данные, обмениваться ими, проводить телеконференции и пр.

ISP третьего поколения (ISP-3) уже предоставляли потребителям услуг возможность устанавливать в точках доступа к Интернету клиентское оборудование, которое решает специфические задачи этих потребителей (например, сетевые базы данных, справочные службы и т.п.). Фактически ISP-3 предоставляли в лизинг потребителям место на стойках своих серверов и мощности каналов связи.

ISP четвертого поколения (ISP-4) стали размещать на своих серверах в точках доступа к Интернету различные традиционно разработанные программные системы. Появилось множество доступных через Интернет приложений, построенных по архитектуре “клиент–сервер”. ISP-4 вырастают до поставщиков приложений (*Application Service Provider — ASP*).

³ Определение облачных вычислений, которое узаконено IEEE, приведено ниже в разделе “Свойства и организация облачных вычислений”.

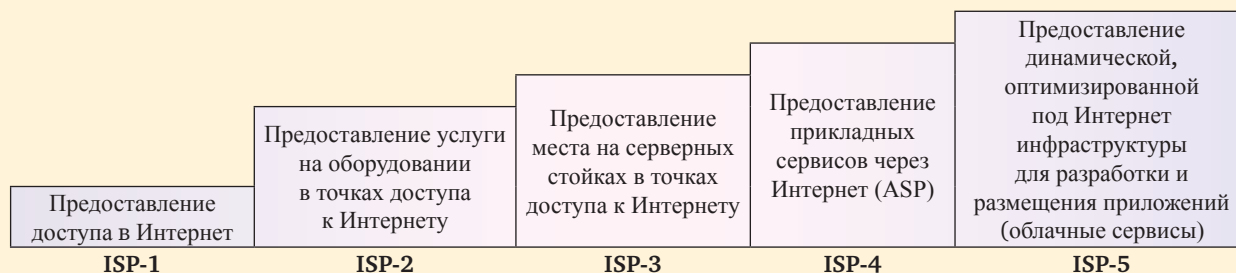


Рис. 2. Четверть века развития интернет-услуг

Примером в области образования может служить созданная в рамках проекта “Информатизация системы образования” [1] единая коллекция цифровых образовательных ресурсов (<http://school-collection.edu.ru/>). Другой известный пример — “е-КМ-Школа” (<http://www.km-school.ru/>), где желающие могут пройти дистанционное обучение.

Сегодня ведущие ISP в очередной раз качественно обновляют предоставляемые ими услуги, превращаются в поставщиков интернет-сервисов пятого поколения (ISP-5). Современные облачные сервисы включают в себя предоставление через Интернет и удаленных вычислительных мощностей, и платформ для разработки приложений, и самих прикладных программных средств (приложения, или App). Огрубляя, можно сказать, что ISP-5 готовы предоставлять своим потребителям в качестве платного сервиса (услуги) любые составляющие всего спектра вычислительных услуг. Клиентам больше не нужно создавать собственные вычислительные мощности, поддерживать серверы с системными и прикладными программными средствами, наращивать производительность персональных вычислительных средств. Провайдеры услуг пятого поколения обещают, что их потребители смогут получить любые необходимые им IT-ресурсы и сервисы через Интернет.

Итак, мы перечислили основные этапы развития интернет-услуг (рис. 2) за прошедшие четверть века.

Начав с предоставления доступа к Интернету (ISP-1), интернет-провайдеры очень скоро стали предоставлять своим клиентам услуги на оборудовании, которое они устанавливали в точках доступа к Интернету (ISP-2). На заре Интернета подавляющее большинство его узлов принадлежало исследовательским центрам, которые ориентировались на обслуживание своих организаций. ISP-2 стали первыми коммерческими сервисными центрами Интернета. Фактически именно тогда и зародились интернет-сервисы.

Развивая эти услуги, ISP-3 стали предоставлять клиентам возможность устанавливать их собственные серверы в точках доступа к Интернету. Этими клиентами были прежде всего те, кто сам предоставлял специфические услуги (удаленный доступ к данным, информирование и т.п.) пользователям Интернета.

Дальнейшее развитие привело к формированию архитектуры “клиент–сервер” и появлению порталов крупных поставщиков различных услуг через Интернет (электронная торговля, заказ гостиниц, дистанционное обучение и т.п.). ISP-4 стали размещать на своих серверах в точках доступа к Интернету различные программные системы. Возникло множество доступных через Интернет приложений, а поставщики интернет-услуг выросли до поставщиков приложений (*Application Service Provider* — ASP).

Следующий шаг на этом пути — предоставление облачных сервисов (ISP-5).

Мир интернет-сервисов

Всемирную паутину (World Wide Web) можно рассматривать как платформу для создания и использования распределенных программных систем на основе веб-сервисов. Веб-сервис (web-service) — программная система для поддержки интероперабельного межкомпьютерного (machine-to-machine) взаимодействия через сеть на основе веб-стандартов⁴.

В основе любого веб-сервиса лежит простая идея. Веб-сервис принимает поступивший запрос, обрабатывает его и возвращает ответ. Обычно веб-сервис принимает один или несколько входных параметров, вызывает логическую процедуру для их обработки и возвращает выходные значения.

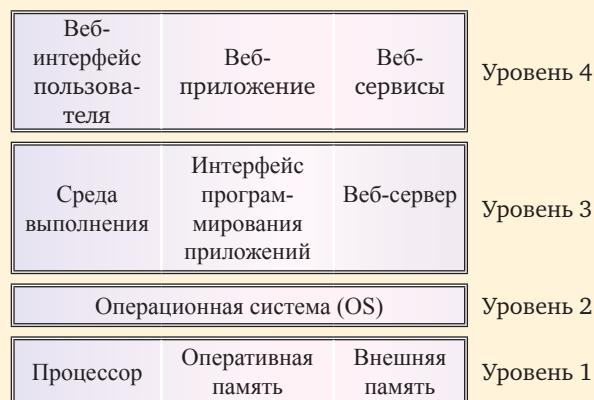


Рис. 3. Стек исполнения веб-сервисов

⁴ Так определяет веб-сервис консорциум W3C (<http://w3c.org.ru/>).

Для разработчика любой веб-сервис является составной частью веб-приложения, которое выполняется с помощью обычного пакета (стека), включающего в себя: аппаратное обеспечение сервера, его операционную систему и платформу разработки приложений (рис. 3 на с. 23). Здесь среда выполнения приложений — вычислительное окружение, доступное во время работы компьютерной программы. В среде выполнения, как правило, невозможно изменить исходный код программы, но можно получить доступ к переменным окружения операционной системы, таблицам объектов и модулей разделяемых библиотек. Веб-сервис (или веб-служба) — это сетевая технология, которая обеспечивает межпрограммное взаимодействие в сети. Консорциум W3C определяет веб-сервис как программную систему, разработанную для поддержки межкомпьютерного (machine-to-machine) взаимодействия через сеть⁵.

Каждый из слоев пакета, который обеспечивает выполнение веб-приложения, можно представить в виде веб-сервиса⁶. Рассмотрим ситуацию, где аппаратное обеспечение и операционная система доступны в Интернете в виде веб-сервиса. Если такой веб-сервис имеется, то можно написать веб-приложение, которое отправит ему запрос и передаст набор параметров, которые он будет использовать при выполнении этого запроса. Операционная система (OS) — это программный интерфейс между техническими средствами компьютера (процессор, устройства ввода-вывода и т.п.) и приложением. В результате веб-сервис, который передает запрос операционной системе, примет запрос на любую работу, которую должны выполнять OS и управляемые ею аппаратные средства. Таким образом, OS вместе с аппаратными средствами превращаются в облачную OS, в доступный на WWW сервис (услугу), который можно предоставлять через Интернет по сходной цене.

Вычислительная инфраструктура как услуга — IaaS

Доступ к облачной OS аналогичен доступу к виртуальной вычислительной системе. С технической точки зрения это доступ к вычислительной инфраструктуре как услуге (*Infrastructure as a Service* — IaaS). Сюда могут входить виртуальные машины, образы диска, файлы для хранения данных, межсетевые экраны, средства балансировки нагрузки, IP-адреса, виртуальные локальные сети, пакеты программного обеспечения и др.



Рис. 4. Инфраструктура как услуга (Infrastructure as a Service)

IaaS принципиально отличается от веб-хостинга, который предлагали ISP-2–ISP-4 (рис. 1 на с. 22). При веб-хостинге на серверах провайдеров исполняются только веб-страницы, которые не имеют доступа к командам нижнего уровня и не могут выполнять действия по запуску виртуальных машин, подключению устройств и т.п. IaaS позволяет управлять вычислительной системой, подключать дополнительные ресурсы из сети и при необходимости выполнять задачи пользователя на теоретически неограниченном количестве виртуальных машин. Вместе с тем использовать IaaS — значит использовать расположенный в облаке сервер. А это значит, что потребитель IaaS берет на себя ответственность за управление работой удаленных серверов, обеспечение их защиты от вирусов и т.п. Таким образом, IaaS — услуга для специалистов в области IT, которые раньше устанавливали и обслуживали физические серверы в организации, а теперь не хотят тратить время и силы на рутинную поддержку инфраструктуры и в первую очередь ее физической составляющей (рис. 4).

Вместо серверов, которые физически размещались на территории организации, специалисты по IT используют удаленные консоли для управления и обслуживания удаленных серверов, которые располагаются в облаке.

Платформа как услуга — PaaS

Все сетевые прикладные программные средства (приложения), которые создаются и применяются в настоящее время, используют платформы для разработки приложений. Среди наиболее известных сегодня платформ — платформы Java и .Net.

Выше рассматривалась ситуация, где аппаратное обеспечение и операционная система были доступны в Интернете в виде веб-сервиса. Рассмотрим ситуацию, где в качестве сервиса предлагается платформа для разработки приложений. С помощью такого сервиса можно разрабатывать и отлаживать сетевое приложение на обычном персональном компьютере. Однако создаваемый код будет запускаться на высокопроизводительной, обладающей самыми широкими возможностями вычислительной системе. При этом язык программирования, инструменты

⁵ <http://www.w3.org/2002/ws/>.

⁶ Идея объяснений заимствована из работы [12].



Рис. 5. Платформа для разработки приложений

разработки (*Software Development Kit* — SDK) и среда выполнения⁷ остаются теми же, что и на обычной платформе для разработки приложений.

Таким образом, в качестве сервиса разработчик получает платформу как услугу (PaaS). Этот сервис включает в себя аппаратные средства и операционную систему, где должно выполняться приложение, среду исполнения приложения, язык программирования, SDK и платформенные компоненты (рис. 5). Все это теперь — платформенные компоненты (большие “кубики”), из которых можно быстро и эффективно “собирать” свое ПО, оптимизированные для использования в облаке и полностью реализующие все его преимущества. PaaS предоставляет разработчикам эластичную, масштабируемую платформу для тестирования и выполнения создаваемых ими приложений.

Одно из главных преимуществ использования PaaS (по сравнению с использованием традиционной организации работ) в том, что разработчикам больше не требуется устанавливать, отлаживать и обслуживать серверы, на которых должны разрабатываться и тестироваться новые приложения. Базовые конфигурации серверов (такие, как IIS и SQL) становятся целостными платформенными компонентами, не требуют от разработчика настройки и обслуживания, их можно сразу использовать. Существуют платформенные компоненты еще более высокого уровня абстракции. Здесь разработчик получает уже не просто отконфигурированный и оптимизированный для облака сервер, а заверченный сервис (например, работа с мобильными устройствами, с медиаконтентом и т.п.). Поставщики PaaS берут на себя все связанные с этим заботы и предоставляют как услугу целостную платформу разработки приложений. Все, чем физически пользуется разработчик, — это обычный персональный компьютер,

⁷ Среда выполнения — вычислительное окружение, доступное во время работы компьютерной программы. В среде выполнения нельзя изменять исходный текст, но можно получить доступ к переменным окружения операционной системы, таблицам объектов и модулей разделяемых библиотек.

подключенный к сети. При этом разрабатываемое приложение легко масштабируется. Оно будет устойчиво функционировать независимо от того, будут ли его использовать несколько десятков или несколько миллионов клиентов. Оно может быть доступно для любого пользователя Интернета, а его надежная работа гарантируется поставщиком облачных услуг.

Распространение PaaS резко сокращает трудоемкость и сроки выполнения прикладных разработок, существенно снижает их стоимость. То, что ранее было доступно лишь крупному бизнесу, становится по силам небольшим организациям со скромными бюджетами. Одним из примеров PaaS могут служить компоненты Microsoft Azure (<http://www.windowsazure.com/ru-ru/overview/what-is-windows-azure/>).

Прикладная программа как услуга — SaaS

Если программное приложение построено на базе сервиса PaaS, его также можно представить как услугу (*Software as a Service* — SaaS). Это показано на рис. 6. Чтобы им воспользоваться, нет нужды устанавливать соответствующий программный пакет на своем компьютере. Достаточно подключенного к Интернету нетбука и браузера, чтобы запустить нужное программное приложение в облаке.



Рис. 6. Программное приложение как услуга (Platform as a Service)

Появление SaaS ведет к революционным изменениям в приобретении и использовании обычных программных продуктов. Эта тихая революция сегодня происходит на наших глазах. Используя SaaS и высокоскоростной канал, уже сегодня можно подписаться на современное программное обеспечение как на услугу. Программу не нужно устанавливать на компьютере пользователя, она не занимает память и не чувствительна к производительности процессора. При этом к ней можно всегда обратиться, когда она требуется, и платить только тогда, когда она используется. Это путь к экономии усилий, технических средств и бюджета (использование облачного программного обеспечения (ПО), как правило, дешевле, чем покупка локальной версии аналогичного ПО и его установка на своем компьютере).

Итак, мы обсудили облачные сервисы (рис. 7). Их формирование стало одной из главных составляющих развития ИТ-услуг, которое привело к появлению облачных вычислений.

Сегодня сервис IaaS (инфраструктура как услуга) вытесняет распространенную во времена ISP-3 практику предоставления потребителям в лизинг мест на стойках своих серверов. Каждый, кому требуется интернет-сервер, может получить его практически сразу. Для этого достаточно заказать соответствующую услугу через Интернет, причем оплачивать лишь фактически используемые вычислительные мощности. Все затраты, связанные с закупкой, установкой, наладкой и поддержанием работы оборудования, отпадают.

Распространение сервиса PaaS (платформа как услуга) революционизирует проектирование и разработку программных приложений, которые доступны через Интернет. PaaS решает проблему масштабирования приложений, предоставляет разработчикам богатые библиотеки готовых решений, существенно упрощает отладку систем и их ввод в эксплуатацию и т.п. То, для чего раньше требовались человеко-месяцы, теперь решается за человеко-недели, а порой и дни. Стоимость разработок сокращается в разы. Освобождаясь от решения рутинных технических задач, разработчики получают возможность сосредоточиться на самих задачах, что переводит веб-технологии на более высокий качественный уровень.



Рис. 7. Пакет облачных сервисов

Новые возможности, которые PaaS создает разработчикам прикладных программ, обеспечивают качественное развитие SaaS. Программные прило-

жения для решения повседневных задач, доступные прежде лишь крупным бизнес-структурам и банкам, становятся доступны массовому потребителю. К таким потребителям относятся и школы. У образовательных учреждений появляется надежда получить по разумной цене крайне необходимую им цифровую инфраструктуру промышленного уровня. Чтобы лучше понять, на чем основана эта надежда, рассмотрим виртуализацию вычислений, которая является ключевой составляющей развития ИТ-услуг, приведших к появлению облачных сервисов.

Виртуализация вычислений

Идея виртуальных вычислений достаточно стара. Впервые она была реализована почти полвека назад компанией IBM при создании операционной системы OS 370 VM и с тех пор стала обычным решением для больших вычислительных машин. Однако распространение дешевых серверов на базе процессоров с архитектурой x86 на долгое время отвлекло внимание массового рынка от этой идеи. В середине 2000-х годов ситуация стала меняться. Экспоненциальный рост производительности серверного оборудования вместе с лавинообразным ростом транзакций, которые должны обрабатывать программы на серверах, возродили интерес к виртуализации вычислений, что в конечном итоге привело к возникновению облачных вычислений.

Виртуализация возникла как ответ на желание разработчиков компьютеров максимально использовать вычислительные возможности высокопроизводительных центральных процессоров. Вспомним традиционную организацию вычислений, которая показана на рис. 8. Между прикладной программой и техническими средствами компьютера находится OS, которая обеспечивает взаимодействие между ними.



Рис. 8. Традиционная организация вычислений: компьютер, OS и прикладные программы

Можно представить и другую вычислительную архитектуру, где на компьютере запускается сразу несколько OS, каждая из которых работает со своими приложениями (App). Комбинацию (OS + App + ... + App), которая является образом запускаемой OS, принято называть виртуальной машиной (VM). Для запуска VM на компьютере между техническими средствами компьютера и самой VM тоже нужна управляющая программа (головная OS). Как правило, это сравнительно небольшая программа, функции которой ограничены

запуском и свертыванием работы виртуальных машин. Такую управляющую программу обычно называют гипервизор (HV). Гипервизор нужен для виртуальных машин так же, как OS для прикладных программ (App).

Итак, виртуализация вычислений — это абстрагирование их от технических средств компьютера. Виртуализация позволяет запускать несколько виртуальных операционных систем с их приложениями (VM) на одной головной вычислительной системе под управлением HV.

Хотя программные средства, позволяющие запускать VM на персональных компьютерах, существуют (например, MS Virtual PC), они используются достаточно редко. Персональный компьютер обычно не предназначен для параллельного выполнения большого количества вычислительных задач. Другое дело — сервер, который может обслуживать запросы многих сотен и тысяч пользователей сети (рис. 9). На одном комплексе технических средств можно запустить несколько виртуальных машин, каждая из которых будет исполнять роль виртуального сервера.

Гипервизор (HV) — системная программа для эффективного управления виртуальными машинами. Это сравнительно небольшая программа, которая заменяет OS и может загружаться непосредственно виртуальной машиной. Такой HV имеет встроенную специализированную OS и использует средства виртуализации, которые встроены в современные серверные процессоры, выпускаемые Intel и AMD. Комбинация такого процессора с HV превращает физический сервер в устройство, на котором можно удаленно загрузить гипервизор и запустить много виртуальных серверов. Это делает идею виртуализации универсальной и очень мощной. Используя широкополосный доступ, можно за считанные секунды загрузить физический сервер и запустить на нем множество гостевых VM. Работой гипервизора легко управлять с помощью центральной консоли, указывая, на каких физических серверах загружаться и какие VM на нем запускать. Это делает функционирование технических средств в центрах обработки данных необычайно динамичным и гибким.

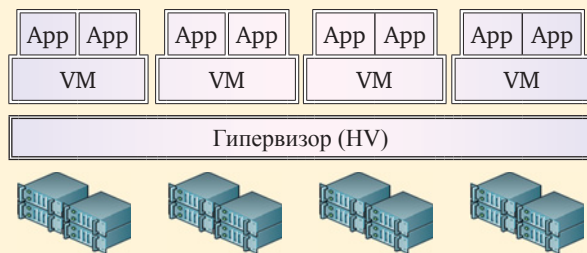


Рис. 9. Гипервизор (HV) поддерживает работу виртуальных машин (VM) на сервере

Перечислим четыре основных причины, которые ведут к виртуализации серверов в центрах обработки данных (ЦОД).

1) Уменьшение времени на восстановление. Восстановить “упавший” веб-сервер, сервер приложений или сервер баз данных, который выполняется

как VM, много быстрее и проще, чем восстановить аналогичный физический сервер. Виртуальный сервер (VM) физически представляет собой всего лишь набор файлов на диске головной операционной системы (HV). Загрузить копию этих файлов (которые фиксируют состояние “упавшего” сервера) на выполнение гораздо быстрее и проще, чем запустить и настроить физический сервер с аналогичными функциями. Администраторы могут предусмотреть серию промежуточных копирований соответствующих VM, что облегчит процесс их восстановления. И, что еще более важно, весь процесс восстановления VM гипервизор может осуществить в автоматическом режиме как одну из стандартных процедур восстановления системы.

2) Балансировка загрузки физических серверов. В ЦОД нередко возникает ситуация, когда одни физические серверы перегружены, в то время как другие простаивают. Виртуализация вычислений позволяет сбалансировать загрузку серверов. Одна из функций гипервизора состоит в том, чтобы автоматически перенести выполнение VM с перегруженного физического сервера на свободный.

3) Уменьшение стоимости используемого оборудования и его обслуживания. Повышение эффективности применения физических серверов за счет балансировки их загрузки позволяет заметно сократить их количество в ЦОД и достаточно полно использовать современную высокопроизводительную вычислительную технику. В результате снижаются производственные расходы на приобретение оборудования, кондиционирование помещений, электроэнергию.

Как известно, серверное оборудование достаточно часто обновляется. До появления виртуализации замена устаревших серверов и установка новых представляла собой весьма длительную и сложную технологическую операцию, которая требовала привлечения большого числа высокооплачиваемых специалистов. Виртуализация вычислений значительно упрощает решение этой задачи. Устаревший сервер просто отключается, а выполнявшиеся на нем VM автоматически восстанавливаются на других физических серверах. Это позволяет существенно снизить затраты на поддержку работы ЦОД.

4) Эффективное управление вычислительными мощностями. Программное обеспечение, которое используется для виртуализации вычислений, имеет центральную консоль. С ее помощью пользователи могут управлять работой, поддерживать и контролировать состояние физических серверов, а также работающих на этих серверах виртуальных машин. Это позволяет системным администраторам легко управлять всей вычислительной инфраструктурой ЦОД, тратить меньше времени на решение возникающих задач. В итоге управление работой ЦОД улучшается и экономятся средства на его эксплуатацию.

Свойства и организация облачных вычислений

Национальный институт стандартов и технологии [13] определяет вычислительное облако как особую модель предоставления услуг, которая обеспечивает удобный и повсеместный доступ по требованию к удаленному пулу конфигурируемых вычислительных ресурсов. Эти ресурсы оперативно предоставляются и высвобождаются по мере необходимости с минимальными эксплуатационными затратами и/или обращениями к провайдеру.

Таким образом, облако — это предоставление провайдером удаленных вычислительных ресурсов и услуг по запросу потребителя. Однако не всякие услуги провайдеров можно назвать облачными.

Основные свойства облака

Ключевые свойства облачных услуг выделены словами: “оперативно предоставляются и высвобождаются”, “минимальные эксплуатационные затраты”, “минимальные обращения к провайдеру”, “удобный и повсеместный доступ”, “пул ресурсов”. Эти свойства обычно называют: масштабируемость, оплата по мере использования, самообслуживание, универсальный доступ по сети, объединение ресурсов, программируемость.

Масштабируемость. Это способность вычислительной системы реагировать на увеличение или уменьшение нагрузки увеличением или снижением производительности путем добавления или высвобождения используемых ресурсов. Масштабируемость обеспечивается путем виртуализации вычислений, которая является важнейшим атрибутом вычислительного облака. Приложение, которое сначала использует один сервер, по мере увеличения нагрузки (например, подключение новых пользователей) может подключать дополнительные ресурсы и использовать сотни серверов. Важно, что такое увеличение масштаба может происходить в том числе автоматически. Когда нагрузка снижается, недостаточно загруженные серверы могут высвободиться от этой работы. Все это может происходить мгновенно, так что пользователи системы этого не замечают.

Свойство динамического разворачивания и сворачивания используемых вычислительных мощностей обычно называют масштабируемостью приложений, или эластичностью обработки данных. Масштабируемость дает возможность приложению получить любое количество ресурсов, которое требуется для его успешного выполнения в каждый текущий момент. Это одно из наиболее привлекательных свойств облачных вычислений. В отличие от веб-хостинга, где подключение дополнительных серверов и настройка топологии сети требуют значительных усилий и времени, облачный сервис позволяет легко управлять конфигурациями серверов и их топологией.

Оплата по мере использования. Виртуализация вычислений и масштабируемость приложений позволяют достаточно точно учитывать ресурсы, которые использует каждое приложение, и оплачивать их с учетом объема использования. Соответственно, потребителю не требуется закупать и эксплуатировать дорогостоящее серверное оборудование, рассчитывая его на максимальную нагрузку. Так, школьные серверы, которые рассчитаны на обслуживание тысяч пользователей в пиковом режиме, практически не загружены в ночное время. Используя облачные сервисы, школа может оплачивать только то время, когда она ими фактически пользуется. В результате вычислительные ресурсы оплачиваются по счетчику, как вода или электроэнергия. Это позволяет перевести инвестиционные расходы (закупка оборудования) в текущие расходы. Школа может оперативно подписаться на требуемые услуги или временно отказаться от их использования.

Оплата по мере использования — лучший способ оптимизировать расходы организации на ИТ. В то же время облачный провайдер может полнее загрузить свои вычислительные ресурсы, обслуживая пользователей из различных временных зон.

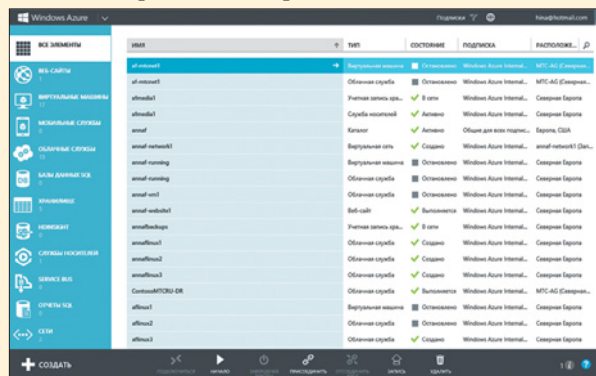


Рис. 10. Пример панели управления облаком в MS Windows Azure

Самообслуживание. Чтобы в полной мере использовать возможности масштабируемости и оплаты по мере использования, потребители услуг должны иметь возможность управлять используемыми ресурсами без посредников, подключая или отключая их в режиме самообслуживания. Панели управления (рис. 10), которые предоставляют потребителям провайдеры облачных услуг, позволяют пользователям самостоятельно управлять вычислительными ресурсами, не обращаясь к персоналу сервис-провайдера. Важно, что для этого пользователю не нужна специальная подготовка. По мере того как приложения перемещаются в облако, самообслуживание становится простым и удобным способом настройки и управления ИТ-инфраструктурой.

Универсальный доступ по сети. Это еще одно обязательное свойство облачных услуг. Необходимые услуги обычно доступны потребителям по сети передачи данных (через Интернет) независимо от используемых терминальных устройств. Это позволяет запустить любую ресурсоемкую задачу

(например, машинный перевод или распознавание речи) с любого мобильного устройства, включая обычный смартфон.

Облачные сервисы являются дешевыми и массовыми. При этом предоставление облачных услуг, как правило, выгоднее хостинга. Это связано с экономией на масштабе. Обслуживая большое количество потребителей, облачный провайдер объединяет все ресурсы в единый пул и динамически перераспределяет мощности между потребителями по мере того, как спрос на эти мощности изменяется. Такое свойство облаков называют **объединение ресурсов**.

Потребители контролируют только основные параметры услуги (объем данных, скорость доступа и т.п.). Иногда они могут влиять на отдельные физические параметры распределения ресурсов (например, указать желаемый центр обработки данных с учетом своего географического положения). Но фактически все физические ресурсы, которые предоставляются потребителю, распределяет поставщик. Благодаря объединению ресурсов и непостоянному характеру их потребления со стороны потребителей, облачные вычисления позволяют поставщикам услуг существенно экономить на масштабах, используя гораздо меньше аппаратных ресурсов, чем это требовалось бы каждому потребителю при выделении ему необходимых аппаратных мощностей в исключительное использование. Автоматизация процедур модификации выделения ресурсов позволяет существенно снизить затраты на абонентское обслуживание.

Экономия на масштабе, которая выгодна поставщику облачных услуг, не конфликтует с интересами потребителя, который этого, как правило, не замечает. Он получает необходимые ему услуги с высоким уровнем доступности (по мере требования) и низким риском неработоспособности. Потребителю облачных услуг не требуется развивать собственную вычислительную инфраструктуру. Все нужные ему приложения быстро масштабируются вычислительной системой облачного провайдера благодаря ее эластичности.

Программируемость. Одно из самых привлекательных свойств облака — его программируемость, которую обеспечивает сервис PaaS. С его появлением у потребителя интернет-услуг появилась возможность без дополнительных затрат программировать, отлаживать и тестировать новые разработки в реальной производственной среде, что значительно сокращает сроки их выполнения и снижает стоимость. Облачные решения могут достаточно динамично реагировать на изменения потребностей пользователей системы, а сами программисты становятся активными потребителями облачных сервисов.

Например, такое облачное решение, как MS Windows Azure, позволяет программистам и архитекторам пользоваться различными традиционными наборами инструментов для разработки приложений (SDK), включая Java, .Net, PHP, Python

и Ruby. Кроме того, разработчики получают возможность решать задачи автоматического масштабирования приложений. По мере необходимости эти приложения могут использовать от нескольких серверов до нескольких тысяч серверов. Разработчики могут использовать как готовые платформенные компоненты, так и создавать дополнительные VM и добавлять их к своим приложениям по требованию.

Облако и Хостинг

Обсуждая четвертьвековую историю развития интернет-услуг (рис. 2), мы видели, как ISP постепенно наращивали свои возможности. Хостинг, который появился на уровне ISP-2, перерос в разработку решений “клиент–сервер” и предоставление пользователям на этой основе программных приложений. Предоставление прикладных сервисов через Интернет (ASP), которые ISP-4 стали называть SaaS, стало завершающим шагом, за которым последовал качественный скачок облачных сервисов. Этот качественный скачок связан с развитием виртуальных вычислений, появлением на рынке IaaS, PaaS и SaaS.

Выделенные выше свойства облачных вычислений позволяют сопоставить возможности облачных вычислений и веб-хостинга (рис. 11). Очевидно, что хостинг не может удовлетворить требованиям масштабируемости в той степени, в какой это позволяет облако. Если это и делается, то далеко не с теми экономическими показателями, которые можно достичь в облаке, где доступно объединение ресурсов.

Свойство	Облачные вычисления	Веб-хостинг
Масштабируемость	Есть	Нет
Объединение ресурсов	Есть	Нет
Программируемость	Есть	Нет
Оплата по мере использования	Есть	Нет
Самообслуживание	Есть	Частично
Универсальный доступ по сети	Есть	Есть

Рис. 11. Сопоставление возможностей облачных вычислений и веб-хостинга

Поддержка среды разработки обходится провайдерам веб-хостинга слишком дорого, так как им нет прямой нужды инвестировать в разработку или закупку инструментов для управления инфраструктурой и SDK. Естественная организация оплаты сервиса по мере его использования является следствием виртуализации и возможна лишь в облаке. Некоторые провайдеры эмулируют оплату за использование отдельных атрибутов, но такие решения во многом искусственны и не являются нормой для веб-хостинга.

Как видно из рис. 11, облачные вычисления качественно отличаются от веб-хостинга, а их поставщики с полным правом могут быть отнесены к новому

поколению интернет-сервис-провайдеров — ISP-5. Отличительные особенности облака: масштабируемость, объединение ресурсов, программируемость, оплата по мере использования и реальное самообслуживание.

Облачные вычисления объединяют в себя самые последние достижения в области технических и программных средств ИТ. Их разработка и вывод на рынок услуг достаточно дороги. Подобные инвестиции доступны лишь лидерам ИТ-индустрии. Среди них такие гиганты, как Amazon, Microsoft и Google. Сегодня эти инвестиции осуществлены, и различные облачные сервисы доступны массовому пользователю. Их услугами могут воспользоваться потребители и в России. Одновременно продолжается быстрое развитие облачных серверов в сторону их унификации и совместимости. Как отмечает Вальтер Вогельс [3], облако перестает раскладываться на четко выраженные слои сервисов. В будущем многие приложения будут собирать разные сервисы у разных провайдеров и совмещать их воедино для максимального удовлетворения запроса конечных потребителей.

Модели развертывания облачных вычислений

Закончив рассматривать основные свойства облака, перечислим базовые модели развертывания облачных вычислений и предоставление к ним доступа потребителям. Эти модели уже сложились. Их обычно называют частным, коммунальным, публичным и гибридным облаками.

Частное облако

Частное облако (private cloud) — вычислительная инфраструктура, которая предназначена для использования в одной организации. Она может включать нескольких потребителей. Например: подразделения организации (расположенные в разных зданиях), ее клиенты и подрядчики. Частное облако может находиться в собственности, управляться или эксплуатироваться как самой организацией, так и внешним агентом. Оно может находиться под юрисдикцией самой организации или быть вне ее юрисдикции. Иначе говоря, частное облако — обычный центр обработки данных (ЦОД) организации, который обладает всеми свойствами облака (см. рис. 11 на с. 29). Такой ЦОД консолидирует и снижает издержки на свою информационную инфраструктуру. Ему требуется меньше высококвалифицированного технического персонала для сопровождения и развития вычислительных ресурсов. Он может эффективнее использовать имеющиеся вычислительные мощности, снижать потребление электроэнергии за счет экономии на серверах и охлаждающих устройствах.

Частное облако дает возможность сотрудникам организации управлять используемыми ресурсами в режиме самообслуживания, гибко предоставлять им необходимые вычислительные мощности и определять новых участников проекта. Частное облако использует все преимущества облачных вы-

числений в пределах заданной организационной границы. Сегодня имеется множество предложений по построению частных облаков от поставщиков оборудования и программного обеспечения для ЦОД. Все они ориентированы прежде всего на крупные организации, которые способны нанять высококвалифицированный персонал для поддержания работоспособности технических и программных средств, которые эксплуатируются в ЦОД.

Коммунальное облако

Коммунальное облако (community cloud)⁸ — вычислительная инфраструктура, которая предназначена для конкретной группы организаций, имеющих общие задачи. У членов этой группы сходная миссия, требования к безопасности, подходы к использованию ИТ и т.п. Коммунальное облако может являться совместной собственностью, управляться или эксплуатироваться одной или несколькими организациями данной группы, а также внешним агентом, находиться под юрисдикцией владельцев или вне их юрисдикции.

Обычно коммунальное облако создают в том случае, когда достаточно большая группа организаций нуждается в аналогичных средствах обработки данных, которые представлены в схожем формате и имеют общее содержание. Например, несколько педагогических вузов могут стать соучредителями коммунального облака. Региональный орган управления образованием может инициировать создание коммунального облака для подчиненных ему школ. Таким образом, он, с одной стороны, освобождает их от необходимости инвестировать в создание собственных информационных инфраструктур, а с другой — создает условия для использования типовых управленческих решений и поддерживает региональный ЦОД. Итак, коммунальное облако — это вариант частного облака, которое выходит за рамки одной организации.

Публичное облако

Публичное облако (public cloud) — вычислительная инфраструктура, которая предназначена для свободного использования самым широким кругом пользователей, включая физических и юридических лиц. Публичным облаком могут порознь или совместно владеть или управлять (в том числе эксплуатировать его) коммерческие, научные и государственные организации. Обычно публичное облако находится под юрисдикцией его владельца — поставщика услуг.

Публичное облако — самое известное воплощение идеи вычислительного облака. Отсюда ошибочное мнение, что облака бывают только публичными.

Создание публичных облаков на глобальном уровне требует огромных инвестиций. Публич-

⁸ Иногда словосочетание “community cloud” также переводят как “общественное облако”.

ное облако обычно использует тысячи серверов, которые работают во многих ЦОД, развернутых в десятках мест по всему миру. Только гигантские транснациональные корпорации типа Microsoft, Amazon или Google имеют возможность создавать и развивать их как составную часть своего бизнеса. Используя публичное облако, потребители могут выбрать место, где собираются использовать необходимые им сервисы. Например, российская компания может развертывать свои приложения на серверах в Восточной Европе, а британская — выбрать сервер в Ирландии. При этом облачные сети доставки контента (CDN) могут автоматически кэшировать соответствующее содержание в ЦОД по всему миру, увеличивая тем самым возможности масштабирования приложений и предлагая наилучшие условия работы для конечных пользователей.

Публичные облака при всех своих преимуществах (простота использования, надежность, низкая стоимость услуг и т.п.) имеют и недостатки. Наиболее обсуждаемой сегодня является проблема ограничения доступа к персональным данным. Чтобы отвечать требованиям законодательства, организации не имеют права хранить и передавать эти данные в зоны вычислительной инфраструктуры, которые находятся вне их юрисдикции. Для решения этой и других подобных проблем предназначено гибридное облако.

Гибридное облако

Гибридное облако (hybrid cloud) — это комбинация из двух или более различных облачных инфраструктур (частных, коммунальных или публичных), каждая из которых остается уникальным объектом. При этом они связаны между собой согласованным набором стандартизованных технологий передачи данных и приложений, что позволяет, например, использовать ресурсы публичного облака для балансировки нагрузки между облаками или использовать CDN для беспрепятственного доступа учащихся к контенту, размещенному в публичном облаке, и т.п.

Гибридное облако сочетает в себе преимущества частного и публичного облаков. При этом оно позволяет исполнять закон, запрещающий хранить информацию о финансовых операциях клиентов, их персональные данные, истории болезни на серверах, которые расположены за пределами данной страны. Кроме того, гибридные облака позволяют своим клиентам при необходимости обеспечить большую гибкость, размещая одни приложения в частном облаке (на собственном оборудовании), а другие — в публичном облаке (например, при организации дистанционного обучения на международном уровне).

Наиболее известным у нас подходом к решению этой задачи является концепция Windows Azure AppFabric, которая используется для создания гибридных облаков по желанию клиентов.

Обновление школы и облачные вычисления

Традиционная массовая школа — сложно устроенное педагогическое производство, которое зародилось два с лишним века назад. В основе организации работы традиционной школы лежало использование самых современных для того времени информационных технологий. Сегодня эти технологии обычно называют старыми, традиционными или “бумажными”. Они сформировались в ходе промышленной революции вместе с книгопечатанием и помогли распространению грамотности. Учебник, школьные тетради, классный журнал, расписание занятий — зримые инструменты традиционной информационной технологии.

Последние несколько лет мы все чаще слышим о цифровой школе, где для обновления образовательного процесса широко используются ИТ. Поставщики вычислительной техники ратуют за переход на цифровые технологии, рекомендуют дать в руки каждому школьнику вместо бумажного — электронный учебник (ультрабук, планшет или ридер). Издатели готовятся доставлять цифровой образовательный контент через Интернет, а работники управления образованием и политики говорят о переходе к “цифровой школе”.

Школа цифрового века

К сожалению, сегодня “цифровая школа” скорее рекламный символ, чем прототип школы цифрового века. В реальности цифровые технологии практически без изменения воспроизводят процессы, которые идут в традиционной школе (рис. 12 на с. 33). А это не позволяет в необходимой мере использовать потенциал ИТ для повышения образовательных результатов и обновления школы. Школа цифрового века не ограничивается заменой бумажных носителей информации на цифровые. Главная цель — появление педагогических практик, использующих новые методы и формы учебной работы, которые расширяют возможности школьников: организовывать и направлять свое учение; повышать свою грамотность; осваивать и использовать практики исследовательской работы; строить ясную картину окружающего мира; в совершенстве владеть родным и иностранными языками; продуктивно коммуницировать и работать в группах; быть ответственным гражданином своей малой родины, страны и мира.

“Повышение результативности процессов учения происходит не в результате улучшения способов обучения, которыми пользуются учителя, а в результате того, что у учащихся появляется больше возможностей выстраивать свое знание”.

Сеймур Пейперт

Реальная информатизация образования, которая ведет к возникновению школы цифрового века, — это необратимый инновационный процесс, который сопряжен с изменением целей, содержания, методов и организационных форм учебно-воспитательной работы. Изменение целей, которые провозглашены в новых образовательных стандартах, требует обновления содержания и расширения спектра методов и организационных форм обучения: систематического использования учебных проектов, активной групповой и индивидуальной работы учащихся, дистанционных образовательных технологий и т.п. А это, в свою очередь, невозможно без обновления традиционной модели работы школы.

Обновление школы связано в первую очередь с расширением привычной рамки классно-урочной системы учебной работы: введение командной работы педагогов, использование различных способов группировки учащихся (в том числе разновозрастных групп переменного состава), переосмысление хронотопа (время-пространства) школы, инновационных способов учебной работы и оценивания [15]. Единицей планирования и управления в школе цифрового века становится не группа учащихся (класс), а каждый отдельный школьник. Цели учебной работы, программа и расписание занятий, подбор самих занятий проводятся исходя из нужд и интересов каждого конкретного школьника. Учителя школы цифрового века сосредотачиваются на использовании педагогических практик, которые активизируют учащихся, вовлекают их в осмысленную учебную работу. Чтобы поддерживать, организовать лично ориентированный учебный процесс, нужно не только обеспечить школьников компьютерами для перехода к работе по технологической модели 1:1⁹. Требуется предоставить им единую цифровую информационную среду (далее — инфосреда), которая позволяет надежно и оперативно решать весь комплекс задач управления коллективным, лично ориентированным образовательным процессом, в который вовлечены многие сотни участников.

Современная цифровая информационная среда должна представлять собой общедоступное (24 × 7 × 365), надежное и гибкое, хорошо структурированное пространство, где доступны все необходимые учебно-методические материалы, имеется возможность для коллективной и индивидуальной работы, накапливаются постоянно обновляемые данные о ходе, результатах и других элементах образовательного процесса. Обязательной составляющей такой среды является постоянно развивающийся комплекс приложений (прикладных программ), которые позволяют обеспечивать все информаци-

онные потребности участников учебно-воспитательного процесса на всех этапах их работы.

Помимо традиционных групповых и индивидуальных учебных занятий, такая среда должна помогать планировать и поддерживать проектную работу школьников, организацию перевернутого обучения¹⁰, использование дистанционных образовательных технологий, тренажеров и т.п. Инфосреда должна содержать (поддерживать использование) инструментов аутентичной оценки образовательных достижений школьников, понятные и признанные всеми участниками образовательного процесса процедуры формирующего и результирующего оценивания.

Среди главных требований к инфосреде: доступность (техническая и экономическая), простота (использования и сопровождения/эксплуатации), универсальность, техническая надежность, защита данных (от несанкционированного доступа и возможных ошибок пользователей), гибкость и простота настройки на решение вновь возникающих задач. По своим техническим характеристикам инфосреда школы (доступность, надежность, оперативность, количество пользователей и т.п.) сравнима с производственными и банковскими информационными системами. Однако разнообразие транзакций и объемы информации, которые используются в образовательном процессе, существенно выше. В результате создание школы цифрового века с использованием производственных информационных систем, которые решают задачи управления учебно-воспитательным процессом, оказывается невозможно.

Решения из облака

Много лет под информатизацией школы понимали оснащение ее компьютерами. Их доступность считалась главным показателем технического оснащения учебных заведений. Появление дешевых носимых устройств решает проблему. Стоимость современных нетбуков и планшетов постоянно снижается. В ближайшее время они станут доступны большинству семей. Это позволяет говорить о том, что переход массовой школы к технологической модели 1:1 становится реальным. Однако перехода к технологической модели 1:1 само по себе недостаточно. Кроме компьютеров нужна цифровая инфосреда, которая позволяет трансформировать учебно-воспитательный процесс. В отличие от транснациональных корпораций и банков общеобразовательные учреждения не имеют возможности заказывать информационные системы, содержать

¹⁰ Перевернутое обучение (flip teaching) — одна из форм смешанного обучения. Учащиеся самостоятельно знакомятся с новым содержанием до занятий (как правило, с помощью видеолекций), в ходе очных занятий с учителем разбирают и закрепляют усвоенное. Перевернутое обучение предоставляет больше возможностей для учебного диалога, гибкой реакции на индивидуальные вопросы или затруднения школьников [10].

⁹ Технологическая модель, при которой отношение количества учащихся к количеству компьютеров равно единице и каждый школьник имеет постоянный доступ к компьютеру (один ученик : один компьютер).

мощные вычислительные центры и дорогостоящий технический персонал. Это было и пока остается главным техническим препятствием на пути построения цифровой школы.

Некоторым управленцам кажется, что ее решает подключение школ к Интернету. Органы управления обязывают школы вести в сети цифровые классные журналы и дневники, которые представляют собой компьютерные аналоги традиционных документов школы. Однако это никак не сказывается на расширении традиционных методов организации образовательного процесса, не ведет к обновлению школы, не повышает результативности ее работы¹¹.

Начавшееся распространение облачных вычислений обещает качественный скачок в развитии инфосреды образования, дает реальную надежду на появление у каждой школы полноценной производственной информационной системы, которая сделает реальным ее переход к личностно ориентированной учебной работе. Это связано с выходом на рынок общедоступных промышленных решений, которые обладают необходимой доступностью, надежностью, экономичностью, простотой использования, универсальностью и возможностью настройки на решение специфических, в том числе вновь возникающих, задач.

Доступность. Современные облачные сервисы доступны через Интернет с помощью браузеров, которые работают практически на всех коммуникационных устройствах (компьютеры, планшеты, смартфоны и т.п.). Таким образом, школьники, их родители и педагоги могут иметь постоянный (24 часа в день, 7 дней в неделю, 365 дней в году) доступ к облачным услугам в школе и дома. Подобно тому, как строительство крупных ГЭС, линий электропередач и создание единой энергосистемы сделало электроснабжение общедоступным, развитие публичных облаков делает общедоступными практически любые вычислительные и информационные услуги.

Главное ограничение для работы с облаком сегодня связано с точками доступа в Интернет. Сегодня далеко не все школы располагают широкополосным доступом в сеть, который обеспечивает потребителям комфортную работу с интернет-сервисом на скорости порядка 10–100 Мбит/сек. Учитывая стремительное развитие цифровой связи (в нашей стране и в мире), можно ожидать, что это положение будет быстро меняться¹².

Надежность. Рассмотренные выше принципы построения облачных вычислений гарантируют надежность предоставления услуг, их безотказность. Зеркалирование информации в ЦОД, ее регулярное

резервное копирование, возможность автоматического перезапуска серверов делают облачные сервисы исключительно надежными. Высококвалифицированный персонал, который могут позволить себе крупные ЦОД, является еще одной гарантией их надежности. Современное серверное оборудование, которое используется для создания гибридных облаков, может обслуживаться удаленно и гарантирует достаточно высокую безотказность предоставления услуг.

Экономичность. Снижение затрат на инфраструктуру школы при переходе в облако связано с отказом от закупки собственных серверов, расходов на их содержание и обслуживание. Глобальный уровень (обслуживание миллионов пользователей) и объединение ресурсов, которое является одним из свойств облачных вычислений, позволяют провайдерам облачных услуг поддерживать достаточно низкие цены на предоставляемые сервисы. Одновременно с этим оплата по мере использования и самообслуживание позволяют потребителям экономно расходовать бюджет. Все это ведет к существенному снижению затрат на ИТ. Программное обеспечение, которое предоставляют облачные технологии, является аппаратно-независимым, что позволяет сократить расходы на приобретение и обслуживание.

Традиционная ИТ-инфраструктура школы фактически не загружена во время праздников и каникул (хотя школа и продолжает нести соответствующие расходы). Масштабирование, которое является одним из ключевых свойств облачных вычислений, позволяет существенно экономить при переходе к облачной инфосреде (рис. 12). Работники школы могут предусмотреть снижение количества используемых серверов при понижении и их увеличение при повышении нагрузки.

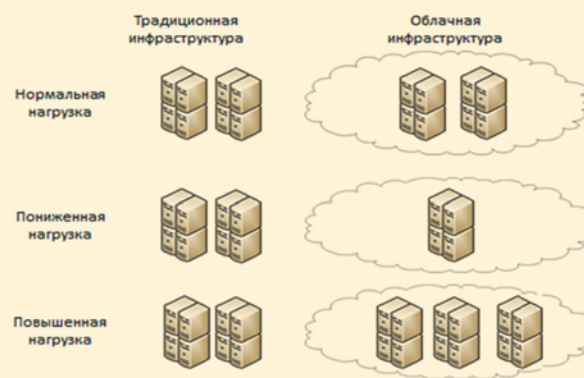


Рис. 12. Изменение использования ресурсов в облаке при изменении нагрузки

Исследования по экономике облака [4] показывают, что облачная платформа обходится потребителям в 40 раз дешевле, чем аналогичная по возможностям традиционная ИТ-инфраструктура. Есть данные о том, что облачный сервис Office 365 обеспечивает 321% возврата по инвестициям и окупается за два месяца использования в организациях

¹¹ Заметим, что это увеличивает нагрузку на работников образовательных учреждений, которые продолжают учет на бумаге и вынуждены переносить данные из бумажных документов на цифровые носители.

¹² Ожидается, что мобильные операторы будут обязаны предоставлять населению услуги доступа в каждом населенном пункте Российской Федерации с количеством жителей более 500 человек.

среднего или небольшого размера [4]. Вопрос о том, какую долю бюджета нашей школы могут занимать расходы на ИТ при использовании облака, пока не имеет ответа и требует специального исследования. Однако ясно, что эти расходы будут как минимум на порядок ниже, чем при развертывании собственной ИТ-инфраструктуры.

Простота использования. Одной из главных проблем, которая встает перед школой, развертывающей свою ИТ-инфраструктуру, является обслуживание вычислительной техники. Производственные ИТ-решения, которые необходимы для поддержания нормальной работы школьной инфосреды, требуют привлечения высококвалифицированных специалистов. Для них требуется слишком высокая заработная плата, а их количество на рынке труда весьма ограничено. Использование облачных сервисов решает эту проблему. Провайдеры облачных услуг берут на себя решение технических проблем, оставляя за работниками школы лишь свойственные им пользовательские функции. Обновить версию платформы или установить новую программу можно за несколько минут. Подобно тому, как современный интерфейс позволяет неспециалисту свободно использовать компьютер для решения практических задач, естественный и доступный интерфейс облачных сервисов позволяет работникам школы обращаться к необходимым услугам, оставаясь на уровне компьютерной грамотности пользователя. А если у пользователя возникают трудности с установкой или использованием программных средств, облачный сервис предлагает им ИТ-поддержку онлайн.

Универсальность. Важное достоинство облачных технологий в том, что они предоставляют организациям с ограниченным бюджетом доступ к возможностям, которыми раньше могли воспользоваться только крупные компании с большим бюджетом.

Учащимся и педагогам нужен доступ к широкому диапазону различных данных и услуг, которые являются собственностью самих участников образовательного процесса, образовательного учреждения, различных поставщиков образовательного контента и услуг, других лиц, сообществ и/или организаций. Работа в облаке позволяет использовать универсальные решения для обеспечения доступа ко всей совокупности таких данных, применять стандартные процедуры взаимных расчетов за предоставление информационных и вычислительных услуг, которые в полной мере обеспечивают интересы и поставщиков, и потребителей. Зарегистрировавшись в облачной инфосреде образовательного учреждения под своим уникальным именем, учащиеся и педагоги получают защищенный доступ и ко всей совокупности необходимых им материалов (личных, групповых, коллективных или покупных). Кроме этого, облачные вычисления делают нормой “виртуальное сотрудничество”. Они позволяют устанавливать связь между любыми точками с доступом к сети, упрощают виртуальные встречи между участника-

ми образовательного процесса, которые работают удаленно, зарубежными партнерами и т.п. Все это создает благоприятные условия для работы поставщиков образовательного контента услуг и облегчает работу образовательных учреждений.

Настройки на решение специфических задач. Облачная PaaS позволяет разработчикам использовать современные средства создания приложения, облегчает и сокращает время на их внедрение. Новые решения становятся доступны всем заинтересованным пользователям, что облегчает знакомство с новыми педагогическими решениями и их освоение.

Кроме этого, облако предоставляет работникам школы новейшие бизнес-инструменты для работы с данными. Накопление данных о ходе учебно-воспитательной работы в облачных хранилищах позволяет использовать для их обработки и анализа промышленные системы типа Business Intelligence [14]. Эти инструменты позволяют использовать собираемые данные для расширения границ педагогической интуиции, помогают анализировать реальное состояние дел, открывать для себя новые связи и отношения между различными наборами данных. Встроенные средства ProClarity помогают осуществлять углубленный анализ и визуализировать результаты, легко готовить отчеты с наглядным представлением полученных материалов, обмениваться ими через Интернет, распространять информацию среди всех заинтересованных лиц, централизованно управлять отчетами и их хранением.

Таким образом, облачные вычисления помогают педагогической науке и практике перейти от качественных к количественным суждениям, использовать их для проведения педагогических исследований и в ходе решения повседневных задач управления личностно ориентированным образовательным процессом.

Завтра уже наступило

Сегодня облачные вычисления перестали быть образом из будущего. Например, компания Comdi использует их для организации интернет-трансляций тысячам участников, что позволяет не только обеспечить высокое качество предоставляемых услуг, но и дает существенную экономию¹³.

Корпорация Microsoft начала широкое распространение облачного сервиса Office 365 в образовательных учреждениях нашей страны. Он включает в себя облачную версию Microsoft Office (Outlook, Word, Excel, PowerPoint, OneNote Web Apps), а также инструменты для совместной работы (Lync Online, SharePoint Online и Exchange Online). Внедрение этой разработки в Финансовом университете [8] позволило расширить возможности мобильной

¹³ См. описание опыта трансляций заседаний международной конференции, проходившей в подмосковном доме отдыха “Покровское” [3], где расходы на трансляцию за два дня работы конференции составили \$120.

работы, увеличились надежность и безопасность системы. Университет получил современные средства интегрированных коммуникаций и оптимизировал затраты на ИТ-инфраструктуру при одновременном повышении доступности и надежности сервисов. Студентам и сотрудникам стало проще работать совместно с коллегами, получать доступ к ресурсам и сервисам университета из любого места и в любое время, используя различные устройства (компьютеры, планшеты, смартфоны). Количество пользователей Office 365 в университете составляет сегодня 95 000 человек, включая студентов и работников вуза, которые территориально разобщены. Теперь, даже если в работе аппаратного обеспечения возникают проблемы, облачные сервисы продолжают работать без сбоев. Интеграция Lync Online и Outlook со службой каталогов Active Directory позволила создать единый каталог учетных записей пользователей, оптимизировать проведение видеоконференций. Использование дистанционных совещаний позволяет сотрудникам более продуктивно организовывать рабочее время. SharePoint Online открыл новые возможности для создания единых стандартизованных учебно-методических материалов.

По мнению проректора по информатизации и коммуникации РЭУ им. Г.В. Плеханова Ольги Гришиной [6], внедрение Office 365 позволяет с минимальными затратами создать гибкую ИТ-инфраструктуру корпоративного уровня, которая отвечает требованиям конкретного образовательного учреждения. РЭУ им. Г.В. Плеханова имеет большое количество филиалов по всей России. Поэтому обеспечение эффективных коммуникаций и удаленного доступа к образовательному контенту — одна из основных задач, стоящих перед вузом. Office 365 дает студентам, преподавателям и администрации надежные средства общения и совместной работы: почтовый сервис, порталную систему, а также аудио- и видеоконференции. В результате студенты из филиалов могут консультироваться с преподавателями из столицы в режиме онлайн, обмениваться документами и совместно работать над ними. В будущем облачный сервис заменит традиционную телефонию и другие средства коммуникации, которые используют в РЭУ им. Г.В. Плеханова. Office 365 становится технологическим ресурсом преподавателей и сотрудников университета, которые могут самостоятельно создавать собственные образовательные сервисы.

Облачный сервис Office 365 стал платформой для информационной системы “Открытое образование”, которая помогает автоматизации образовательного процесса в российских школах [5]. С ее помощью можно вести удаленное обучение, поддерживать оперативную связь с родителями, вести электронный учет и т.п. Система “Открытое образование” поддерживает интерактивное удаленное обучение, почтовый сервис, а также портал учебного заведения, который содержит архивы учебных материалов (в том числе видеозаписи уроков), школьное расписание со

списками домашних заданий к каждому уроку, данные о посещаемости и успеваемости, электронные журналы, дневники и т.п. Большая часть функций системы (включая удаленное обучение) доступна с мобильных устройств на современных платформах — Windows Phone, iOS, Android. Это позволяет школьникам и учителям пользоваться системой, находясь вдали от домашнего или рабочего компьютера.

По мнению Ирины Безугловой, директора по развитию компании Prof IT, разработавшей систему “Открытое образование”, Office 365 обеспечил их разработке простоту внедрения. Система не требует администрирования на уровне школы, что особенно важно тем, кто не имеет штатной ИТ-поддержки на местах.

Перечисленные примеры показывают, что облачные сервисы начали внедряться в работу образовательных учреждений в нашей стране. Одновременно с этим разработчики облачных платформ продолжают создавать новые виды услуг, которые могут оказать заметное влияние на организацию образовательного процесса.

Новые разработки в России

Итак, мы обсудили основные идеи, лежащие в основе облачных сервисов, рассмотрели их основные свойства, перечислили несколько примеров их использования для решения задач в образовании. Можно заключить, что сегодня работникам управления школой пора перенести свое внимание с оснащения школ компьютерами на формирование ее цифровой информационной среды как составной части инновационной работы по переходу к новым образовательным стандартам. Теперь уже не требуется создавать в каждой школе собственный центр обработки данных. Облачные вычисления позволяют решать эту задачу по-новому, повышая качество и снижая стоимость соответствующих услуг.

Сегодня все больше и больше провайдеров предлагают пакетные облачные решения, которые включают в себя технические средства (оконечные устройства и т.п.), программные приложения, системы управления учебным процессом и т.п. Требуются специальные исследования, которые позволят сопоставить их практичность и цену для отдельных учебных заведений и их групп в реальных условиях. Всех, кто занимается информатизацией школы, ждет очень большая работа.

Сегодня идет поиск новой, пригодной для массового использования модели школы с индивидуализированной системой учебной работы, которая гарантирует достижение каждым учащимся максимально высоких результатов учебно-воспитательной работы. Для того чтобы перейти от отдельных экспериментов к воплощению такой модели в жизнь, требуется опережающая разработка дидактических и организационно-педагогических вопросов построения новой образовательной среды, которая в полной мере использует возможности в ИКТ-насыщенной

образовательной среде и обеспечивает выполнение требований новых ФГОС. Среди этих вопросов:

- внутришкольные ориентиры (нормы) образовательных достижений школьников,
- типовые инструменты формирующего и констатирующего (итогового) оценивания, которые должны действовать на уровне школы,
- новые регламенты работы школы и педагогов в условиях индивидуализированной (лично ориентированной) образовательной работы.

Нормы и регламенты, которые определяют сегодня работу массовой школы, в своей основе опираются на установки и требования, которые были сформулированы десятилетия назад в Постановлении ЦК ВКП(б) от 25.08.1932, где сказано: "...основной формой организации учебной работы в начальной и средней школе должен являться урок с данной группой учащихся со строго определенным расписанием занятий и твердым составом учащихся". Индивидуализированная система учебной работы требует качественного пересмотра традиционных нормативных документов, которые ориентируют работу школы на выполнение учебной программы (проведение заданного количества учебных занятий в классах, поурочное планирование работы учителей, расчет классов-комплектов и т.п.), а не на достижение каждым школьником заданных образовательных результатов. Сегодня они требуют пересмотра.

Ключевыми для перехода к индивидуализированной модели организации образовательного процесса являются шесть составляющих.

Внутришкольные нормативы результативности, цели учебной работы, хорошо понятные самим учащимся, их родителям и педагогам. Будучи операционально¹⁴ представлены, хорошо измеряемы и соотнесены с образовательными областями и ступенями работы школы на каждом этапе этой работы (вплоть до учебного модуля и/или темы), они создают основу для надежного промежуточного/формирующего и итогового/констатирующего оценивания. Их наличие позволяет начать движение к индивидуализации образовательного процесса, сменить управленческую парадигму со стратегии управления образовательным процессом (часы на изучение отдельных образовательных дисциплин, типовые учебные планы и т.п.) к стратегии управления образовательными результатами, на деле решать задачу индивидуализации образовательного процесса. Без хорошо (операционально) определенных образовательных результатов невозможно предоставить каждому учащемуся свой собственный темп движения по материалу, выявить и поддержать талантливых детей в условиях школы.

¹⁴ Операционально представленными здесь называются такие образовательные результаты (цели учебной работы), которые выражены через соответствующие операции (иногда говорят — действия). Каждый, успешно прошедший обучение, должен быть способен продемонстрировать выполнение этих операций.

Инструменты аутентичной оценки уровней образовательных достижений школьников, которые обеспечивают объективные, понятные и признанные всеми участниками образовательного процесса практически действенные процедуры формирующего и результирующего оценивания.

Мониторинг, систематическое оценивание индивидуальных образовательных достижений, который добросовестно проводится и поддержан всеми участниками образовательного процесса. Такой мониторинг обеспечивает, в свою очередь, формирование личностной позиции учащегося как субъекта образовательного процесса, ведет к обновлению действующих сегодня приемов учебно-воспитательной работы, функциональных обязанностей работников школы, включая отказ от привычной поурочной оплаты труда педагогов. Результаты мониторинга:

- доступны всем заинтересованным сторонам и
- служат действенной основой для совершенствования повседневной учебной работы, обновления используемых педагогических практик и учебно-методических материалов.

Вариативные учебно-методические материалы, цифровые образовательные ресурсы и новые педагогические практики, которые поддерживают современные высокорезультативные педагогические практики и помогают участникам учебно-воспитательного процесса достигать желаемых образовательных результатов на выбранных ими уровнях с использованием различных форм учебной работы (фронтальной, групповой, индивидуальной).

Профессиональное развитие и методическая поддержка педагогов, которые организуют и обеспечивают учебную работу школьников по достижению необходимых образовательных результатов. Подобно тому, как школа ориентируется на индивидуализированную работу учащихся, методическая поддержка должна отвечать на индивидуальные запросы педагогов, помогая им осваивать новые результативные педагогические практики, которые обеспечивают успешную учебную работу школьников.

Общедоступная, надежная и экономически оправданная ИКТ-насыщенная образовательная среда (школьный портал и мобильные рабочие места участников образовательного процесса), которая реально поддерживает и упрощает работу по осуществлению лично ориентированного образовательного процесса: управление индивидуальными образовательными траекториями, организацию индивидуальной и групповой работы учащихся (в том числе с цифровыми образовательными ресурсами), постоянный автоматизированный мониторинг хода и результатов образовательного процесса.

Чтобы перечисленные выше составляющие слились в устойчивой практике работы школы, школа должна стать "обучающейся организацией"¹⁵. Здесь нужна эффективная система последовательного

¹⁵ Об обучающихся организациях см., например, работу [17].

выявления и освоения необходимых нововведений, которая:

- переводит образовательные учреждения в инновационный цикл развития (формирование видения, изучение опыта, вовлечение, планирование, реализация, рефлексия и оценка),
- широко открыта для использования лучших отечественных и зарубежных разработок, опирается на международно признанную экспертизу,
- включает в себя систематическую оценку доказательной результативности осуществляемых нововведений,
- открыто демонстрирует все промежуточные и итоговые результаты выполняемых работ общественности, морально и материально поддерживает исполнителей за результативное, качественное и своевременное выполнение работ.

В основе такой системы лежит ясное видение смысла, содержания и этапов проводимой работы всеми учащимися, их родителями, педагогами и другими членами местного образовательного сообщества, без всесторонней поддержки которых успешное выполнение работ по трансформации школы невозможно. Она связана в том числе с решением задач профессионального роста педагогов, которые решают свои задачи в новой образовательной среде¹⁶.

В июне 2013 года, в рамках Петербургского международного экономического форума-2013 Ассоциация “Школа Сколково” и компания Microsoft в России подписали соглашение о сотрудничестве по развитию информационной образовательной среды в рамках программы *School of Digital Age* (Школа Цифрового Века¹⁷), или SODA. Ассоциация и Майкрософт подчеркнули необходимость разработки и проверки на практике действенности современных информационных сервисов для поддержки образовательного процесса, которые обеспечивают переход к индивидуализированной системе обучения.

Проект SODA предусматривает разработку педагогических, организационных и технических решений, обеспечивающих повышение образовательных результатов на основе индивидуализации образовательного процесса. Его главная цель — разработка и апробация новой модели работы общеобразовательных учреждений, которая обеспечивает:

- достижение высоких образовательных результатов (в том числе новых) каждым школьником;
- предоставление каждому учащемуся равного доступа к качественному образованию;

¹⁶ Соответствующие требования зафиксированы в новых рекомендациях ЮНЕСКО по ИКТ-компетентности учителей [9].

¹⁷ **От редакции.** Название программы свидетельствует о некорректном заимствовании названия широкомасштабного проекта “Школа цифрового века”, реализуемого Издательским домом “Первое сентября” с 2011 года. Название зарегистрировано Роскомнадзором (эл. № ФС77-53232).

- безболезненное обновление содержания, методов и организационных форм работы;
- интеграцию учебных дисциплин (межпредметные связи) и взаимодействие между преподавателями различных предметных областей;
- управление качеством учебно-воспитательной работы в школе.

В рамках проекта предполагается разработать и внедрить в работу общеобразовательных учреждений:

- модульно-зачетную систему обучения, которая использует модульные учебно-методические материалы, обновленную базу и новые регламенты работы школы;
- сформировать требования к обновленным правам и обязанностям участников образовательного процесса;
- выстроить ИКТ-насыщенную образовательную среду, которая основана на облачных сервисах и технологической модели “1:1”.

Ассоциация намерена в сотрудничестве с партнерами способствовать:

- предоставлению работникам образовательных учреждений поддержки и помощи в разработке новой модели школы и использовании ИКТ в образовательном процессе;
- разработке и апробации нового поколения ИКТ средств поддержки профессиональной деятельности педагога, которые создают условия для перехода на индивидуализированную систему обучения;
- разработке и апробации учебно-методических комплексов и рекомендаций по их интеграции в рамках школ Ассоциации;
- профессиональному развитию педагогов в части использования ИКТ-сервисов для индивидуализации образовательного процесса;
- проведению конференций и встреч работников образования, организованных с целью обмена опытом, знаниями и новыми методами и формами образовательной работы.

Майкрософт намерен:

- прилагать усилия по выработке требований к архитектуре ИКТ-насыщенной среды школы с индивидуализированным обучением. Целью этих усилий будет обеспечение безопасного использования ИКТ в образовательном процессе, развитие инновационных методов и форм обучения, повышение эффективности работы педагогов и повышение качества образования;
- поддерживать использование ИКТ-сервисов индивидуализации учебного процесса учителями школ Ассоциации и партнеров путем разработки, распространения, освоения и поддержки новых инструментов и технологий, а также предоставления дополнительных материалов и консультаций с российскими и международными экспертами;
- способствовать развитию взаимодействия с использованием разрабатываемых и существующих образовательных интернет-порталов, способ-

ствовать развитию онлайн-сообществ педагогов-исследователей, инновационных работников школ Ассоциации. С помощью международного портала для учителей “Партнерство в образовании” Майкрософт намерен содействовать в предоставлении доступа к:

- образовательно-информационному контенту и источникам информации онлайн;
- средам обмена опытом;
- современным методикам с использованием ИКТ;
- онлайн-инструментам для педагогов-исследователей и практиков с целью улучшения методов преподавания, повышения эффективности решения исследовательских и проектных задач;
- коммуникационным сервисам для общения между педагогами-исследователями, работниками школ, а также родителями, учащимися;
- сервисам и виртуальным средствам методической поддержки педагогов-исследователей и других работников образования, включая ИКТ-специалистов.

В следующей публикации мы рассмотрим различные решения, которые предлагает проект SODA.

Литература

1. Авдеева С.М., Уваров А.Ю. Российская школа на пути к информационному обществу: проект “Информатизация системы образования” // Вопросы образования, 2005, № 3.
2. Асмолов А.Г., Семенов А.Л., Уваров А.Ю. Российская школа и новые информационные технологии: взгляд в следующее десятилетие. М.: Некспринт, 2010.
3. Облачные сервисы. Взгляд из России / Е.Гребнева (ред.). М.: СNews, 2011.
4. Облачный офис сегодня — безоблачное будущее завтра: Office 365 в ИГУМО и ИТ, 10.03.2013 (<http://slon.ru/office365/articles/igumo/>).
5. Office 365 помогает модернизировать процесс обучения в российских школах. // MS, Пресс-центр, 27.05.2013 (http://www.microsoft.com/ru-ru/news/archive/2013/05/New_education_level_with_Office_365.aspx).
6. РЭУ им. Г.В. Плеханова создает единую информационную среду для студентов и преподавателей. // MS, Пресс-центр, 24.06.2013 (http://www.microsoft.com/ru-ru/news/archive/2013/06/informacionnaja_sreda_dlja_studentov_i_prepodavatelej.aspx).
7. Совместная программа Ассоциации “Школа Сколково” и компании Microsoft позволит реализовать в российских школах новую образовательную концепцию. Ассоциация “Школа Сколково”, 24.06.2013 (http://community.sk.ru/foundation/shkolkovo/b/mnews/archive/2013/06/21/podpisano-soglasenie-o-namereniyah-mezhdu-associaciy-_2200_shkola-skolkovo_2200_-i-kompaniy-maykrosoft.aspx).
8. Специалисты Финансового университета остановили свой выбор на Office 365. // MS, Примеры внедрения, 04.06.2013 (http://www.microsoft.com/rus/casestudies/Case_Study_Detail.aspx?casestudyid=710000002599).
9. Уваров А.Ю. Структура ИКТ-компетентности учителей и требования к их подготовке: рекомендации ЮНЕСКО // Информатика и образование, № 1, 2013.
10. Bergmann J., Sams A. Flip your classroom: reach every student in every class every day. ISTE/ASCD, 2012.
11. Ferenstein G. The Flipped Classroom Boosts Grades 5% // District Administration, September 18th, 2013 (<http://techcrunch.com/2013/09/18/the-flipped-classroom-boosts-grades-5-why-thats-as-big-as-we-can-expect/>).
12. Gupta A. Demystifying the Cloud. Janakiram & Associates, 2012 (<http://www.getcloudready.com/ebooks>).
13. Mell P., Grance T. The NIST Definition of Cloud Computing. NIST Special Publication 800-145, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, September 2011 (<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>).
14. Microsoft Business Intelligence Office 365, September 2011 (http://blogs.msdn.com/b/microsoft_business_intelligence1/archive/2013/09/17/microsoft-power-bi-for-office-365-new-samples-video-launched-view-now.aspx).
15. OECD. Innovative Learning Environments, Educational Research and Innovation, OECD Publishing, 2013 (<http://dx.doi.org/10.1787/9789264203488-en>).
16. Pierce D. FCC Votes to Revisit e-Rate // eSchool News. No 7, Vol. 16, 2013 (<http://www.eschoolnews.com>).
17. Schon D. Beyond the Stable State. Public and Private Learning in a Changing Society. Harmondsworth: Penguin, 1973.
18. Winkler K. What Amazon Shows Us about the Future of Online Learning // EdCetera: straight talk on ed-tech, Oct 4, 2013 (<http://edcetera.rafter.com/what-amazon-shows-us-about-the-future-of-online-learning/>).

MimioClassroom

Весь спектр интерактивного оборудования
для школы и детского сада



Интерактивная приставка
MimioBoard



Интерактивный
короткофокусный проектор
MimioProjector

Интерактивная доска
MimioBoard

ME78 — диагональ 78''
ME87 — диагональ 87''



Документ-камера
MimioView



Автоматическое конспектирование урока
MimioCapture



Система автоматизации тестирования
MimioVote

Все оборудование поставляется с ПО MimioStudio, позволяющим создавать интерактивные уроки, проводить тестирования и голосования. MimioStudio является центром интеграции всего интерактивного оборудования класса и позволяет использовать оборудование иных производителей. Совместимо с Microsoft Windows, Linux и MacOS.

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



Схема Горнера

Д.М. Златопольский,
Москва

► В математике “схемой Горнера¹” (или “правилом Горнера”, “методом Горнера”) называют алгоритм вычисления значения многочлена $a_n x^n + a_{n-1} x^{n-1} + \dots + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$, записанного в виде суммы одночленов при заданном значении переменной x . Этот метод позволяет решить ряд других задач.

В информатике понятие “схема Горнера” используется при переводе чисел из недесятичных систем в десятичную.

Пусть дано число 2345_p , записанное в системе счисления с основанием p .

Если вспомнить так называемую “развернутую форму” записи чисел, то можно сказать, что десятичный эквивалент заданного числа будет равен:

$$2 \cdot p^3 + 3 \cdot p^2 + 4 \cdot p^1 + 5 \cdot p^0.$$

Указанное выражение можно записать по-другому:

$$\begin{aligned} & 2 \cdot p \cdot p \cdot p + 3 \cdot p \cdot p + 4 \cdot p + 5 = \\ & = (2 \cdot p \cdot p + 3 \cdot p + 4)p + 5 = \\ & = ((2 \cdot p + 3)p + 4)p + 5 = \\ & = (((0 \cdot p + 2)p + 3)p + 4)p + 5. \end{aligned}$$

Анализ показывает, что схема вычислений следующая: первая цифра заданного числа умножается на основание p исходной системы счисления, к произведению прибавляется вторая цифра, сумма умножается на основание, к произведению прибавляется третья цифра — и т.д. до прибавления последней цифры.

Эту схему можно записать в виде такого алгоритма — для каждой цифры заданного числа, начиная с первой:

- 1) “старое” значение рассчитываемой (искомой) величины умножается на основание p ;
- 2) к произведению прибавляется данная цифра;
- 3) полученная сумма используется как “старое” значение для следующей цифры.

При этом для первой цифры “старое” значение искомой величины равно нулю.

В программе, реализующей перевод p -ичного числа в десятичную систему по схеме Горнера, используем следующие переменные величины:

- *число* — заданное число;
- p — основание системы счисления, в которой оно записано;
- n — искомое десятичное число;
- *цифры* — массив с данными целого типа, в котором будут храниться цифры заданного числа. Размер этого массива

¹ Горнер Вильямс Джордж (1786–1837) — английский математик.

ва следует определить с учетом возможной длины записи числа.

Примем, что заданное число представляет собой последовательность символов-цифр (является строковой величиной).

```

алг Перевод_чисел_в_десятичную_систему_
    по_схеме_Горнера
нач лит число, цел n, p, i, лог успех, цел
таб цифры[1:20]
    |Ввод исходных данных
вывод нс, "Введите исходное число "
вывод число
вывод нс, "Введите основание системы
    счисления, в которой оно записано"
ввод p
    |Заполняем массив цифры
нц для i от 1 до длин(число)
    |Определяем очередную i-ю цифру
    |и записываем ее в массив
    цифры[i] := лит_в_цел(число[i], успех)
кц
    |Реализуем схему Горнера
    n := 0
нц для i от 1 до длин(число)
    n := n * p + цифры[i]
кц
    |Выводим результат
вывод нс, "Соответствующее число
    в десятичной системе счисления: ", n
кон

```

Язык Паскаль

```

{Перевод чисел в десятичную систему
по схеме Горнера}
Var chislo: string; n, uspeh: integer;
    p, i: byte;
    zifri: array [1..20] of integer;
BEGIN
{Ввод исходных данных}
write('Введите исходное число ');
readln(chislo);
write('Введите основание системы
счисления, в которой оно записано ');
readln(p);
{Заполняем массив zifri}
for i := 1 to length(chislo) do
    {Определяем очередную i-ю цифру
и записываем ее в массив}
    Val(chislo[i], zifri[i], uspeh);
{Реализуем схему Горнера}
n := 0;
for i := 1 to length(chislo) do
    n := n * p + zifri[i];
{Выводим результат}
writeln('Соответствующее число
в десятичной системе счисления: ', n)
END.

```

Задание для самостоятельной работы учащихся

1. Разработайте варианты программы для перевода целого p -ичного числа в десятичную систему счисления по схеме Горнера:

- 1) в котором массив *цифры* не используется;
- 2) в котором исходное число задано как целое десятичное, например, троичное 2021₃ как 2021.

Использование при расчетах схемы Горнера минимизирует количество арифметических действий и позволяет избежать возведения в степень.

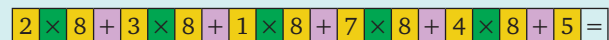
Рассмотрим перевод восьмеричного числа 231745₈. Запишем его в развернутой форме и преобразуем полученную сумму к эквивалентной скобочной форме:

$$231745_8 = 2 \cdot 8^5 + 3 \cdot 8^4 + 1 \cdot 8^3 + 7 \cdot 8^2 + 4 \cdot 8 + 5 =$$

$$= (((((2 \cdot 8 + 3) \cdot 8 + 1) \cdot 8 + 7) \cdot 8 + 4) \cdot 8 + 5 =$$

$$78\ 821_{10}.$$

Схема Горнера позволяет использовать для перевода чисел в десятичную систему "простой" (не инженерный/научный) калькулятор. При расчетах на калькуляторе нужно последовательно слева направо выполнять умножения и сложения. Порядок нажатия клавиш на калькуляторе для рассматриваемого примера будет таким:



Всего будет выполнено пять умножений и пять сложений (при этом будет также 11 нажатий клавиш с цифрами и одно нажатие на клавишу со знаком равенства).

В общем виде алгебраический многочлен n -й степени и его преобразование к скобочной форме выглядят так:

$$a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0 =$$

$$= (((((a_n x + a_{n-1})x + a_{n-2})x + \dots + a_2)x + a_1)x + a_0.$$

Из этой формулы следует, что алгебраический многочлен n -й степени, а следовательно и десятичный эквивалент p -ичного числа, можно вычислить за n операций умножения и n операций сложения. Это оптимальный способ вычисления.

Задания для самостоятельной работы учащихся

2. Определите количество умножений, количество сложений, количество нажатий клавиш с цифрами при переводе восьмеричного числа 231745₈ в десятичную систему без использования схемы Горнера на калькуляторе:

- 1) без функции возведения в степень;
- 2) с такой функцией.

В обоих случаях учтите также нажатия клавиш <M+> (сохранение и суммирование частных произведений в памяти) и <MR> (вывод значения из памяти).

3. Переведите в десятичную систему счисления описанным способом с использованием схемы Горнера следующие числа:

- а) 10101₂;
- б) 2121₃.

4. В ячейки B2:I2 вводятся цифры заданного целого двоичного числа. Подготовьте лист для перевода этого числа в десятичную систему счисления без использования схемы Горнера:

	A	B	C	D	E	F	G	H	I
1	Введите двоичные цифры числа:	0	1	1					
2	Соответствующее десятичное число равно:								

Указания по выполнению

1. Вспомогательные расчеты проведите вне зоны видимости листа.

2. Используйте функцию СУММПРОИЗВ².

5. В ячейки B2:I2 вводятся цифры заданного целого двоичного числа. Подготовьте лист для перевода этого числа в десятичную систему счисления с использованием схемы Горнера:

	A	B	C	D	E	F	G	H	I
1	Введите двоичные цифры числа:	0	1	1					
2	Соответствующее десятичное число равно:								

6. Подготовьте лист для перевода в десятичную систему счисления с использованием схемы Горнера целого числа, заданного в системе счисления с основанием p ($2 \leq p \leq 9$) в виде набора цифр:

	A	B	C	D	E	F	G	H	I
1	Введите цифры числа:	0	1	1					
2	Введите основание системы p :								
3	Соответствующее десятичное число равно:								

Схему Горнера можно применить и для перевода в десятичную систему дробных чисел. Напомним, что один из алгоритмов перевода правильных конечных p -ичных дробей в десятичную систему состоит в следующем [1].

1. Пронумеровать все цифры дробной части слева направо, начиная с 1.

2. Каждую цифру дробной части умножить на p^{-k} , где k — номер цифры.

3. Результаты сложить.

Примечание. Все арифметические действия проводятся в десятичной системе.

Если значения p^{-k} назвать “весомостью k -го дробного разряда”, то можно также сказать, что искомым результатом равен сумме произведений каждой цифры дробной части заданной дроби на весомость ее разряда.

Пример 1. Переведем число $0,2012_3$ в десятичную систему счисления.

Решение

$$0,2012_3 = 2 \cdot 3^{-1} + 1 \cdot 3^{-2} + 2 \cdot 3^{-3} = 0,728395061728_{10}.$$

Когда $p > 9$, каждая цифра дробной части предварительно переводится в десятичную систему.

Пример 2. Переведем число $0,А0В2_{16}$ в десятичную систему счисления.

Решение

$$0,А0В2_{16} = 0, [10_{10}] [0_{10}] [11_{10}] [2_{10}] = 10 \cdot 16^{-1} + 11 \cdot 16^{-2} + 2 \cdot 16^{-3} = 0,627716064453_{10}.$$

Для того чтобы получить схему Горнера для вычисления правильной p -ичной дроби $0,a_1a_2\dots a_k$, выпишем в развернутой форме цифры дроби в обратном порядке с учетом весомости соответствующих разрядов:

² При изучении электронной таблицы Oracle OpenOffice.org Calc здесь и далее следует использовать соответствующие функции.

$$\begin{aligned} & a_k p^{-k} + a_{k-1} p^{-k+1} + \dots + a_1 p^{-1} = \\ & (a_k p^{-k+1} + a_{k-1} p^{-k+2} + \dots + a_1) p^{-1} = \\ & ((a_k p^{-k+2} + a_{k-1} p^{-k+3} + \dots + a_2) p^{-1}) p^{-1} = \dots \\ & = (\dots((a_k p^{-1} + a_{k-1}) p^{-1} + a_{k-2}) p^{-1} + a_{k-3}) p^{-1} + \dots + a_1) p^{-1}. \end{aligned}$$

Можно сказать, что схема Горнера для дробей “работает” так: последняя цифра p -ичной дроби делится на p , к ней прибавляется предпоследняя цифра, сумма делится на p , к ней прибавляется предпредпоследняя цифра — и т.д. до первой цифры дроби, после чего сумма также делится на p .

Обратим внимание, что описанный механизм работы несколько отличается от работы схемы в случае перевода целых чисел (см. выше).

Покажем использование схемы Горнера на примере двоичного числа $0,1101_2$.

Запишем число в развернутой форме и выполним тождественные преобразования, приводящие выражение к скобочной форме:

$$\begin{aligned} 0,1101_2 &= 1 \cdot 2^{-4} + 0 \cdot 2^{-3} + 1 \cdot 2^{-2} + 1 \cdot 2^{-1} = \\ &= (1 \cdot 2^{-3} + 0 \cdot 2^{-2} + 1 \cdot 2^{-1} + 1) \cdot 2^{-1} = \\ &= ((1 \cdot 2^{-2} + 0 \cdot 2^{-1} + 1) \cdot 2^{-1} + 1) \cdot 2^{-1} = \\ &= (((1 \cdot 2^{-1} + 0) \cdot 2^{-1} + 1) \cdot 2^{-1} + 1) \cdot 2^{-1} = \\ &= (((1 : 2 + 0) : 2 + 1) : 2 + 1) : 2 = \\ &= \left(\frac{1}{2} \cdot \frac{1}{2} + 1\right) : 2 + 1 : 2 = (1,25 : 2 + 1) : 2 = \\ &= (0,625 + 1) : 2 = 0,8125_{10}. \end{aligned}$$

Полученное выражение $((1 \cdot 2^{-1} + 0) \cdot 2^{-1} + 1) \cdot 2^{-1} + 1) \cdot 2^{-1}$, эквивалентное следующему $((1 / 2 + 0) / 2 + 1) / 2 + 1) / 2$, может быть рассчитано на калькуляторе. Клавиши нажимаются в таком порядке:

$$1 \ / \ 2 \ + \ 0 \ / \ 2 \ + \ 1 \ / \ 2 \ + \ 1 \ / \ 2 \ =$$

Всего будет выполнено четыре операции деления и три операции сложения (при этом будет также семь нажатий клавиш с цифрами и одно нажатие на клавишу со знаком равенства).

Прибавление нуля (в общем случае — нулей) можно не выполнять, тогда число операций сократится.

Задания для самостоятельной работы учащихся

7. Определите количество умножений, количество сложений, количество нажатий клавиш с цифрами при переводе двоичного числа $0,1101_2$ в десятичную систему без использования схемы Горнера на калькуляторе:

- 1) без функции возведения в степень;
- 2) с такой функцией.

В обоих случаях учтите также нажатия клавиш $\langle M+ \rangle$ (сохранение и суммирование частных произведений в памяти) и $\langle MR \rangle$ (вывод значения из памяти).

8. Переведите в десятичную систему счисления описанным способом с использованием схемы Горнера следующие числа:

- а) $0,10101_2$; б) $0,2121_3$.

9. Подготовьте лист электронной таблицы для перевода p -ичной дроби, заданной в виде отдельных цифр дробной части, в десятичную систему счисления без использования схемы Горнера:

	A	B	C	D	E	F	G	H	I
1	Введите цифры дроби	0	1	1					
2	Введите основание системы счисления p	2							
3	Соответствующая десятичная дробь равна	0,375							

Указания по выполнению

1. Заданные цифры должны вводиться в столбцы В–I (не более восьми цифр).

2. Вспомогательные расчеты проведите вне зоны видимости листа.

3. Используйте функцию СУММПРОИЗВ.

10. Подготовьте лист электронной таблицы для перевода p -ичной дроби, заданной в виде отдельных цифр дробной части, в десятичную систему счисления с использованием схемы Горнера.

11. Разработайте программу для перевода заданной правильной конечной p -ичной дроби в десятичную систему счисления описанным способом:

- без использования схемы Горнера;
- с ее использованием.

Литература

1. Андреева Е.В., Босова Л.Л., Фалина И.Н. Математические основы информатики. М.: Бином. Лаборатория знаний, 2005.

2. Семакин И.Г. Каким мне представляется профильный курс информатики. / Информатика, № 4/2009.

Ответы на задания для самостоятельной работы учащихся

3.

- $10101_2 = 21_{10}$;
- $2121_3 = 70_{10}$.

4. Если весомости разрядов записать, например, в строке 42 (см. ниже), то искомое значение в ячейке В2 может быть рассчитано по формуле: =СУММПРОИЗВ(В1:И1;В42:И42). Расчет весомостей может быть проведен следующим образом:

1) в ячейки В41:И41 вводятся номера разрядов (это можно сделать, используя автозаполнение);

2) в ячейку В42 вводится формула =2^В41, которая затем распространяется (копируется) на остальные ячейки.

	A	B	C	D	E	F	G	H	I
1	Введите двоичные цифры числа:	0	1	0	1	1	1	0	0
2	Соответствующее десятичное число равно:								
...									
41	Номер разряда	7	6	5	4	3	2	1	0
42	Весомость разряда	128	64	32	16	8	4	2	1

5. Вспомогательные расчеты, согласно схеме Горнера, можно провести, например, в строке 41:

	A	B	C	D	E	F	G	H	I
1	Введите двоичные цифры числа:	0	0	1	0	1	0	1	1
2	Соответствующее десятичное число равно:	43							
...									
41		0	0	1	2	5	10	21	43

В ячейку В41 вводится формула =A42*2+В1, которая затем распространяется (копируется) на остальные ячейки 41-й строки. Искомое значение в ячейке В2 равно значению в ячейке И41.

Приведенный пример наглядно демонстрирует преимущества использования для расчетов схемы Горнера.

6. Формула в ячейке В41: =A41*\$B2+В1 распространяется (копируется) на остальные ячейки 41-й строки.

	A	B	C	D	E	F	G	H	I
1	Введите цифры числа:	0	0	1	0	1	0	1	1
2	Введите основание системы p :	3							
3	Соответствующее десятичное число равно:	274							
...									
41		0	0	1	3	10	30	91	274

8.

- $0,10101_2 = 0,65625_{10}$;
- $0,2121_3 = 0,86419753_{10}$.

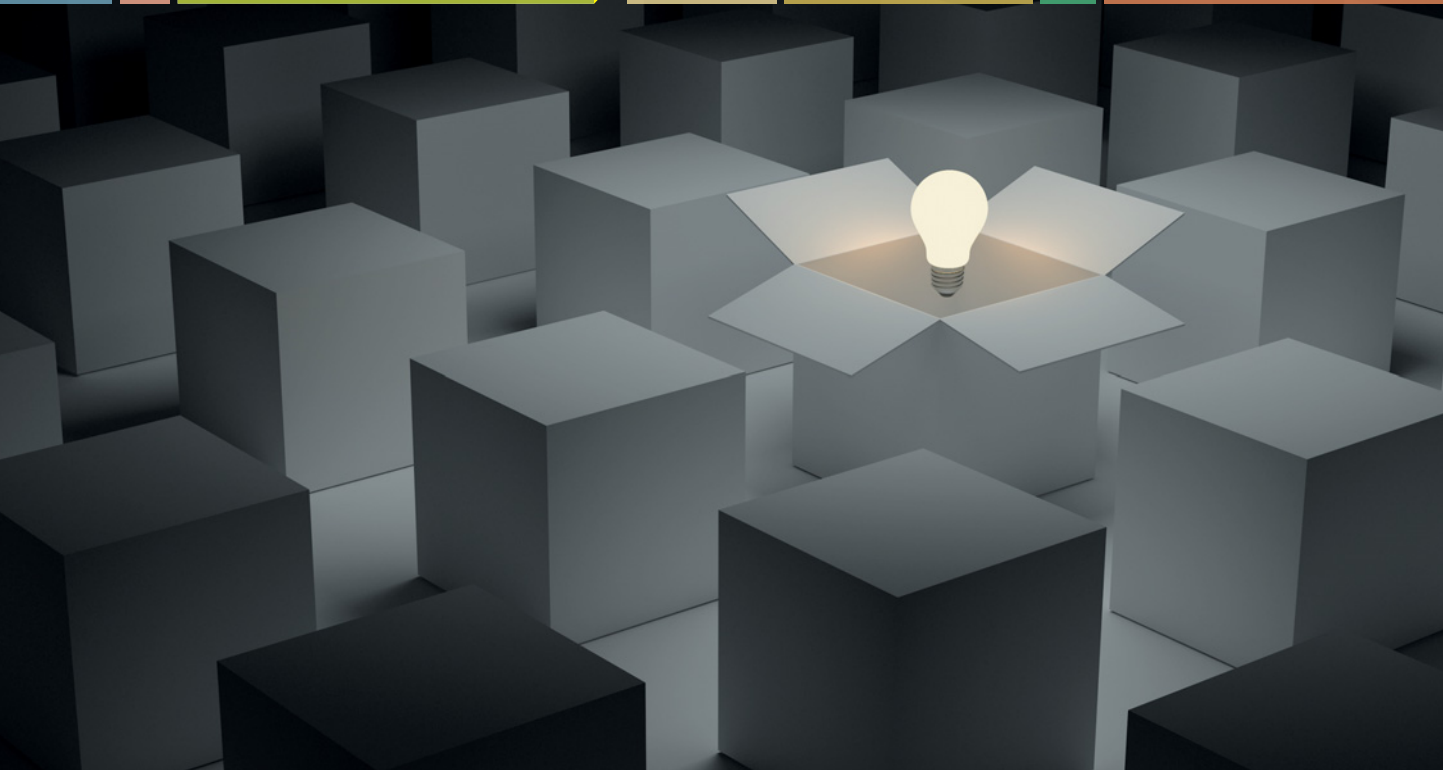
9. Если весомости разрядов записать, например, в строке 42 (см. ниже), то искомое значение в ячейке В3 может быть рассчитано по формуле: =СУММПРОИЗВ(В1:И1;В42:И42). Расчет весомостей может быть проведен следующим образом:

1) в ячейки В41:И41 вводятся номера разрядов (это можно сделать, используя автозаполнение);

2) в ячейку В42 вводится формула =СТЕПЕНЬ(\$B2; -С21), которая затем распространяется (копируется) на остальные ячейки.

Таблица к заданию 9

	A	B	C	D	E	F	G	H	I
1	Введите цифры правильной p -ичной дроби	0	1	1					
2	Введите основание системы счисления p	2							
3	Соответствующая десятичная дробь равна	0,375							
...									
41	Номер разряда	1	2	3	4	5	6	7	8
42	Весомость разряда	0,5	0,25	0,125	0,0625	0,03125	0,01562	0,00781	0,0039



Вопросы по информатике для проведения школьных конкурсов “Что? Где? Когда?” и “Брейн-ринг”

Д.М. Златопольский,
Москва

► 1. Вы, наверное, знаете, что такое “черный ящик”? Так называют устройство для автоматической записи данных во время эксплуатации самолета, вертолета, водного судна и т.п. В случае аварии или неисправной работы эта информация позволит выяснить возможную причину произошедшего.

Что в названии прибора не соответствует его особенностям?

Вариант вопроса

Вы, наверное, знаете, что такое “черный ящик”? Так называют устройство для автоматической записи данных во время эксплуатации самолета, вертолета, водного судна и т.п. В случае аварии или неисправной работы эта информация позволит выяснить возможную причину произошедшего. Несмотря на название прибора, чаще всего его окрашивают в оранжевый цвет. Почему?

2. Хотя это слово не является названием числа (как *два*, *пять*, *сто* и т.п.), оно иногда используется при счете, наряду с другими словами. Что это за слово?

Вариант вопроса

Хотя это слово не является названием числа (как *три*, *пять*, *сто* и т.п.), оно иногда используется при счете, наряду со словом “два”. Что это за слово?

3. Ведущий (обращаясь к одному из участников конкурса): “Вы знаете, что такое *декада*? А теперь посмотрите на табличку:




3	Триада
4	?
10	Декада

Какое слово должно стоять на месте вопросительного знака?”

4. Эти два понятия, используемые в информатике, имеют одну и ту же вторую часть, а первая в одном из этих понятий



происходит от латинского слова, соответствующего слову *объединять*, а во втором — от слова, соответствующего слову *разобцать*. Что это за понятия?

5. Какой алгоритмической конструкции, по вашему мнению, соответствует последнее изображение в табличке?

Алгоритмическая конструкция	Условное изображение
Следование (линейный алгоритм)	
Ветвление (разветвляющийся алгоритм)	
Цикл (циклический алгоритм)	
?	

Вариант вопроса

Какое изображение, по вашему мнению, соответствует последней алгоритмической конструкции в табличке?

Алгоритмическая конструкция	Условное изображение
Следование (линейный алгоритм)	
Ветвление (разветвляющийся алгоритм)	
Цикл (циклический алгоритм)	
Цикл в цикле (вложенный цикл)	?

6. Ведущий: “Вы знаете, что такое пишущая машинка? Если нет, то скажу, что это было устройство для печати текста на бумаге. Оно представляло собой клавиатуру, при резком нажатии на клавиши которой ударный молоточек с соответствующим символом ударял через красящую ленту по листу бумаги, печатая на нем символ. После этого бумага смещалась влево на одну позицию и очередной символ мог быть напечатан рядом.

А теперь внимание — вопрос. В результате очередного усовершенствования конструкции была добавлена одна клавиша, а количество печатаемых символов увеличилось в два раза. Что делала эта клавиша?”

7. Вы, конечно, знаете, что арифметика — это наука о числах и действиях над ними. А как назывался механический прибор, позволявший выполнять четыре арифметических действия над целыми числами?

8. Существует одна из ненаучных гипотез происхождения начертания современных арабских цифр. Ее идею демонстрирует следующее изображение:



В чем заключается эта гипотеза?

9. Один любитель информатики проживал по адресу: ул. Кременчугская, 153, корпус 6. Но когда он

смотрел на вывеску с номером дома, он утверждал, что живет в доме номер 69. Как выглядела вывеска?

Вариант вопроса

Один любитель информатики проживал по адресу: ул. Кременчугская, 153, корпус 6. Но когда он смотрел на вывеску с номером дома, он утверждал, что живет в доме номер 69. Почему?

10. Один журналист приехал в гостиницу, не успев забронировать номер по Интернету (при таком бронировании стоимость проживания меньше). В числе услуг, предоставляемых гостиницей, было:

- 1) бесплатный завтрак;
- 2) бесплатная доставка в номер ужина;
- 3) бесплатный Wi-Fi;
- 4) бесплатная подача в номер свежих газет и журналов

и другие.

В результате он все же смог обеспечить меньшую стоимость проживания. Как?

Вариант вопроса

Один журналист приехал в гостиницу, не успев забронировать номер по Интернету (при таком бронировании стоимость проживания меньше). Как обычно, он взял с собой ноутбук.

В числе услуг, предоставляемых гостиницей, было:

- 1) бесплатный завтрак;
- 2) бесплатная доставка в номер ужина;
- 3) бесплатный Wi-Fi;
- 4) бесплатная подача в номер свежих газет и журналов

и другие.

В результате журналист все же смог обеспечить меньшую стоимость проживания в гостинице. Как?

11. В этом выражении (задание предлагается участникам конкурса в письменном виде):

$$101 - 102 = 1$$

переместите одну цифру так, чтобы оно стало верным.

Вариант вопроса

То же задание, но с добавлением условия “Операция возведения в степень применяться не должна”.

12. Как известно, для проверки правильности программы, проводящей некоторые расчеты, и поиска возможных ошибок следует провести контрольные расчеты “вручную” — используя калькулятор или т.п. Но на ЭВМ первых поколений даже при расчетах по правильной программе были возможны ошибки из-за случайных аппаратных сбоев (скачков напряжения и других причин). Для них проверка правильности результатов проводилась по-другому. Как именно?

Вариант вопроса

Как известно, для проверки правильности программы, проводящей некоторые расчеты, и поиска возможных ошибок следует провести контрольные расчеты “вручную” — используя калькулятор или т.п. Но на ЭВМ первых поколений даже при правильной программе иногда расчет проводился дважды. Почему?

13. В электронно-вычислительных машинах первых поколений программа оформлялась в виде номеров ячеек, в которых размещались команды для решения задачи. Номера ячеек указывались относительно некоторого начального номера, например, в виде, показанном в таблице

Номер ячейки	Команда в ячейке
$a + 0$	
$a + 1$	
$a + 2$	
$a + 3$	
$a + 4$	
$a + 5$	
$a + 6$	
$a + 7$	
$a + 10$	
$a + 11$	
...	

Почему номер ячейки, оформленный в таблице полужирным начертанием и большим кеглем, именно такой?

14. Когда число, представляемое в компьютере как 11100000, меньше числа, представляемого как 00000111?

15. Вы, наверное, слышали о головоломке “Ханойские башни”. Ее простейший вариант изображен на картинке¹.



Напомню, что в ней имеются еще два вертикальных стержня. Задача состоит в том, чтобы перенести диски с исходного стержня на какой-то другой, используя третий стержень как промежуточный. При этом должны соблюдаться три условия:

- 1) за один ход можно переносить лишь один диск;
- 2) нельзя класть больший диск на меньший;
- 3) снятый диск нельзя отложить в сторону — он должен быть надет на один из стержней.

А теперь внимание — вопрос: “Какое отношение к головоломке “Ханойские башни” имеет следующая табличка?”:

1
2
1
3
1
2
1

¹ При наличии головоломки ее можно предъявить участникам конкурса. Можно также использовать монеты разного диаметра.

Вариант вопроса

Какое отношение к головоломке “Ханойские башни” имеет следующая последовательность чисел:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

16. Ученик в документе текстового редактора Microsoft Word подготовил верное равенство, скопировал его в буфер обмена, а потом вставил скопированный фрагмент в презентацию Microsoft PowerPoint. Этот фрагмент стал выглядеть так:

$$102 = 210$$

Почему?

Вариант вопроса

Посмотрите, пожалуйста, на табличку

$$102 = 210$$

Как, не меняя цифры местами, получить верное равенство?

17. Вопрос (в двух вариантах) аналогичен предыдущему, но используется табличка вида:

$$112 = 310$$

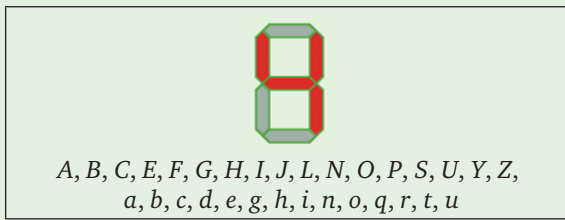
18. Академик Андрей Петрович Ершов — один из основоположников школьной информатики. В 1958 г. вышла его книга “Программирующая программа для быстродействующей электронной счетной машины”.

Как называют программу такого типа в настоящее время?

Вариант вопроса

Как называют системную программу такого типа в настоящее время?

19. Посмотрите, пожалуйста, на картинку²:



Что изображено на ней?

20. Посмотрите, пожалуйста, на картинку:



Что изображено на ней?

Ответы

1. Цвет — как правило, он не черный, а оранжевый (чтобы его было легче разыскать после аварии).

Ответ на вариант вопроса

Чтобы его было легче разыскать после аварии.

² Соответствующее изображение показывается участникам конкурса и при задании следующих вопросов.


2. Это слово “раз”, одно из значений которого — “один”. Примеры его использования:

1) “раз, два и обчёлся”; 2) “раз, два — взяли” и т.п.

3. Тетрада. Так в информатике называют группу из четырех бит (например, при переводе двоичного числа в шестнадцатеричную систему счисления). Слово происходит от *тетра* — части сложных слов, означающей “четыре” (например, тетраэдр и т.п.).

4. Конъюнкция и дизъюнкция — логические операции.

5. Цикл в цикле (вложенный цикл).

6. Новая клавиша позволяла делать то, что на клавиатуре персональных компьютеров делает клавиша <CapsLock> (или однократно — клавиша , а именно — меняла регистр букв. До описанного усовершенствования машинка печатала только большие (прописные) буквы. В результате усовершенствования на каждом ударном молоточке помещались по две литеры, одна под другой, и поэтому могли быть напечатаны как большие, так и маленькие буквы.

Источник — Шилов В.В. “Клавиши, клавиши — это вам не три струны (мифы о клавиатуре)”. Потенциал, № 9/2011.

7. Арифмометр. Название прибора происходит от двух греческих слов, одно из которых — “число”, “счет”, а второе — “мера”, “измеритель”.

Примечание. Учащиеся должны вспомнить такие термины, как *геометрия*, *фотометрия*, *антропометрия* и др., вторая часть которых соответствует по значению слову “измерение”, а также приборы амперметр, вольтметр, омметр и т.д.

8. Гипотеза заключается в том, что количество углов (острых или прямых) в изображении каждой цифры соответствует ее числовому значению: 1 — один угол, 2 — два угла и т.д.



Видно, что для цифры 9 гипотеза не соблюдается.

9. На вывеске номер дома был указан в виде 153_6 . В информатике в виде нижнего индекса указывают основание системы счисления, в которой записано число. Шестеричному числу 153_6 соответствует десятичное значение 69_{10} .

10. Журналист, подключившись по бесплатному Wi-Fi к Интернету, забронировал номер, после чего, согласно брони, оформил проживание по меньшей стоимости.

11. Возможны два варианта ответа:

1) $101 - 10^2 = 1$;

2) $110 - 102 = 1$ в троичной системе счисления.

Ответ на вариант вопроса — только с использованием троичной системы счисления.

12. Когда на результат расчетов могли повлиять случайные аппаратные сбои, проводился так называемый “двойной счет” — задача решалась дважды, и результаты сравнивались между собой. Тожественность результатов говорила о том, что в процессе вычислений машина не допускала случайных сбоев.

Ответ на вариант вопроса

Это делалось в случаях, когда на результат расчетов могли повлиять случайные аппаратные сбои (скачков напряжения и т.п.). В таких случаях зада-

ча решалась дважды, и результаты сравнивались между собой. Тожественность результатов говорила о том, что в процессе вычислений машина не допускала случайных сбоев.

13. После номера $a + 7$ следовал номер $a + 10$, так как запись велась в восьмеричной системе счисления ($8_{10} = 10_8$).

14. Это может быть, когда приведенные представления — это представления чисел в 8-разрядной ячейке со знаком. Первое представление соответствует отрицательному числу (старший, знаковый, разряд равен 1), второе — положительному.

15. Если диски, начиная с самого маленького, обозначить, соответственно, 1, 2 и 3, то последовательность их переключивания при решении головоломки совпадает с последовательностью чисел в показанной таблице.

Ответ на вариант вопроса

Если дисков — четыре и если обозначить их, начиная с самого маленького, соответственно, 1, 2, 3 и 4, то последовательность переключивания дисков при решении головоломки совпадает с показанной последовательностью чисел.

16. При копировании был изменен формат символов. В текстовом редакторе Microsoft Word числа 2 и 10 были представлены в виде подстрочного знака (нижнего индекса), которым обозначается основание системы счисления, в которой записываются числа:

$$10_2 = 2_{10}$$

Ответ на вариант вопроса

Надо числа 2 и 10 представить в виде подстрочного знака (нижнего индекса):







$$10_2 = 2_{10}$$

Примечание. Вариант ответа $1^{02} = 21^0$ не является правильным, так как в математике запись 1^{02} не используется.

17. $11_2 = 3_{10}$ или $1^{12} = 31^0$.

18. Транслятор (системная программа, переводящая прикладную программу на машинный язык).

19. На рисунке слева изображен так называемый “семисегментный индикатор” — прибор, используемый для указания цифр. Иногда он применяется и для отображения букв. Но лишь немногие из букв имеют интуитивно понятное семисегментное представление. Перечень таких латинских букв и приведен на рисунке. Так как количество букв ограничено, то семисегментные индикаторы используют только для отображения простейших сообщений:

	On
	Off
	Stop
	Close
	Play
	Pause

20. На рисунке приведен перечень русских букв, которые имеют интуитивно понятное семисегментное представление (см. также ответ на предыдущий вопрос).



СЕМИНАР

Полярные координаты

Д.М. Златопольский,
Москва

► Как правило, точку на плоскости представляют парой чисел x и y . Эта пара означает расстояние этой точки до неких осей. Число x задает расстояние по горизонтали, y — по вертикали. Пара (x, y) называется декартовыми координатами, или просто координатами точки. Свое название координаты получили по имени французского математика Рене Декарта¹ (1596–1650), которому мы обязаны этим методом описания местонахождения точки.

Но есть и другой способ определения расположения точек на плоскости — задание так называемых “полярных координат”.

Возьмем на плоскости точку O (называемую *полусом*) и выходящую из этой точки полупрямую (называемую *полярной осью*). Если на этой прямой задать масштаб и положительное направление, то мы определим полярную систему координат (рис. 1).

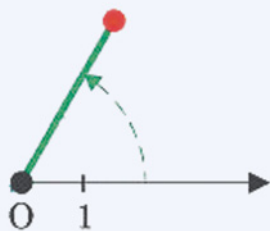


Рис. 1

Любую точку на плоскости теперь можно определить парой чисел: R — расстоянием до точки от полюса и φ — углом между полярной осью и прямой, соединяющей полюс и данную точку (угол φ измеряется в направлении против часовой стрелки от оси).

¹ Именно Декарту принадлежит известное высказывание “Я мыслю, значит, я существую”.

Будем обозначать точку с полярными координатами в виде (R, φ) . На рис. 2 приведены две точки — с координатами $(3, 60^\circ)$ и $(4, 210^\circ)$.

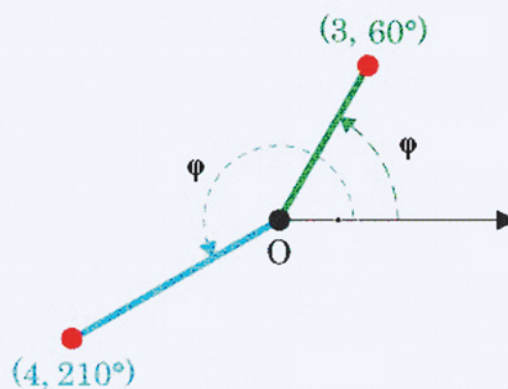


Рис. 2

Допускается, что значения R и φ могут быть и отрицательными. Тогда три пары координат определяют одну и ту же точку: $(1, 90)$, $(-1, -90)$, $(-1, 270)$.

Интересно, что полярные координаты используют... пчелы! Медоносные пчелы применяют их для обмена информацией об источниках пищи. Найдя новый источник (например, цветочную клумбу), пчела-разведчица возвращается в улей, приносит образец найденной пищи и исполняет своеобразный танец, на языке которого рассказывает, где находится клумба. Танец состоит в



том, что пчела, покачиваясь с боку на бок, прочерчивает прямую и затем, описав плавную кривую, возвращается в начальную точку. Затем она перемещается по тому же участку прямой, но возвращается по кривой в другом направлении. Весь процесс повторяется несколько раз. При этом “танце” длина отрезка указывает расстояние до цветочной клумбы, а направление этой прямой — направление (угол), под которым надо лететь. Таким образом пчела-разведчица сообщает другим пчелам полярные координаты нового источника пищи.

Функции, в которых используются полярные координаты, будем называть *функциями в полярных координатах*. Например, $R = \cos(\varphi)$ — функция в полярных координатах. Здесь для каждого значения φ из некоторого диапазона строится точка с полярными координатами (R, φ) .

Как получить график функции $R = \cos(\varphi)$ в полярных координатах? Можно, конечно, воспользоваться транспортиром и линейкой, получать необходимые точки на плоскости непосредственно в таких координатах:

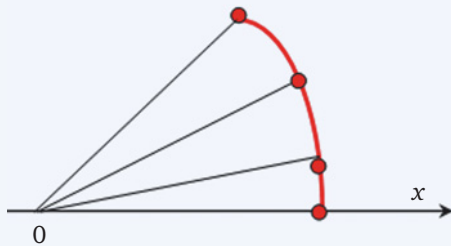


Рис. 3

Можно (а может, даже и удобнее?) снова обратиться к декартовым координатам. Вспомнив о синусе и косинусе угла φ , можем сказать, что точка (R, φ) в полярных координатах — это то же самое, что $(R \times \cos(\varphi), R \times \sin(\varphi))$ в декартовых координатах, и именно ее мы можем построить (см. рис. 4).

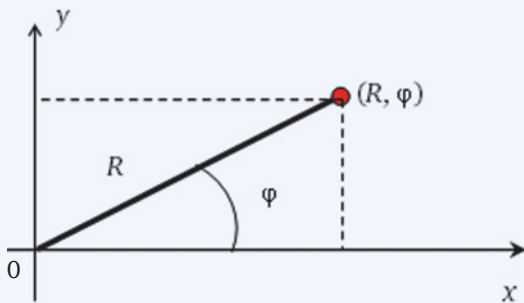


Рис. 4

Но, конечно, построение графиков функций в полярных координатах вручную — дело достаточно трудоемкое. Удобнее использовать для этой цели компьютер. К сожалению, в электронных таблицах (Microsoft Excel, Oracle OpenOffice Calc и др.) нет возможности строить такие графики.

Единственная возможность — использовать графические возможности языков программирования. Предлагаем читателям, владеющим каким-либо языком программирования, получить в полярных координатах график, например, функции $R = \sin(\varphi)$. Методика разработки программы приведена в следующей статье. Здесь же обратим внимание, что график этой функции в полярных координатах совсем не похож на график функции $y = \sin(x)$ в декартовых координатах (его называют “синусоидой”). Например, график функции $R = 1 + 2\cos(2\varphi)$ выглядит так.

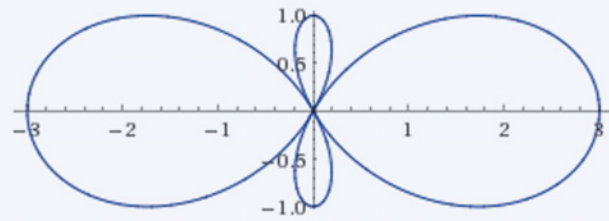


Рис. 5

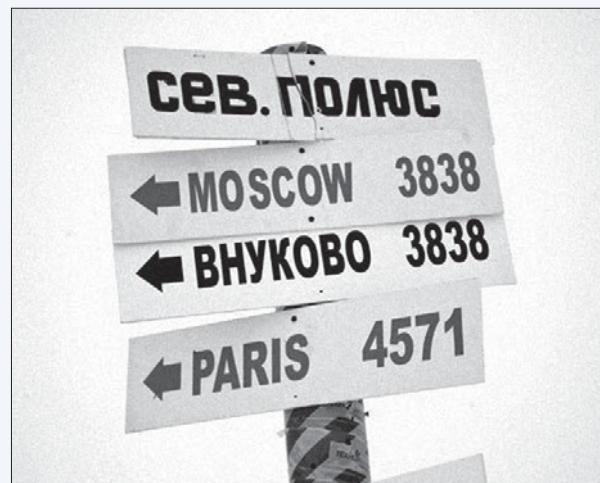
В Интернете имеется ряд сайтов, на которых можно получить график нужной функции в полярных координатах. Попробуйте найти их — мы уверены, что вы получите удовольствие от результатов своих экспериментов.

Еще об одном интересном эксперименте, косвенно связанном с полярными координатами, рассказано в следующей статье.

Задание для самостоятельной работы

Установите, как выглядит в полярных координатах график функции:

- 1) $R = 1$;
- 2) $R = \varphi$.



Литература

1. Коснёвски Ч. Занимательная математика и персональный компьютер. М.: Мир, 1987.

Построение графиков функций в полярных координатах

Альберт Михайлов,
ученик гимназии № 1530
г. Москвы

Основные этапы программы построения графиков в полярных координатах функций в общем случае следующие:

1. Инициализация графического режима
2. Цикл для углов от 0 до max градусов
 - 2.1. Пересчет значения угла в радианы (именно в этих единицах выражается угол при расчете синуса и косинуса)
 - 2.2. Расчет значения R
 - 2.3. Определение декартовых координат x и y
 - 2.4. Рисование точки с координатами x , y

— где max — некоторое максимальное значение угла.

В программе на языке Паскаль используем основные величины:

— $angle$ — угол между полярной осью и прямой, соединяющей полюс и данную точку, в градусах (в первой статье выпуска он обозначен как ϕ);

R — соответствующее значение расстояния до полюса;

rad — угол $angle$, выраженный в радианах (360 градусов = 2π радиан);

x, y — декартовы координаты, соответствующие полученным полярным (см. рис. 4 в первой статье). На этапе 2.3 надо учесть, что значения синуса и косинуса углов не превышают 1 и не меньше -1 , поэтому следует использовать коэффициент увеличения размера графика (например, равный 100). Кроме того, здесь требуется пересчет, связанный с различными точками начала и разными направлениями осей координат в графическом режиме системы программирования и принятыми в декартовых координатах:

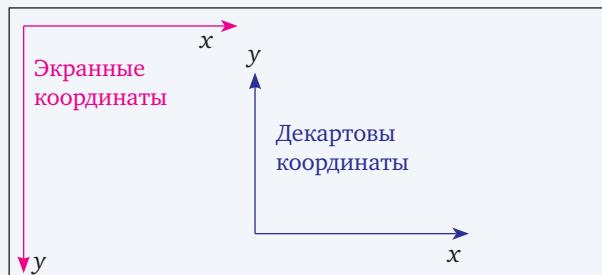


Рис. 1

Например, для графического режима 640 на 480 пикселей точка начала декартовых координат (0, 0) в центре экрана будет находиться в точке с экранными координатами (320, 240).

Формулы для пересчета, учитывающие оба эти обстоятельства:

$$x = r \times \cos(rad) \times 100 + 320;$$

$$y = 240 - 100 \times r \times \sin(rad),$$

где 100 — коэффициент увеличения размера графика.

Вся программа, например для функции $R = \sin(6\phi)$, имеет вид:

```
Uses Graph, CRT;
```

```
Var angle, gd, gm: integer; R, rad, x, y: real;
```

```
BEGIN
```

```
{Инициализация графического режима}
```

```
gd := detect;
```

```
initgraph(gd, gm, ' ');
```

```
{Цикл для всех углов}
```

```
For angle := 0 To 359 Do
```

```
  Begin
```

```
    {Перевод градусов в радианы}
```

```
    rad := 2 * Pi * angle/360;
```

```
    {Расчет значения R}
```

```
    R := sin(6 * rad);
```

```
    {Расчет соответствующих декартовых координат}
```

```
    x := R * cos(rad) * 100 + 320;
```

```
    y := 240 - 100 * R * sin(rad);
```

```
    {Очередная точка графика}
```

```
    PutPixel(Trunc(x), Trunc(y), 1)
```

```
  End;
```

```
{Приостановка программы до нажатия любой клавиши}
```

```
Repeat
```

```
Until Keypressed;
```

```
CloseGraph
```

```
END.
```

В книге, указанной в предыдущей статье, приведено большое число оригинальных графиков. Одна из групп графиков показана на рис. 2. Они получены как график функции $R = \sin(6\phi)$ в полярных координатах, построенный по декартовым координатам (как в описанной чуть выше программе), но в формулах для расчета x и y при этом использованы дополнительные коэффициенты a и b :

$$x = R \times \cos(rad \times a) \times 100 + 320;$$

$$y = 240 - 100 \times R \times \sin(rad \times b).$$

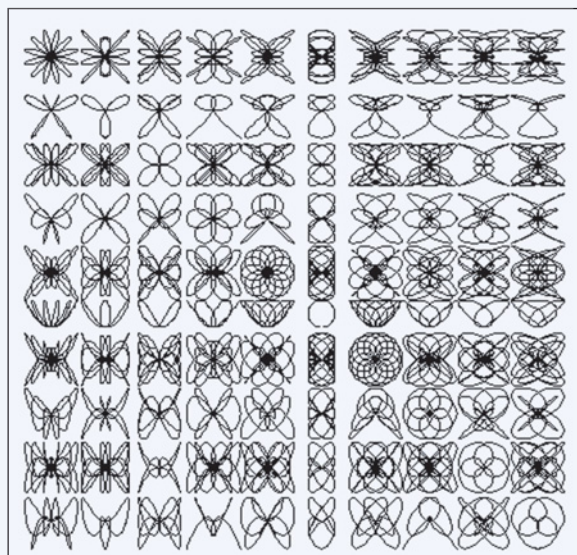


Рис. 2

К сожалению, в книге не приводятся значения коэффициентов для того или иного графика. Я провел эксперимент и установил эти значения. Значения коэффициентов a и b можно представить в виде таблицы, соответствующей условной таблице, в которой изображены графики:

a	b				
	1	2	3	...	10
1					
2					
3					
...					
10					

ЭТО ПОЛЕЗНО ЗНАТЬ

Древние системы нумерации и счет предметов

Г.И. Фалин,
Москва

Каждая система записи чисел представляет их с помощью определенных знаков по определенным правилам. Простейший способ записать число заключается в повторении отрезков, точек и т.п. столько раз, сколько в этом числе содержится единиц. Элементы такой примитивной системы прослеживаются в записи небольших чисел всех известных нам древних систем счисления. В используемой до сих пор римской системе нумерации именно так обозначаются числа от 1 до 3: 1 обозначается как I, 2 — как II, 3 — как III. Конечно, для достаточно больших чисел этот способ нумерации неудобен. Поэтому неудивительно, что многие народы для записи группы из 5 или 10 символов, обозначающих один предмет, стали использовать особый символ. Например, в аттической нумерации (Древняя Греция, первое тысячелетие до нашей эры) числа от 1 до 9 обозначались следующим образом:

Десятичная нумерация	1	2	3	4	5	6	7	8	9
Аттическая нумерация					Г	Г	Г	Г	Г

Для числа 10 в аттической нумерации используется новый символ: Δ, так что, например, 11 обозначается как Δ|, а 16 — как ΔГ|. В такой системе счисления числа 1, 5, 10 и т.п. называются узловыми, а все остальные числа получаются из узловых с помощью единственной операции сложения.

Индейцы майя в повседневной жизни для записи чисел от 1 до 19 использовали похожий принцип: единица обозначалась точкой, а число 5 — горизонтальной чертой. Но располагались эти символы не так, как в аттической системе нумерации; принцип ясен из следующей таблицы:

Предлагаю читателям, используя эти же значения коэффициентов, получить графики, относящиеся к функциям $R = \sin(5\varphi)$ и $R = \sin(8\varphi)$. Уверен — вам понравится...

От редакции

1. Полученные графики (например, в виде так называемых “скриншотов” — “снимков экрана”) присылайте в редакцию. Рассмотрите также другие функции.

Представившие графики с оригинальными (красивыми, необычными) изображениями будут награждены дипломами.

2. На диске к данному номеру журнала представлен разработанный автором статьи проект Borland Delphi, иллюстрирующий построение графиков с коэффициентами a и b для ряда функций.

Десятичная нумерация	1	2	3	7	10	12	15
Нумерация майя	⋯	══	⋯	═══

В Древнем Египте использовалась иероглифическая система нумерации чисел, в которой числа от 1 до 9 записывались соответствующим количеством вертикальных черточек, расположенных как показано в следующей таблице:

Десятичная нумерация	1	2	3	4	5	6	7	8	9
Египетская иероглифическая нумерация									

Для записи числа 10 использовался символ “∩”, так что, например, число 15 изображалось пиктограммой ∩ |||, а число 50 — пиктограммой ∩ ∩ ∩ ∩ ∩ ||. Следующим узловым числом было 100;

для него использовался рисунок, похожий на греческую букву “ρ”, так что, например, число 232 изображалось как ρ ρ ∩ ∩ ∩ ||.

Описанные выше системы нумерации были одними из первых шагов человечества на длинном пути, который привел к появлению современной десятичной позиционной системы счисления. Подробнее о системах нумерации древних народов и “примитивных” племен, зарождении позиционных систем счисления и т.п. можно прочитать в [1].

Интересно отметить, что при всей своей примитивности эти древние системы нумерации исключительно удобны при счете предметов. С определенной модификацией они используются до сих пор, в частности, в описательной статистике, элементы которой сейчас включены в школьные программы как в России, так и за рубежом.

В статистике первичная информация для подсчета часто представляется в виде неупорядоченных последовательностей из значений анализируемой статистической величины. Такую последовательность могут образовывать, например, ответы на определенный вопрос в виде пачки заполненных анкет социологического опроса, оценки по разным предметам учеников разных классов, расположенные в алфавитном порядке фамилий, и т.п.

По этим данным на практике нужно получить различные количественные показатели (среднее значение и т.п.), построить графики или диаграммы и т.п.

Важным начальным этапом в обработке статистических данных является разбиение диапазона возможных значений анализируемой переменной на несколько промежутков (их называют “интервалами группировки”, или “классами группировки”) и подсчет числа значений, попадающих в каждый промежуток.

Приведем такой пример. По результатам измерения времени (в минутах), которое 100 выбранных наугад школьников тратят на дорогу в школу, может появиться такой список: 27, 52, 43, 38, 47, 8, 21, 40, ... Нужно определить количество значений времени, попавших в тот или иной интервал времени.

Примем следующие классы группировки: 3–10, 11–18, 19–26, 27–34, 35–42, 43–50, 51–58.

В учебниках по статистике, включая соответствующие разделы школьных учебников (например, в [2]), для подсчета числа значений, попадающих в каждый класс, рекомендуется использовать следующий метод.

Сначала нужно нарисовать таблицу из трех столбцов; в первом указаны все классы группировки, во втором — проводится подсчет числа значений, попадающих в соответствующий класс, а в третьем — для каждого класса группировки записывается окончательный результат подсчета:

Класс группировки (диапазон времени)	Подсчет	Количество чисел в классе группировки
3–10		
11–18		
...		
51–58		

Если статистические данные принимают небольшое число значений, группировка не производится, т.е. просто подсчитывается, сколько раз повторяется каждое значение. Пример — подсчет количества двоек, троек, четверок и пятерок среди 60 оценок учеников по некоторому предмету:

Класс группировки (оценка)	Подсчет	Количество чисел в классе группировки
2		
3		
4		
5		

Как заполнить второй столбец (“Подсчет”) приведенных таблиц?

Можно просто, последовательно просматривая элементы анализируемой последовательности значений статистической переменной (время или оценки), для каждого значения определять, в какой класс группировки оно попадает, и рисовать вертикальные “черточки” (“|”) в соответствующей строке, а потом подсчитать их количество для того или иного класса группировки. Но есть и более рациональный способ подсчета.

После того как в строке нарисовано четыре вертикальные черточки, пятая черточка рисуется горизонтально (или наклонно), перечеркивая группу из четырех ранее нарисованных черточек: |||| или |||/ .

В результате число значений, попавших в тот или иной класс, представляется в виде рисунка (пиктограммы) из нескольких фигур |||| и нескольких (но не более четырех) вертикальных черточек. Переход от этой пиктографической записи к привычной нам десятичной системе нумерации не вызывает труда.

Например, пиктограмма $\text{||||} \text{||||} \text{||||} \text{||}$ представляет число:

3×5 (три группы вида ||||) + 2 (вертикальные черточки) = 17.

Для контроля ошибок при счете полезно сложить найденные во втором столбце числа; сумма должна быть равна общему количеству чисел в анализируемой последовательности.

Чтобы проиллюстрировать описанную процедуру, рассмотрим следующую задачу (это слегка отредактированная задача № 10 демонстрационного варианта британского экзамена по математике на получение аттестата о среднем образовании [3]).

Задача

Петя попросил 30 своих друзей указать любимый сорт мороженого и получил следующие ответы: сливочное, шоколадное, сливочное, ванильное, шоколадное, шоколадное, сливочное, шоколадное, фруктовое, шоколадное, ванильное, сливочное, шоколадное, ванильное, шоколадное, сливочное, ванильное, шоколадное, шоколадное, ванильное, шоколадное, шоколадное, шоколадное, сливочное, шоколадное, фруктовое, шоколадное, сливочное, ванильное, шоколадное. Затем Петя обработал эти результаты и нарисовал круговую диаграмму, представляющую результаты опроса.

В приведенной ниже таблице заполнена только одна строка.

Сорт мороженого	Подсчет	Частота	Угол диаграммы
Сливочное	$\text{ } \text{ }$	7	84°
Шоколадное			
Ванильное			
Фруктовое			
	<i>Итого:</i>	30	360°

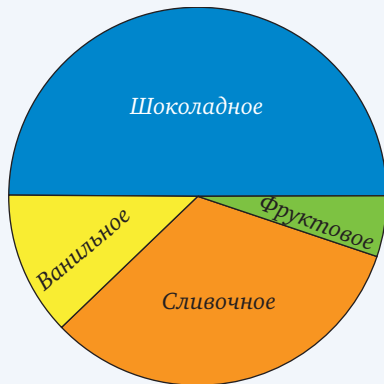
Заполните таблицу до конца и нарисуйте круговую диаграмму, представляющую результаты опроса.

Решение

Используя для подсчета изложенный выше метод, мы получим следующую таблицу:

Сорт мороженого	Подсчет	Частота	Угол диаграммы
Сливочное		7	84°
Шоколадное		15	180°
Ванильное		6	72°
Фруктовое		2	24°
	<i>Итого:</i>	30	360°

Соответствующая круговая диаграмма имеет вид:



Для двумерных переменных (когда каждый элемент набора данных состоит из двух значений) приходится рисовать две таблицы. В первой проводится подсчет с использованием пиктограмм, а во второй записываются результаты подсчета в обычном виде. Мы проиллюстрируем это с помощью следующей задачи (это задача № 2 из раздела В британского письменного экзамена по статистике на получение аттестата о среднем образовании [4]; текст задачи слегка отредактирован).

Задача

На свободное рабочее место претендовало 40 человек. Для 30 из них информация о поле и возрасте приведена в следующей таблице:

Возрастная группа (лет)	До 20 лет	20–30 лет	Старше 30 лет
Мужчина			
Женщина			

Возраст и пол остальных 10 человек указан ниже (м — мужчина, ж — женщина): 22-м, 35-м, 22-ж, 28-м, 23-ж, 41-м, 26-м, 33-ж, 29-ж, 32-м.

(а) Используйте эту информацию, чтобы создать таблицу, аналогичную приведенной выше, для группы из всех 40 человек.

Из всей группы претендентов случайно выбирается один человек.

(b) (i) Какой пол и возрастная группа наиболее вероятны для этого человека.

(c) (ii) Найдите вероятность того, что выбрана женщина.

Решение

Полностью заполненная таблица приведена ниже.

Возрастная группа (лет)	До 20 лет	20–30 лет	Старше 30 лет
Мужчина			
Женщина			

Ей соответствует следующая таблица, в которой указано число претендентов данного пола и возрастной группы:

Возрастная группа (лет)	До 20 лет	20–30 лет	Старше 30 лет
Мужчина	7	14	4
Женщина	5	7	3

Если из 40 претендентов случайно выбран один, то наиболее вероятно, что это мужчина в возрасте от 20 до 30 лет. Поскольку среди претендентов 15 женщин, вероятность того, что выбрана женщина, равна $\frac{15}{40} = 37,5\%$.

При всей простоте описанного метода у него есть один существенный недостаток, отмеченный в классическом учебнике по описательной статистике [5]. При быстрой работе легко допустить ошибку в оценке числа вертикальных черточек в очередной группе и провести горизонтальную черту, когда нарисовано не четыре, а три или пять вертикальных черточек. Иначе говоря, вместо |||| (всего пять предметов) можно по ошибке нарисовать ||| (всего четыре предмета) или ||||| (всего шесть предметов). Гораздо удобнее использовать другие схемы подсчета.

Самая простая из них выглядит следующим образом. Последовательности нужно, как и раньше, отмечать черточкой, но формировать из черточек не фигуру ||||, а квадрат с диагоналями. Таким образом, вертикальная черта | символизирует 1 предмет, фигура L — 2, фигура □ — 3, фигура □ — 4, фигура □ — 5, а фигура □ — 6 предметов. В этом случае невозможно ошибиться при рисовании полной группы □. Применяя этот метод к рассматриваемому в первой задаче (про мороженое), мы получим таблицу:

Сорт мороженого	Подсчет	Частота
Сливочное	□	7
Шоколадное	□ □ □	15
Ванильное	□	6
Фруктовое	L	2
	<i>Итого:</i>	30

Подсчет общего количества значений, попавших в тот или иной класс группировки, производится по той же схеме, что и для первого метода. Например, для класса “шоколадное” это количество равно 2×6 (две группы вида \boxtimes) + 3 (группа вида \square) = 15. После небольшой тренировки можно научиться определять количество предметов, обозначаемое неполными группами $|$, \perp , \square , \square , \boxtimes , на глаз, по визуальному образу фигуры, т.е. вообще без подсчета. Конечно, при счете шестерками нужно иметь лучшие навыки устного счета, чем при счете пятерками, особенно если число объектов в классе группировки несколько десятков.

Можно немного модифицировать эту схему так, чтобы полная группа состояла из 10 объектов. Например, в [5] предлагается не сразу рисовать квадрат, а сначала поставить четыре точки в его вершинах. В результате мы получим специфическую систему нумерации, изображенную на рис. 1.

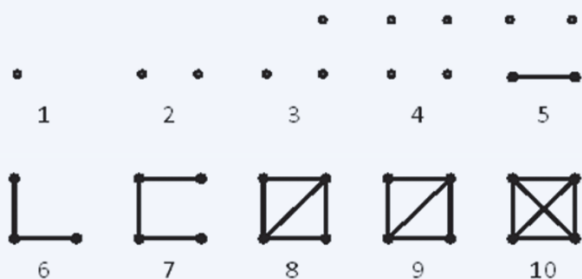


Рис. 1

С равным успехом можно дорисовать по два отрезка на противоположных сторонах квадрата так, чтобы получились треугольники. В результате мы получим систему нумерации, изображенную на рис. 2.

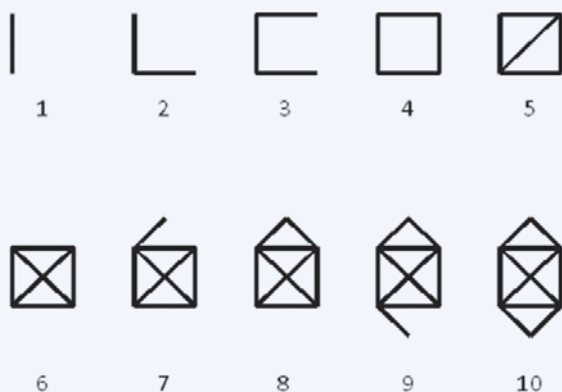


Рис. 2

Используя эти системы нумерации, например, число 23 можно записать так, как показано на рис. 3.



Система рис. 1



Система рис. 2

Рис. 3

В заключение еще раз скажем, что полученные по описанным методикам данные используются затем для расчета различных статистических показателей, построения графиков и т.п.

Задания для самостоятельной работы

1. Приняв, что полный список продолжительностей “похода” в школу, о котором шла речь выше, следующий:

27, 52, 43, 38, 47, 8, 21, 40, 32, 53, 35, 28, 40, 18, 31, 45, 24, 30, 37, 15, 39, 34, 48, 25, 30, 7, 32, 12, 26, 35, 48, 19, 33, 26, 17, 30, 42, 22, 53, 28, 42, 36, 23, 10, 34, 46, 16, 29, 35, 52, 41, 32, 21, 39, 55, 25, 29, 8, 36, 44, 26, 55, 34, 19, 42, 54, 27, 10, 45, 20, 31, 50, 18, 9, 41, 14, 38, 40, 23, 49, 33, 15, 24, 46, 36, 28, 32, 37, 51, 20, 29, 47, 33, 27, 41, 22, 39, 40,

— получите итоговые таблицы, составленные с использованием трех систем нумерации, описанных в статье.

2. Используя электронную таблицу Microsoft Excel или другую, постройте круговую диаграмму, иллюстрирующую распределение времени.

Литература

1. Башмакова И.Г., Юшкевич А.П. Происхождение систем счисления. В кн.: Энциклопедия элементарной математики. М. – Л.: Государственное издательство теоретико-технической литературы, 1951.
2. Алгебра. 9-й класс. Часть 1. Учебник для учащихся общеобразовательных учреждений. / А.Г. Мордкович, П.В. Семенов. М.: Мнемозина, 2010.
3. AQA GCSE Mathematics (Linear) B 4365/1F. Specimen Paper. Foundation Tier. 2012 Specification.
4. Edexcel GCSE Statistics 1389/1F. Paper 1F. Foundation Tier. 18 June 2008.
5. Тьюки Дж. Анализ результатов наблюдений. Разведочный анализ. Пер. с англ. М.: Мир, 1981.

От редакции. Результаты выполнения предложенных заданий присылайте в редакцию.

Папки и файлы с масками

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

Предлагаю читателям выполнить три задания, связанные с файлами и папками на диске.

Задание 1

Пользователь работал в **Проводнике** и переходил от одной папки к другой. Четыре момента его работы показаны в окнах на *рис. 1–4*.

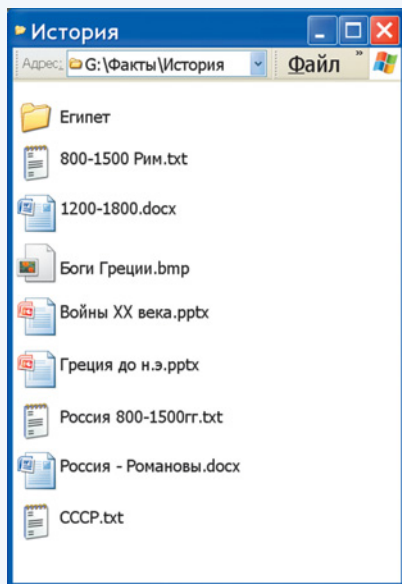


Рис. 1

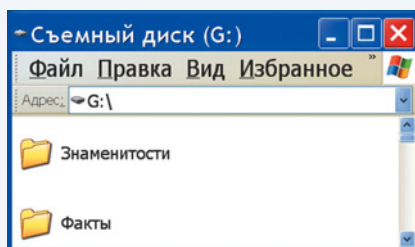


Рис. 2

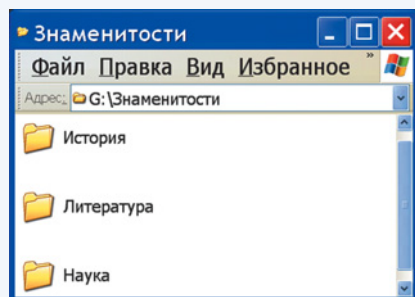


Рис. 3

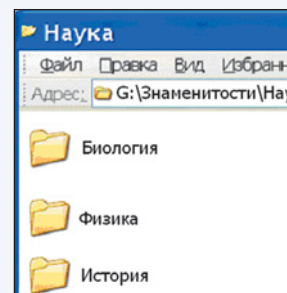


Рис. 4

Постройте фрагмент дерева папок, с которым работал пользователь.



Задание 2

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

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

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*.

От редакции

- Если вы не знакомы с понятием “маска файлов”, то найдите соответствующую информацию в Интернете или в других источниках.
- Ответы, пожалуйста, присылайте в редакцию (можно выполнять не все задания).

Необычный маршрут

Митя вышел из точки А плоской равнины и прошел 1 м — на юг, 2 м — на запад, 4 м — на север, 6 м — на восток, 7 м — на юг, 8 м — на запад, 10 м — на север, 12 м — на восток, 13 м — на юг, 14 м — на запад, 16 м — на север, 18 м — на восток и т.д.

На каком расстоянии от точки А будет Митя после 100 “шагов” (после прохождения 100 отрезков)?

- Какова длина последнего, сотого отрезка пути?
- Какой общий путь пройдет Митя за 100 “шагов”?

Лишний солдат и бидоны с медом

Рассмотрим две задачи, при решении которых можно использовать электронную таблицу (Microsoft Excel или др.).

Задача 1 “Генерал и рядовой Байтиков”

Генерал построил солдат в колонну по 4, но при этом только рядовой Байтиков остался один в шеренге. Тогда генерал построил солдат в колонну по 5, и снова один Байтиков остался “лишним”. Когда же в колонне по 6 опять только Байтиков остался один, генерал посулил ему наряд на уборку территории вне очереди, после чего в колонне по 7 Байтиков “нашел” себе место в шеренге с другими, и никого “лишнего” не осталось. Какое наименьшее количество солдат могло быть у генерала? Известно, что искомое количество не более 350.

Оформим лист так, как показано на *рис. 1*:

	A	B	C	D	E
1		Остаток 4	Остаток 5	Остаток 6	Остаток 7
2	5				
3	6				
4	7				
...					
346	349				
347	350				
348					

Рис. 1

В столбце A запишем значения числа солдат, в столбцах B, C, D и E — рассчитаем остатки от деления того или иного числа солдат, соответственно, на 4, 5, 6 и 7. В электронной таблице Microsoft Excel для расчета остатка используется функция ОСТАТ, в Oracle OpenOffice.org Calc — функция MOD.

Формулы в ячейках можно получить следующим образом (на примере программы Microsoft Excel). В ячейку B2 ввести =ОСТАТ(A5;4), в ячейку C2 — =ОСТАТ(A5;5), в ячейку D2 — =ОСТАТ(A5;6), в ячейку E2 — =ОСТАТ(A5;7). Их можно распространить (скопировать) на остальные строки.

Но, оказывается, можно “вручную” вводить только одну (!) формулу. Для этого в строке 348 следует ввести числа 4–7 (см. *рис. 2*). Тогда формула в ячейке B2: =ОСТАТ(\$A2;B\$29), в которой использованы так называемые “смешанные”, или “комбинированные”, ссылки (с одним символом “\$”), может быть скопирована во все остальные ячейки диапазона B2:E347. Результат показан на *рис. 2*.

	A	B	C	D	E
1		Остаток 4	Остаток 5	Остаток 6	Остаток 7
2	5	1	0	5	5
3	6	2	1	0	6
...					
346	349	0	0	0	0
347	350	2	0	2	0
348		4	5	6	7

Рис. 2

После этого искомое значение будет в строке, в которой в столбцах B–E остатки будут равны, соответственно, 1, 1, 1 и 0. Для вывода в столбце F искомого значения следует использовать функции ЕСЛИ и И. Соответствующую формулу введите в ячейку G2 и распространите (скопируйте) ее на остальные строки.

Результат (число солдат) присылайте в редакцию.

Заметим также, что задачу можно решить с помощью электронной таблицы, записав при этом формулы в гораздо меньшее число ячеек. Для этого предварительно следует исследовать общий вид искомого числа. Оформите лист с учетом этого и также представьте его в редакцию.

Задача 2 “Бидоны с медом”

Собранный мед заполняет несколько 50-литровых бидонов. Если его разлить в 40-литровые бидоны, то понадобится на 5 бидонов больше, и один из них останется неполным. Если же этот мед разлить в 70-литровые бидоны, то понадобится на 4 бидона меньше, и один из них тоже останется неполным. Сколько 50-литровых бидонов заполняет собранный мед?

Решите задачу, используя электронную таблицу.

Три учителя

В старших классах школы работают три учителя: Иванов, Петров и Сидоров. Каждый из них преподает по два предмета, так что в расписании у них всего шесть предметов: математика, физика, химия, история, литература, английский язык. Сидоров — самый молодой из них. Учитель химии старше учителя истории. Все трое — учитель химии, учитель физики и Петров — любят играть в шахматы. Когда между собой играют учитель литературы и учитель английского языка, то Сидоров болеет за последнего. Петров не знает английского языка и не преподаёт математику.

Кто какие предметы преподаёт?

Литература

1. Богомолова О.Б. Логические задачи. М.: БИНОМ. Лаборатория знаний, 2005.

ПОИСК ИНФОРМАЦИИ**Четыре вопроса**

Найдите в Интернете или по другим источникам информации ответы на следующие вопросы.

1–2. Какая песня какой эстрадной звезды стала “японской национальной песней”?

3. Какую пряность больше всего не любят клопы и тараканы?

4. Из какого дерева, согласно скандинавским легендам, боги сотворили первую женщину?

Томас Хэрриот
и двоичная система счисления

Марат Эйнуллаев,
Константин Якушев,
ученики гимназии № 1530 г. Москвы

В большинстве источников можно найти информацию о том, что современная двоичная система была впервые описана великим немецким ученым Готфридом Вильгельмом Лейбницем в работе “Explication de l’Arithmétique Binaire” (1703 г). Но профессор В.В. Шилов, занимающийся вопросами истории счета и вычислительной техники, сообщил нашему научному руководителю, Дмитрию Михайловичу Златопольскому, что это не совсем так. Двоичную систему описывали и до Лейбница. В качестве примера В.В. Шилов привел ссылку на работы Томаса Хэрриота.

Томас Хэрриот (1560–1621) — английский математик, географ и астроном, воспитанник Оксфордского университета, составитель (1586 г.) ценного описания и карты исследованной им части Северной Америки (в настоящее время — штат Северная Каролина США), карты Луны, которую он наблюдал через зрительную трубу в одно время с Галилеем, и, наконец, автор труда “Применение аналитического искусства к решению алгебраических уравнений”, изданного в Лондоне через 10 лет после его смерти в 1631 году.



В последнем сочинении, во многом примыкающем к алгебраическим трудам Виета², Хэрриот поставил себе задачей изложить “аналитическое искусство” своего предшественника легче и проще для понимания и применения³.

Он оставил несколько тысяч страниц неопубликованных рукописей (правда, об этом не было известно вплоть до появления в 1951 г. работы [1]). Выдержки из последней статьи приведены в книге [2]. Мы ознакомились с этой книгой и перевели главу, связанную с Т.Хэрриотом, на русский язык.

В одной из рукописей Хэрриота приведена без комментариев следующая таблица:

² Франсуа Виет — французский математик XVI века, основоположник символической алгебры, установивший свойства корней приведенного квадратного уравнения: их сумма равна второму коэффициенту, взятому с противоположным знаком, а произведение — свободному члену уравнения. — Прим. ред.

³ Хэрриот также первым предложил использовать математические знаки “>” (больше) и “<” (меньше). — Прим. ред.

		16	16
1	1	17	16 + 1
2	2	18	16 + 2
3	2 + 1	19	16 + 2 + 1
4	4	20	16 + 4
5	4 + 1	21	16 + 4 + 1
6	4 + 2	22	16 + 4 + 2
7	4 + 2 + 1	23	16 + 4 + 2 + 1
8	8	24	16 + 8
9	8 + 1	25	16 + 8 + 1
10	8 + 2	26	16 + 8 + 2
11	8 + 2 + 1	27	16 + 8 + 2 + 1
12	8 + 4	28	16 + 8 + 4
13	8 + 4 + 1	29	16 + 8 + 4 + 1
14	8 + 4 + 2	30	16 + 8 + 4 + 2
15	8 + 4 + 2 + 1	31	16 + 8 + 4 + 2 + 1

Можно сказать, что в таблице приведены представления чисел от 1 до 31 в виде суммы степеней числа 2 (1, 2, 4 и т.д.). Как известно, такие представления используются при переводе десятичных чисел в двоичную систему счисления методом выделения максимальных степеней двойки.

На следующих примерно ста страницах рукописи приведены различные варианты табличного представления всех подмножеств из n предметов для $n = 1, 2, 3, 4$ и 5 (после чего следует сокращение “и т.д.”). Первая такая таблица имеет вид:

a	1	a	15	a	31	
		b		b		
a	3	c		c		
b		d		d		
ab		ab		e		
		ac		ab		ac
		ad		ac		ad
		bc	ae	bc		
a	7	bd	bd	bd		
b		cd	be	cd		
c		abc	ce	de		
ab		abd	abc	abd		
ac		acd	abd	abe		
bc	bcd	acd	acd			
abc	abcd	bcd	ace	ade		
		abcde	bcd	bce		
			bde	cde		
			abcde	abce		
			abde	abde		
			acde	acde		
			bcde	bcde		
			abcde	abcde		

Анализ показывает, что в ней приведены представления десятичных чисел с использованием цифр a , b и c , равных, соответственно, 1, 2 и 4, и каждое число равно сумме значений цифр. Например, число 3 записывается в виде ab ($1 + 2$), число 6 — как bc ($2 + 4$), число 25 — ade ($1 + 8 + 16$) и т.п.

Всю таблицу можно представить в виде:

a	1
-----	---

a	1
b	2
ab	$1 + 2 = 3$

a	1
b	2
c	4
ab	$1 + 2 = 3$
ac	$1 + 4 = 5$
bc	$2 + 4 = 6$
abc	$1 + 2 + 4 = 7$

a	1
b	2
c	4
d	8
ab	$1 + 2 = 3$
ac	$1 + 4 = 5$
ad	$1 + 8 = 9$
bc	$2 + 4 = 6$
bd	$2 + 8 = 10$
cd	$4 + 8 = 12$
abc	$1 + 2 + 4 = 7$
abd	$1 + 2 + 8 = 11$
acd	$1 + 4 + 8 = 13$
bcd	$2 + 4 + 8 = 14$
$abcd$	$1 + 2 + 4 + 8 = 15$

a	1
b	2
c	4
d	8
e	16
ab	$1 + 2 = 3$
ac	$1 + 4 = 5$
ad	$1 + 8 = 9$
ae	$1 + 16 = 17$
bc	$2 + 4 = 6$
bd	$2 + 8 = 10$
be	$2 + 16 = 18$
ce	$4 + 16 = 20$
de	$8 + 16 = 24$
abc	$1 + 2 + 4 = 7$
abd	$1 + 2 + 8 = 11$
abe	$1 + 2 + 16 = 19$
acd	$1 + 4 + 8 = 13$
ace	$1 + 4 + 16 = 21$
ade	$1 + 8 + 16 = 25$
bcd	$2 + 4 + 8 = 14$
bce	$2 + 4 + 16 = 22$
cde	$4 + 8 + 16 = 28$
$abcd$	$1 + 2 + 4 + 8 = 15$
$abce$	$1 + 2 + 4 + 16 = 23$
$abde$	$1 + 2 + 8 + 16 = 27$
$acde$	$1 + 4 + 8 + 16 = 29$
$bcde$	$2 + 4 + 8 + 16 = 30$
$abcde$	$1 + 2 + 4 + 8 + 16 = 31$

Видно, что приведены представления чисел в виде суммы степеней числа 2.

Через несколько страниц Хэрриот описывает другой вариант представления чисел. Он также приводит примеры для случаев от $n = 1$ до $n = 5$ (и также с фразой “и т.д.”). Чтобы проиллюстрировать этот вариант, покажем случай для $n = 3$:

--+	}
++-	
+++	
-+-	
7	

+++	}
++-	
+--	
7	

Видимо, Хэрриот имел в виду следующее.

Если условно обозначить столбцы a , b и c , то

- знак “+” соответствует тому, что “Да, этот символ (буква) используется в комбинации чисел”,
- знак “-” — что “Нет, не используется”.

В конце Хэрриот приводит такой вариант таблицы:

Рассуждая аналогично, но считая, что $a = 4$, $b = 2$, $c = 1$, можно утверждать, что в последней таблице представлены записи десятичных чисел, равные сумме элементов, обозначенных знаком “+”. Такое представление совпадает с двоичной записью чисел 7, 6, ..., 0:

111 110 101 100 011 010 001 000

В записях Хэрриота имеются также примеры вычислений в двоичной системе счисления с ис-

пользованием цифр 0 и 1. О них мы планируем рассказать читателям раздела “В мир информатики” в будущем.

Подводя итог, можно сделать вывод о том, что Томас Хэрриот описал использование двоичной системы для представления чисел за много лет до Лейбница.

Литература

1. Shirley J.W. Binary numeration before Leibniz // American Journal of Physics, 1951. Vol. 19.

2. Glaser Anton. History of binary and other nondecimal numeration, 1981.

От редакции. Предлагаем читателям составить таблицу записи чисел 32, 33, ... 63 с использованием цифр a, b, c, d и e , равных, соответственно, 1, 2, 4, 8, 16 и 32, аналогичную оформленную темно-зеленым цветом в конце статьи. Результат, пожалуйста, присылайте в редакцию.

Считаем так, как Пифагор и Архимед

В разделе “В мир информатики” публиковались материалы о греческой алфавитной системе счисления — способе записи чисел с помощью греческих букв. Так как такая система не была позиционной, то вычисления на ней были сложнее, чем при

использовании десятичной системы (попробуйте, например, выполнить сложение двух чисел, записанных римскими цифрами). Но ведь древние греки считали! Как?

Наш постоянный автор, кандидат технических наук М.А. Цайгер, в своем блоге в Живом Журнале (LiveJournal) открыл класс обучения древнегреческому счету “Считаем как Пифагор и Архимед”.

Вы можете присоединиться к числу тех, кто хочет освоить технику древнегреческого счета и научиться считать так, как считали Пифагор и Архимед, — складывать, вычитать и умножать числа, записанные в греческой системе нумерации, не используя калькулятор. Планируется даже проведение соревнований.

Для кого-то это может быть забавой, а для кого-то — стать подспорьем в работе по переводу старинных греческих математических и технических текстов на современный язык. В любом случае, как говорится, “лишних знаний не бывает”.

Чтобы присоединиться к сообществу, т.е. к классу учащихся, зарегистрируйтесь в LiveJournal, затем войдите в блог по адресу <http://archimedes-com.livejournal.com>, найдите опцию “Записаться в сообщество” и подтвердите свое желание стать членом сообщества. Сообщение об этом будет передано автору, который свяжется с вами и расскажет о дальнейших действиях.

ЯПОНСКИЙ УГОЛОК

Два sudoku

Решите, пожалуйста, две японские головоломки “судоку”:

1) простую:

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

Судоку без цифр

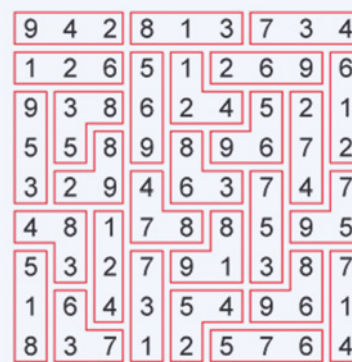
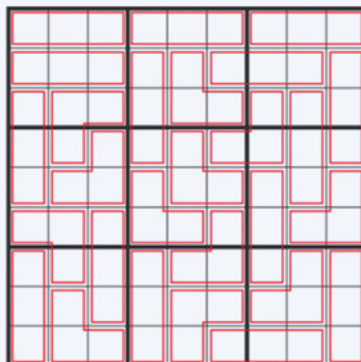
Не пугайтесь, уважаемый читатель, мы не предлагаем решить судоку, в котором не известна ни одна цифра ☺. Попробуйте расставить все блоки, имеющиеся на правом рисунке, в левом так, чтобы форма блоков совпадала и при этом соблюдались правила “классического” судоку (в каждом столбце, в каждой строке и в каждом квадратике 3×3 цифры не должны повторяться).



2) сложную:

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

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



Равенства

В какой системе счисления справедливы равенства:

- а) $3 \times 4 = 10$;
- б) $5 \times 5 = 31$;
- в) $2 \times 2 = 100$?

Серебряная цепочка

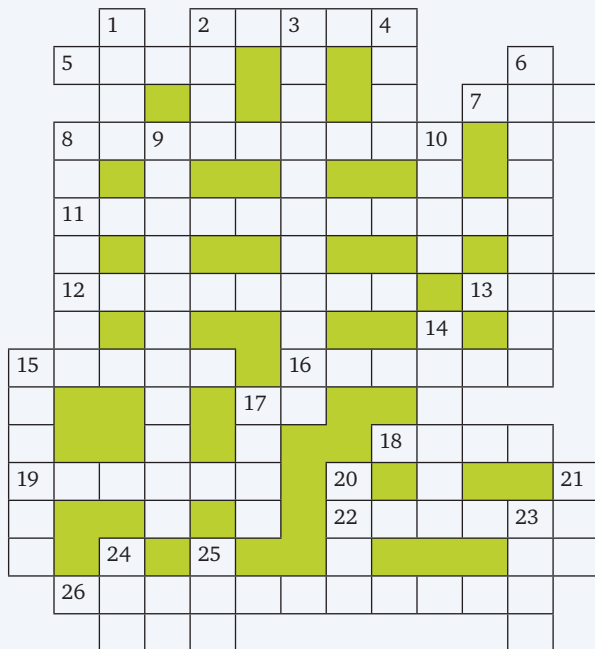
В гостиницу приехал путешественник. Выяснилось, что деньги он забыл дома. У него была лишь серебряная цепочка из семи звеньев. Он договорился с хозяином гостиницы о том, что за каждый день пребывания в гостинице будет расплачиваться одним звеном цепочки. Какое звено цепочки надо аккуратно расцепить, чтобы прожить в гостинице семь дней и ежедневно расплачиваться с хозяином? (Хозяин может давать сдачу звеньями, полученными им ранее.)

Кто есть кто?

Представьте себе, что, прогуливаясь по острову Двоичный Код, вы встретили двух людей — А и В. Каждый из них либо рыцарь (всегда говорит правду), либо лжец (всегда врет). А высказывает следующее утверждение: “По крайней мере один из нас лжец”. Можно ли определить, кто есть кто?

Кроссворд

Решите, пожалуйста, кроссворд.



По горизонтали

- 2. Последовательность символов, предназначенная для чтения человеком.
- 5. Компонент данных типа запись.

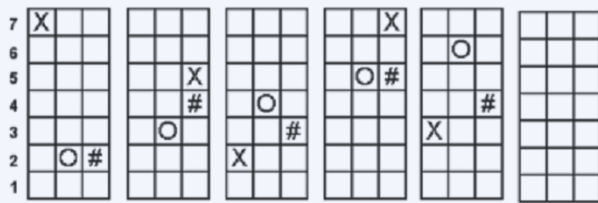
- 7. Задний план, на котором изображается символ на экране или элементы слайда презентации.
- 8. Серия устройств в ее отношении к предшествующим или последующим сериям.
- 11. Устройство или человек, которые могут выполнять определенные действия (по алгоритму).
- 12. Трехэлектродный полупроводниковый прибор, который может находиться в одном из двух устойчивых состояний: выключенном — с высоким сопротивлением и включенном — с низким.
- 13. Название буквы древнерусского алфавита, напоминающей твердый знак.
- 15. Разрешение на возможность выполнения тех или иных действий на компьютере.
- 16. Указатель места на экране.
- 18. Разновидность носителя информации.
- 19. Устройство управления работой шины персонального компьютера, а также спортивный судья.
- 22. Изображение, иллюстрирующее зависимость одной величины от другой.
- 26. Электронное вычислительное устройство.

По вертикали

- 1. Язык программирования для начинающих его изучать (и не только).
- 2. Часть оператора цикла (операторы, повторяемые при его выполнении).
- 3. Наука об общих закономерностях процессов управления и передачи информации.
- 4. Почти антоним названию “виги”.
- 6. Универсальное электронное устройство для обработки информации.
- 8. Нестандартное устройство для вывода информации в компьютере.
- 9. Создание второго экземпляра файла, символа, фрагмента текста.
- 10. Город в Липецкой области, которому присвоено звание “Город воинской славы”.
- 14. Значение переменной величины или константы логического типа (русский вариант).
- 15. Перемещение бумаги в принтере при печати.
- 17. Точка подключения внешних устройств к внутренней шине микропроцессора.
- 20. Геометрическая фигура, состоящая из двух лучей.
- 21. Буква греческого алфавита.
- 23. Популярный вид компьютерных программ.
- 24. Величина изменения значения переменной цикла.
- 25. Французская единица измерения расстояния на море, используемая в названии одного из романов Жюль Верна.

Последняя табличка

Определите, как должна выглядеть последняя табличка, чтобы закон изменения положения символов во всех табличках сохранился:



Числовой ребус “Трижды ШЕСТЬ”

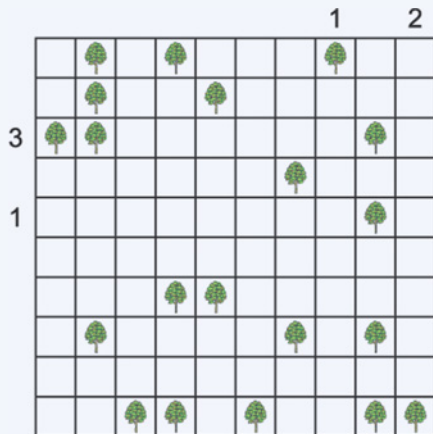
В этом числовом ребусе:

$$\begin{array}{r}
 \text{Ш Е С Т Ь} \\
 + \text{Ш Е С Т Ь} \\
 \hline
 \text{Ш Е С Т Ь} \\
 \text{С У М М А}
 \end{array}$$

Как обычно в таких головоломках, одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры. Решите его.

Деревья и скамейки

Группа деревьев растет так, как показано на рисунке:



Возле каждого дерева (в соседней с ним по стороне условной клетке) нужно установить скамейку так, чтобы скамейки не соприкасались между собой даже углом. Числа слева и сверху означают количество деревьев в соответствующем ряду. Можно ли решить задачу?

На шахматной доске

Имеется шахматная доска с обозначением клеток согласно стандартной шахматной нотации ($a1$ — нижняя левая, ..., $h8$ — верхняя правая):

a8	b8	c8	d8	e8	f8	g8	h8
a7	b7	c7	d7	e7	f7	g7	h7
a6	b6	c6	d6	e6	f6	g6	h6
a5	b5	c5	d5	e5	f5	g5	h5
a4	b4	c4	d4	e4	f4	g4	h4
a3	b3	c3	d3	e3	f3	g3	h3
a2	b2	c2	d2	e2	f2	g2	h2
a1	b1	c1	d1	e1	f1	g1	h1

Из некоторой начальной клетки нужно проложить маршрут 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. Например, из клетки $d3$ допустимы ходы на клетки $c3$, $d2$, $c2$.

Перечислите все такие маршруты, ведущие из начальной клетки $c3$. Ответ, пожалуйста, оформите с использованием следующей системы обозначения ходов:

Л — ход влево;

Н — ход вниз;

Д — ход по диагонали (вниз–влево).

Квадрат из костей домино

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”.

О карандаше

Предлагаем учащимся 1–7-х классов ответить на вопрос: “Сколько граней у шестигранного карандаша?”

Учитель: “Почему ты опоздал на урок?”

Ученик: “Потому что он начался раньше, чем я вошел в класс”.

Ответы, решения, разъяснения к заданиям, опубликованным в сентябрьском выпуске “В мир информатики”

Здесь редакция публикует ответы на задания, являющиеся, так сказать, “регулярными” — публикуемыми в каждом выпуске (и списки приславших ответы на них). Решения остальных, “оригинальных”, заданий из сентябрьского выпуска будут опубликованы в феврале.

Вопросы в рубрике “Поиск информации”

Ответы

1. Плиний Старший считал эликсиром молодости и средством от плохого дыхания абсент.
2. Согласно легенде, первый американский флаг шила швея Бетси Росс из Филадельфии.
3. Александром Сергеевичем Пушкиным овладела “ужасная тоска” в марте.
4. В России первую хирургическую операцию под наркозом провел 19 февраля 1847 года Федор Иноземцев в г. Риге. Были представлены также правильные ответы, связанные с первой операцией под наркозом, проведенной за рубежом.

Ответы представили:

- Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;
- Бельских Марина, Бойкова Юлия, Борникова Елена, Вышегородских Анастасия и Шамаев Евгений, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;
- Березин Василий, 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.**;
- Глазкова Екатерина, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;
- Евграфова Ксения, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;
- Заева Кристина, Республика Башкортостан, г. Уфа, школа № 54 (Центр дистанционного обучения), учитель **Искандарова А.Р.**;
- Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;
- Кононенко Александра, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;
- Лопаткина София, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;
- Новикова Анна и Потапова Алевтина, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Соловьева Марина, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;
- Федосеева Анастасия, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Задания в рубрике “Для эрудитов”

Ответы

1. Спутницей автора детективных романов Татьяна Устиновой в обувном магазине во время истории, о которой шла речь в вопросе, была телеведущая Елена Малышева (вариант ответа Б).
2. В парке, открытом американским кинорежиссером Фрэнсисом Фордом Coppолой в Калифорнии, в темноте дегустируют вино (вариант ответа В).
3. Памятник сантехнику поставлен в г. Перми (вариант ответа А).
4. Считается, что именно желтый цвет способен улучшать память и способствует ясной мысли, помогает сконцентрироваться (вариант ответа В).
5. При оформлении гардеробной в одном из популярных австралийских торговых центров было использовано более двух с половиной центнеров шоколада (вариант ответа Б).

Правильные ответы прислали:

- Андрющенко Александр, Остроухова Валерия, Пономаренко Анастасия и Уткина Ксения, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;
- Баранова Елизавета, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;
- Бельских Марина, Бойкова Юлия, Борникова Елена, Вышегородских Анастасия и Шамаев Евгений, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;
- Березин Василий и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;
- Бородюк Анна и Василенко Татьяна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;
- Глазкова Екатерина, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;
- Громова Дарья, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;
- Заева Кристина, Республика Башкортостан, г. Уфа, школа № 54 (Центр дистанционного обучения), учитель **Искандарова А.Р.**;
- Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;
- Кононенко Александра, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;
- Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;
- Новикова Анна и Потапова Алевтина, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Трептау Татьяна, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;
- Федосеева Анастасия, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**

Кроссворд

Ответы

По горизонтали: 1. Ось. 3. Имя. 8. Основание. 9. Цифра. 10. Ответ. 12. Копирование. 15. Архив. 16. “Кисть”. 19. Спам. 21. Кадр. 22. Стиль.

По вертикали: 2. Сканер. 4. Магнит. 5. Монитор. 6. Два. 7. Деление. 11. Код. 13. Истина. 14. Африка. 16. Рост. 18. Тире. 20. Мат. 21. “Кол”.

Ответы прислали:

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

— Баранова Елизавета, средняя школа поселка Осинка, Алтайский край, учитель **Евдокимова А.И.**;

— Барановская Татьяна и Жукова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Глазкова Екатерина, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Дикарева Елизавета и Калинина Ирина, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Добрякова Светлана, Козина Мария и Чернова Наталья, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

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

— Кононенко Александра, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Карпова Вера, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Семенова Юлия, Республика Татарстан, г. Альметьевск, гимназия № 5, учитель **Уливанова Р.М.** (Юлия и другие ученики Розалии Минировны впервые приняли участие в наших конкурсах);

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

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

Решение японских головоломок “судоку” представили:

— Абросимова Дарья и Русскова Мария, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Глазкова Екатерина, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

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

— Есипова Мария, Круглякова Мария и Яснова Дарья, средняя школа поселка Осинка, Алтайский край, учитель **Евдокимова А.И.**;

— Зорина Елена, Республика Башкортостан, г. Уфа, школа № 18, учитель **Искандарова А.Р.**;

— Кабанова Владислава, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Караваева Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

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

Ответы на задания, опубликованные в августовском выпуске “В мир информатики”, прислали также:

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

Ответы на вопросы, предложенные в рубриках “Поиск информации” и “Для эрудитов”:

— Наделяев Денис, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н (учитель **Краснёнкова Л.А.**) — решение задачи “Бальзам”; Сурин Алексей из той же школы представил правильные ответы на вопросы, предложенные в рубрике “Поиск информации”.

ВНИМАНИЕ! КОНКУРС

Конкурс № 107

Для перевода десятичного числа 241 762 в двоичную систему методом последовательного деления числа на основание понадобится выполнить 17 операций деления. Предложите методику перевода, согласно которой можно решить задачу за гораздо меньшее число операций деления. Методика должна быть такой, чтобы ее можно было продемонстрировать, например, у классной доски.

Ответ отправьте в редакцию до 5 февраля по адресу: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика” или по электронной почте: vmi@1september.ru. Пожалуйста, четко ука-

жите в ответе свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.

За участие в конкурсе № 102 дипломами будут награждены следующие читатели:

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

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

— Калачев Андрей, г. Москва, школа № 827, учитель **Липина В.М.**

Поздравляем!



Общероссийский проект Школа цифрового века

Интернет-обеспечение проекта – Издательский дом «ПЕРВОЕ СЕНТЯБРЯ»

2014/15

Прием заявок на следующий учебный год открыт

Срок действия проекта в 2014/15 учебном году –
с 1 августа 2014 года по 30 июня 2015 года.

Стоимость участия в проекте для образовательного учреждения –
6 тысяч рублей за весь учебный год.

Стоимость не зависит от количества педагогических работников,
желающих принять участие в проекте.

Материалы проекта предоставляются каждому
педагогическому работнику образовательного учреждения.

Предметно-методические издания

- **23 журнала** по всем учебным дисциплинам и направлениям школьной жизни с дополнительными материалами для практического использования (презентации, раздаточные материалы, образовательное видео)
- **методические брошюры**
- **журнал для родителей**
- **общероссийская педагогическая газета «Первое сентября»**

Курсы повышения квалификации

- **дистанционные** 36-часовые курсы по общей педагогике
- **модульные** 6-часовые курсы
 - цикл «Навыки профессиональной и личной эффективности»
 - цикл «Инклюзивный подход в образовании»

Прием заявок и подробности на сайте
digital.1september.ru

Участие образовательного учреждения и педагогических работников
в проекте удостоверяется соответствующими документами



СОЧИ 20-14

ВКЛАД ШКОЛЬНИКОВ В ЗИМНЮЮ ОЛИМПИАДУ

с 1 сентября по 1 февраля проходит Всероссийская акция
«СПОРТИВНЫЙ ЛОНГМОБ СОЧИ 20-14»
За четыре месяца школьники страны преодолели 500 000 километров!

НАША ЦЕЛЬ:

900 000 КИЛОМЕТРОВ – 22 ОЛИМПИЙСКИХ ЭКВАТОРА!

ПРИСОЕДИНЯЙТЕСЬ!

Расскажите о проекте коллегам!
Проведите еще один забег!

Стартуйте в любой день до 1 февраля!

Пешком, бегом, на лыжах, вплавь... Выбирайте!

материалы акции* на сайте

Longmob.1september.ru

ПОКАЖЕМ НАШУ СИЛУ!

Формула участия: организуйте 20-минутный забег со стартом в 14 часов дня и внесите суммарную дистанцию, пройденную всеми участниками забега, на сайт Longmob.1september.ru

На сайте дистанции забегов складываются в олимпийские экваторы (по 40 тысяч километров каждый).



ДИСТАНЦИОННЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

(с учетом требований ФГОС)

15 января заканчивается прием заявок на второй поток 2013/14 учебного года

образовательные программы:

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

Стоимость – 2990 руб.

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

Стоимость – 2390 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