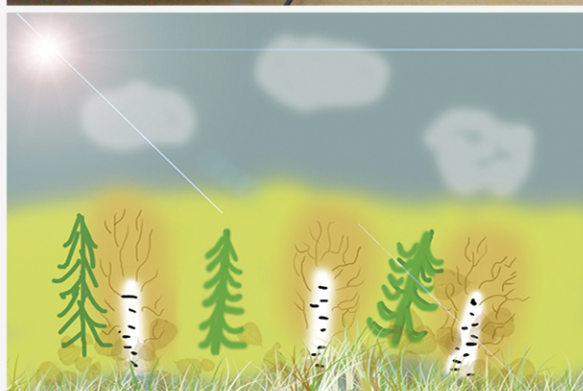
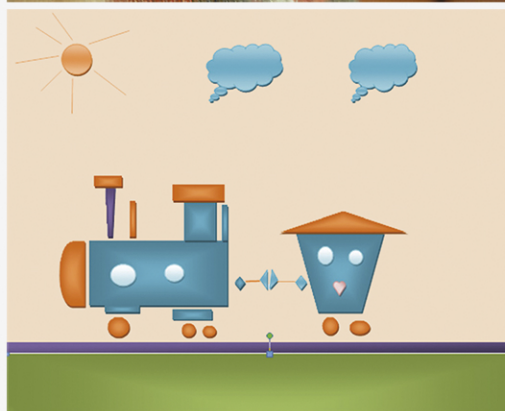
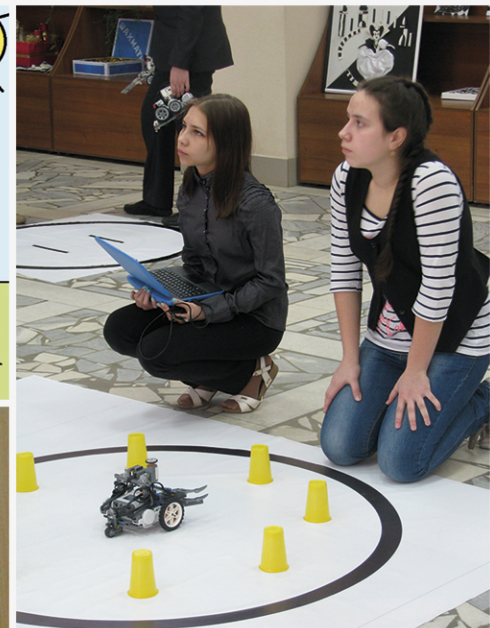


ИНФОРМАТИКА В ШКОЛЕ

№ 9'2015

ISSN 2221-1993

www.infojournal.ru





XVI МЕЖДУНАРОДНАЯ НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

Основные направления работы:

- Технологическая модернизация бизнеса и образования с использованием инновационных решений и сервисов фирмы «1С»
- Новые организационно-технологические решения фирмы «1С» для глобальных образовательных сообществ
- Индивидуализация психолого-педагогической работы с учащимся в условиях глобальной образовательной среды
- Использование технологической платформы «1С:Предприятие» для проектно-изыскательской работы преподавателей и студентов

Мероприятия в рамках конференции:

- Пленарные и секционные заседания
- Мастер-классы по использованию программных продуктов фирмы «1С»
- Вернисаж программных и методических разработок
- Тестирование «1С:Профессионал» по программным продуктам «1С:Предприятие 8»
- Отбор в программу У.М.Н.И.К. Фонда содействия развитию малых предприятий

В 2015 году в конференции приняли участие более 1700 человек.

Подробнее о тематиках конференции и условиях участия см. сайт www.1c.ru/educonf

Участие бесплатное для всех сотрудников образовательных организаций и органов управления образованием (проживание оплачивается отдельно).

Обязательная предварительная регистрация открыта до 01 февраля 2016 года на сайте <http://www.1c.ru/educonf>



ФИРМА «1С»
Оргкомитет конференции:
Тел./факс: +7 (495) 688-90-02
Email: npk@1c.ru
www.1c.ru/educonf

02–03 февраля 2016 г.
Гостиница «Космос»,
Москва, проспект Мира, д. 150

Редакционный совет

Баракина Т. В.
Бешенков С. А.
Босова Л. Л.
Дергачева Л. М.
Заславская О. Ю.
Захарова Т. Б.
Кириченко И. Б.
Кравцова А. Ю.
Кузнецов А. А.
Левченко И. В.
Окулов С. М.
Рыбаков Д. С.

Редакция

Дергачева Л. М.
главный редактор
Кириченко И. Б.
*заместитель
главного редактора*
Губкин В. А.
Коптева С. А.
Лукичева И. А.
Меркулова Н. И.
Федотов Д. В.
Шарапкова Л. М.

Адрес редакции:

119121, г. Москва,
ул. Погодинская, д. 8, оф. 222
Телефон/факс: (495) 364-95-97
E-mail: readinfo@infojournal.ru
URL: <http://www.infojournal.ru>

Подписные индексы

в каталоге «Роспечать»:
для индивидуальных подписчиков — 81407
для предприятий и организаций — 81408

Подписано в печать 20.11.2015.
Формат 60×90¹/₈. Усл. печ. л. 8,0.
Тираж 2000 экз. Заказ № 0900.

Отпечатано в типографии
ООО «ГЕО-Полиграф»
141290, Московская область,
г. Красноармейск, ул. Свердлова, д. 1.

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

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

© «Образование и Информатика», 2015

ИНФОРМАТИКА В ШКОЛЕ

№ 9(112) ноябрь 2015

Содержание

От редакции 2

УРОКИ ИНФОРМАТИКИ

Босова Л. Л., Босова А. Ю. Практикум по информатике. VII класс..... 4

Сорокина Т. Е. Поурочные разработки к программному модулю
по информатике для V класса «Пропедевтика программирования
со Scratch» 17

МЕТОДИЧЕСКАЯ КОПИЛКА

Дьякова В. В. Сетевой проект «От победы регионов к Великой Победе»...29

СУПЕРКОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ШКОЛЕ

Еремин Е. А. Как наглядно продемонстрировать школьникам
преимущества многопроцессорных систем.....32

ЗАДАЧИ

Дмитриев В. Л. Об эффективных решениях в задачах
по программированию37

ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Слинкина И. Н. Олимпиадные задачи по теоретическим основам
информатики для школьников и студентов41

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Александрова О. В. Интегрированное занятие-викторина
«Увлекательное путешествие в волшебный мир сказок»49

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

Кильдишов В. Д. Использование MS Excel для изучения основ
стеганографии53

Очков В. Ф., Богомолова Е. П. Немного о рекурсии, числовых рядах
и фракталах56

Зацепина Е. Д., Мишина А. С. Системы счисления: это может быть
интересно!60



Уважаемые читатели!

Последний месяц осени — время, когда в самом разгаре подписная кампания на следующий год. Издательство «Образование и Информатика» предлагает вам разные варианты подписки на наши издания — журналы «Информатика в школе» и «Информатика и образование».

Быстрый и удобный способ получать журналы сразу после их выхода из печати — **электронная подписка**. Подписавшись на электронные версии журналов, вы не только получите возможность оперативно знакомиться с новыми выпусками наших изданий, но и существенно сэкономите. Вы можете подписаться как на будущие выпуски — 2016 года, так и на ранее вышедшие номера журналов — 2014 и 2015 годов. (Доступ к электронным версиям более ранних выпусков предоставляется бесплатно всем зарегистрированным пользователям нашего сайта.)

На печатные версии журналов можно подписаться:

- в отделениях «Почты России» по каталогу агентства «Роспечать»;
- через агентство «Урал-Пресс»;
- непосредственно в редакции.

Подписка в редакции — самый дешевый и удобный способ получать наши издания в печатном виде: журналы будут доставляться вам заказной бандеролью. Кроме того, читателям, оформившим подписку в редакции на печатные версии журналов, бесплатно предоставляется доступ к электронным версиям.

Подробную информацию о подписке на издания ИНФО см. на нашем сайте: <http://www.infojournal.ru> в разделе «Подписка».

В текущем полугодии сайт ИНФО был существенно обновлен как по функционалу, так и по контенту. Мы постарались учесть пожелания наших пользователей и добавили много нового, в частности:

- подробно изложены правила редакционной работы со статьями;
- для каждого выпуска журнала можно скачать его содержание в формате pdf;
- вся сопроводительная информация к статье вводится авторами в полях электронной формы, что позволяет значительно сократить сроки обработки присланных материалов;
- приобрести отдельные выпуски журналов и оформить подписку теперь можно непосредственно с сайта.

Изменения, как содержательные, так и организационные, коснулись и **конкурсов, проводимых издательством «Образование и Информатика»**: материалы на конкурсы участникам следует отправлять через электронную форму на сайте ИНФО, а среди номинаций традиционного конкурса ИНФО появилась специальная онлайн-номинация. Подробная информация о **XII Всероссийском конкурсе научно-практических работ по методике обучения информатике и информатизации образования ИНФО-2015** — на с. 3 и на сайте ИНФО в разделе «Конкурсы».

Уже в течение пяти лет обложка журнала «Информатика в школе» создается исключительно из работ участников конкурса цифровых изображений и фотографий. Уважаемые читатели, напоминаем вам, что **работы на X Всероссийский конкурс ФОТО 1-2016 можно присылать до 15 декабря включительно, принять участие в конкурсе могут как учащиеся, так и педагоги**. О требованиях к представляемым на конкурс работам можно прочитать на сайте ИНФО.

Традиционно в первом полугодии каждого года проводятся тематические конкурсы для учителей и учащихся. Информация об очередном конкурсе 2016 года будет опубликована в первых номерах журналов «Информатика в школе» и «Информатика и образование».

Уважаемые коллеги! Приглашаем вас и ваших учеников к участию в конкурсах нашего издательства. Мы всегда рады видеть среди конкурсантов знакомые имена постоянных участников — наших замечательных авторов, благодаря которым журнал «Информатика в школе» живет и развивается. Но хотелось бы подчеркнуть, что мы ждем и новых участников, — пожалуйста, присылайте свои материалы, делитесь своими наработками. Даже если это будет ваш первый опыт подготовки журнальной публикации, мы вместе постараемся, чтобы он стал удачным началом большого пути. Дерзайте, и у вас все получится!

Редакция журнала
«Информатика в школе»

XII ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ПРАКТИЧЕСКИХ РАБОТ ПО МЕТОДИКЕ ОБУЧЕНИЯ ИНФОРМАТИКЕ И ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ ИНФО-2015

**Издательство «Образование и Информатика»,
Всероссийское научно-методическое общество педагогов
объявляют о проведении
в 2015 году конкурса по следующим номинациям:**

- **Урок информатики — тридцать лет спустя.**
- **Опыт работы по ФГОС.**
- **Информатизация образовательной организации. Использование электронных ресурсов.**
- **Методическая копилка учителя информатики (специальная онлайн-номинация).**

Руководит конкурсом **Организационный комитет** (далее — Оргкомитет), состоящий из представителей Российской академии образования, ведущих методистов, членов Всероссийского научно-методического общества педагогов, членов редакционных советов журналов «Информатика и образование» и «Информатика в школе», сотрудников объединенной редакции журналов.

Цели и задачи конкурса

1. Выявление и поддержка талантливых педагогов, методистов, руководителей образовательных учреждений и органов управления образованием, использующих в профессиональной деятельности информационно-коммуникационные технологии.
2. Включение педагогов, методистов, руководителей образовательных учреждений и органов управления образованием в деятельность по разработке нового содержания образования, новых педагогических технологий, методик обучения и управления образованием.
3. Создание информационно-образовательного пространства на страницах журналов «Информатика и образование» и «Информатика в школе» по обмену и распространению опыта использования средств информационно-коммуникационных технологий в педагогической деятельности и в области управления образованием.
4. Повышение информационной культуры и информационно-коммуникационной компетентности всех участников образовательного процесса — учащихся, педагогов, родителей.

Конкурс проводится с 1 октября по 20 декабря 2015 года.

Работы на конкурс принимаются до 20 декабря 2015 года включительно. Работы, присланные позже этой даты, к участию в конкурсе допускаться не будут. Подача работ производится только через заполнение формы заявки на сайте ИНФО (необходима предварительная регистрация на сайте или авторизация для зарегистрированных пользователей).

Итоги конкурса будут опубликованы на сайте издательства «Образование и Информатика» (<http://www.infojournal.ru/>), а также в номерах 1–2016 журналов «Информатика и образование» и «Информатика в школе».

Лучшие работы будут опубликованы в журналах «Информатика и образование» и «Информатика в школе».

Победители получают:

- диплом от Всероссийского научно-методического общества педагогов и издательства «Образование и Информатика»;
- электронную подписку на журналы «Информатика и образование» и «Информатика в школе» на 2016 год;
- по одному печатному экземпляру журналов «Информатика и образование» № 1–2016 и «Информатика в школе» № 1–2016, в которых будут опубликованы итоги конкурса;
- авторский печатный экземпляр журнала с опубликованной работой.

Подробную информацию о конкурсе вы можете найти на сайте ИНФО:

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

Контакты Оргкомитета

Телефон: (495) 364-95-97

E-mail: readinfo@infojournal.ru

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



Л. Л. Босова,
Федеральный институт развития образования, Москва,

А. Ю. Босова,
Учебно-методический центр Истринского района Московской области

ПРАКТИКУМ ПО ИНФОРМАТИКЕ. VII КЛАСС*

Аннотация

Формирование умений и навыков использования информационных технологий невозможно без организации индивидуальных практических работ учащихся с современным аппаратным и программным обеспечением. В статье представлено детальное описание 15 практических работ по информационным технологиям, ориентированных на учащихся VII класса.

Ключевые слова: информационные технологии, практическая работа, основная школа.

Контактная информация

Босова Людмила Леонидовна, заслуженный учитель РФ, доктор пед. наук, главный научный сотрудник Федерального института развития образования, Москва; *адрес:* 129319, г. Москва, ул. Черняховского, д. 9, стр. 1; *телефон:* (499) 152-73-41; *e-mail:* akull@mail.ru

L. L. Bosova,
Federal Institute of Education Development,
Moscow,

A. Yu. Bosova,
Training and Methodical Centre, Istra District,
Moscow Region

PRACTICAL WORKS ON INFORMATICS. 7TH GRADE

Abstract

The formation of the skills in use of information technologies is impossible without the organization of individual student practical work with modern hardware and software. Detailed description of the 15 practical works in information technologies, aimed at students of the 7th grade is presented in the article.

Keywords: information technologies, practical work, secondary school.

Изучение информационных технологий — неотъемлемая часть современного школьного курса информатики. В Примерную основную образовательную программу основного общего образования [3] включен раздел «Использование программных систем и сервисов», предполагающий формирование у выпускников основной школы не только общих представлений, но и ряда конкретных практических умений по темам: «Файловая система», «Подготовка текстов и демонстрационных материалов», «Электронные (динамические) таблицы», «Базы данных. Поиск информации», «Работа в информационном пространстве. Информационно-коммуникационные технологии». Формирование таких умений невозможно без организации индивидуальных практических работ учащихся с современным аппаратным и программным обеспечением. Наша авторская программа курса информатики для основной школы [1] предполагает изучение в VII классе многих вопросов, связанных с базовыми информационными технологиями.

В помощь педагогам, работающим по этой программе, разработано детальное описание пятнадцати практических работ, соответствующих содержанию, рекомендуемому к изучению в VII классе (см. табл. на с. 5).


Каждая работа содержит задания разных уровней сложности. Это позволит учителю учесть возможности и интересы каждого ученика и путем подбора тех или иных заданий сформировать для него индивидуальную образовательную траекторию.

К ряду практических работ подготовлены файлы-заготовки, их можно скачать на сайте ИНФО:

http://infojournal.ru/journals/school/school_09-2015/

Работа 1. Поиск информации в сети Интернет

Задание 1. Знакомство с браузером.

1. Займите рабочее место за компьютером.
2. Подключитесь к Интернету, запустив браузер Chrome (пиктограмма ) или другой (по указанию учителя). Для этого можно:
 - установить указатель мыши на соответствующую пиктограмму;
 - дважды быстро нажать и отпустить левую кнопку мыши.
3. Ознакомьтесь с интерфейсом программы. Для этого нужно последовательно установить указатель мыши на каждый из элементов окна браузера и прочесть всплывающие надписи.
4. Впишите в рамки (рис. 1) номера — названия элементов окна браузера:
 - 1) Адресная строка (поле *Адрес*).
 - 2) Кнопка *Закрыть*.

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_09-2015/

Таблица

№ п/п	Название работы	Формируемые умения
1	Поиск информации в сети Интернет	<ul style="list-style-type: none"> • Перемещаться по WWW с помощью браузера; • искать информацию точно по фразе, по всем словам, по любому из слов; • искать текстовую, графическую и видеоинформацию; • делать закладки; • сохранять найденные текстовые и графические материалы; • использовать поисковую систему для получения необходимой информации
2	Компьютеры и их история	<ul style="list-style-type: none"> • Искать информацию в сети Интернет; • использовать онлайн-инструменты
3	Устройства персонального компьютера	<ul style="list-style-type: none"> • Подключать внешние устройства персонального компьютера; • определять имя, тип процессора и его характеристики, характеристики оперативной памяти, тип используемой операционной системы компьютера
4	Программное обеспечение компьютера	<ul style="list-style-type: none"> • Выполнять классификацию программного обеспечения; • подбирать платное и бесплатное программное обеспечение для решения той или иной задачи; • выполнять проверку на вирусы
5	Работа с объектами файловой системы	<ul style="list-style-type: none"> • Извлекать файлы из архива и создавать архивы; • находить файл по его полному имени; • выполнять сортировку файлов по имени, типу и размеру; • создавать дерево каталогов; • удалять, копировать и перемещать файлы
6	Настройка пользовательского интерфейса	<ul style="list-style-type: none"> • Изменять свойства рабочего стола; • изменять свойства панели задач; • упорядочивать объекты на рабочем столе; • изменять размеры и перемещать окна приложений; • переключаться между приложениями
7	Обработка и создание растровых изображений	<ul style="list-style-type: none"> • Создавать изображения в графическом редакторе; • редактировать готовые изображения, в том числе удаляя их фрагменты и добавляя надписи; • создавать изображения с помощью выделения и преобразования их фрагментов; • получать и преобразовывать копию экрана компьютера; • выполнять художественную обработку графических изображений
8	Создание векторных изображений	<ul style="list-style-type: none"> • Создавать изображения средствами векторной графики; • использовать автофигуры для создания графических изображений; • строить произвольные фигуры с помощью кривых; • строить несложные объемные изображения; • писать программы для исполнителя Рисователь
9	Создание текстовых документов	<ul style="list-style-type: none"> • Вводить тексты на русском и английском языках; • редактировать текстовые документы (удалять, вставлять и заменять отдельные символы и фрагменты текста; искать и заменять, копировать и перемещать фрагменты текста; склеивать и разрезать строки); • форматировать текстовые документы (изменять свойства отдельных символов, абзацев и всего документа); • вставлять в текст специальные символы и формулы; • создавать списки, таблицы, схемы; • вставлять рисунки и делать надписи к отдельным элементам рисунка
10	Подготовка реферата «История развития компьютерной техники»	<ul style="list-style-type: none"> • Создавать многостраничные текстовые документы; • вставлять в текстовые документы таблицы и изображения, колонтитулы и сноски; • создавать оглавление; • устанавливать параметры страницы; • оформлять реферат в соответствии с заданными требованиями
11	Компьютерный перевод текстов	<ul style="list-style-type: none"> • Пользоваться онлайн-переводчиками
12	Сканирование и распознавание текстовых документов	<ul style="list-style-type: none"> • Сканировать текстовые документы; • выполнять распознавание отсканированных текстовых документов с помощью онлайн-сервиса
13	Разработка презентации	<ul style="list-style-type: none"> • Создавать компьютерные презентации, включать в презентацию аудиовизуальные объекты
14	Создание анимации	<ul style="list-style-type: none"> • Использовать различные инструменты для создания простых анимаций, представляющих собой последовательность кадров, сменяющихся многократно
15	Создание видеофильма	<ul style="list-style-type: none"> • Создавать простые видеофильмы

- 3) Кнопка *Развернуть (Восстановить)*.
- 4) Кнопка *Свернуть*.
- 5) Вкладка.
- 6) Удалить вкладку.
- 7) Добавить новую вкладку.
- 8) Панель закладок.
- 9) Настройка и управление.
- 10) Вернуться.
- 11) Перейти вперед.
- 12) Обновить страницу.
- 13) Добавить страницу в закладки.

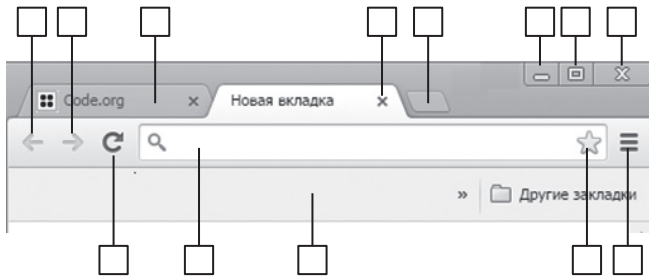


Рис. 1

5. В поле *Адрес* введите: <http://www.whatbrowser.org/> и нажмите клавишу **Enter**. Внимательно изучите содержание этой страницы. Будьте готовы ответить на вопросы:
 - Что такое браузер?
 - Какой браузер вы используете?
 - Почему браузер желательно регулярно обновлять?
6. Закройте вкладку или вернитесь назад (щелкните мышью на соответствующих кнопках окна браузера).
7. В поле *Адрес* введите: <http://bebras.ru/> и нажмите клавишу **Enter** — вы окажетесь на сайте международного конкурса по информатике «Бобер». Добавьте просматриваемую страницу в закладки — щелкните мышью на значке звездочки ☆.
8. Перейдите в раздел «Примеры соревнований», щелкнув мышью на соответствующем пункте вертикального меню. В разделе «Примеры соревнований» найдите пробный тур для учащихся VII—VIII классов и перейдите по гиперссылке «Начать». При наличии времени попытайтесь выполнить задания. Если на уроке времени для выполнения заданий нет, то вернитесь к их выполнению дома.
9. Завершите работу с программами и компьютером (по указанию учителя).

Задание 2. Знакомство с поисковой системой.

1. Запустите браузер Chrome или другой (по указанию учителя).
2. Если в рабочей зоне браузера вы видите изображение, представленное на рисунке 2, то это значит, что загружена поисковая система Google.



Введите поисковый запрос или URL



Рис. 2

Если такого изображения нет, то загрузите поисковую систему Google (наберите в поле *Адрес*: google.ru).

3. Введите в строку поиска запрос: **конкурс кит** (можете воспользоваться голосовым вводом запроса). Обратите внимание на результаты поиска (рис. 3).

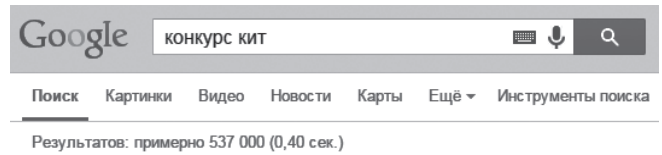


Рис. 3

4. Зайдите на главную страницу сайта конкурса «Компьютеры. Информатика. Технологии». Добавьте страницу в закладки.
5. Найдите на главной странице ссылку на задания последнего прошедшего конкурса. Откройте задания для учеников VI—VII классов. Сохраните задания в своей личной папке (пиктограмма 📁). Закройте вкладку с заданиями. Закройте вкладку с главной страницей конкурса «Компьютеры. Информатика. Технологии».
6. Вернитесь на страницу с результатами запроса **конкурс кит**. Перейдите к картинкам, найденным по этому запросу. Вы увидите множество небольших картинок в режиме предварительного просмотра (небольшие изображения).
7. Выберите по своему усмотрению одну из картинок с изображением символа конкурса «Компьютеры. Информатика. Технологии». Щелкните на выбранной картинке левой кнопкой мыши, чтобы увидеть ее в реальном размере. Сохраните выбранное изображение в своей личной папке:
 - подведите указатель мыши к картинке;
 - нажмите правую кнопку мыши и выберите в контекстном меню команду *Сохранить рисунок как*;
 - сохраните графический файл в нужной папке, не меняя типа файла.
8. Перейдите на страницу сайта, на которой размещена сохраненная вами картинка, и добавьте ее в закладки. Это позволит вам быстро найти и указать ссылку на ресурс, откуда была взята картинка, если вы захотите разместить ее на собственном сайте. Это необходимо делать для того, чтобы не нарушать авторские права.

Задание 3. Поисковые запросы.

1. Запустите браузер Chrome или другой (по указанию учителя).
2. Выполните поиск точно по фразе. Поочередно в строку поиска введите следующие фразы-запросы (фраза вводится в кавычках):
 - 1) "месяц — серебряный шар со свечою внутри"
 - 2) "на смену декаблям приходят январь"
 - 3) "красные цветы мои в садике завяли все"
 - 4) "день промыт, как стекло, только этого мало"

Выясните, из каких произведений эти строки и кто их автор.

По результатам поиска заполните таблицу:

№ п/п	Фраза	Произведение	Автор
1	месяц — серебряный шар со свечою внутри		
2	на смену декаблям приходят январь		
3	красные цветы мои в садике завяли все		
4	день промыт, как стекло, только этого мало		

3. Выполните поиск по словам. Используя поисковую систему, заполните следующую таблицу:

№ п/п	Запрос	Количество найденных страниц
1	«канарейки и щеглы»	
2	канарейки и щеглы	
3	канарейки & щеглы	
4	канарейки щеглы	
5	канарейки -щеглы	
6	-канарейки щеглы	

4. Используя поисковую систему, заполните таблицу о людях, изменивших наш мир:

№ п/п	Фамилия, имя, отчество	Годы жизни	Род занятий
1	Пирогов Николай Иванович		
2	Королев Сергей Павлович		
3	Выготский Лев Семенович		
4	Папанин Иван Дмитриевич		
5	Высоцкий Владимир Семенович		

Задание 4. Поиск видеоинформации.

1. Запустите браузер Chrome или другой (по указанию учителя).
2. Перейдите на страницу: <http://www.youtube.com>. Введите в окно поиска название видео: «Социальные сети и дети».
3. Щелкните на найденном видео — начнется его воспроизведение. Внимательно просмотрите видеоролик «Соцсети. Не ведите себя как дети!» от начала до конца — это поможет вам избежать возможных опасностей при использовании сети Интернет.
4. Завершите работу с программами и компьютером (по указанию учителя).

Теперь вы умеете:

- перемещаться по WWW с помощью браузера;
- искать информацию точно по фразе; по всем словам; по любому из слов;

- искать текстовую, графическую и видеоинформацию;
- делать закладки;
- сохранять найденные текстовые и графические материалы;
- использовать поисковую систему для получения необходимой информации.

Работа 2. Компьютеры и их история

Задание 1. Лента времени.

1. Посетите Виртуальный музей информатики: <http://informat444.narod.ru/museum/>, другие интернет-ресурсы и найдите там информацию о следующих выдающихся ученых и изобретателях, работы которых сделали возможным появление компьютеров:
 - Блез Паскаль,
 - Герман Холлерит,
 - Готфрид Вильгельм Лейбниц,
 - Джон Непер,
 - Жозеф Мари Жаккард,
 - Клод Шапп,
 - Пафнутий Львович Чебышев,
 - Самуэль Морзе,
 - Чарлз Бэббидж.
2. Создайте ленту времени, на которой разместите портреты этих людей и укажите годы их жизни. Для создания ленты времени можно воспользоваться сервисом TimeRime*: <http://timerime.com/>, имеющимся в свободном доступе в сети Интернет.

Задание 2. Кроссворд.**

1. В сети Интернет найдите информацию, необходимую для разгадывания кроссворда, составленного из фамилий людей, внесших значительный вклад в развитие вычислительной техники.
2. Разгадайте кроссворд.

		2			5	6		8
			3	4			7	
1								
	Е	Е	Ш	Ш	Е	Е	К	К

* Методика работы с этим сервисом доступно изложена на сайте: <https://docs.google.com/document/pub?id=1VjMAv8wR46xk1Qf8s6DqYAtkV9i7qlbT1lvq51DF9G4>

** Идея заимствована из разработки внеклассного мероприятия С. Н. Серых.

Вопросы:

- 1) Ученый, имя которого связано с созданием лаборатории по разработке ЭВМ, названной МЭСМ (Малая электронная счетная машина); создатель первого компьютера в континентальной Европе.
 - 2) Под его руководством были разработаны ЭВМ «Стрела», «Урал-1».
 - 3) Он является одним из зачинателей теоретического и системного программирования, создателем сибирской школы информатики. Его существенный вклад в становление информатики как новой отрасли науки и нового феномена общественной жизни широко признан в нашей стране и за рубежом.
 - 4) Один из основоположников современных информационных технологий и мыслитель-футуролог, впервые предложивший механизм перекрестных ссылок, аналогичный используемому в гипертексте.
 - 5) Один из первых исследователей человеко-машинного интерфейса и изобретатель мыши.
 - 6) Автор проекта первой электронно-счетной машины.
 - 7) Изобретатель, впервые продемонстрировавший работу устройства под управлением перфокарт.
 - 8) Изобретатель счетного устройства. В его честь назван язык программирования.
3. Придумайте и создайте свой интерактивный кроссворд по истории компьютера, состоящий не менее чем из пяти слов. Используйте онлайн-инструмент Фабрика кроссвордов.

Задание 3. Архитектура фон Неймана.

1. На сайте Единой коллекции цифровых образовательных ресурсов: <http://sc.edu.ru/> найдите ресурс «Программа-тренажер «Устройство компьютера — 1»».
2. Выполните задание. Результат покажите учителю.

Задание 4. Суперкомпьютеры.

1. Зайдите на страницу: <http://www.youtube.com>
2. Введите в окно поиска название видео: «Сделано в России. Суперкомпьютер Ломоносов. РБК».
3. Посмотрите видеоролик.
4. Завершите работу с программами и компьютером (по указанию учителя).

Теперь вы умеете:

- искать информацию в сети Интернет;
- использовать онлайн-инструменты.

Работа 3. Устройства персонального компьютера**Задание 1. Подключение внешних устройств персонального компьютера*.**

1. Выключите компьютер (обязательно выньте вилку шнура питания из розетки).
2. Выключите монитор.
3. Отсоедините монитор от компьютера.
4. Отсоедините мышь и клавиатуру.

5. Отсоедините все остальные имеющиеся внешние устройства и кабели (принтер, сканер, колонки, сетевую кабель и т. д.).
6. Результат покажите учителю.
7. Подключите к компьютеру монитор.
8. Подключите к компьютеру мышь и клавиатуру.
9. Подключите к компьютеру прочие внешние устройства и кабели (принтер, сканер, колонки, сетевую кабель и т. д.).
10. Подключите шнур питания компьютера к сети.
11. Включите компьютер и монитор.
12. Войдите в Интернет. На сайте Единой коллекции цифровых образовательных ресурсов: <http://sc.edu.ru/> найдите ресурс «Программа-тренажер «Устройство компьютера — 2»». Выполните задание. Результат покажите учителю.

Задание 2. Свойства компьютера.

1. Познакомьтесь со свойствами компьютера, к которому вы имеете доступ в школе и (или) дома. Для этого можно:
 - 1) вызвать контекстное меню для объекта *Мой компьютер* (установить указатель мыши на объект, нажать и отпустить правую кнопку мыши);
 - 2) выбрать пункт *Свойства*;
 - 3) прочитать имя компьютера, тип процессора и его характеристики, характеристики оперативной памяти, тип используемой операционной системы.
2. Представьте полученную информацию в следующей таблице:

Свойство	Школьный компьютер	Домашний компьютер
Имя компьютера		
Процессор		
Оперативная память (размер)		
Тип операционной системы		

Задание 3. Устройства внешней памяти.

1. Познакомьтесь с устройствами внешней памяти компьютера, к которому вы имеете доступ в школе и (или) дома. Для этого можно:
 - 1) открыть объект *Мой компьютер* (установить указатель мыши на объект, дважды быстро нажать и отпустить левую кнопку мыши);
 - 2) открыть контекстное меню диска С:;
 - 3) выбрать пункт *Свойства*;
 - 4) изучить информацию в открывшемся окне;
 - 5) закрыть окно;
 - 6) изучить информацию обо всех имеющихся на компьютере устройствах внешней памяти.
2. Представьте информацию о диске С: в следующей таблице:

Емкость диска С:	Школьный компьютер	Домашний компьютер
Всего		
Занято		
Свободно		

* Идея задания заимствована из практикума И. Г. Семакина.

Задание 4. Пары.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/1204571>
2. Рассмотрите карточки с сокращенными названиями компьютерных устройств и их графическими изображениями. Перетащите соответствующие карточки одну на другую (карточки должны соединиться). При затруднении определения пары воспользуйтесь материалом учебника [2].
3. После выполнения задания щелкните мышью на кнопке *Перепроверить решение*, покажите результат учителю.

Теперь вы умеете:

- подключать внешние устройства персонального компьютера;
- определять имя, тип процессора и его характеристики, характеристики оперативной памяти, тип используемой операционной системы компьютера.

Работа 4. Программное обеспечение компьютера**Задание 1. Основные понятия.**

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/1115111>
2. Постройте истинные утверждения, соединив соответствующие пары.
3. После выполнения задания щелкните мышью на кнопке *Перепроверить решение*, покажите результат учителю.

Задание 2. Операционные системы.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/1350946>
2. Выясните достоинства и недостатки самых популярных операционных систем.
3. Покажите результат учителю.

Задание 3. Классификация программного обеспечения.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/256185>
2. Выполните классификацию программного обеспечения, отнеся его к одной из четырех групп.
3. После выполнения задания щелкните мышью на кнопке *Перепроверить решение*, покажите результат учителю.

Задание 4. Значки программ.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/1029324>
2. Составьте пары: название программы — значок программы.
3. Покажите результат учителю.

Задание 5. Исследование программного обеспечения компьютера.

1. Выясните, какая операционная система установлена на компьютере, к которому вы имеете доступ в школе и (или) дома.

Компьютер	Операционная система
Школьный	
Домашний	

Запишите последовательность действий, с помощью которой вы это установили: _____

2. Выясните, какое программное обеспечение (ПО) установлено на компьютере, к которому вы имеете доступ в школе и (или) дома.

ПО	В школе	Дома
Программы для создания текстовых документов		
Программы для работы с изображениями		
Программы для создания презентаций		
Программы для работы с видео и звуком		
Программы для выполнения вычислений		
Программы для виртуального общения		
Антивирусы		
Архиваторы		

Запишите последовательность действий, с помощью которой вы это установили: _____

Задание 6. Правовые нормы использования программного обеспечения.

1. Выйдите в Интернет.
2. Найдите информацию о платных и бесплатных программах, позволяющих решать следующие задачи:

Задача	Платные программы	Бесплатные программы
Доступ к ресурсам компьютера		
Создание текстовых документов		
Обработка фотографий		
Создание графических изображений		
Создание презентаций		
Создание видеороликов		
Обработка звука		
Выполнение расчетов		
Виртуальное общение		
Антивирусная защита		

Запишите названия найденных программ в таблице.

Задание 7. Защита от вирусов.

1. Запустите антивирусную программу, установленную на компьютере.
2. Выберите следующие настройки: лечить, а если лечение не удастся, зараженный файл поместить в папку карантина.
3. Выбрать устройство (по указанию учителя) и проверить его на наличие вирусов.

Теперь вы умеете:

- выполнять классификацию программного обеспечения;
- подбирать платное и бесплатное программное обеспечение для решения той или иной задачи;
- выполнять проверку на вирусы.

Работа 5. Работа с объектами файловой системы**Задание 1. Извлечение файлов из архива.**

1. На рабочем столе найдите архив **ФАЙЛЫ** (ФАЙЛЫ).
2. Вызовите контекстное меню этого объекта (щелчок правой кнопкой мыши) и выясните размер архива.
3. Вызовите контекстное меню архива **ФАЙЛЫ** и выберите в нем пункт *Извлечь в ФАЙЛЫ* (рис. 4).

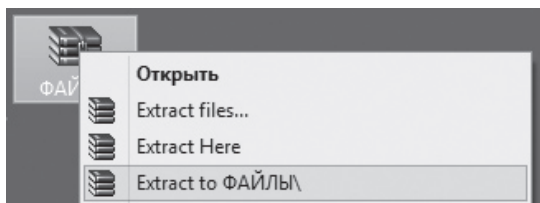


Рис. 4

4. Вызовите контекстное меню появившейся папки **ФАЙЛЫ** и выясните ее размер.
5. Запишите размер архива **ФАЙЛЫ** и размер папки **ФАЙЛЫ**:

Размер архива	
Размер папки	

6. Переместите папку **ФАЙЛЫ** в свою личную папку.

Задание 2. Путь к файлу. Полное имя файла*.

1. Выберите один из путей к файлу (один из вариантов 1–11).
2. Откройте соответствующий файл, прочтите в нем фрагмент крылатого выражения и полный путь к файлу с продолжением этого крылатого выражения.
3. Запишите фрагмент крылатого выражения и следуйте по указанному пути.
4. Повторяйте п. 2–3, пока не «соберете» крылатое выражение полностью.

Выполните 5–6 вариантов заданий из предложенных (по своему выбору).

Вариант 1	C:\DIRTEST\TEST2\SYSTEM\T1.EXE
Крылатая фраза:	
Вариант 2	C:\DIRTEST\WORK\FRIENDS\VLADIMIR\PIC3.GIF
Крылатая фраза:	
Вариант 3	C:\DIRTEST\TEST3\SCHOOL\TEXT1.TXT
Крылатая фраза:	
Вариант 4	C:\DIRTEST\TEST3\HOME\W21.HTM
Крылатая фраза:	
Вариант 5	C:\DIRTEST\TEST1\MYDOC\N1\NNN.EXE
Крылатая фраза:	
Вариант 6	C:\DIRTEST\WORK\MY1\STORY\PIC1.GIF
Крылатая фраза:	
Вариант 7	C:\DIRTEST\TEST1\NEWDOC\1\NEW.TXT
Крылатая фраза:	
Вариант 8	C:\DIRTEST\MYDAY\NOW\B2\PAGE2.HTM
Крылатая фраза:	
Вариант 9	C:\DIRTEST\LONG\1\PRG1.EXE
Крылатая фраза:	
Вариант 10	C:\DIRTEST\MYDAY\NEWDAY\PIC12.GIF
Крылатая фраза:	
Вариант 11	C:\DIRTEST\WORK\MY1\STORY\MY1.TXT
Крылатая фраза:	

Задание 3. Имена файлов.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/424457>
2. Определите, каким может, а каким не может быть имя файла.
3. После выполнения задания щелкните мышью на кнопке *Перепроверить решение*, покажите результат учителю.

* Использована идея учителя Н. Л. Короткова: <http://nk-inform.pnrod.ru/>

Задание 4. Работа с файлами.

1. Перейдите в каталог **ФАЙЛЫ**.
2. Упорядочивая файлы по имени, типу или размеру, ответьте на следующие вопросы:
 - 1) Сколько в каталоге **ФАЙЛЫ** содержится файлов, размер которых превышает 1000 Кб?

 - 2) Сколько в каталоге **ФАЙЛЫ** содержится рисунков?

 - 3) Сколько в каталоге **ФАЙЛЫ** содержится файлов, имя которых начинается с русской буквы «П»?

3. Удалите из каталога **ФАЙЛЫ** все файлы, имеющие расширение **TXT**.
4. Переименуйте файлы с произведениями А. П. Чехова (всего 10 файлов), указав в новом имени порядковый номер и фамилию автора (например, *1Чехов.doc*).

Задание 5. Создание дерева каталогов.

1. В каталоге **ФАЙЛЫ** создайте дерево каталогов следующей структуры (рис. 5):

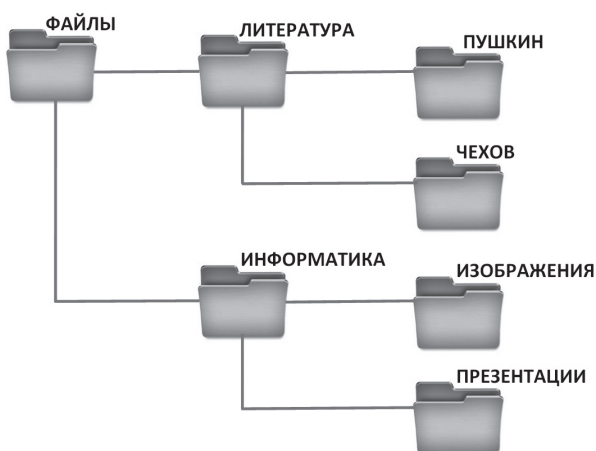


Рис. 5

2. В подкаталоги **ПУШКИН**, **ЧЕХОВ**, **ИЗОБРАЖЕНИЯ**, **ПРЕЗЕНТАЦИИ** переместите соответствующие файлы из каталога **ФАЙЛЫ**.
3. Определите размер всех графических файлов, содержащихся в подкаталоге **ЛИТЕРАТУРА**.
4. Заархивируйте содержимое каталога **ИНФОРМАТИКА**; полученный архив разместите в каталоге **ФАЙЛЫ**.

Теперь вы умеете:

- извлекать файлы из архива и создавать архивы;
- находить файл по его полному имени;
- выполнять сортировку файлов по имени, типу и размеру;
- создавать дерево каталогов;
- удалять, копировать и перемещать файлы.

Работа 6. Настройка пользовательского интерфейса**Задание 1. Оформление рабочего стола.**

1. Откройте контекстное меню рабочего стола, щелкнув правой кнопкой мыши на его любом свободном участке.

2. Выберите в контекстном меню пункт **Вид**. Измените размеры значков.
3. Выберите в контекстном меню пункт **Сортировка**. Упорядочите все значки, расположенные на рабочем столе, по имени.
4. Выберите в контекстном меню пункт **Персонализация (Свойства)**.
5. Измените по своему усмотрению тему рабочего стола; измените отдельно каждый из компонентов темы рабочего стола: фоновый рисунок, цвет границ окон и панели задач, звуки и заставку.
6. Установите по своему усмотрению несколько гаджетов рабочего стола — настраиваемых мини-программ, позволяющих отображать данные без необходимости открывать новое окно (Погода, Часы, Календарь и др.).
7. Верните измененные параметры в исходное положение.

Задание 2. Объекты рабочего стола.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/1179245>
2. Установите соответствие между отмеченными объектами рабочего стола и их названиями.
3. После выполнения задания щелкните мышью на кнопке **Перепроверить решение**, покажите результат учителю.

Задание 3. Панель задач.

1. По всплывающим подсказкам узнайте назначение всех значков и кнопок, имеющихся на панели задач.
2. Откройте контекстное меню панели задач.
3. Измените положение панели задач.
4. Верните измененные параметры в исходное положение.

Задание 4. Окно.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/498261>
2. Укажите элемент окна, с помощью которого можно выполнить то или иное действие.
3. Покажите результат учителю.

Задание 5. Операции с окнами.

1. Запустите программу Блокнот и выполните с ее окном следующие операции:
 - измените размер окна;
 - переместите окно в правый нижний угол рабочего стола;
 - разверните окно на весь экран;
 - сверните окно в кнопку панели задач;
 - восстановите предыдущий размер окна;
 - закройте окно программы.
2. Запустите программы Калькулятор, WordPad, Paint и выполните следующие операции с окнами:
 - разместите окна открытых программ одно под другим (стопкой);
 - разместите окна открытых программ рядом;
 - разместите окна открытых программ каскадом;
 - сверните все окна;
 - переключитесь между окнами программ.
3. Закройте окно программы Paint.
4. С помощью программы Калькулятор выполните требуемые вычисления. Ответы (в форме раз-

вернутых предложений) запишите в программе WordPad и сохраните в личной папке в файле с именем *Человек.doc*.

- Количество сердцебиений человека за год составляет 36 800 000. Определите среднюю скорость биения сердца в ударах в минуту.
- Через печень человека в течение одной минуты протекает 1,5 литра крови. Подсчитайте, какое количество крови пропускает печень за сутки.
- Нервные импульсы в человеческом теле передаются со скоростью примерно 90 метров в секунду. Выразите эту скорость в километрах в час.

Задание 6. Объекты графического интерфейса.

1. Выйдите в Интернет и перейдите на страницу: <http://learningapps.org/700856>
2. Установите соответствие между объектами графического интерфейса и их названиями.
3. После выполнения задания щелкните мышью на кнопке *Перепроверить решение*, покажите результат учителю.

Теперь вы умеете:

- изменять свойства рабочего стола;
- изменять свойства панели задач;
- упорядочивать объекты на рабочем столе;
- изменять размеры и перемещать окна приложений;
- переключаться между приложениями.

Работа 7. Обработка и создание растровых изображений

Задание 1. Работа с графическими примитивами

1. Запустите графический редактор Paint.
2. Установите размеры области для рисования: ширина — 1024 точки, высота — 512 точек.
3. Повторите рисунок 6, используя инструменты *Линия*, *Прямоугольник*, *Скругленный прямоугольник* и *Эллипс*.



Рис. 6

4. Сохраните результат работы в личной папке:
 - в файле *p1.bmp* как 24-разрядный рисунок;
 - в файле *p2.bmp* как 256-цветный рисунок;
 - в файле *p3.bmp* как 16-цветный рисунок;
 - в файле *p4.bmp* как монохромный рисунок;
 - в файле *p5.jpeg*;
 - в файле *p5.gif*.

Сравните размеры полученных файлов и качество сохраненных в них изображений.

Задание 2. Выделение и удаление фрагментов.

1. В графическом редакторе откройте файл *Устройства.bmp* (рис. 7).



Рис. 7

2. Оставьте на рисунке только устройства ввода, а все лишнее удалите с помощью инструментов *Выделение* и *Выделение произвольной области*.
3. Сохраните рисунок в личной папке под именем *Устройства_ввода*.

Задание 3. Перемещение фрагментов.

1. В графическом редакторе откройте файл *Сказка.bmp* (рис. 8).



Рис. 8

2. Поочередно выделите прямоугольные прозрачные фрагменты и переместите их так, чтобы сказочные персонажи обрели свой истинный вид.
3. Сохраните результат работы в личной папке.

Задание 4. Преобразование фрагментов.

1. В графическом редакторе откройте файл *Стрекоза.bmp* (рис. 9).

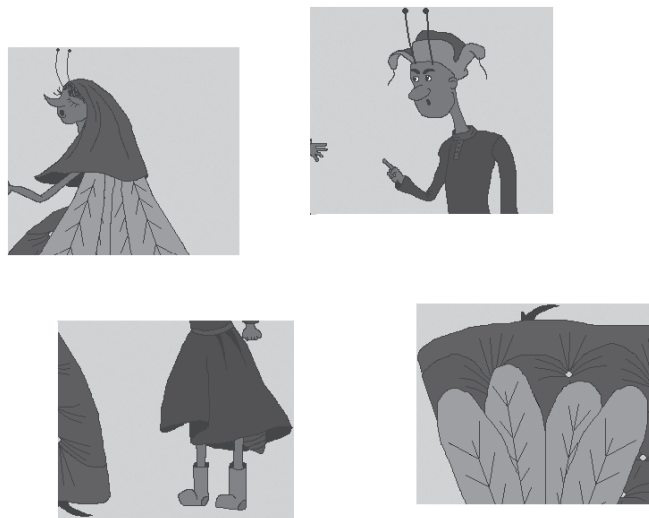


Рис. 9

2. Поочередно выделите прямоугольные фрагменты, при необходимости поверните и переместите их так, чтобы получилась иллюстрация к басне И. А. Крылова «Стрекоза и муравей».
3. Сохраните результат работы под тем же именем в личной папке.

Задание 5. Конструирование сложных объектов из графических примитивов.

1. Запустите графический редактор.
2. Изобразите один из рисунков 10, 11.

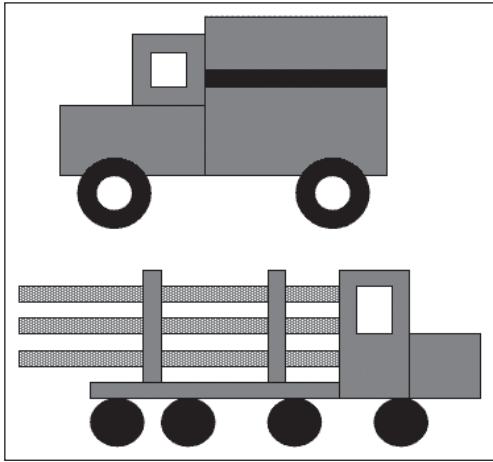


Рис. 10

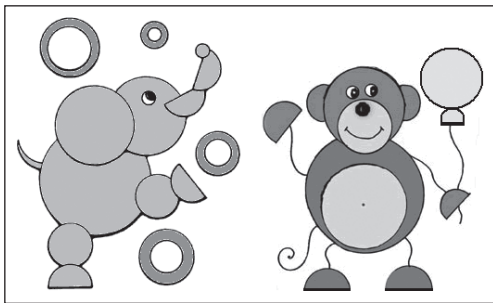


Рис. 11

3. Сохраните результат работы в личной папке под именем *Мой_рисунок*.

Задание 6. Создание надписей.

1. В графическом редакторе откройте файл *Панель.bmp* (рис. 12).

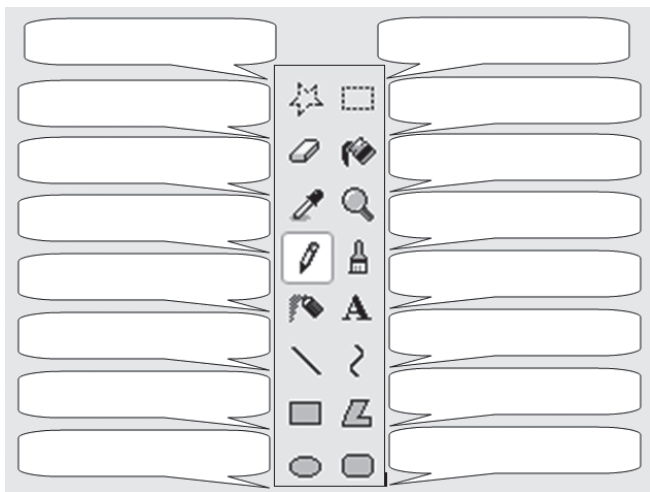


Рис. 12

2. Подпишите инструменты графического редактора Paint.
3. Сохраните рисунок в личной папке в файле *Панель1.bmp*.

Задание 7. Копирование фрагментов.

1. Рассмотрите представленный на рисунке 13 орнамент. Выделите в нем повторяющийся фрагмент.

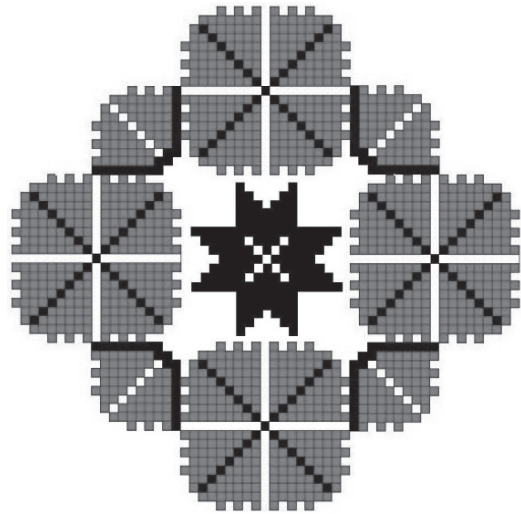


Рис. 13

2. Изобразите повторяющийся фрагмент в графическом редакторе.
3. Путем копирования и поворотов фрагмента воспроизведите орнамент.
4. Сохраните рисунок в личной папке под именем *Орнамент*.

Задание 8. Работа с несколькими файлами.

1. В графическом редакторе откройте файл *Схема.bmp* (рис. 14).



Рис. 14

2. Проиллюстрируйте схему, добавив в нее изображения соответствующих устройств из файлов *Оперативная Память.bmp*, *Винчестер.bmp*, *Диск.bmp*, *Дискета.bmp*, *Флэшка.bmp*. Для удобства откройте каждый из этих файлов в новых окнах. Копируйте нужные изображения в буфер обмена и вставляйте в нужные места схемы.
3. Сохраните полученный результат в личной папке под именем *Схема1*.

Задание 9. Получение копии экрана.

1. Запустите графический редактор Paint, минимизируйте его окно и сделайте копию этого окна (Alt+PrintScreen).
2. Разместите полученное изображение в рабочем поле графического редактора Paint (команда *Правка, Вставить*) и подпишите основные элементы интерфейса.
3. Сохраните результат работы в личной папке под именем *Paint.bmp*.

Задание 10. Художественная обработка изображений.

1. Запустите графический редактор Gimp.
2. Откройте в графическом редакторе файл *tamont.jpeg*.
3. Примените к исходному изображению несколько различных фильтров; получите не менее четырех вариантов обработки исходного изображения.
4. Сохраните результаты работы в файлах *tamont1.jpeg*, *tamont2.jpeg*, *tamont3.jpeg* и *tamont4.jpeg*.

Теперь вы умеете:

- создавать изображения в графическом редакторе;
- редактировать готовые изображения, в том числе удаляя их фрагменты и добавляя надписи;
- создавать изображения с помощью выделения и преобразования их фрагментов;
- получать и преобразовывать копию экрана компьютера;
- выполнять художественную обработку графических изображений.

Работа 8. Создание векторных изображений**Задание 1. Создание изображений из автофигур.**

1. Средствами рисования, имеющимися в текстовом процессоре, создайте изображение (рис. 15).



Рис. 15

2. Сохраните результат работы в личной папке под именем *Елка*.

Задание 2. Работа с кривыми.

1. Откройте в графическом редакторе OpenOffice.org Draw файл *Шаблон.odg* (рис. 16).
2. С помощью инструмента *Кривая* попробуйте как можно более точно воспроизвести контур рисунка сплошными линиями. При необходимости используйте возможности добавления и изменения узлов на кривых.

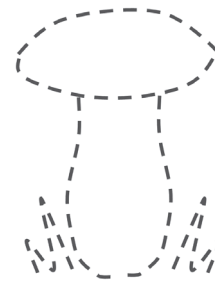


Рис. 16

3. Сохраните результат работы в личной папке под именем *Грибок*.

Задание 3. Создание блок-схем.

1. Используя автофигуры, начертите блок-схему (рис. 17).

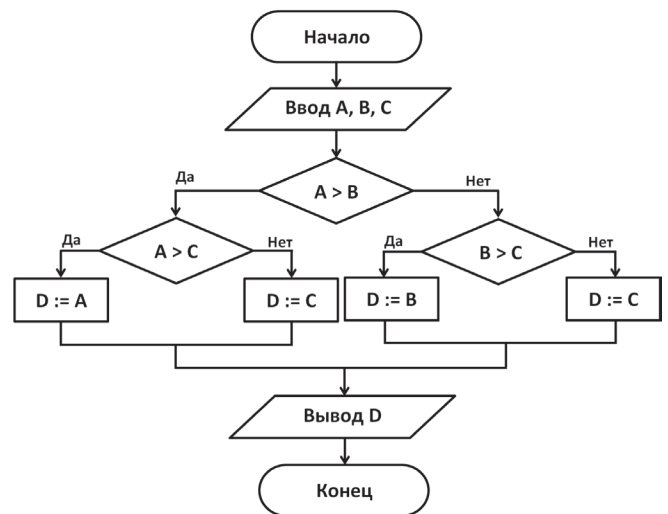


Рис. 17

2. Сохраните результат работы в личной папке под именем *Блок_схема*.
3. Как вы думаете, что получится на выходе, если в качестве значений A, B, C введены числа 15, 10 и 20 соответственно? Запишите ответ: _____

Задание 4. Трехмерные изображения.

1. Используйте возможности имеющегося в вашем распоряжении графического редактора для создания трехмерного изображения (рис. 18).
2. Сохраните результат работы в личной папке под именем *Кубики*.

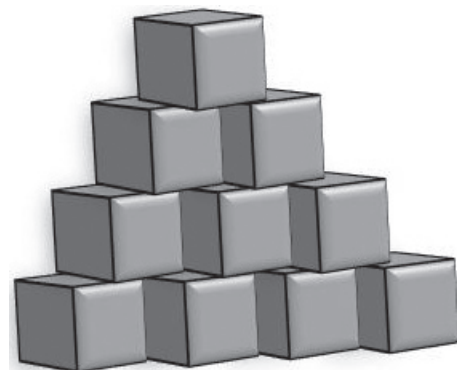


Рис. 18

Задание 5. Масштабирование растровых и векторных изображений.

1. В графическом редакторе Paint постройте изображение (рис. 19).



Рис. 19

- Сохраните результат работы в личной папке как *24-разрядный рисунок* (тип файла).
- Выделите любой фрагмент рисунка. Несколько раз уменьшите и увеличьте выделенный фрагмент. Понаблюдайте за тем, как операции масштабирования влияют на качество изображения.
- Выполните такой же рисунок в графическом редакторе OpenOffice.org Draw. Сохраните результат работы в личной папке как *Рисунок ODF* (тип файла).
- Выделите любой фрагмент рисунка. Несколько раз уменьшите и увеличьте выделенный фрагмент. Понаблюдайте за тем, как операции масштабирования влияют на качество изображения.
- Завершите работу с графическими редакторами.

Задание 6. Знакомство с исполнителем Рисователь.

1. Изучите основные команды исполнителя Рисователь:

Команда	Описание	Пример
перо (толщина, цвет)	Устанавливает ширину пера в пикселях и цвет — название любого цвета радуги, а также черный, белый и серый	перо (5, "синий")
кисть (цвет)	Устанавливает цвет заливки внутренней области рисуемой фигуры — название любого цвета радуги, а также черный, белый и серый	кисть ("красный")
линия (x1, y1, x2, y2)	Проводит линию из точки с координатами (x1, y1) в точку с координатами (x2, y2) в соответствии с состоянием пера	линия (100, 10, 200, 10)
линия в точку (x1, y1)	Проводит линию из текущей точки в точку с координатами (x1, y1) в соответствии с состоянием пера	линия (100, 10)

Команда	Описание	Пример
окружность (x, y, r)	Проводит контур окружности радиуса r с центром в точке (x, y) в соответствии с состоянием пера и закрашивает ее внутреннюю область в соответствии с состоянием кисти	окружность (300, 200, 50)
прямоугольник (x1, y1, x2, y2)	Проводит контур прямоугольника в соответствии с состоянием пера и закрашивает его внутреннюю область в соответствии с состоянием кисти; (x1, y1) и (x2, y2) — вершины диагонали прямоугольника со сторонами, параллельными осям координат	прямоугольник (10, 10, 100, 100)
эллипс (x1, y1, x2, y2)	Рисует эллипс, вписанный в соответствующий прямоугольник	эллипс (50, 10, 100, 30)
залить (x, y)	В соответствии с состоянием кисти закрашивает ограниченную область, внутри которой находится точка с заданными координатами	залить (10, 50)

2. Запустите среду «КуМир», введите и запустите на выполнение программу рисования кораблика для исполнителя Рисователь (рис. 20).

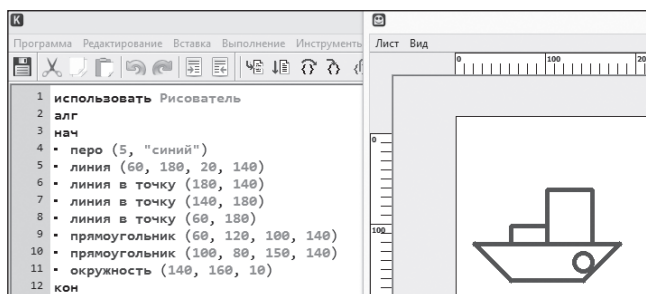


Рис. 20

- Попробуйте внести изменения в программу, экспериментируя с заливкой, цветом, толщиной линий и т. д.
- Сохраните результат работы в личной папке.

Задание 7. Программа для исполнителя Рисователь.

1. Самостоятельно разработайте для исполнителя Рисователь программу рисования одного из объектов (рис. 21).



Рис. 21

2. Сохраните результат работы в личной папке.

Задание 8. Знакомство со средой программирования PascalABC.NET.

1. Запустите среду программирования PascalABC.NET, введите и запустите на выполнение программу (рис. 22).

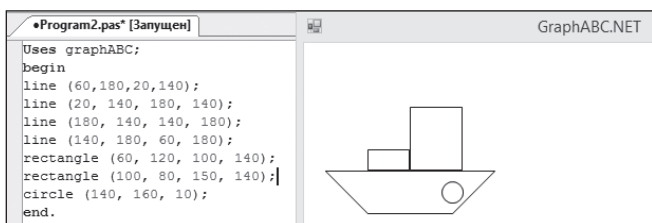


Рис. 22

2. Выясните, какой смысл имеют используемые в программе команды (операторы).
3. Попробуйте внести изменения в рисунок.
4. Сохраните результат работы в личной папке.

Теперь вы умеете:

- создавать изображения средствами векторной графики;

- использовать автофигуры для создания графических изображений;
- строить произвольные фигуры с помощью кривых;
- строить несложные объемные изображения;
- писать программы для исполнителя Рисователь.

Литературные и интернет-источники

1. Босова Л. Л., Босова А. Ю. Информатика. Программа для основной школы: 5–6 классы. 7–9 классы. М.: БИНОМ. Лаборатория знаний, 2014.

2. Босова Л. Л., Босова А. Ю. Информатика: учебник для 7 класса. 2-е изд., испр. М.: БИНОМ. Лаборатория знаний, 2014.

3. Примерная основная образовательная программа основного общего образования. Одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 8 апреля 2015 г. № 1/15). http://минобрнауки.рф/документы/938/файл/4587/POOP_OOO_reestr_2015_01.doc

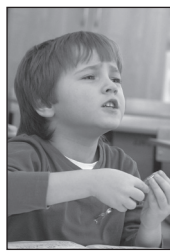
Окончание следует

КОНКУРСЫ

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



В. П. Кушникова,
учитель информатики гимназии № 64
имени В. А. Котельникова, г. Липецк



Никита Шарьгин,
ученик средней общеобразова-
вательной школы № 31,
г. Нижнекамск,
Республика Татарстан



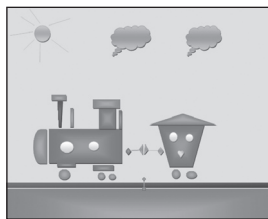
М. Н. Шарьгина,
воспитатель по интеллекту-
альному развитию центра
развития ребенка — детского
сада № 91, г. Нижнекамск,
Республика Татарстан



С. А. Иванова,
учитель начальных
классов
«Перфект-гимназии»,
г. Уссурийск,
Приморский край



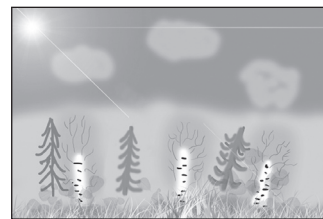
Альфия Хасаншина,
ученица средней общеобразова-
тельной школы № 8, г. Ноябрьск,
Ямало-Ненецкий автономный округ



Анна Перова,
ученица средней общеобразова-
вательной школы № 26,
г. Сочи, Краснодарский край



Анастасия Хохлова,
ученица средней общеобразова-
вательной школы № 8,
г. Ноябрьск, Ямало-Ненецкий
автономный округ



Никита Гордеев,
ученик средней общеобразова-
тельной школы № 18 имени
Виталия Яковлевича Алексеева,
г. Сургут, Ханты-Мансийский
автономный округ — Югра

Уважаемые авторы и читатели!

Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» в первом полугодии 2016 года.

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

Т. Е. Сорокина,

школа с углубленным изучением английского языка № 1319, Москва

ПОУРОЧНЫЕ РАЗРАБОТКИ К ПРОГРАММНОМУ МОДУЛЮ ПО ИНФОРМАТИКЕ ДЛЯ V КЛАССА «ПРОПЕДЕВТИКА ПРОГРАММИРОВАНИЯ СО SCRATCH»*

Аннотация

В статье представлены поурочные методические разработки к программному модулю «Пропедевтика программирования со Scratch» для использования в курсе информатики основной школы для пятого класса.

Ключевые слова: Scratch, пропедевтика программирования, программная среда, дополнительное образование, проектная деятельность, свободное программное обеспечение, алгоритмика, обучение программированию, информатика, STEAM-образование, «перевернутый» урок.

Контактная информация

Сорокина Татьяна Евгеньевна, учитель информатики школы с углубленным изучением английского языка № 1319, Москва; адрес: 109469, г. Москва, Мячковский бул., д. 21; телефон: (495) 658-61-35; e-mail: sorokina1240@yandex.ru

T. E. Sorokina,
School 1319, Moscow

LESSONS OF THE MODULE ON INFORMATICS FOR 5TH CLASS "PROPAEDEUTICS OF PROGRAMMING WITH SCRATCH"

Abstract

The article presents lessons of the module "Propaedeutics of programming with Scratch" for use in the course of informatics in secondary school for the fifth class.

Keywords: Scratch, propaedeutics of programming, software environment, extra-curriculum education, project activity, free software, algorithmics, teaching of programming languages, informatics, STEAM education, flipped lesson.

Урок 5. Линейный алгоритм.

Рисование линий исполнителем среды Scratch

Планируемые образовательные результаты:

- **предметные** — закрепление представлений о линейном алгоритме, блок-схеме линейного алгоритма, основных графических примитивах векторного редактора Draw свободного офисного пакета LibreOffice;
- **метапредметные** — формирование представления о компьютере как универсальном устройстве для создания векторных графических моделей; умение корректировать свои действия, вносить изменения в программу и отлаживать ее в соответствии с изменяющимися условиями; формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенции);
- **личностные** — формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершенных учебных задач; формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий.

Решаемые учебные задачи:

- формирование навыков решения поставленной задачи в виде последовательного выполнения команд;
- приобретение навыков изображения блок-схемы линейного алгоритма средствами редактора векторной графики;
- использование навыков изменения параметров команд программной среды Scratch для выполнения поставленной задачи;
- умение применять созданную программу для широкого спектра возможных задач.

Основные понятия, изучаемые на уроке:

- последовательное выполнение команд;
- линейный алгоритм;
- блок-схема линейного алгоритма;
- изменение параметров пера.

Необходимые средства ИКТ:

- компьютер учителя с установленной программной средой Scratch;
- мультимедийный проектор, экран или интерактивная доска;
- компьютеры учащихся с установленной программной средой Scratch;
- свободное программное обеспечение — офисный пакет LibreOffice, установленный на компьютеры учителя и учеников.

Электронные образовательные ресурсы, размещенные в сети Интернет:

- проекты на сайте интернет-сообщества Scratch: <http://scratch.mit.edu/>
- видеурок «Линейный алгоритм. Рисование линий исполнителем Scratch»: https://youtu.be/dG_rdHpzfMg

Предлагаемая методика изложения темы урока.

На уроке продолжается изучение линейных алгоритмов.

* Продолжение. Начало см.: Информатика в школе. 2015. № 8.

Для актуализации полученных знаний можно предложить следующие **задания** (на 5 мин):

- Дайте определение алгоритма.
- Приведите формы записи алгоритма.
- Назовите и нарисуйте графический примитив, который используется для обозначения начала и конца в блок-схеме.
- Напишите, что характерно для линейного алгоритма.
- Запишите «горячие» клавиши, которые используются для копирования и вставки объектов.

Далее рекомендуется **проверить домашнее задание** на создание блок-схемы линейного алгоритма.

В продолжение рассмотренных алгоритмов следует **всем вместе создать простой линейный алгоритм**.

Постановка задачи: средствами векторного графического редактора LibreOffice Draw создать блок-схему линейного алгоритма для перемещения исполнителя по экрану с опущенным пером на 150 шагов с установкой белого цвета и размера пера 20.

Блок-схема линейного алгоритма для поставленной задачи будет выглядеть следующим образом:

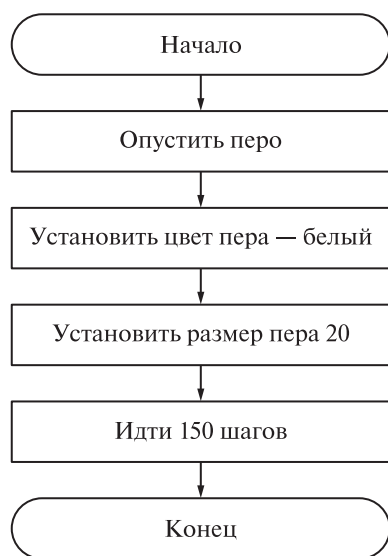


Рис. 1

Если учебный кабинет оборудован интерактивной доской, то возможна организация совместной работы учащихся по созданию блок-схемы линейного алгоритма поставленной задачи. В противном случае ученики по очереди могут создавать элементы блок-схемы на компьютере учителя с демонстрацией результата на экране.

Далее предлагается провести **практическую работу**. Обучающиеся смогут реализовать на практике совместно разработанный алгоритм средствами программной среды Scratch.

После того как программа будет создана, нужно дать ученикам возможность увидеть, как можно использовать полученную программу.

Мини-проект «Зебра». Путем перемещения исполнителя и многократного запуска программы рисуются белые полоски на сером фоне, похожие на «зебру» — место пешеходного перехода.

Для реализации мини-проекта «Зебра» обучающимся нужно:

- изменить фон — нарисовать на нем прямоугольник и залить его серым цветом;

- изменить костюм исполнителя, нарисовав красный кружок, — для удобства рисования белых линий.

Здесь учащиеся будут использовать ранее полученные знания по работе с инструментами встроенного растрового графического редактора.

Результат работы должен выглядеть следующим образом:

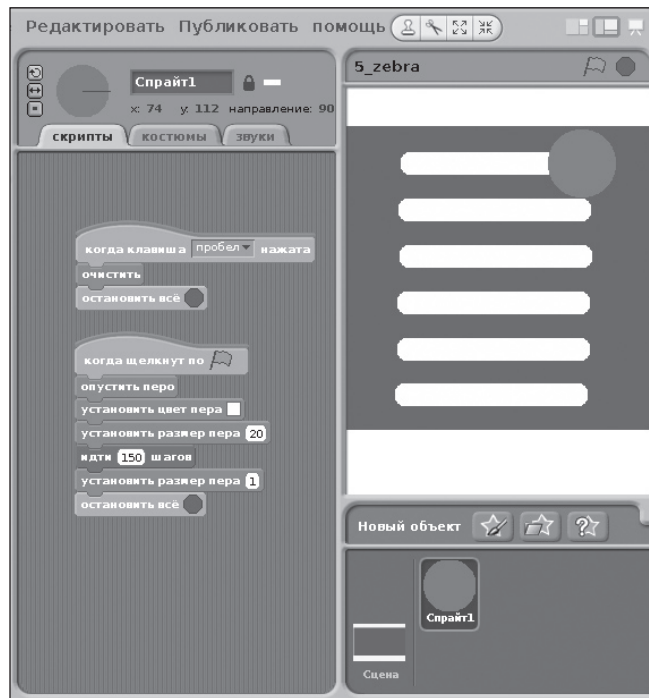


Рис. 2

Дополнительные задания для особо одаренных детей, выполнивших основное задание.

Задание. Для тех учащихся, которые быстро справились с мини-проектом «Зебра», можно предложить решение задач, которые внешне будут отличаться от предыдущей, но выполняться могут на основе подобной программы (рис. 3, 4). Чтобы получить результат, изображенный на рисунке 4, нужно после каждого запуска программы уменьшать размер пера и смещать исполнителя вправо и вверх.

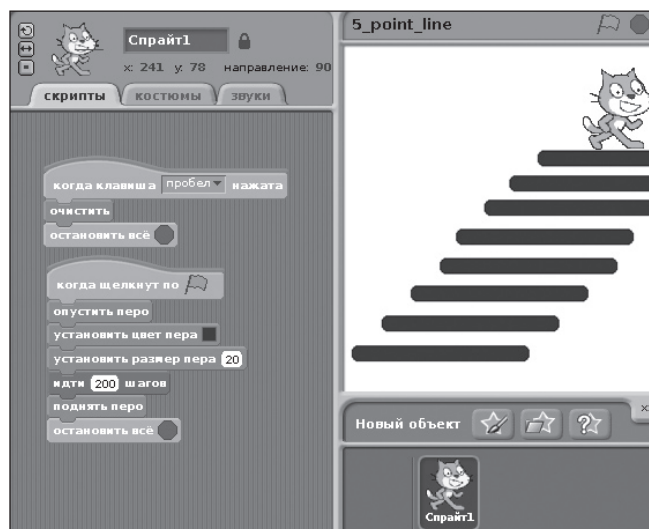


Рис. 3

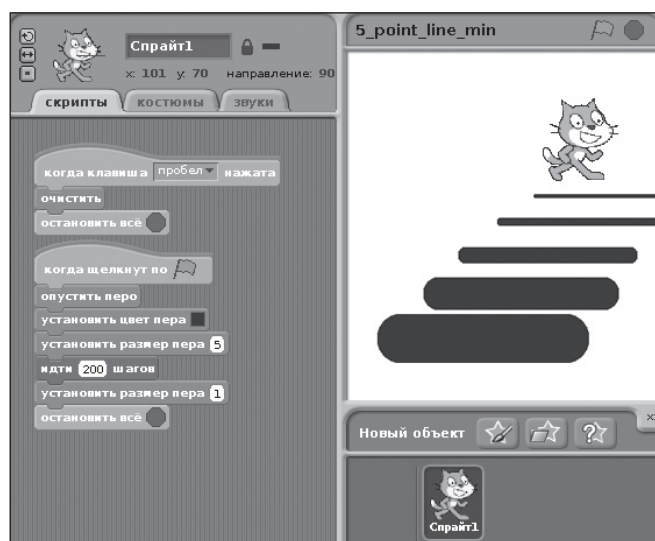


Рис. 4

Домашнее задание.

Задание. В качестве домашнего задания можно предложить обучающимся дальнейшие эксперименты с написанной программой. Возможны варианты создания мини-проектов «Штанга» (рис. 5) или «Пирамидка» (рис. 6).

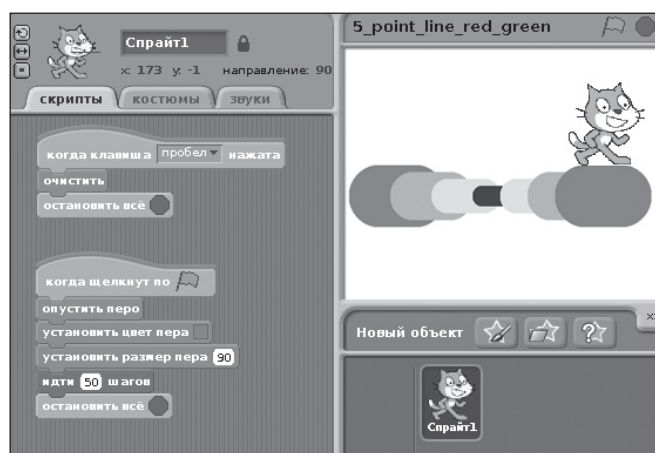


Рис. 5

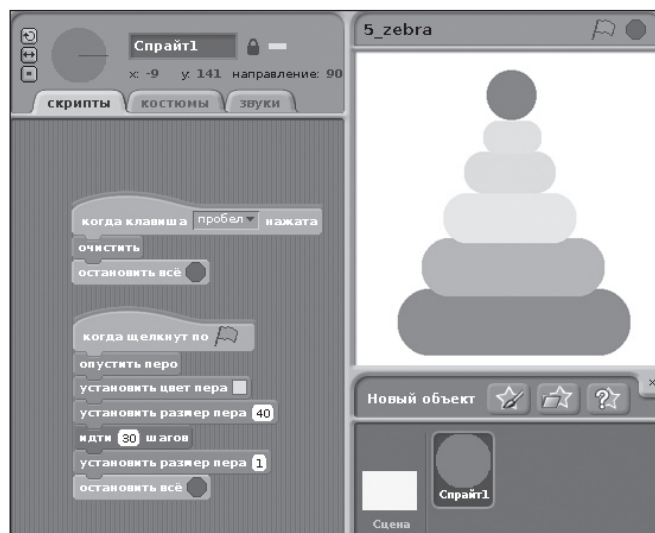


Рис. 6

Урок 6. Линейный алгоритм. Исполнитель среды Scratch рисует квадраты и прямоугольники линейно

Планируемые образовательные результаты:

- *предметные* — развитие пространственных представлений, навыков геометрических построений и моделирования процессов, развитие изобразительных умений с помощью средств ИКТ; закрепление представлений о линейном алгоритме; формирование навыков использования инструментов растрового редактора для создания изображений;
- *метапредметные* — умение корректировать свои действия, вносить изменения в программу и отлаживать ее в соответствии с изменяющимися условиями; формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенции);
- *личностные* — формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершенных учебных задач; развитие эстетического сознания через творческую деятельность на базе визуализированной среды программирования.

Решаемые учебные задачи:

- формирование навыков создания алгоритма для рисования исполнителем квадрата путем последовательного выполнения команд;
- приобретение навыков изображения блок-схемы линейного алгоритма в виде двух колонок блоков команд;
- использование навыков создания графических изображений в программной среде Scratch;
- тренировка навыков использования векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем;
- формирование навыков выбора нужного значения из предлагаемого списка вариантов;
- формирование навыков отладки программы для получения верного результата.

Основные понятия, изучаемые на уроке:

- команда *повернуть в направление*;
- пошаговое выполнение программы для ее отладки;
- единичный шаг редактирования;
- центр костюма исполнителя Scratch.

Необходимые средства ИКТ:

- компьютер учителя с установленной программной средой Scratch;
- мультимедийный проектор, экран или интерактивная доска;
- компьютеры учащихся с установленной программной средой Scratch;
- свободное программное обеспечение — офисный пакет LibreOffice, установленный на компьютеры учителя и учеников.

Электронные образовательные ресурсы, размещенные в сети Интернет:

- проекты на сайте интернет-сообщества Scratch: <http://scratch.mit.edu/>
- видеоурок «Линейный алгоритм. Scratch рисует квадраты и прямоугольники линейно»: <https://youtu.be/LxYtQZmHRMs>

Предлагаемая методика изложения темы урока.

На уроке продолжается изучение линейных алгоритмов. Кроме того, он является подготовительным перед изучением циклических конструкций.

Это первый из серии уроков, посвященных написанию программ для рисования исполнителем геометрических фигур. Учитывая, что, в соответствии с ФГОС НОО по математике, к пятому классу обучающиеся уже умеют «исследовать, распознавать и изображать геометрические фигуры», в начале урока рекомендуется **актуализировать знания, полученные в начальной школе.** Можно провести **экспресс-опрос** обучающихся, используя рисунок 7.

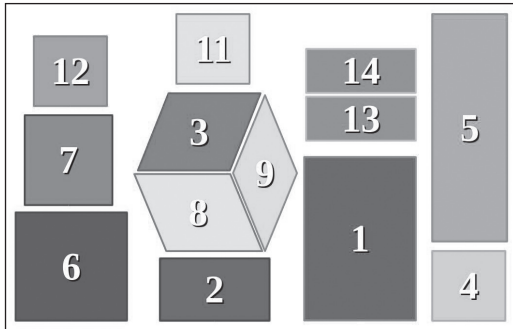


Рис. 7

Можно предложить обучающимся следующие вопросы и задания:

- Перечислите геометрические фигуры, изображенные на рисунке, называя их номера.
- Сколько прямоугольников представлено на рисунке?
- Сколько квадратов представлено на рисунке?
- Назовите номера одинаковых квадратов.
- Какой номер у самого большого из представленных квадратов?
- На что похожа фигура, состоящая из фигур с номерами 3, 8, 9?
- Какие еще сходства взаимного расположения фигур можно увидеть на рисунке?

В качестве **проверки домашнего задания** можно организовать небольшое обсуждение полученных результатов по созданию вариантов мини-проектов «Штанга» и «Пирамидка».

Далее следует провести **объяснение нового материала.**

Постановка задачи: написать программу для исполнителя, чтобы он, двигаясь по пути, совпадающему со сторонами квадрата, прочертил геометрическую фигуру квадрат, длина стороны которого равна 100 шагам.

Поскольку исполнитель рисует тогда, когда движется, то нужно понять, какие команды он должен выполнять, чтобы при движении получился квадрат. Можно провести **эксперимент** — вызвать желающего ученика и предложить ему шагами пройти так, чтобы путь движения представлял собой квадрат, например со стороной три шага (рис. 8). Обычно школьникам нравится перевоплощаться в исполнителя и они с удовольствием включаются в эксперимент. Можно повторить эксперимент дважды.

После того как движение по сторонам квадрата завершено, испытуемый вернулся в точку старта и всем стало понятно, как составить алгоритм решения задачи, можно

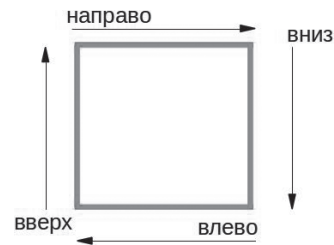


Рис. 8

предложить ученикам нарисовать блок-схему линейного алгоритма рисования квадрата в тетради (рис. 9).

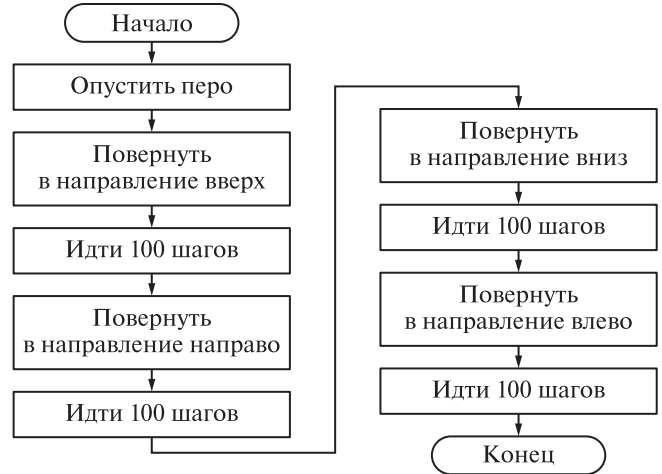


Рис. 9

Правильный вариант блок-схемы алгоритма можно показать на слайде. Если обучающиеся сидят парно, то они могут поменяться тетрадями и провести взаимопроверку решения.

Нужно сделать акцент на том, что при большом количестве блоков в линейном алгоритме его можно рисовать, как показано на рисунке 9, условно разделив лист на две колонки.

Далее нужно объяснить, где в программной среде Scratch находится новая команда.

Команда синего цвета *повернуть в направление* из блока *движение* может принимать четыре различных значения — *направо, влево, вверх и вниз* (рис. 10).

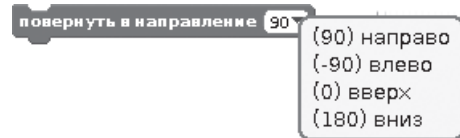


Рис. 10

Следует обратить внимание обучающихся на то, что слева от направления указаны цифры, показывающие угол в градусах. Однако на данном этапе рекомендуется выбирать направление, которое описано в словесной форме. Это связано с тем, что данный урок — шестой по счету и проводится в первом полугодии пятого класса, а градусная мера угла является одной из тем предметной области «Математика» и в отдельных учебниках попадает в программу второго полугодия пятого класса [4].

Теперь все необходимые для рисования квадрата команды известны и можно собирать программу по блок-схеме линейного алгоритма (рис. 11).

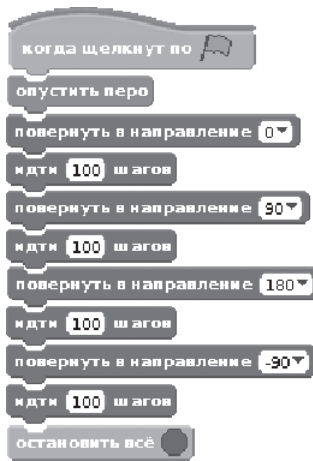


Рис. 11

После составления программы из команд рекомендуется изменить костюм исполнителя, нарисовав стрелочку во встроенном растровом редакторе. Следует показать ученикам, где находится центр костюма и для чего он используется.

Удобно в качестве костюма использовать стрелочку, чтобы при пошаговом выполнении программы было видно, в каком направлении в текущий момент находится исполнитель.

Движение в определенном заданном направлении очень важно для дальнейшего понимания, поэтому следует на данном уроке показать возможности отладки программы, используя пошаговое выполнение.

На этом этапе изучения рекомендуется не привязывать начало программы к конкретной точке на координатной плоскости. Достаточно просто переместить исполнителя в точку начала программы.

После объяснения нового материала обучающиеся могут выполнить **практическую работу**.

Выполнение практической работы — **мини-проекта «Квадрат линейно»** — можно организовать **в виде соревнования на скорость составления программы**.

После получения готовой работающей программы нужно дать возможность обучающимся поэкспериментировать с изменением параметров в программе: цвета, размера пера, количества шагов. Далее можно провести обсуждение того, кто какие результаты получил из одной и той же программы, что удивило и что понравилось.

Этот урок очень важен для понимания, поскольку понятие градусной меры не всегда просто для восприятия обучающимися.

Данный урок является подготовительным к изучению команд организации цикла, поэтому нужно сделать акцент на повторении отдельных частей программы с некоторыми изменениями. Нужно формировать у обучающихся умение видеть повторяющиеся фрагменты программы.

Команды, изученные на уроке

Командный блок	Название команды	Отображение команды в среде программирования Scratch
движение	повернуть в направлении	

Дополнительные задания для особо одаренных детей, выполнивших основное задание.

Задание 1. Тем учащимся, кто раньше всех выполнил мини-проект «Квадрат линейно», можно предложить написать программу для рисования двух квадратов так, чтобы второй квадрат был нарисован правее и выше (рис. 12).

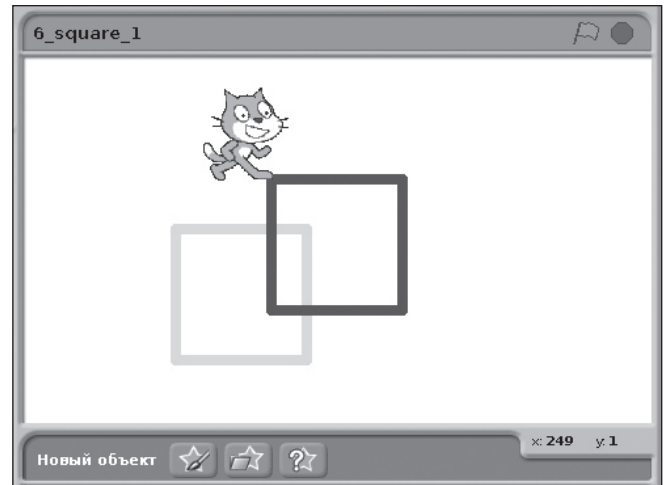


Рис. 12

Задание 2. Тем, кому понятно рисование квадрата, можно предложить подумать, какие изменения нужно внести в программу, чтобы был нарисован не квадрат, а прямоугольник (рис. 13).

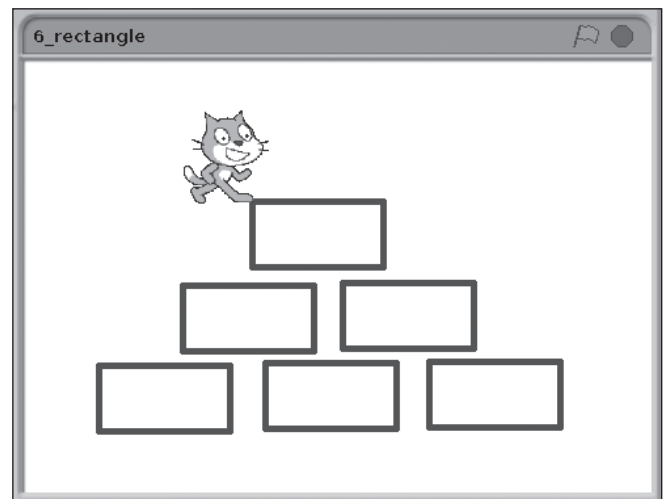


Рис. 13

Домашнее задание.

Задание 1. Придумать алгоритм и написать программу для рисования исполнителем лесенки.

Задание 2. Придумать самостоятельно задачу на использование изученной команды *повернуть в направлении*.

Урок 7. Конечный цикл. Исполнитель среды Scratch рисует квадраты и линии

Планируемые образовательные результаты:

- предметные — формирование представлений о циклическом алгоритме; развитие навыков

геометрических построений и моделирования процессов; формирование навыков выделения циклически повторяющихся этапов решения задачи; развитие изобразительных умений с помощью средств ИКТ; формирование навыков использования инструментов растрового редактора для создания изображений;

- *метапредметные* — умение выделять повторяющиеся процессы; формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенции); формирование умения оптимизировать решение;
- *личностные* — формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых учебных задач; развитие эстетического сознания через творческую деятельность на базе визуализированной среды программирования.

Решаемые учебные задачи:

- формирование навыков сохранения созданных программ для дальнейшего использования;
- приобретение навыков рисования блок-схемы циклического алгоритма;
- тренировка навыков использования векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем;
- тренировка навыка использования команд поворота на прямой угол (90°) по часовой и против часовой стрелки;
- формирование навыков использования циклического алгоритма для рисования исполнителем квадрата;
- формирование навыков оптимизации линейного алгоритма за счет использования циклической конструкции в программной среде Scratch.

Основные понятия, изучаемые на уроке:

- команда *открыть* из пункта меню *File*;
- команда *Сохранить как* из пункта меню *File*;
- циклический алгоритм;
- блок-схема циклического алгоритма;
- команда *повернуться на 90° по часовой стрелке*;
- команда *повернуться на 90° против часовой стрелки*;
- команда конечного цикла *повторить 10*.

Необходимые средства ИКТ:

- компьютер учителя с установленной программной средой Scratch;
- мультимедийный проектор, экран или интерактивная доска;
- компьютеры учащихся с установленной программной средой Scratch;
- свободное программное обеспечение — офисный пакет LibreOffice, установленный на компьютеры учителя и учеников.

Электронные образовательные ресурсы, размещенные в сети Интернет:

- проекты на сайте интернет-сообщества Scratch: <http://scratch.mit.edu/>
- видеоролик «Конечный цикл. Scratch рисует квадраты, линии»: https://youtu.be/fdwRg_1EVu0

Предлагаемая методика изложения темы урока.

Урок является первым в изучении циклических конструкций. На этом уроке нужно сравнить программы с использованием цикла и программы с последовательным выполнением команд.

В начале урока рекомендуется актуализировать знания, полученные на предыдущем уроке.

Это можно сделать, **поставив задачу, сходную с домашним заданием**, например, такую: создать программу для исполнителя среды Scratch, чтобы он нарисовал две ступеньки; высота и глубина каждой ступеньки составляет 100 шагов.

После получения решения можно провести **работу в парах по проверке домашнего задания**.

Далее следует перейти к **объяснению нового материала**.

Следует познакомить обучающихся с новыми командами из блока *движение*:

- командой *повернуться на 90° по часовой стрелке*;
- командой *повернуться на 90° против часовой стрелки*.

Далее вновь рассматриваем задачу о рисовании квадрата из предыдущего урока.

Постановка задачи: написать программу для исполнителя, чтобы он, двигаясь по пути, совпадающему со сторонами квадрата, прочертил геометрическую фигуру квадрат, длина стороны которого равна 100 шагам, применяя команду поворота на 90° по часовой стрелке.

При пошаговом разборе алгоритма рекомендуется обратить внимание обучающихся на то, что в каждой вершине вместо поворота в соответствующем направлении можно использовать команду *повернуться на 90° по часовой стрелке* (рис. 14), и результат от этого не изменится.

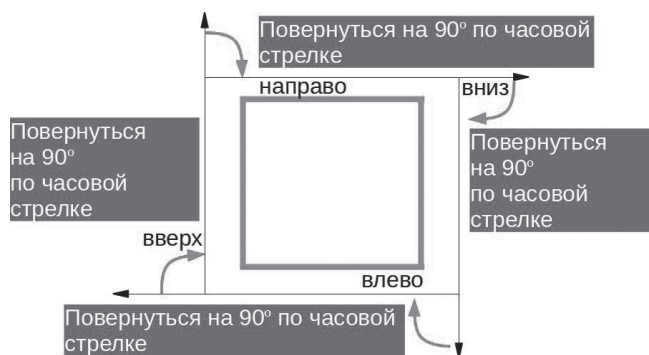


Рис. 14

Затем нужно перейти к **работе в программной среде Scratch**. Для этого надо открыть на учительском компьютере сохранённую задачу, выполненную на предыдущем уроке. Следует показать обучающимся, что это делается с помощью выбора команды *открыть* в пункте меню *File*. Здесь нужно обратить внимание учеников, насколько важно сохранять свои работы, и объяснить, что для сохранения готовой работы нужно выбрать пункт меню *File*, а затем в выпадающем списке в диалоговом окне надо написать название файла и указать место, где этот файл будет храниться. Важно обратить внимание учеников на то, что название файлу нужно давать таким, чтобы оно соответствовало смыслу того, что в нем содержится.

Когда учащиеся откроют программу решения задачи рисования квадрата из предыдущего урока, для того чтобы напомнить решение, рекомендуется запустить эту программу.

Теперь можно заменить все команды *повернуть в направление* на команды *повернуться на 90° по часовой стрелке*. Полученная в результате замены программа для рисования квадрата со стороной 100 шагов будет содержать восемь команд. Затем нужно дать возможность обучающимся внимательно посмотреть на последовательность команд. Нет ли какой-то закономерности? Закономерность состоит в том, что четыре раза повторяется пара команд (рис. 15).



Рис. 15

В таких случаях можно использовать команду из желтого блока управления *повторить 10*.

Нужно объяснить, что в этой команде вместо числа 10 нужно набрать с клавиатуры число 4, поскольку мы хотим, чтобы исполнитель повторил выполнение двух команд четыре раза. Далее рекомендуется пояснить, что команды, обеспечивающие повторяющееся выполнение одной или нескольких команд, называются циклическими, и показать, что использование команды цикла всегда сокращает количество команд в программе, а значит, делает ее более красивой (сравните рис. 16, 17). В более короткой программе легче обнаружить ошибку или неточность. Нужно стремиться к написанию коротких программ.

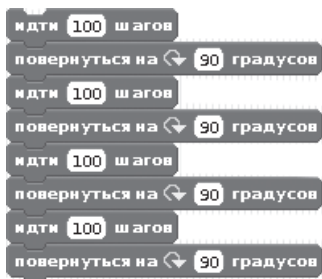


Рис. 16

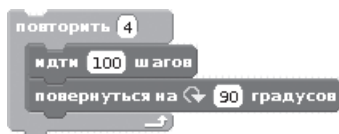


Рис. 17

При запуске новой программы рекомендуется показать еще один способ замедления выполнения команд программы для демонстрации ее работы. Обычно для этой цели используется команда *ждать 1 секунд*, рассмотренная на первом уроке.

Далее нужно открыть векторный графический редактор LibreOffice Draw и показать, каким образом отображается в блок-схеме фрагмент программы, выполняемый командой цикла (рис. 18).

После завершения разбора задач и объяснения нового материала обучающиеся могут приступить к выполнению **практической работы** — *мини-проекта «Квадрат циклически»*.



Рис. 18

После получения готовой работающей программы нужно дать возможность обучающимся поэкспериментировать с изменением параметров: цвета, размера пера, количества шагов.

Далее можно провести с ребятами **обсуждение**:

- какой из способов решения задачи рисования квадрата понравился им больше всего;
- какие команды показались интересными;
- какие еще задачи можно решить с использованием команды организации цикла *повторить 10*.

Этот урок очень важен, поскольку на нем обучающиеся впервые встречаются с командой организации цикла.

Команды, изученные на уроке

Командный блок	Название команды	Отображение команды в среде программирования Scratch
движение	повернуться на 90° по часовой стрелке	
движение	повернуться на 90° против часовой стрелки	
контроль	повторить 10	

Дополнительные задания для особо одаренных детей, выполнивших основное задание.

Задание 1. Тем учащимся, кто раньше всех выполнил мини-проект «Квадрат циклически», можно предложить написать программу для задачи, которая была решена на втором уроке, когда за счет уменьшения размера пера на 50 получились круги, уменьшающиеся по размеру и цвету (рис. 19). Задачу нужно решать, используя команду организации цикла *повторить 10*.

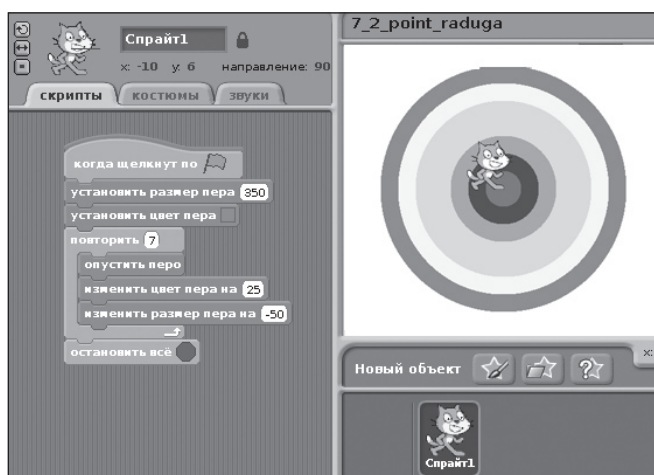


Рис. 19

Задание 2. Можно предложить решение задач рисования исполнителем линий, отличающихся от сплошной: например, нарисовать пунктирную линию (рис. 20) или штрих-пунктирную (рис. 21).

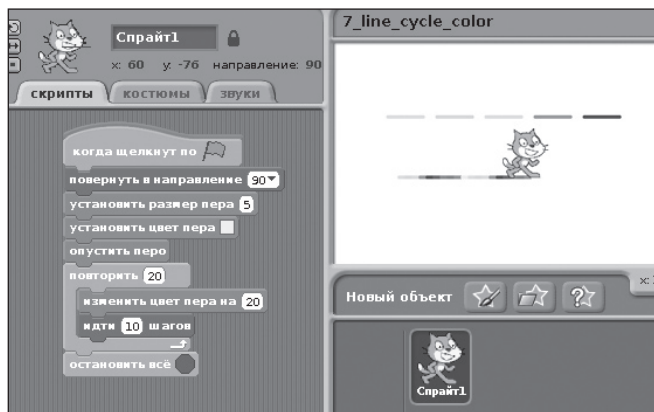


Рис. 20



Рис. 21

Домашнее задание.

Задание 1. Написать программу для рисования исполнителем квадрата, но с использованием команды *вернуться на 90° против часовой стрелки*.

Задание 2. Придумать программу для рисования пунктирной линии розового цвета.

Задание 3. Придумать задачи, для решения которых можно использовать циклический алгоритм.

Урок 8. Конечный цикл. Исполнитель среды Scratch рисует несколько линий и фигур. Копирование фрагментов программы

Планируемые образовательные результаты:

- **предметные** — закрепление представлений о циклическом алгоритме; развитие навыков геометрических построений и моделирования процессов, формирование навыков выделения циклически повторяющихся этапов решения задачи; развитие изобразительных умений с помощью средств ИКТ; совершенствование навыков использования инструментов растрового редактора для создания изображений; формирование навыков копирования фрагментов программы;
- **метапредметные** — умение выделять повторяющиеся процессы; формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенции); формирование умения оптимизировать решение;
- **личностные** — формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых учебных задач; развитие эстетического сознания через творческую деятельность на базе визуализированной среды программирования.

Решаемые учебные задачи:

- тренировка навыков использования операции цикла для решения учебных задач;
- формирование пространственных представлений о двумерных фигурах;
- тренировка навыков применения поворота на прямой угол (90°) при создании геометрических фигур и перемещении исполнителя;
- тренировка навыков оптимизации линейного алгоритма за счет использования циклической конструкции в программной среде Scratch.

Основные понятия, изучаемые на уроке:

- тело цикла;
- конечный цикл;
- блок-схема бесконечного цикла;
- имя костюма;
- имя спрайта;
- изменение костюма исполнителя;
- копирование фрагмента программы.

Необходимые средства ИКТ:

- компьютер учителя с установленной программной средой Scratch;
- мультимедийный проектор, экран или интерактивная доска;
- компьютеры учащихся с установленной программной средой Scratch;
- свободное программное обеспечение — офисный пакет LibreOffice, установленный на компьютеры учителя и учеников.

Электронные образовательные ресурсы, размещенные в сети Интернет:

- проекты на сайте интернет-сообщества Scratch: <http://scratch.mit.edu/>
- видеоролик «Конечный цикл. Scratch рисует несколько линий и фигур»: <https://youtu.be/PTcCvOc0F1A>

Предлагаемая методика изложения темы урока.

На уроке продолжается изучение циклических конструкций. На этом уроке следует расширить представление обучающихся об использовании конструкций организации цикла.

В начале урока рекомендуется **актуализировать знания, полученные на предыдущем уроке.**

Можно провести небольшой *опрос*, организованный или на листочках, или с использованием инструмента Google-формы.

Вопросы могут быть следующими:

- С помощью каких действий можно открыть ранее сохраненный проект?
- Вы написали программу для рисования квадрата. Какое имя наилучшим образом подходит для названия файла при сохранении?
 - Рисунок
 - Квадрат
 - Программа
 - Котенок
- Сколько градусов содержит прямой угол?
- Какие изменения нужно внести в программу, чтобы исполнитель среды Scratch рисовал квадрат в противоположную сторону (рис. 22)?



Рис. 22

- Как изменится результат выполнения программы из предыдущего вопроса, если использовать команду *идти 50 шагов*?

Объяснение материала урока можно начать со сравнения двух уже изученных команд, с помощью которых можно организовать цикл в программной среде Scratch (рис. 23). Команда для организации бесконечного цикла была изучена на первом уроке.



Рис. 23

Далее рекомендуется показать различие этих команд. Для этого можно привести сравнение блок-схем алгоритмов в обоих случаях (рис. 24).

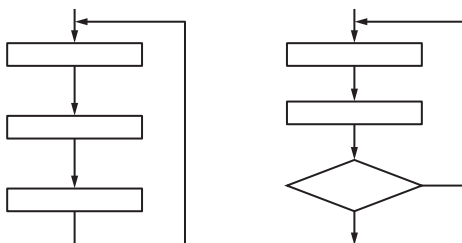


Рис. 24

Затем можно ввести понятие тела цикла и определить, что такое конечный цикл.

Основная задача этого урока — научить применять конструкцию конечного цикла, при этом показать, что использовать цикл можно не только для рисования геометрических фигур и линий, но также для смены внешности исполнителя среды Scratch.

Теперь следует перейти к **рассмотрению вариантов задач**, в которых используется конструкция конечного цикла или цикла со счетчиком.

Первая задача — рисование трех пунктирных розовых линий на расстоянии 40 шагов одна от другой, где длина штриха 50 шагов, а расстояние между штрихами 20 шагов (рис. 25).

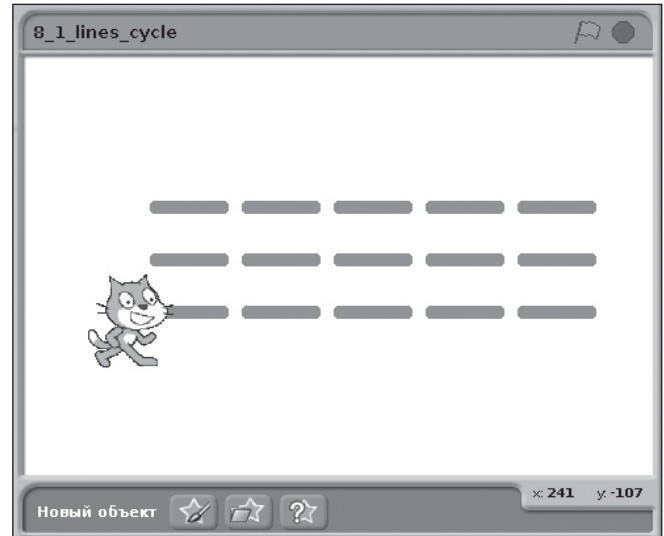


Рис. 25

При рассмотрении этой задачи важно показать, что все решение можно разделить на три этапа:

- 1) составить циклический алгоритм для рисования одной линии;
- 2) переместить исполнителя среды Scratch в начальную точку для рисования второй линии;
- 3) копировать два раза команды из п. 1 и 2.

Над рисованием одной пунктирной линии обучающиеся уже думали — это задача, которая была предложена для домашней заготовки. Поэтому у каждого из учеников уже должен быть свой вариант решения. При разборе задачи важно показать, что есть повторяющиеся действия и именно они будут описываться командами, которые нужно поместить в тело цикла. Таких повторений в данной команде пять. Поэтому количество необходимых по условию задачи повторений нужно поставить в качестве значения в команду *повторить* (рис. 26).

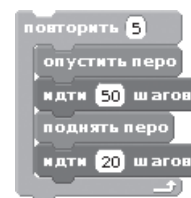


Рис. 26

Для выполнения второго этапа решения задачи нужно посчитать количество шагов, чтобы вернуть исполнителя в исходную точку.

Очень важно, чтобы обучающиеся сами научились действовать как исполнитель среды Scratch, поэтому нужно вместе посчитать количество пройденных шагов от начала рисования линии (рис. 27).

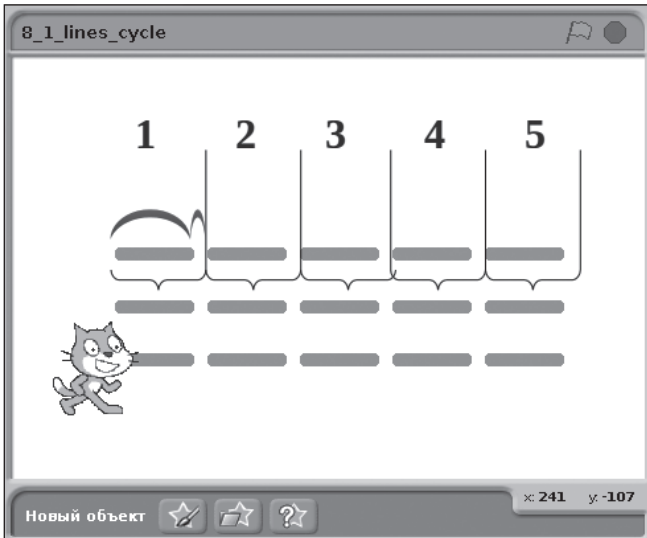


Рис. 27

Общая длина пути вычисляется по формуле:
 $(50 \text{ шагов} + 20 \text{ шагов}) * 5 \text{ повторений} = 350 \text{ шагов}$.

Для того чтобы вернуться в начало первой линии, нужно повернуть исполнителя в противоположную сторону или в направлении влево. Теперь осталось переместить нашего исполнителя в точку начала рисования второй линии. Для этого достаточно повернуть в направлении вниз и идти 40 шагов. Мы выполнили два этапа решения задачи. Теперь можно просто скопировать полученный фрагмент программы два раза.

Здесь нужно показать возможность использования операции копирования фрагмента программы с помощью кнопки *дублировать* (рис. 28).



Рис. 28

Вторая задача — рисование ступеней: нужно нарисовать четыре ступеньки, глубина и высота каждой из которых составляет 50 шагов (рис. 29).

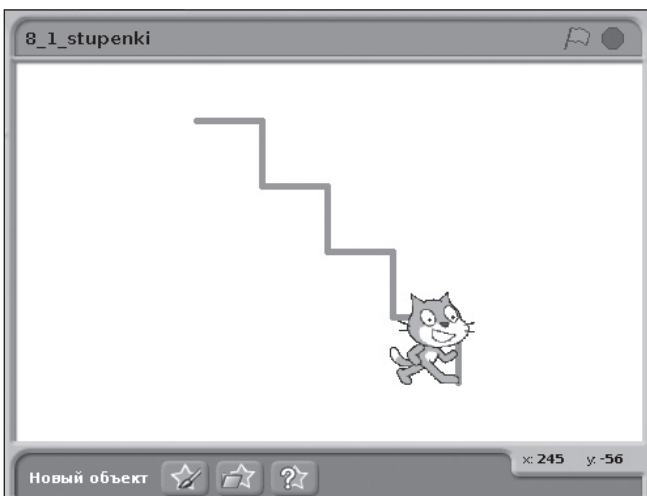


Рис. 29

При разборе задачи важно, чтобы обучающиеся поняли, что основное отличие рисования ступеней от рисования квадрата заключается в том, что меняется направление угла поворота. Если при рисовании квадрата все четыре раза поворот осуществлялся в одном направлении, например *по часовой стрелке*, то в случае со ступенями после изображения двух сторон нужно изменить направление поворота, и оно будет *против часовой стрелки* (рис. 30).

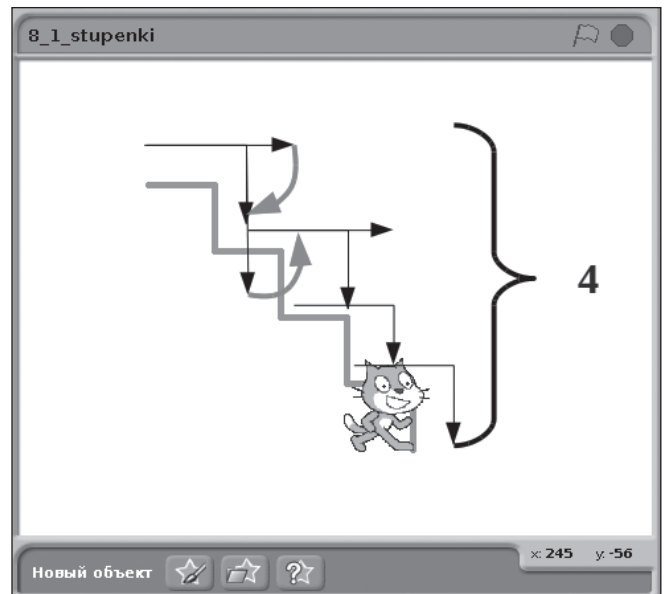


Рис. 30

Следующее, что нужно объяснить в решении задач на организацию конечного цикла, это то, что циклическая конструкция может использоваться не только для рисования геометрических фигур. Для примера можно рассмотреть *задачу с цифрами*: нужно, чтобы последовательно появлялись пять цифр от 1 до 5 на расстоянии 85 шагов друг от друга (рис. 31). Цифры — это раскрашенные импортированные костюмы исполнителя среды Scratch. Для решения этой задачи понадобятся команды из фиолетового блока *внешность*.

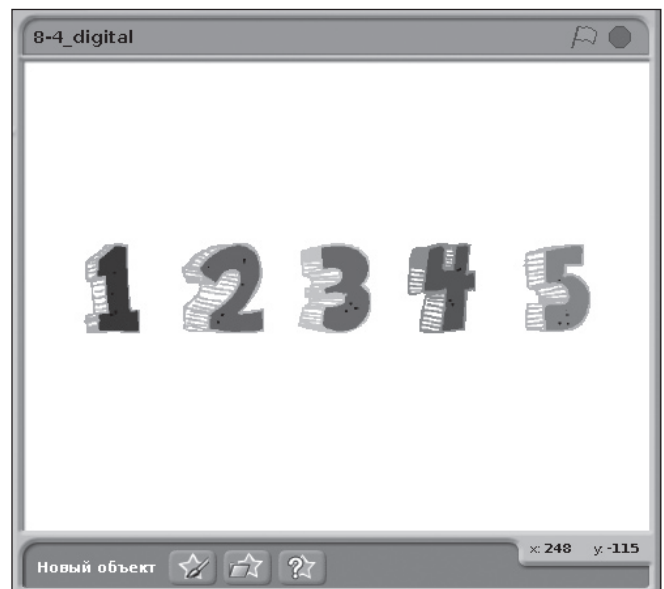


Рис. 31

При объяснении использования команд изменения внешности нужно указать, что каждому костюму можно задать свое имя. Можно изменять костюм, каждый раз используя команду *перейти к костюму*. Но в том случае, когда костюмы расположены друг за другом, можно последовательно их менять, поместив команду *следующий костюм* в тело цикла. В нашей задаче исполнитель перемещается от левого к правому краю и через равное число шагов оставляет отпечаток своего текущего костюма с использованием команды *печатать*.

После получения готовой работающей программы нужно дать возможность обучающимся поэкспериментировать с изменением параметров: цвета, размера пера, количества шагов, вариантов костюмов.

Команды, изученные на уроке

Командный блок	Название команды	Отображение команды в среде программирования Scratch
внешность	следующий костюм	следующий костюм
внешность	перейти к костюму	перейти к костюму Пять

Дополнительные задания для особо одаренных детей, выполнивших основное задание.

Задание 1. Тем учащимся, кто раньше всех решил задачу с лесенкой, можно предложить изменить решение задачи таким образом, чтобы высота ступенек была меньше глубины.

Задание 2. При решении задачи с костюмами можно предложить учащимся выбрать костюмы другого начертания из библиотеки костюмов, можно иначе их раскрасить в графическом редакторе. Примером выполнения может служить вариант, представленный на рисунке 32.

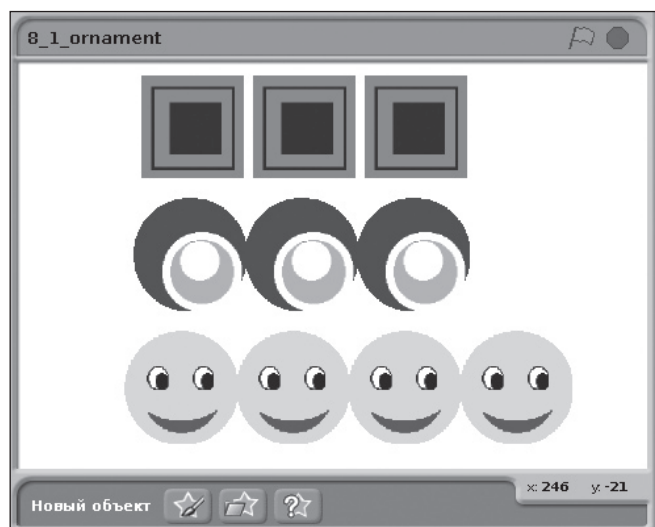


Рис. 32

Домашнее задание.

Задание 1. Используя программу для рисования ступеней и импортируя костюм баскетбольного мяча из стандартной библиотеки, создать изображение, представленное на рисунке 33 (программа запускалась дважды).

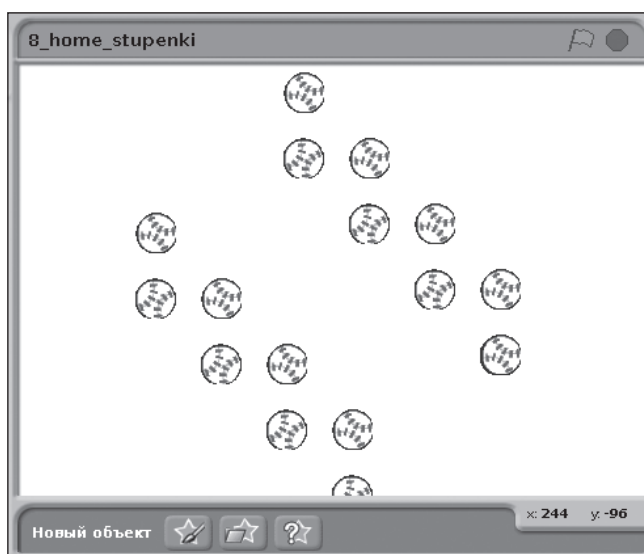


Рис. 33

Задание 2. Изменить задачу с цифрами так, чтобы получилась картинка с увеличивающимися в размере звездочками (рис. 34) (использовать команду *изменить размер*).

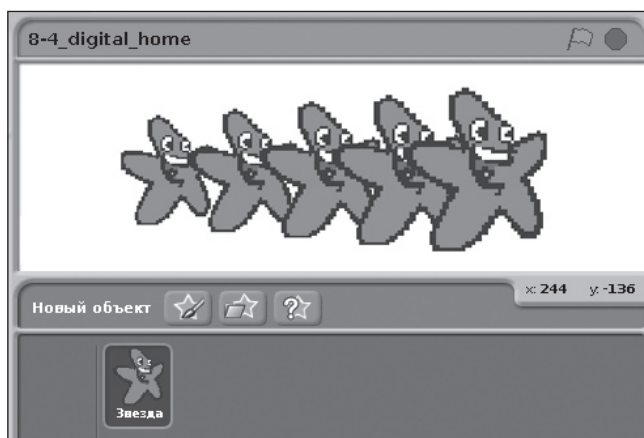


Рис. 34

Задание 3. Составить программу для решения задачи, когда исполнитель рисует уменьшающиеся квадраты со сторонами 100, 80, 60 и 40 шагов из одной точки (рис. 35).

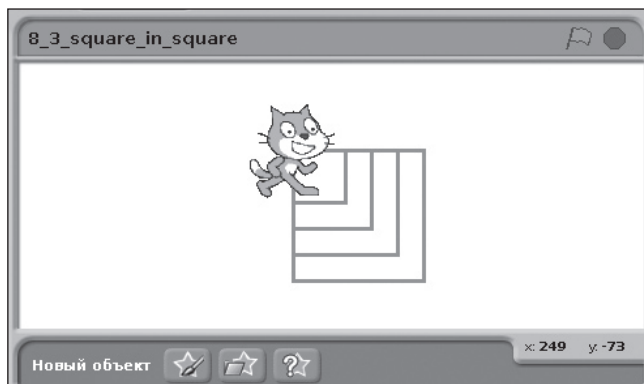


Рис. 35

Задание 4. Создать программы с использованием циклических конструкций для решенных ранее задач (рис. 36, 37).

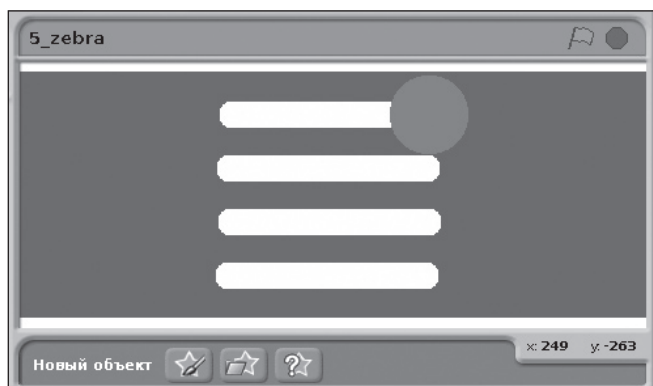


Рис. 36

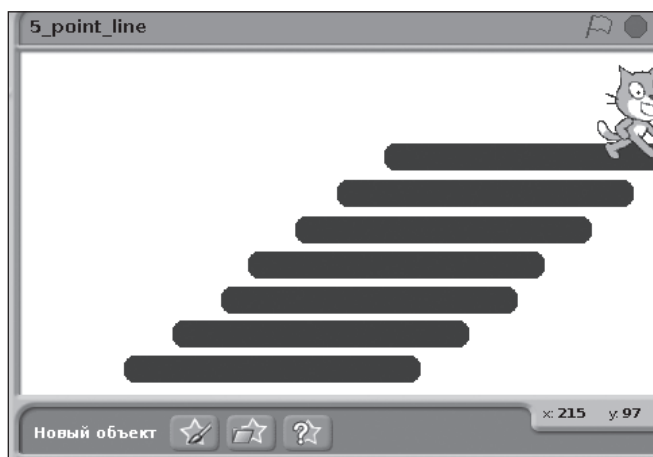


Рис. 37

Задание 5. Придумать свою задачу на пройденную тему.

Литературные и интернет-источники

1. Атлас новых профессий. <http://atlas100.ru/>
2. Босова Л. Л., Босова А. Ю. Информатика: учебник для 5 класса. М.: БИНОМ. Лаборатория знаний, 2014.
3. Босова Л. Л., Босова А. Ю. Информатика: учебник для 6 класса. М.: БИНОМ. Лаборатория знаний, 2014.
4. Виленкин Н. Я., Жохов В. И. и др. Математика: учебник для 5 класса: 31-е изд., стер. М.: Мнемозина, 2013.
5. Реформа инновационной системы США. От STEM к STEAM-образованию. <http://www.almavest.ru/ru/favorite/2013/05/14/384/>
6. Сорокина Т. Е. Визуальная среда Scratch как средство мотивации учащихся основной школы к изучению программирования // Информатика и образование. 2015. № 5.
7. Сорокина Т. Е. Пропедевтика программирования со Scratch // Слово учителю. Сетевое издание ГМЦ. М., 2014. <http://slovo.mosmetod.ru/avtorskie-materialy/item/238-sorokina-t-e-propedevtika-programmirovaniya-so-scratch>
8. Стратегия развития отрасли информационных технологий в Российской Федерации на 2014–2020 годы и на перспективу до 2025 года (утверждена распоряжением Правительства РФ от 1 ноября 2013 г. № 2026-п). http://www.consultant.ru/document/cons_doc_LAW_154161/
9. Уроки по Scratch. <https://www.youtube.com/playlist?list=PLMIhDclNR1GsZ9CJBZESbm7k3Xpr7aw>
10. Intel предложила создать в России ассоциацию в поддержку научно-технического образования // Учительская газета. 9 октября 2014 года. <http://www.ug.ru/news/13092>
11. Scratch. <https://scratch.mit.edu/>

Окончание следует

НОВОСТИ

Робототехника с детского сада

В FabLab Санкт-Петербургского государственного политехнического университета прошел фестиваль робототехники для детей «Робот Фест Политех», организатором которого выступил образовательный проект ScratchDuino. С азами программирования и создания роботов познакомились дети от пяти лет, школьники, студенты, а также их родители.

«В существующей системе образования изучают, как пользоваться роботами, а не как их создавать. Следствием такого обучения будет еще большая зависимость от стран, лидирующих в области робототехники. Чтобы перестроить консервативную по определению систему образования, потребуется 10–15 лет. Поэтому мы запустили проект, который предполагает создание сети кружков для детей, проведение фестивалей и олимпиад в области робототехники. Проект призван улучшить качество обучения в области робототехники, инженерных специальностей, информационных технологий, естественнонаучных дисциплин», — рассказал Павел Фролов, продюсер проекта.

Использование исключительно свободного программного и аппаратного обеспечения позволяет учащимся разобраться в роботах до винтика и до байта. Кроме того, созданная методика дает возможность вовлекать в процесс обучения детей с самых малых лет. Главное требование — это умение читать по-русски. Даже умение

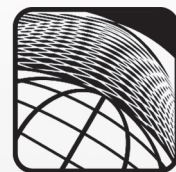
пользоваться компьютерной мышкой не обязательно. Следовательно, создание своего робота доступно даже совсем маленькому ребенку, а если он чуть постарше, то самостоятельно сможет развести, вытравить и распаять печатную плату, напечатать на 3D-принтере детали корпуса, запрограммировать микроконтроллеры. Фактически, пожалуй, только микропроцессоры нужны заводского изготовления и двигатели.

Можно использовать и некоторые типовые массово производимые комплектующие. По словам Фролова, не менее половины таких комплектующих уже выпускается в России. Однако, например, чипы для двигателей пока приходится закупать в Китае. Из инструментов программирования используется язык Scratch (с поддержкой программирования роботов, адаптированный под Android), Blockly (Google) и Arduino IDE.

Какие негативные последствия могут быть от умения детей самостоятельно создавать роботов, инициаторы проекта пока не задумываются, считая, что нужно быть оптимистами. Однако на занятиях детям обязательно объясняют, что роботы-убийцы — это плохо.

В рамках ScratchDuino продвигаются два главных продукта — робоконструкторы для обучения, доступные как школам, так и частным лицам, и кружки робототехники для детей от пяти лет, с собственной методологией обучения.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)



В. В. Дьякова,

гимназия № 1 Центрального района Волгограда

СЕТЕВОЙ ПРОЕКТ «ОТ ПОБЕДЫ РЕГИОНОВ К ВЕЛИКОЙ ПОБЕДЕ»

Аннотация

В статье рассмотрен опыт реализации проекта «От победы регионов к Великой Победе». Группа учащихся изучила теоретический материал, посвященный основным вехам Великой Отечественной войны, выполнила практические задания (выпуск школьной газеты, создание презентаций, создание и заполнение сайта), организовала и провела акцию «Бессмертный полк в гимназии».

Ключевые слова: проект, 70-летие Победы в Великой Отечественной войне, сайт, акция, «Бессмертный полк».

Контактная информация

Дьякова Валентина Владимировна,
учитель информатики гимназии № 1 Центрального района Волгограда; *адрес:* 400050, г. Волгоград, ул. Ткачева, д. 10А; *телефон:* (844-2) 37-37-67; *e-mail:* mou_gimnazia_1@mail.ru

V. V. Dyakova,
Gymnasium 1 of Central District of Volgograd

NETWORK PROJECT "FROM THE VICTORY OF THE REGIONS TO THE GREAT VICTORY"

Abstract

The article describes the experience of the project "From the victory of the regions to the Great Victory". Group of students was studying theoretical material on the major landmarks of the Great Patriotic War, carried out practical tasks (issue of the school newspaper, making presentations, creating and filling the web-site), organized and conducted the action "Immortal Regiment in the gymnasium".

Keywords: project, 70th anniversary of Victory in the Great Patriotic War, site, promotion, "Immortal Regiment".

В 2015 году мы отмечаем знаменательную дату — 70-летие Победы в Великой Отечественной войне. Патриотическое воспитание подрастающего поколения всегда являлось одной из важнейших задач общества. Создание проекта «От победы регионов к Великой Победе» направлено на работу по воспитанию у учащихся чувства гордости за свой народ, уважения к его свершениям и достойным страницам истории, предполагает привлечение учащихся к изучению важнейших событий Великой Отечественной войны.

Цели проекта:

- формирование представлений о Великой Отечественной войне на основе уже имеющихся знаний;
- сохранение преемственности поколений;
- формирование у учащихся уважения к военной истории России;
- формирование гражданской позиции;
- воспитание патриотизма и чувства гордости за свою Родину.

Задачи проекта:

- систематизировать, обобщить и закрепить знания учащихся по теме «Великая Отечественная война 1941–1945 годов»; сформировать у учащихся понимание исторического значения победы нашего народа в Великой Отечественной войне;
- пробудить интерес у учащихся к изучению истории своей семьи; провести акцию «Бессмертный полк в гимназии»;
- организовать партнерское взаимодействие с представителями образовательного учреждения Санкт-Петербурга — НОШ № 300;
- создать материальный продукт (сайт, газету, презентации, буклеты).

Срок реализации проекта: июнь 2014 года — май 2015 года.

Содержание проекта.

Создание проекта «От победы регионов к Великой Победе» направлено на работу по воспитанию у учащихся чувства гордости за свой народ, уважения к его свершениям и достойным страницам истории, предполагает привлечение учащихся к изучению важнейших событий Великой Отечественной войны 1941–1945 годов.

Группа учащихся изучила теоретический материал, посвященный основным вехам Великой Отечественной войны, выполнила практические задания (выпуск школьной газеты, создание презентаций, создание и заполнение сайта), организовала и провела акцию «Бессмертный полк в гимназии».

В результате работы над проектом были созданы сайт, газета, 37 презентаций, посвященных основным датам Великой Отечественной войны, 20 инфографик, 3 постера.

В проекте рассмотрены такие вопросы, как «Битва за Москву», «Сталинградская битва», «Блокада Ленинграда», «Битва за Кавказ», «Битва за Днепр», «Курская битва», «Операция «Багратион»», «Висло-Одерская операция», «Штурм

Берлина», а также «Ордена и медали ВОВ», «Пионеры-герои», «Молодая гвардия».

Материалы для раздела «Блокада Ленинграда» («Письма в блокадный Ленинград», газета «Дорогами славы», викторина) предоставлены учащимися НОШ № 300 Санкт-Петербурга.

В рамках проекта была организована акция «Бессмертный полк в гимназии». Ребята приносили фотографии своих родственников — участников Великой Отечественной войны (бабушек, дедушек, прабабушек, прадедушек), а также их награды. Все материалы были размещены на сайте проекта «От победы регионов к Великой Победе»: <http://greatvictory70.jimdo.com/> в разделе «Бессмертный полк».

Ребята подготовили небольшие рассказы о своем родственнике-ветеране, дорогом для них человеке, не ограничиваясь сухими строчками биографии. Учащиеся записывали свои личные воспоминания или воспоминания своих родителей, родственников, выражая свое отношение к солдату, к празднику Победы. Ведь, кроме самих ребят, больше никому передать эту память о конкретном человеке потомкам. Каждая строчка для нас имеет ценность. На данный момент в полку зарегистрировано 28 человек.

План реализации проекта.

1-й этап — подготовительный.

- Формирование проектной группы;
- определение объекта исследования;
- формулирование темы проектной работы;
- определение проблемной задачи;
- выдвижение гипотез решения проблемной задачи;
- обоснование методов исследования;
- разработка плана исследования;
- поиск информации, изучение литературы по исследуемой проблеме.

Учитель информатики непосредственно организует и контролирует осуществление учащимися проектного обучения, выполняет следующие функции:

- проведение консультаций с участниками проекта;
- руководство работой над проектом в рамках согласованного объекта исследования;
- методическая поддержка проектного обучения;
- планирование совместно с учащимися работы в течение всего проектного периода с учетом применения средств ИКТ.

2-й этап — основной.

Таблица

Мероприятия 2-го этапа проекта

№ п/п	Мероприятие	Время проведения	Результат
1	Подготовка к разработке проекта. Просмотр примеров проектов	2 июня	Поиск информации
2	Подготовка к разработке проекта. Пример планирования содержания и этапов проекта	3 июня	Поиск информации
3	Знакомство с требованиями к содержанию и организации проведения проекта. Формулировка темы проекта	4 июня	Поиск информации
4	Составление списка информационных ресурсов для проекта и его материалов. Библиографические правила цитирования источников	5 июня	Информация об основных событиях ВОВ
5	Разработка проекта «От победы регионов к Великой Победе»	10 июня	Презентации по ВОВ
6	Разработка проекта «От победы регионов к Великой Победе». Проведение акции «Бессмертный полк в гимназии»	11 июня — 24 июня	Презентации по ВОВ. Сайт «От победы регионов к Великой Победе»: http://greatvictory70.jimdo.com/
7	Разработка специального выпуска школьной газеты «Мы»	25 июня	Специальный выпуск газеты «Мы»
8	Разработка специального выпуска школьной газеты «Мы»	26 июня	Специальный выпуск газеты «Мы»
9	Проведение акции «Бессмертный полк в гимназии»	В течение года	Информация на сайте http://greatvictory70.jimdo.com/
10	Партнерское взаимодействие с представителями образовательного учреждения Санкт-Петербурга — НОШ № 300	Сентябрь 2014 года — май 2015 года	Материалы о блокаде Ленинграда представлены в разделе «Блокада Ленинграда»

3-й этап — заключительный.

На наш взгляд, презентация учащимися результатов своей деятельности позволяет им глубже осознать полученный в ходе исследования новый опыт, почувствовать значимость проделанной работы и овладеть навыками публичного общения. Поэтому проект участвовал в конкурсах «Электронный Волгоград», «Позитивный

контент», «Компьютерный Олимп», «Моя страна — моя Россия». Сайт «От победы регионов к Великой Победе» награжден издательским домом «Красная звезда» памятной медалью фронтового корреспондента Константина Симонова. Результаты проекта были представлены на XI Волгоградском областном образовательном форуме «Образование-2015».

Механизм реализации проекта и схема управления проектом в рамках территории.

Механизм заключается в ответственной работе учащихся — разработчиков проекта.

Схема управления: руководитель проекта отслеживает все происходящие события, связанные с реализацией проекта, и в порядке кооперации принимает решения вместе с группой разработчиков.

Кадровое обеспечение проекта с описанием количественного и качественного потенциала команды проекта: участники проекта — это инициативная группа единомышленников: учащиеся девятых классов, учитель информатики.

Критерии оценки эффективности проекта:

- положительные отзывы пользователей сайта, участников проекта и акции «Бессмертный полк в гимназии»;
- посещаемость сайта;
- широкое освещение проекта в средствах массовой информации.

Предполагаемые конечные результаты, перспективы развития, долгосрочный эффект:

- побуждение интереса у учащихся к изучению истории своей семьи;

- реализация данного проекта на всей территории Волгограда и Волгоградской области;
- закладка организационной основы для обоснования и запуска следующего аналогичного, но более широкого по проблемному диапазону проекта.

Порядок контроля и оценки результатов проекта:

- авторский надзор;
- соответствие утвержденному плану реализации проекта.

Финансово-экономическое обоснование проекта: организационные, технические и другие ресурсы предоставляются гимназией № 1. Финансовое сопровождение осуществляется силами руководителя и разработчиков проекта.

Литературные и интернет-источники

1. Новые педагогические и информационные технологии в системе образования / под ред. Е. С. Полат. М.: Академия, 2000.

2. Сайт межрегионального историко-патриотического движения «Бессмертный полк» — общероссийской гражданской инициативы. <http://moypolk.ru/>

НОВОСТИ

Сборная России завоевала две золотых и две серебряных награды на Всемирной олимпиаде по робототехнике

На XII Всемирной олимпиаде по робототехнике (World Robot Olympiad — WRO), которая прошла с 6 по 8 ноября в Катаре, российские команды завоевали четыре комплекта наград: два «золота» в основной категории средней возрастной группы и в творческой категории старшей возрастной группы, а также две серебряные медали в творческой категории младшей возрастной группы и в студенческом зачете. В соревнованиях по футболу роботов российская команда стала четвертой, уступив в матче за третье место соперникам из Гонконга. Кроме того, российские школьники заняли с шестого по восьмое места в творческой и девятое место в основной категории младшей возрастной группы. Отдельным призом Creativity Award отмечен проект из Санкт-Петербурга.

Результаты команд из России:

1-е место:

- основная категория (средняя возрастная группа): состязание «Поиск сокровищ», команда Dzeta, Москва, тренер — Сергей Мустафин;
- творческая категория (старшая возрастная группа): команда BURITOS, Санкт-Петербург, тренер — Игорь Лосицкий.

2-е место:

- творческая категория (младшая группа): команда Robofriends, Санкт-Петербург, тренер — Сергей Филиппов;
- студенческая категория (задание «Боулинг»): команда PML30-students, Санкт-Петербург, тренеры — Дмитрий и Екатерина Лузины.

Creativity Award — специальный приз за техническую сложность проекта:

- команда DRL, Санкт-Петербург, тренер — Сергей Филиппов.

4-е место:

- футбол роботов: команда Hexagon, Москва, тренер — Сергей Мустафин.

«Поздравляю нашу команду с высокими результатами, продемонстрированными на крупнейших международных робототехнических соревнованиях, — сказал глава Минкомсвязи России Николай Никифоров. — Это передовое направление научно-технического творчества, и я рад, что наша молодежь в очередной раз показала, насколько талантливо она может решать самые сложные инженерные задачи. Главное — это не столько занятые места и набранные очки. Важно, чтобы наши олимпийцы продолжили развиваться в области радиоэлектроники, программирования, закончили российские вузы и стали лидерами отечественных компаний ИКТ-сектора. Это наша элита, наш золотой кадровый резерв сегодня. Нам нужно готовить как можно больше высококвалифицированных специалистов по ключевым направлениям инфокоммуникационных технологий, ведь в центре любой разработки стоит прежде всего человек, инженер».

Напомним, что защищать честь страны на международной арене отправились победители российского этапа Всемирной олимпиады роботов, который в июне 2015 года собрал 800 детей из 43 регионов страны в «Университете Иннополис», расположенном в Республике Татарстан.

(По материалам официального сайта Минкомсвязи России)



Е. А. Еремин,

Пермский государственный гуманитарно-педагогический университет

КАК НАГЛЯДНО ПРОДЕМОНСТРИРОВАТЬ ШКОЛЬНИКАМ ПРЕИМУЩЕСТВА МНОГОПРОЦЕССОРНЫХ СИСТЕМ*

Аннотация

В качестве наглядного примера параллельной обработки данных в статье рассматривается задача о составлении частотного словаря. Для демонстрации исполнения алгоритмов решений написана программа-имитатор, визуализирующая ход вычислений на экране обычного ПК. Материал интуитивно понятен, вместе с программной поддержкой он может быть использован при объяснении школьникам преимуществ параллельных вычислительных технологий.

Ключевые слова: параллельная обработка, многопроцессорные системы, пример, содержание курса, частотный словарь.

Контактная информация

Еремин Евгений Александрович, канд. физ.-мат. наук, доцент, доцент кафедры мультимедийной дидактики и ИТО, Пермский государственный гуманитарно-педагогический университет; *адрес:* 614990, г. Пермь, ул. Сибирская, д. 24; *телефон:* (342) 238-63-31; *e-mail:* eremin@pspu.ru

Е. А. Eremin,
Perm State Humanitarian Pedagogical
University

HOW TO DEMONSTRATE PUPILS VISUALLY THE ADVANTAGES OF MULTIPROCESSOR SYSTEMS

Abstract

A problem of working out a concordance is considered in the article as a demonstrative example of parallel data processing. The software imitator, visualizing a progress of calculations on an ordinary PC's screen, is developed to demonstrate an execution of solution's algorithms. Material is intuitively clear, together with the software support it may be used for explanation of parallel technologies advantages to pupils at school.

Keywords: parallel processing, multiprocessor systems, example, course content, concordance.

За последние годы потребность в понимании идей параллельных вычислительных технологий существенно возросла из-за массового проникновения многоядерных процессоров в повсеместную электронную технику. При этом, как показывает опыт, нужны новые методики преподавания, поскольку традиционное образование не всегда обеспечивает понимание новых «непривычных» технологий. «Требуется изменение образа мысли в отношении вычислений, т. е. то, что называется сдвигом парадигмы, прежде чем программисты станут инстинктивно разрабатывать параллельные методы решения вычислительных задач» [6].

Уже в школе необходимо закладывать основы параллельного подхода к решению задач. Отсюда вытекает, что «разработка целостной методики пропедевтической подготовки учащихся основной школы в области параллельных вычислений — задача нескольких ближайших лет» [2]. Все более активно обсуждается вопрос о необходимости изучения основ параллельного программирования в педагогических вузах [8].

К сожалению, несмотря на наличие целого ряда интересных и оригинальных публикаций [1–3, 7], бросаются в глаза два принципиальных обстоятельства. Во-первых, везде речь идет в основном о начальной школе. А во-вторых, понятие параллельного программирования вводится на базе обыденных представлений учащихся: несколько человек выполняют общую работу — строят дом, вскапывают грядку и т. д. Параллельный алгоритм при этом немного упрощенно определяется как «алгоритм для одновременной работы нескольких исполнителей» [1, 3]. Ни в коем случае не отрицая полезности «жизненных аналогий», все же хотелось бы как-то видеть во всех разговорах о параллельности вычислений и саму вычислительную технику.

Итак, мы вынуждены констатировать наличие существенного разрыва между имеющимися «описательными» материалами для изучения параллельных технологий в школе и реальными технологиями программирования (вроде MPI). В настоящей работе сделана попытка заполнить образовавшийся пробел: описана простая задача, с помощью которой можно наглядно продемонстрировать школьникам преимущества решения на многопроцессорной вычислительной технике. Разработано учебное программное обеспечение, визуализирующее процесс обработки данных. Благодаря интуитивной понятности задачи описанный пример вполне пригоден для изучения основ параллельных технологий в рамках школьного курса информатики.

Постановка задачи и алгоритмы ее решения

Задача. Имеется некоторый текст на английском языке (английский выбран только потому, что в его алфавите меньше букв, чем в русском, и процесс легче визуализировать).

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_09-2015/

Требуется составить *частотный словарь* текста, т. е. алфавитный список всех имеющихся в тексте слов с указанием, сколько раз встречается каждое слово.

Классическое «однопроцессорное» решение задачи состоит в сортировке массива и объединении одинаковых слов.

При *параллельной* обработке предполагается получить выигрыш за счет распределения работы между 26 процессорами, каждый из которых будет выполнять сортировку слов на фиксированную букву.

Сама по себе проблема, конечно, не нова: она обсуждалась, например, в книге [6]. Но там представлено решение с 26 «самостоятельными» потоками, а у нас работой 26 процессоров-сортировщиков будет руководить дополнительный процессор.

Выбранная задача очень наглядна и интуитивно понятна всем, что позволяет рекомендовать ее для изложения учебного материала о преимуществах параллельной обработки данных.

Будем использовать для решения следующий алгоритм обработки (см. также рис. 1), аналогичный классической сортировке вставкой [4, 5].

Извлекаем из текста очередное слово W и по очереди сравниваем его со словами, уже имеющимися в упорядоченном списке S . Возможны три ситуации:

- 1) $W = S[m]$: рассматриваемое слово совпало с одним из слов списка — тогда просто увеличиваем счетчик $c[m]$ для этого слова на единицу;
- 2) $W < S[m]$: в списке нашлось слово, которое по алфавиту должно стоять после проверяемого — сдвигаем все оставшиеся слова на одну позицию, а на освободившееся место помещаем текущее слово с единичным значением счетчика;
- 3) удалось дойти до конца списка, и ни одно из предыдущих условий не выполнилось — вносим слово W в конец списка (тоже с единичным значением счетчика).

В двух последних случаях длина списка увеличивается на одно слово. Упорядоченность списка по алфавиту все время сохраняется.

Процесс обработки продолжается до тех пор, пока слова в исходном тексте не закончатся.

Заметим, что описанный выше алгоритм не является оптимальным: в частности, используя вместо перебора элементов метод дихотомии (деления списка пополам), можно получить результат значительно быстрее. Тем не менее продемонстрируем, что даже наш медленный учебный алгоритм при параллельном исполнении способен обработать небольшой текст за приемлемое время.

Пусть теперь в нашей воображаемой параллельной вычислительной машине имеется 27 процессоров. Один из них для краткости будем называть *центральным* (ЦП), а все остальные, работающие под его управлением, — *локальными процессорами* (ЛП).

Центральный процессор считывает из текста очередное слово и по первой букве определяет, какому из локальных процессоров передать это слово на обработку. Так, все слова на букву «а» попадают в ЛП1, на «b» — в ЛП2, а на «z» — в ЛП26. Каждый из локальных процессоров сортирует слова по описанному выше алгоритму, но секрет заключается в том, что списки у них многократно короче, чем в случае обработки полного текста одним процессором. Именно это и является главной причиной ускорения обработки.

Оценка ускорения параллельных вычислений

Важным моментом при сравнении производительности является выбор единицы измерения работы, проделанной над данными. Во многих случаях за единицу принимают машинную команду. Часто для более грубой оценки используется время выполнения операций, которые «вносят наибольший вклад в общее время выполнения алгоритма» [5]. Для нашей задачи естественно принять в качестве такой единицы времени обработку одного слова. Примером подобных команд могут служить извлечение очередного слова из текста, сравнение двух слов, перемещение слова внутри списка и т. д.

Операция над словом — это комплексное действие, которое состоит из многих элементарных машинных команд. Как правило, это небольшая циклическая программа, которая помимо прочего включает в себя арифметические операции над целыми числами: увеличение адреса рассматриваемого символа текста, изменение значения счетчика. Отсюда следует, что для оценочных расчетов временем выполнения отдельной арифметической операции по сравнению с операцией над словом можно пренебрегать. Аналогичным образом при оценке времени обработки вещественных данных обычно пренебрегают входящими в программу целочисленными командами.

Зная количество наиболее трудоемких строковых операций для последовательного и параллельного алгоритмов, легко найти ускорение вычислений как частное этих чисел.

Для начала подсчитаем количество операций для некоторого упрощенного случая. Прежде всего, предположим, что обрабатываемый текст не содержит ни одного одинакового слова. Кроме того, примем, что слова, начинающиеся с одинаковой буквы, находятся в тексте настолько далеко, что в момент передачи нового слова соответствующий локальный процессор уже успел обработать старое (иными словами, ЦП при передаче не приходится ожидать, пока ЛП закончит обработку предыдущего слова).

Покажем, как делается оценка числа строковых операций, на примере основной части алгоритма — вставки очередного слова в список уже имеющихся. Алгоритм, написанный на часто используемом в школьном курсе информатики псевдокоде, приведен на рисунке 1. Серым фоном в тексте отмечены строковые операции (напомним, что временем выполнения всех остальных команд мы обоснованно пренебрегаем).

Легко увидеть, что полное количество операций во всем приведенном фрагменте, который обрабатывает одно слово, равняется:

$$m + 1 + (n - m) + 1 = n + 2.$$

Значение n (число слов в отсортированном списке) с ростом длины списка увеличивается от 1 до полного числа слов в тексте N . В результате время выполнения каждой последующей итерации возрастает на 1. Для цикла по всему тексту получаем сумму арифметической прогрессии, которая, согласно известной формуле, может быть вычислена как:

$$(3 + [N + 2]) \cdot N / 2 = N \cdot (N + 5) / 2.$$

Для последовательного алгоритма остается к полученной величине прибавить количество дополнительных

```

//блок 1: ищем в списке слов S позицию m
для слова W;
//m циклов по 1 оп., всего m оп.
m = 1
пока (S[m] < W) И (m ≤ n)
  иц m=m+1
кц
//блок 2: выбор варианта продолжения; 1 оп.
если S[m] = W
  то c[m] = c[m]+1
  выход
иначе n = n+1
  если m = n
    то перейти к блоку 4
  все
все
//блок 3: сдвигаем последние слова и их
счетчики;
//n-m циклов по 1 оп., всего n-m оп.
k = n-1
пока k ≥ m
  иц S[k+1] = S[k]; c[k+1] = c[k]
  k=k-1
кц
//блок 4: занести слово W и установить
счетчик; 1 оп.
S[m] = W; c[m] = 1

```

Рис. 1. Алгоритм вставки нового слова W в имеющийся массив S

операций по выборке слов из текста. Каждая итерация цикла содержит по две строковые операции, которые связаны с выборкой очередного слова, и выполняется $N + 1$ раз — последняя соответствует «пустому» слову после окончания текста. Приведем подобные члены, получим окончательный результат для количества операций в однопроцессорном алгоритме:

$$T1 = (N^2 + 9N + 4) / 2.$$

Подсчитывая количество операций в параллельном алгоритме, необходимо как-то определить время обмена между ЦП и ЛП. Введем условный коэффициент Cio , который показывает время пересылки строки из одного процессора в другой. Например, при $Cio = 2$ время обмена полагаем равным времени двух строковых операций. В дальнейшем будем считать Cio свободным параметром задачи.

Первая часть параллельного алгоритма считывает из текста по одному слову и отправляет их нужному ЛП. Длительность выполнения этой части алгоритма оценивается как:

$$2 \cdot (N + 1) + N \cdot Cio.$$

Временем сортировки в ЛП в рассматриваемом предельном случае пренебрегаем, что следует из исходного допущения о расположении слов в тексте.

Наконец, завершается параллельный алгоритм сбором результатов сортировки в память ЦП. Этот этап занимает заметное время:

$$N \cdot Cio + 26 \cdot Cio.$$

Первое слагаемое описывает копирование всех слов из ЛП в ЦП, а второе — последний обмен с каждым из процессоров, в ходе которого сообщается, что больше

данных нет. Суммарное время работы 27-процессорного параллельного алгоритма после тривиальных упрощений принимает вид:

$$T27 = 2 \cdot (Cio + 1) \cdot N + 26 \cdot Cio + 2.$$

Важно подчеркнуть, что $T27$ оказывается линейной функцией от N , а не квадратичной, как $T1$.

Общее ускорение благодаря применению параллельного алгоритма равно

$$S = T1 / T27.$$

Таким образом, несложные вычисления (не выходящие за рамки школьной алгебры!) позволили нам вычислить примерный эффект от использования многопроцессорной системы. Например, для $Cio = 2$, подставляя в формулы значение $N = 100$, получаем:

$$S \approx 8,34,$$

а при $N = 200$:

$$S \approx 16,67.$$

Отчетливо видно, что чем длиннее текст, тем больше выигрыш от использования параллельного алгоритма. Следует, правда, помнить, что для достаточно больших текстов сделанные при выводе формул предположения перестают выполняться, а значит, сами формулы уже не работают.

Программное обеспечение

Оценочные формулы — это, конечно, здорово, но все же в реальных текстах много повторяющихся слов. Для выбранного алгоритма сортировки очевидно, что уже имеющееся в списке слово обрабатывается быстрее (не нужен сдвиг элементов массива), причем тем быстрее, чем ближе это слово к началу списка. Как оценить выигрыш от параллельных вычислений в общем случае? Для ответа на этот вопрос была написана специальная программа-имитатор. Она моделирует 27-процессорную систему, описанную выше, причем за время одного «виртуального такта» система выполняет ровно одну операцию над строкой. Отсюда общее количество тактов, за которые выполняются последовательный и параллельный алгоритмы, есть $T1$ и $T27$ соответственно. Если в тексте более сотни слов, то, наблюдая за работой программы, легко увидеть, что параллельный алгоритм действительно завершается много быстрее, чем последовательный.

Программа визуализирует на экране состояние каждого из процессоров, и учащийся может наглядно наблюдать, как идет процесс сортировки. Некоторое представление о том, как выглядит программа, дает рисунок 2.

На экране создаются четыре окна:

- 1) Первое (на рисунке оно слева сверху) содержит органы управления виртуальной машиной и визуализирует работу параллельного алгоритма: подсвечивает в тексте обрабатываемое слово, показывает сборку итогового словаря.
- 2) Второе (справа сверху) похожим образом демонстрирует работу последовательного алгоритма. Заметим, что оба алгоритма стартуют одновременно, так что результат их «соревнования» наглядно виден.
- 3) Третье, самое большое окно (в центральной части экрана) отображает состояние всех 26 локальных процессоров. Если установить медленную ско-

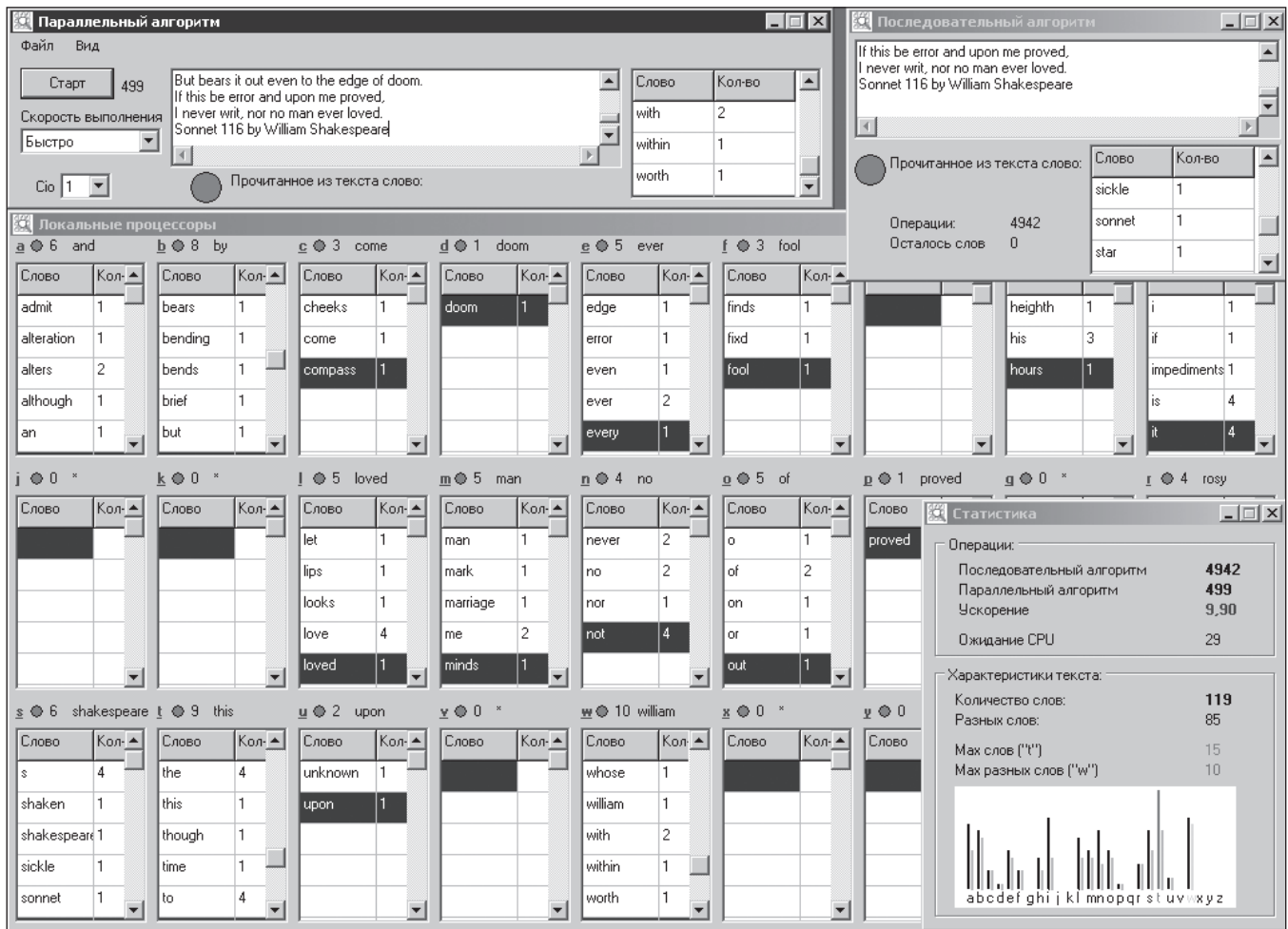


Рис. 2. Вид экрана программы-имитатора 27-процессорной системы

рость исполнения, то можно отчетливо видеть, как ЛП обрабатывает каждое слово: проходит по массиву, сравнивая с его элементами текущее слово, отодвигает «хвост» массива и т. д.

- 4) Наконец, последнее, четвертое окно (справа внизу) отображает различные статистические данные. Прежде всего, это время работы $T1$ и $T27$ и их отношение S , характеризующее параллельное ускорение. Программа также подсчитывает общее время, которое тратит ЦП, ожидая ЛП. Оказывается, с ростом N эта величина растет, но гораздо медленнее, чем $T1$.

Кроме вычислительных характеристик ПО также определяет некоторые параметры самого текста: общее количество слов в тексте, число разных слов, а также строит распределение слов по буквам алфавита. Все эти параметры позволяют понять, как содержимое текста влияет на скорость параллельной обработки.

К программе приложено несколько тестов:

- 1) Тест T0 — это «искусственный» тест, содержащий повторяющиеся в алфавитном порядке комбинации сначала из одной, а затем из двух, трех и т. д. одинаковых букв. Этот тест полностью моделирует предельный случай, описанный выше. Остальные тесты представляют собой реальные тексты на английском языке.
- 2) Тест T1 — это сонет У. Шекспира номер 116.
- 3) Тест T2 — начало первой главы романа Р. Бредбери «451 градус по Фаренгейту».

- 4) Тест T3 — фрагмент документации на процессор Cell.

Программа также позволяет загрузить и обработать любой другой текстовый файл.

Поскольку ПО работает на обычном ПК с операционной системой Windows, его можно использовать при изучении параллельных технологий в любом учебном заведении.

Что можно продемонстрировать на задаче

Анализируя описываемую задачу и используя программу-имитатор, можно продемонстрировать школьникам целый ряд важных идей, связанных с параллельной обработкой данных на многопроцессорных компьютерах.

Во-первых, можно убедительно показать, что параллельная обработка может дать значительный выигрыш по сравнению с последовательной. Кроме того, величина ускорения обработки, как обычно, зависит от параметров задачи. В нашем примере это размер задачи (число слов в тексте) и параметр Cio , характеризующий скорость обмена между ЦП и ЛП.

Во-вторых, можно обратить внимание учеников на основные препятствия, которые затрудняют параллельную реализацию алгоритмов. В [6] выделены четыре возможные причины потери производительности, и все их можно увидеть в нашей задаче и проанализировать. Причины эти следующие:

- 1) увеличение времени выполнения из-за передачи данных из ЦП в ЛП и результатов в обратном направлении;
- 2) необходимость синхронизации параллельных процессов (в нашей задаче ЦП иногда приходится ожидать, пока ЛП не закончит обработку предыдущего слова);
- 3) возникновение в параллельных алгоритмах дополнительных действий, которых не было в последовательном алгоритме (помимо копирований и ожиданий, упомянутых выше, можно привести в качестве примера действия ЦП по выбору ЛП при помощи анализа первой буквы слова — ничего подобного в последовательном алгоритме не было);
- 4) и, наконец, дополнительный расход памяти (помимо общего массива в ЦП каждый ЛП создает свой собственный массив слов).

В-третьих, рассмотренная задача позволяет дать школьникам некоторое представление о методах оценки времени исполнения программ или их фрагментов. В статье показано, как это легко можно сделать, не выходя за рамки школьной алгебры. По мнению автора, применение подобных оценочных вычислений очень полезно для развития мышления школьников.

Примеры заданий

Завершая обсуждение, отметим, что рассмотренная проблема составления частотного словаря хороша не только как иллюстрация к изложению нового материала. Ее еще можно использовать и как основу для решения на уроке множества самых разнообразных задач. Приведем в качестве примера некоторые из них.

1. Пользуясь электронными таблицами, вычислите по выведенным выше формулам значения T_1 , T_{27} и S для разных значений N . Постройте графики для $C_{io} = 1, 2$, и 3 . Найдите области, где $T_{27} > T_1$, а $S < 1$.
2. Выберите некоторый текст и, пользуясь результатами программы-имитатора, постройте график $S(N)$ при $C_{io} = 2$, взяв 10 различных значений N . (Для уменьшения N текст достаточно укоротить.) Сделайте выводы.
3. Для некоторого текста постройте семейство графиков $S(N)$ со значениями C_{io} , равными 1, 2 и 3. Объясните, как влияет данный параметр на величину параллельного ускорения S . Обратите внимание на тот факт, что рост C_{io} снижает время ожидания ЦП.
4. Сравнив значения T_1 и T_{27} для нескольких текстов одинаковой длины при фиксированном C_{io} , определите, какое из них сильнее влияет на изменение ускорения S . Зависит ли результат от N ?
5. Пользуясь окном статистики, проанализируйте получающиеся результаты для общего времени ожидания ЦП при разных значениях параметров задачи. Сопоставьте это время с общим временем T_{27} . Существенны ли потери?
6. Сравните результаты работы программы для конкретного текста с полученными по формулам для предельного случая. Оцените примерное значение N , ниже которого расхождение не превышает 10 %.

7. Измерьте время выполнения программой последовательного и параллельного алгоритмов в секундах и сравните отношение полученных значений с величиной S .
8. Определите, зависит ли ускорение S для демонстрационного теста T_0 (см. описание в тексте статьи) от порядка следования слов. Результаты объясните.
9. Как влияют сочетания слов вроде «to the tree» (т. е. начинающиеся на одну букву) на время выполнения последовательного и параллельного алгоритмов?
10. Изучите подробнее случай, когда в тексте очень много слов (или даже все) начинаются на одну и ту же букву. Сделайте вывод.
11. Экспериментально проверьте, как влияет на значения T_1 , T_{27} и S количество повторяющихся слов в тексте. Объясните результаты.
12. Попробуйте составить текст (не обязательно осмысленный), для которого значение S максимально близко к полученной для предельного случая формуле. (Значения N и C_{io} зафиксировать.) Подумайте, можно ли «превзойти» значение S , предсказанное формулой.
13. Попробуйте получить противоположный эффект — подбором слов в тексте добиться минимально возможного значения S . Сравните свои результаты с результатами соседей.
14. Поэкспериментируйте с разными по содержанию текстами одинаковой длины. Удалось ли вам увидеть какие-нибудь закономерности для S ?

Таким образом, в статье предложена простая методика демонстрации школьникам основ параллельных технологий обработки данных. Рассмотренный пример нагляден и легко понимается, что позволяет рекомендовать его для рассмотрения в школьном курсе информатики. Наличие программной поддержки дает возможность проводить практические занятия по изученному теоретическому материалу.

Литература

1. Алгоритмика: учебное пособие / А. К. Звонкин, С. К. Ландо, А. Л. Семенов, А. Х. Шень. М.: Институт новых технологий образования, 1994.
2. Босова Л. Л. Параллельные алгоритмы в начальной и основной школе // Информатика в школе. 2015. № 2.
3. Дуванов А. А. Азы информатики. Знакомимся с компьютером. Книга для ученика. 5 класс. СПб.: БХВ-Петербург, 2007.
4. Кнут Д. Э. Искусство программирования. Т. 3. Сортировка и поиск. М.: Вильямс, 2007.
5. Левитин А. В. Алгоритмы. Введение в разработку и анализ. М.: Вильямс, 2006.
6. Лин К., Снайдер Л. Принципы параллельного программирования. М.: Изд-во Московского университета, 2013.
7. Плаксин М. А. О методике начального знакомства с параллельными вычислениями в средней школе // Информатика в школе: прошлое, настоящее и будущее: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании. Пермь: ПГНИУ, 2014.
8. Сокольская М. А. Содержание курса «Основы параллельного программирования» // Вестник Красноярского государственного педагогического университета им. В. П. Астафьева. 2011. Т. 1. Вып. 3.



В. Л. Дмитриев,
Стерлитамакский филиал Башкирского государственного университета

ОБ ЭФФЕКТИВНЫХ РЕШЕНИЯХ В ЗАДАЧАХ ПО ПРОГРАММИРОВАНИЮ*

Аннотация

В статье рассматриваются вопросы поиска эффективных алгоритмов при решении некоторого класса задач в процессе обучения программированию. Под эффективностью алгоритма понимается выполнение требований по минимальному использованию памяти программой и скорости работы алгоритма. Представлены решения одной из авторских задач.

Ключевые слова: программирование, эффективность, скорость вычислений, алгоритм, обучение.

Контактная информация

Дмитриев Владислав Леонидович,
канд. физ.-мат. наук, доцент, доцент кафедры информатики и программирования Стерлитамакского филиала Башкирского государственного университета; *адрес:* 453103, Республика Башкортостан, г. Стерлитамак, пр-т Ленина, д. 49; *телефон:* (347-3) 43-45-67; *e-mail:* admwell@yandex.ru

V. L. Dmitriev,
Sterlitamak Branch of the Bashkir State University

ABOUT EFFECTIVE SOLUTIONS IN PROGRAMMING TASKS

Abstract

The article deals with the problems of finding efficient algorithms for solving a some class of tasks in the process of training in programming. Under the efficiency of the algorithm is understood the fulfillment of the requirements for the minimum memory usage of the program and the speed of the algorithm. The solutions of one of the author's tasks are presented in the article.

Keywords: programming, efficiency, computing speed, algorithm, training.

Программирование является важнейшей областью компьютерного мира, ключевым элементом в создании любого приложения. Развитие программирования идет в основном в направлении совершенствования сред разработки. Так, например, в свое время появились поддержка технологии объектно-ориентированного программирования и такие средства разработки, как Java, средства быстрой визуальной разработки и отладки Delphi, C++ Builder, Visual Basic, Visual Fortran и др. Кроме этого созданная корпорацией Borland новая .Net-оболочка (для универсальной платформы Microsoft.Net) поддерживает несколько языков программирования (Delphi, C++, C#, Visual Basic). Это означает, что первоначальное изучение какого-либо из языков программирования с постепенным совершенствованием навыков и повышением мастерства позволит в дальнейшем без особых трудностей осваивать новые платформы и операционные системы и переносить на них существующие программы без существенных модификаций исходных текстов [4].

Программированию на различных языках посвящено огромное множество книг и публикаций как российских, так и зарубежных авторов. К сожалению, в большей части из них в качестве примеров рассматриваются лишь типовые задачи, многие из которых очень просты в реализации. При этом в большинстве случаев при изучении программирования (как в школе, так и в вузе) редко обращается внимание на поиск наиболее эффективных решений той или иной задачи. По крайней мере, этот вопрос обычно не рассматривается, если предложенный алгоритм решения задачи дает более или менее хорошие результаты в плане скорости вычислений (при этом зачастую об экономии памяти речь вообще не идет). Стоит отметить, что в школьных учебниках информатики оптимизации программного кода практически не уделяется внимания (хотя некоторые указания на существование лучших решений даются, например, в [7]).

Однако многие задачи, даже являющиеся достаточно простыми, могут быть решены более эффективными алгоритмами. Простота таких задач, очевидно, и является причиной того, что решение, не лежащее на поверхности, а требующее некоторых мыслительных операций, так и остается ненайденным [1].

С целью привлечения внимания к проблемам поиска и реализации эффективных алгоритмов автором в последние годы организуется **городская олимпиада по программированию в дистанционно-очной форме среди студентов и школьников города Стерлитамак Республики Башкортостан**.

Дистанционный тур проходит в течение шести дней. Задания (по одному в день) высылаются участникам ежедневно в одно и то же время (определенное положением о проведении олимпиады) на адреса электронной почты, и в течение двух часов после рассылки заданий участник должен прислать ре-

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_09-2015/

шение. По окончании дистанционного тура участники, набравшие определенное число баллов, допускаются к очному туру олимпиады.

Продолжительность **очного тура** — три часа, участникам предлагаются три задачи. При подведении итогов олимпиады оценивается эффективность использования памяти, скорость работы алгоритма, его оригинальность и, конечно, время решения задачи участником.

Результаты олимпиады показывают, что только около 20 % студентов и школьников могут предложить более или менее эффективные алгоритмы решения некоторых задач. О высокой степени эффективности речь здесь вообще не идет.

Опыт разбора задач олимпиад показал, что задачи на поиск эффективных алгоритмов пользуются большим успехом у участников, развивают и стимулируют их мыслительные и конструктивные способности [5], алгоритмическое мышление, способствуют развитию творческих способностей.

Некоторые задачи, в которых присутствуют элементы эффективных решений, рассмотрены в работах [2, 3, 6], проблемам поиска и реализации эффективных алгоритмов посвящена работа [1].

Рассмотрим теперь задачу, демонстрирующую возможности использования эффективных алгоритмов. Похожие задачи могут быть предложены как при обучении программированию, так и при проведении олимпиад (при оценке задач олимпиады максимальный балл будет давать именно оригинальный эффективный алгоритм). Входные и выходные данные задачи размещаются в файлах.

Задача.

Жили-были в селении отдаленном дед да бабка с внучкой и собачкой Жучкой. И было у деда хобби — любил он репку растить. Поначалу, конечно, дела у деда шли не так хорошо, как хотелось бы — слишком маленькая репка по массе вырастала — всего 89 грамм каждая, и приходилось сажать ее много и ухаживать за ней долго — ладно хоть внучка да Жучка помогали.

Но со временем надоело внучке и Жучке деду помогать — репку они не любили и не ели вовсе, а все время свое на ней почем зря убивали. Но вот как-то раз не успел дед вовремя семян купить, и пришлось ему репку из семян своей же репки сеять. И случилось чудо — репка-то из своих семян больше выросла — по целых 135 грамм: то ли у деда за много лет мичуринские способности открылись, то ли соседство с атомной станцией не прошло даром.

Обрадовался дед и решил впредь репку только из своих семян сажать. И уже через год масса каждой репки стала еще больше. Раз так дела хорошо пошли, решил дед поставки репки магазину местному открыть, да вот беда: договор поставок заключать надо, а деду для этого нужно знать, какой урожай у него на годы вперед будет. Хорошо хоть внучка, как за репкой смотреть перестала, в вуз на физико-математический факультет поступила. Поэтому сразу же нашла закономерность в величинах масс репки.

Оказалось, что число, выражающее массу репки, таково, что для него выполняется следующее простое правило: если, например, A, B, C — цифры, составляющие число ABC , то сумма соответствующих степеней этих цифр дает само число: $ABC = A^1 + B^2 + C^3$.

Так, например, $89 = 8^1 + 9^2$, $135 = 1^1 + 3^2 + 5^3$.

Так как вручную находить следующие значения масс репки долго и неудобно, написала внучка программу сначала на языке Turbo Pascal, а затем и на C++ ради интереса переделала.

Повторите работу внучки и напишите программу, которая позволит предсказать массу репки на несколько лет вперед.

Входные данные. В единственной строке файла *input.txt* указано число (не превышающее 10^8), до которого нужно искать возможную массу репки.

Выходные данные. В выходной файл *output.txt* нужно вывести последовательность чисел, выражающих массу репки в следующие годы.

В таблице приведены *тестовые примеры*, показывающие содержимое входного и выходного файлов задачи.

Таблица

Содержимое входного и выходного файлов задачи

№ п/п	<i>input.txt</i>	<i>output.txt</i>
1	200	175
2	550	175 518
3	1400	175 518 598 1306

Решение.

Рассмотрим варианты решения данной задачи с использованием системы программирования Delphi (будем создавать консольное приложение).

Самый простой вариант решения задачи — это открыть цикл от 10 (например) и до указанного во входном файле числа. В цикле переводить очередное число в строку для более удобного выделения цифр числа, выполнять требуемые операции и искать массы репки. Такой код будет иметь следующий вид:

```

program N1;
{$APPTYPE CONSOLE}
uses SysUtils, DateUtils;
var
  i, n, s, max: longint;
  a, j, k, nn, c: integer;
  ss: string;
  f: text;
  t1, t2: TDateTime;
begin
  t1:=Now;
  assign(f, 'input.txt');
  reset(f);
  read(f, max);
  close(f);
  assign(f, 'output.txt');
  rewrite(f);
  for i:=10 to max do
  begin
    s:=0;
    str(i,ss);
    nn:=length(ss);
    for j:=1 to nn do
    begin
      val(ss[j], a, c);
      n:=1;
      for k:=1 to j do
        n:=n*a;
      s:=s+n;
    end;
  end;

```

```

    if s=i then writeln(f, i);
  end;
close(f);
t2:=Now;
writeln('Time ', MillisecondsBetween(t2,
t1)/1000:5:3, ' s. ');
write('Program is complete');
readln;
end.

```

Здесь переменные $t1$ и $t2$ введены для того, чтобы подсчитать время, в течение которого программа выдает решение.

Тестирование этой версии программы и следующих проведем на компьютере с процессором Core i3-3240 (3,4 ГГц).

Взяв в качестве входного значения 10^8 , получим время работы 103,6 секунды.

Теперь попробуем усовершенствовать алгоритм. Очевидно, одной из слабых сторон приведенного выше алгоритма является повторный подсчет степеней цифр, составляющих числа, поэтому введем двумерный массив для хранения предварительно подсчитанных степеней всех цифр. Опишем функцию вычисления степени цифры, а также откажемся от использования процедуры *val*. В итоге код программы будет иметь следующий вид:

```

program N2;
{$APPTYPE CONSOLE}
uses SysUtils, DateUtils;
var
  i, m, s, max: longint;
  k, j: integer;
  a: array[1..10, 1..10] of longint;
  ss: string;
  f: text;
  t1, t2: TDateTime;

function stepen(x, y: byte): longint;
var
  i: byte;
  z: longint;
begin
  z:=1;
  for i:=1 to y do
    z:=z*x;
  stepen:=z;
end;

begin
  t1:=Now;
  assign(f, 'input.txt');
  reset(f);
  read(f, max);
  close(f);
  assign(f, 'output.txt');
  rewrite(f);
  for i:=0 to 9 do
    for j:=1 to 10 do
      a[i+1, j]:=stepen(i, j);
    for i:=10 to max do
      begin
        if i mod 10 <> 0 then
          begin
            m:=i;
            s:=0;
            str(i, ss);
            k:=length(ss);
            while m<>0 do
              begin
                j:=m mod 10;
                m:=m div 10;
                s:=s+a[j+1, k];

```

```

            k:=k-1;
            if s>i then m:=0;
          end;
          if s=i then writeln(f, i);
        end;
      end;
    close(f);
    t2:=Now;
    writeln('Time ', MillisecondsBetween(t2,
t1)/1000:5:3, ' s. ');
    write('Program is complete');
    readln;
  end.

```

Взяв в качестве входного значения 10^8 , получим время работы уже 18,7 секунды, т. е. усовершенствованный код работает примерно в 5,5 раз быстрее.

Можно написать код еще эффективнее, если избавиться в программе от процедуры *str*, которая используется в программе для получения количества цифр числа. Сделать это можно, введя дополнительную переменную v , смысл которой понятен из кода программы, приведенной ниже:

```

program N3;
{$APPTYPE CONSOLE}
uses SysUtils, DateUtils;
var
  i, m, s, v, max: longint;
  k, j, n: integer;
  a: array[1..10, 1..10] of longint;
  f: text;
  t1, t2: TDateTime;

function stepen(x, y: byte): longint;
var
  i: byte;
  z: longint;
begin
  z:=1;
  for i:=1 to y do
    z:=z*x;
  stepen:=z;
end;

begin
  t1:=Now;
  assign(f, 'input.txt');
  reset(f);
  read(f, max);
  close(f);
  assign(f, 'output.txt');
  rewrite(f);
  for i:=0 to 9 do
    for j:=1 to 10 do
      a[i+1, j]:=stepen(i, j);
    n:=2; v:=100;
    for i:=10 to max do
      begin
        if i mod 10 <> 0 then
          begin
            m:=i;
            s:=0;
            k:=n;
            while m<>0 do
              begin
                j:=m mod 10;
                m:=m div 10;
                s:=s+a[j+1, k];
                k:=k-1;
                if s>i then m:=0;
              end;
            if s=i then writeln(f, i);
          end;

```

```

else if i mod v = 0 then
begin
  v:=v*10;
  n:=n+1;
end;
end;
close(f);
t2:=Now;
writeln('Time ', MillisecondsBetween(t2,
t1)/1000:5:3, ' s. ');
write('Program is complete');
readln;
end.

```

Снова взяв в качестве входного значения 10^8 , получим время работы 6,0 секунд, т. е. еще в три раза быстрее.

Таким образом, усовершенствованный код эффективнее первоначального примерно в 17 раз.

Рассмотренный пример показывает, насколько эффективным может быть решение той или иной задачи: остается научить учащихся заниматься поиском таких решений.

Мы считаем, что одной из главных целей при изучении программирования должно стать приобретение

учащимися навыков конструктивного мышления, т. е. умения видеть различные варианты реализации поставленной задачи.

Литература

1. *Дмитриев В. Л.* Об эффективных алгоритмах решения ряда задач при обучении программированию // Профильная школа. 2014. Т. 2. № 3.
2. *Дмитриев В. Л.* Опыт отбора задач для активизации интереса школьников к изучению программирования // Информатика в школе. 2015. № 4.
3. *Дмитриев В. Л.* Поэтапная разработка программы в среде Turbo Pascal на примере поиска пути с использованием волнового алгоритма // Информатика и образование. 2013. № 8.
4. *Дмитриев В. Л.* Теория и практика решения задач по программированию. Ч. 1. Уфа: РИЦ БашГУ, 2007.
5. *Дмитриев В. Л., Ахмадеева Р. З.* Развитие конструктивного мышления при изучении программирования // Информатика и образование. 2009. № 2.
6. *Окулов С. М.* Программирование в алгоритмах. М.: БИНОМ. Лаборатория знаний, 2002.
7. *Поляков К. Ю., Еремин Е. А.* Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2013.

НОВОСТИ

«Яндекс» открывает информагентство с роботами вместо журналистов

«Яндекс» открывает собственное информационное агентство для СМИ, новости в котором будут написаны не журналистами, а сгенерированы поисковыми роботами. Первое время информационные сообщения будут выходить только по темам погоды и пробок, а со временем список разделов будет дополнен. Поисковые роботы будут генерировать информационные

сообщения на основе мониторинга и анализа данных, собираемых с различных сайтов. Подписка на сервис будет бесплатной. Новости будут представлены в разных форматах: графики для телеканалов, для радиостанций — специальный интерфейс с текстами, для информагентств, газет, журналов и сайтов — лента уведомлений.

«Умная» щетка превращает чистку зубов в игру

Создатели зубной щетки Kolibree, поставки которой начались в феврале благодаря поддержке народных инвесторов с Kickstarter, выпустили новую, еще более «умную» модель. Среди усовершенствований — улучшенная связь со смартфоном по Bluetooth и более точное отслеживание движений. Кроме того, значительно доработано приложение щетки. Теперь в нем есть игра Go Pirate 2, которая позволяет зарабатывать баллы и значки за правильную чистку зубов и отображает таблицу лидеров, чтобы члены

семьи могли состязаться друг с другом в тщательности гигиены полости рта. Приложение также отображает «индекс Kolibree» — рейтинг, подсчитываемый с учетом частоты, регулярности и качества чистки зубов. Кроме того, есть режим, наглядно демонстрирующий, какие области при чистке были пропущены. Можно также настраивать выдачу напоминаний и чистить зубы в режиме отсутствия связи со смартфоном, синхронизируя данные в любое время по желанию.

Зубы, напечатанные на 3D-принтере, защитят полость рта от бактерий

Одним из перспективных направлений использования 3D-принтеров может оказаться протезирование зубов. По прогнозам, 3D-сканирование и 3D-печать уже в скором будущем придут на смену традиционным методам изготовления имплантатов. Однако в Университете Гронингена (Нидерланды) разработали антимикробный пластик, который

можно использовать для 3D-печати искусственных зубов. Новый материал изготовлен из уже используемых в стоматологии полимерных смол с добавлением четвертичных аммониевых солей, которые обладают антибактериальным эффектом. Голландские исследователи утверждают, что для человека подобный пластик совершенно безвреден.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)



И. Н. Слинкина,

дипломант конкурса «Изучаем информатику с героями любимых книг»,
Шадринский государственный педагогический институт, Курганская область

ОЛИМПИАДНЫЕ ЗАДАЧИ ПО ТЕОРЕТИЧЕСКИМ ОСНОВАМ ИНФОРМАТИКИ ДЛЯ ШКОЛЬНИКОВ И СТУДЕНТОВ*

Аннотация

В статье приводятся олимпиадные задачи (с решениями) по теоретическим основам информатики со сказочными сюжетами. Они предлагались участникам Всероссийского студенческого форума «Актуальные проблемы теории и методики информатики, математики и экономики». Задачи могут быть использованы для проведения конкурсов и олимпиад как для школьников, так и для студентов педагогических вузов, обучающихся по направлению подготовки «Педагогическое образование (профиль: “Информатика”)».

Ключевые слова: олимпиада, олимпиадные задачи, теоретические основы информатики, сказочный сюжет, занимательные задачи, олимпиадная задача по теоретическим основам информатики, теория множеств, теория кодирования, логические задачи, круги Эйлера—Венна, задача линейного программирования, двойственные задачи, транспортная задача, системы счисления.

Контактная информация

Слинкина Ирина Николаевна, канд. пед. наук, доцент, декан факультета информатики, математики и физики Шадринского государственного педагогического института, Курганская область; адрес: 641870, Курганская область, г. Шадринск, ул. К. Либкнехта, д. 3; телефон: (352-53) 3-63-69; e-mail: slinkinain@mail.ru

I. N. Slinkina,

Shadrinsk State Pedagogical Institute,
Kurgan Region

OLYMPIAD TASKS ON THE THEORETICAL FOUNDATIONS OF INFORMATICS FOR SCHOOL-CHILDREN AND STUDENTS

Abstract

The article presents the tasks on the theoretical foundations of informatics with fairy tales subjects and the examples they're to be solved. The presented tasks were given to the participants of the All-Russian Student Forum "Actual problems of the theory and methodology of informatics, mathematics and economics". The tasks can be used to hold competitions or academic Olympiad among school students or among the students of pedagogical institutes acquiring the "Informatics" subject-oriented bachelor's program within the "Pedagogical Education" Bachelor Program.

Keywords: Olympiad task on theoretical foundations of informatics, set theory, coding theory, logic problems, Eulerian circles, liner programming problem, dual problems, transport problem, numerical systems.

Задача 6

И откуда у нас в Подберезовке появился этот Вовка Кулибин?! Вот если бы точно не помнили, как его, мелкого и голоштанного, из кузницы крапивой выгоняли, когда он пытался к подкове какую-то фигуринку прикрутить, чтоб, говорит, «скорость у кобылы увеличить», точно бы подумали, что засланец он из другого мира к нам в Подберезовку волшебством бабы-ягинским заброшенный. И мир-то такой чудной есть! Там таких, как Вовка, очень, говорят, много. Все что-то там придумывают, проводят, собирают... Советник Бабы-яги по стратегическим вопросам Елена Устинова сказала, что мир тот загадочный называется «ШеГеПеИ». Что сие означает — загадка.

А Вовка-то Кулибин накликал из деревни мальчишек и учит их чему-то. Собрали за зиму чудо чудное, диво дивное — самодвижущую повозку. Назвали как-то не по-русски — «Броневик». Что это за механизма такая — никто не ведает... Едет он по главной улице Подберезовки, трещит, шумит, курей да малых детушек распугивает. Но ведь быстро едет, только на телеге догнать и можно! А слова-то какие Вовка своим пацанятам говорит! Не иначе как ругает их за что-то! «Трансмиссия, — говорит, — карбюратор!» И еще: «Инжектор, ступица, подшипник...» Это мы под окошком их мастерской вместе с Пашкой Рыжим сидели да на бумажку эти ругательные слова и записывали. А что, блеснем потом перед мужиками ерундичией. А еще он (Вовка то есть) пытается как-то, за что не ведаем! Дает им железью рогатую, «ключ на 17» называется, и заставляет какой-то нагар у свечи чистить иль менять их, мол, «пробило»...

Мы с Пашкой, пока те все искупаться ушли после трудового дня, заглянули в этот броневик. Дык, где там можно свечку поставить, не поняли. Ни лампадки нет, ни подсвечника, ни тарелки какой иль кружки — одни железки. А как свечку «пробить» можно? Не иначе, как шилом в бок. Пашка говорит, что наверняка они свечку ставят, чтоб механизма божественную силу получила! А кормят-то транспортину эту каким-то «горючим». Витька-подмастерье сказал, что это «бензин» или «огненная вода». Дык ни фиги! Какая это огненная вода, когда она хоть и горит, а воняет неправильно! Как пить-то ее? Вон тетка Матрена огненную воду продает, дык она хоть и воняет, да пить-то ее одно удовольствие. Не медовуха, конечно, но не эта дрянь! Мы с Пашкой ихнюю бензину попробовали. Два дня потом животом маялись!

Раз в месяц Вовка Кулибин загоняет механизму эту в большой сарай и (это он сам так говорит) «ставит на диагностику». Берут штучки какие-то волшебные, прикладывают их к разным местам броневика. Витька говорит, что какие-то «параметры» меряют. Потом относят их к Вовке. Тот поколдует над ними полчаса, посчитает, перерисует, с табличками сверит, в формулу поставит и говорит, здорова ли механизма

* Продолжение. Начало см.: Информатика в школе. 2015. № 6, 8.

иль лечить ее надо. Если болеет она, то чем и как лечить. Мудрено все это! Поэтому только Вовка и может с ентим справиться. Умный он больно! Табличка у него на стене висит, а ключ от нее, как и положено: в дубовом ларце — в зайце — в утке — в яйце. Все Вовка своими руками сделал! Рукастый мужик он, мастеровой. Только уж фантазий у него не по нашим подберезовским головам...

Подглядели мы и их «ключ». Сперва думали, что уж если и заяц такой красивый, и утка резная, и яйцо узорчатое сделано, то уж ключ не иначе как резной да затейливый. Оказалось — бумажка какая-то. Где ж замок-то такой найти и куда в него эту бумажку толкать, нам не ведомо. Так вы, люди добрые, наверняка разберетесь. Говорят, что числа должны получиться «с плавающей запятой». Куда эта запятая поплывет и что это за скотинка такая «плавающая запятая», мы не знаем. Еще в заклинии про какого-то типа твердят, не иначе как иностранец! Мы списали по буквам с бумажки, вот так получилось: «тип Single», и еще что-то там про фри паскаль (по буквам: Free Pascal). Может, город какой, может, родители, типа того. Неведомо нам.

Вот сами таблицы (*подсказка*: это прямая последовательность байтов из памяти):

0	4	8	A	0	9	4	4	X1
0	0	0	A	1	A	4	4	X2
0	0	5	8	2	A	C	4	X3
0	0	A	0	7	3	C	4	X4
0	0	9	4	D	3	4	3	X5
0	0	B	0	A	A	C	3	X6
0	0	C	A	5	0	4	4	X7
0	0	2	3	2	3	C	4	X8

Вторая таблица (если врукопашную считать):

4	4	0	9	8	A	0	4	X1
4	4	1	A	0	A	0	0	X2
C	4	2	A	5	8	0	0	X3
C	4	7	3	A	0	0	0	X4
4	3	D	3	9	4	0	0	X5
C	3	A	A	B	0	0	0	X6
4	4	5	0	C	A	0	0	X7
C	4	2	3	2	3	0	0	X8

A в ключе формула:

$$X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8.$$

1. Эти две таблицы были оставлены потомкам великим механиком Ларгусом. Волшебное число S , полученное по формуле из ключа, поможет определить, болеет ли машинка и чем. Посчитайте, люди добрые, это число «в идеальном» исполнении, т. е. когда данные взяты из таблички.

2. После очередной диагностики мальчишки Вовки Кулибина получили следующие данные:

$$\begin{aligned} X1 &= 84.15625; \\ X2 &= 650.375; \\ X3 &= -197.15625; \\ X4 &= 612.15625; \\ X5 &= -222.546875; \\ X6 &= -419.546875; \\ X7 &= 613.375; \\ X8 &= -754.15625. \end{aligned}$$

Если полученное волшебное число больше «идеального», то составляем табличку и преобразуем ее так: каждую цифру четного столбца инвертируем и считаем новые значения переменных. Если же полученное число меньше «идеального», то составляем табличку, инвертируем каждую цифру нечетного столбца и делаем обратный расчет переменных. Если, подставив в формулу, получим меньшее значение, то ремонт затянется, так как поломки есть и они существенные. Если же большее, то поломка незначительная и быстро справимся.

Решение.

1. Найдем значения переменных. Можно воспользоваться стандартными алгоритмами перевода (см., например, <http://www.titorov.ru/index.php/distant/inform-theoretical/549-mashen-code>, <http://comp-science.narod.ru/Cod/cod.html> или в учебниках А. В. Могилева «Информатика», Б. Е. Стариченко «Теоретические основы информатики», в учебниках по структурам и алгоритмам обработки данных), можно написать небольшую программку на любом языке программирования, поддерживающем данный тип, и выделить содержимое соответствующих байтов. Необходимо только помнить, что при размещении данных в памяти байты переставляются, т. е. формально они получаются в порядке 1 — 2 — 3 — 4, а в памяти они хранятся как 4 — 3 — 2 — 1.

$$\begin{aligned} X1 &= 550.1565; \\ X2 &= 616.15625; \\ X3 &= -681.375; \\ X4 &= -974.5; \\ X5 &= 423.15625; \\ X6 &= -341.375; \\ X7 &= 835.15625; \\ X8 &= -652.546875. \end{aligned}$$

Подставив полученные значения в формулу, получим: $S = -225.171875$.

2. $S = 366.65625$, $S > -225.171875$. Это означает, что необходимо идти по первому пути.

Составим таблицу:

0	0	5	0	A	8	4	2	X1
0	0	9	8	2	2	4	4	X2
0	0	2	8	4	5	C	3	X3
0	0	0	A	1	9	4	4	X4
0	0	8	C	5	E	C	3	X5
0	0	C	6	D	1	C	3	X6
0	0	5	8	1	9	4	4	X7
0	0	8	A	3	C	C	4	X8

Или:

4	2	A	8	5	0	0	0	X1
4	4	2	2	9	8	0	0	X2
C	3	4	5	2	8	0	0	X3
4	4	1	9	0	A	0	0	X4
C	3	5	E	8	C	0	0	X5
C	3	D	1	C	6	0	0	X6
4	4	1	9	5	8	0	0	X7
C	4	3	C	8	A	0	0	X8

Преобразуем таблицу:

0	F	5	F	A	7	4	D	X1
0	F	9	7	2	D	4	B	X2
0	F	2	7	4	A	C	C	X3
0	F	0	5	1	6	4	C	X4
0	F	8	3	5	1	C	C	X5
0	F	C	9	D	E	C	C	X6
0	F	5	7	1	6	4	B	X7
0	F	8	5	3	3	C	B	X8

Вторая таблица строится стандартным методом. Переведем шестнадцатеричный код в число с плавающей запятой.

$$X1 = 351003100;$$

$$X2 = 11376399;$$

$$X3 = -52993084;$$

$$X4 = 39326780;$$

$$X5 = -54922300;$$

$$X6 = -11680370;$$

$$X7 = 9852687;$$

$$X8 = -11765007.$$

Вычислим S : $S = 280198205$. S намного больше идеального, следовательно, мальчишки справятся с поломкой быстро.

Задача 7

— Опять кого-то принесло... Не иначе, как опять Иван-царевич. Что за наказание! Как царевич — так дурак, как дурак — так умница редкостная! — думала Баба-яга, выглядывая в окошко.

В этот раз не повезло... Царевич...

— Избушка, избушка, повернись к лесу задом, ко мне передом! — надрывно прокричал Ванька.

— Ну-ну, кричи, касатик, кричи... Сказки-то не только почитать надо, но и почитать, там ведь все написано, только внимательным будь да усидчивым!

— Избушка, избушка, повернись к лесу задом, ко мне передом!

Через минуту:

— Избушка, избушка...

После десятиминутных завываний царевич взвыл:

— Ну что ты не шевелишься совсем!?

— А зачем ей шевелиться, если она и так к тебе передом повернутая стоит! Неуч, у избушки зад от переда отличить не можешь!

— Здравствуйте, уважаемая Баба-яга. Как поживаете? Как здоровье ваше драгоценное? Как погода? Не будете ли вы так любезны ответить мне на несколько вопросов? Не сочтете ли за труд удовлетворить мое любопытство? Не могли бы вы рассеять мои сомнения?..

— Блин, еще и интеллигент на мою голову! Сейчас, пока до сути доберется, будет минут тридцать ножкой шаркать! — проворчала Баба-яга тихонько. И громко крикнула в окно:

— Короче, Склифосовский! Могу!

— Что вы говорите! Как я рад! Я вам очень признателен! Моя благодарность не имеет границ! Спасибо вам огромное!..

— Ближе к делу!

— Я даже не знаю, с чего начать...

— С начала!

— Украл у меня Кошей Бессмертный невесту мою, Аленушку-умницу...

— Ну, слава Богу, в семье хоть один нормальный человек будет... И что дальше?

— Она прислала мне письмо загадочное и велела в случае чего к вам идти. Ну вот, я не понял: что такое «в случае чего»? Зачем к вам идти, если весточку от нее я получил?

— А ты, горемычный, письмо-то хоть прочитал?

— Нет, а зачем? Читать велено не было. Велено же было получить — я получил. Только вот со «случаем *чего*» не разобрался. Дык к вам пришел.

— Горе ты горькое! И что в тебе только Аленка-то нашла...

— Как чего? Я красив, богат, умен...

— Ну, с последним трудно не согласиться, — пробурчала Баба-яга. — Давай сюда письмо. А сейчас кот мой Василий тебя к делу приспособит. Дров там нарубить, баньку истопить, в избе прибраться... Делать-то что-нибудь умеешь?

— Умею! Царь-батюшка сказал, что если уж не головой, то руками должен уметь все делать! Все сделаю, бабушка, в лучшем виде! Не извольте беспокоиться. Всегда рад помочь. Я не могу оставить без ответа вашу мольбу о помощи...

— Заткнись... Иди, работай!

— Так, умница-то умница, а что за письмецо? Пока не пойму... Попробуем расшифровать в лоб. Шестнадцатеричный код. Переводим в двоичный. Явно ничего сложного. Будем шифровать обычным кодом: 32 буквы, 5 символов код, пробел — 00000, «е» и «ё» — один символ, «и» и «й» — тоже. Два ноля последних отбросим. Явно лишние. Блин, а не получается! Но количество букв-то правильное! Вроде и в буквы переводится корректно... Есть тут какая-то западня... Какая же, интересно? «В случае чего», «в случае *чего*»... Знаю я один шифр со сдвигом на слово. Попробуем его реализовать. Шифр называется шифром Виженера с использованием ключевого слова «чего». Получилось! Бедный царевич... (ПыСы. Пробел не считать при кодировании шифром Виженера, а букв 33, как положено!)

3D6224B41530C11472C0920178341530C07A0509ADC8601
FEA4462605F590296807380B9VB6019D33181530C

Решение.

1. Переводим сообщение из шестнадцатеричного кода в двоичный. Получим:

```
0011 1101 0110 0010 0010 0100 1011 0100 0001 0101
0011 0000 1100 0001 0001 0100 0111 0010 1100 0000
1001 0010 0000 0001 0111 1000 0011 0100 0001 0101
0011 0000 1100 0000 0111 1010 0000 0101 0000 1001
1010 1101 1100 1000 0110 0000 0001 1111 1110 1010
0100 0100 0110 0010 0110 0000 0101 1111 0101 1001
0000 0010 1001 0110 1000 0000 0111 0011 1000 0000
1011 1001 1011 1011 0110 0000 0001 1001 1101 0011
0011 0001 1000 0001 0101 0011 0000 1100
```

2. Уберем два последних ноля и разделим по пять знаков:

```
00111 10101 10001 00010 01001 01101 00000 10101 00110
00011 00000 10001 01000 11100 10110 00000 10010 01000
00000 10111 10000 01101 00000 10101 00110 00011 00000
00111 10100 00001 01000 01001 10101 10111 00100 00110
00000 00111 11111 01010 01000 10001 10001 00110 00000
10111 11010 11001 00000 01010 01011 01000 00000 11100
11100 00000 10111 00110 11101 10110 00000 00110 01110
10011 00110 00110 00000 10101 00110 00011
```

3. Составим кодовую таблицу.

а	б	в	г	д	е	ё
00001	00010	00011	00100	00101	00110	—
ж	з	и	й	к	л	м
00111	01000	01001	—	01010	01011	01100
н	о	п	р	с	т	у
01101	01110	01111	10000	10001	10010	10011
ф	х	ц	ч	ш	щ	ъ
10100	10101	10110	10111	11000	11001	11010
ы	ь	э	ю	я	пробел	
11011	11100	11101	11110	11111	00000	

4. Переведем двоичный код, полученный на шаге 2, в текст:

жхсайн хев сзыч тз чрн хев жфбзйхчге жякзссе чыщ клз
ьь чеэц еоуее хев

5. Осуществим расшифровку с помощью кода Виженера с ключевым словом «чего» (при расшифровке удаляем пробелы)*:

Проститцкменянеишитцквстретиладругогнтчкуженет
воаленатцк

Понятно, что вместо «тцк» должно быть «тчк», и «ц» мы получили из-за того, что приняли «ё» и «е» за одну букву.

Ответ. Послание Алены к Ивану-царевичу: «прости тчк меня не ищи тчк встретила другого тчк уже не твоя алена тчк».

* Для расшифровки можно воспользоваться, например, онлайнным калькулятором для шифра Виженера: <http://planetcalc.ru/2468/>

Задача 8

Зачастили что-то гости дорогие к Бабе-яге. Кто, и правда, дорогой, в смысле — долгожданный, добрый и душевный, а иногда такие «дорогие» встречаются, что сам бы приплатил, лишь бы их не видеть... Вот один из таких «дорогих» и прибыл с утречка, ближе к обеду (может, даже к ужину) к Бабе-яге. Права качать. Лучше бы воду качал или нефть на крайняк, толку бы больше было! Чтoб не орал да глотку не драл, вышла к нему Баба-яга самолично.

— Что надо, красавчик?

— Несознательные у нас пенсионеры пошли! Нет, чтобы накормить, напоить, в баньке попарить...

— Ну, за этим надо к Марье-искуснице ехать. Десять верст на север, тринадцать на восток, три дня лесом, и ты на месте! А ко мне только с проблемами да загадками ходят. Ну, говори, дело пытаешь аль от дела лытаешь?

— Че сразу так, по сказочному-то? Ладно, дело у меня к тебе есть тайное, *государево!* Он мне весточку велел везти к другу заклитому султану шамаханскому, но надо письмецо энто как-то странно перерисовать. Слушай, а тебе, старой карге, поди и слова-то такие неведомы? На фига я три дня скакал?

— Что, скакалку отобрали через три дня-то? До меня за день добраться можно, если тайные тропы знать!

— Я собачьими тропами не хаживал никогда! Царских я кровей, а не холопских! Езжу только верстовыми дорогами!

— А ну, говори слова свои заветные, а то сейчас на каргу да на старую обижушь и превращу тебя в мухомор!

— Не посмеешь! Я царевич тридесятого царства тридевятиого государства! У меня коней — табун, злата-серебра — полны закрома, каменьев самоцветных не счесть!

— Ума бы лучше да уважительности к старшим! Пшел вон!

— Как ты смеешь, старая калоша, мне приказывать! Да я, да тебя...

Не успел царевич слова молвить, превратила его Баба-яга в мухомор.

— Говорил я ему, вежливым надо быть да воспитанным... А он: «Я царевич!» да «Я царский сын!» И, эх...

— А ты кто? — удивилась Баба-яга.

— Да я кузнеца Миколы средний сын — Васятка.

— А сколь лет-то тебе?

— Скоро шестнадцать стукнет.

— Не прибило бы, когда стукать будет... Да ладно, шучу. И что ты у петуха это гарвардского делаешь?

— Головой за него работаю. Думаю да советы даю. Да не слушается он. Вот и приходится его вытаскивать из передрыг всяких. Хоть и надоедно, да с малолетства мы с ним вместе, привык я. Меня даже тенью его звали, да потом попустились... Помоги, бабушка, сделай милость. Без тебя мы не справимся. Да меня научи, чтобы мы больше тебя по пустякам не дергали.

— Слова-то заветные знаешь?

— Знаю! «Неравномерный код с разделителем». Тебе говорит это о чем-то?

— Конечно! Книги мудрые да волшебные читать надобно, вот и ты про все это знать будешь! Текст давай!

— Даю...

F962B3161A036588C344E1A0286282A60689658D1B9A0E9E686E0

— Ух ты, садись-ка ко мне поближе... Будем вместе задачку решать! Переведем в двоичный код. Дальше... Надо переводить... Возьмем стандартный случай. Условимся, что разделителем отдельных кодов букв будет последовательность 00 (признак конца знака), а разделителем слов — 000 (признак конца слова — пробел). Правила построения кодов:

- код признака конца знака может быть включен в код буквы, поскольку не существует отдельно (т. е. коды всех букв будут заканчиваться на 00);
- коды букв не должны содержать двух и более нулей подряд в середине (иначе они будут восприниматься как конец знака);
- код буквы (кроме пробела) всегда должен начинаться с 1;
- разделителю слов (000) всегда предшествует признак конца знака, при этом реализуется последовательность 00000 (т. е., если в конце кода встречается комбинация «...000» или «...0000», она не воспринимается как разделитель слов); следовательно, коды букв могут оканчиваться на 0 или 00 (до признака конца знака).

Вот и все правила. Но что-то мне здесь не нравится. Сильно просто будет, если мы стандартный код с разделителем возьмем! Мы, знаешь, как сделаем — мы не будем учитывать вероятность появления символа в тексте, а возьмем русский алфавит в перевозданном виде! У нас так классно получится (в конце фразы — пробел)! Потом переведем в шестнадцатеричный код. Только когда будешь отправлять шамаханскому царю, про эту особенность напомнить не забудь. И про то, что коды должны быть последовательны, без пропусков и в возрастающем порядке. Понял?

— Все понял! Сделаю! Спасибо тебе, Баба-яга, выручила. А как мне получить царевича-то обратно?

— А вот ответ от царя шамаханского придет, ты его расшифруешь правильно и получишь своего царевича. Договорились?

— А у меня, что, выбор есть? Привык я к нему...

Прошла неделя-другая, возвращается Васятка к Бабе-яге с таким посланием:

A9810B19634602CB9C056628741D2B30F366E0B2E701042C658D180B2E70159C2C0E96549E0

— Перевел я, бабушка! Проверь, пожалуйста!

— Молодец! Получай своего олуха обратно! Только, это... Минут через пять он расколдуется, дык ты его под локотки хватай да тикай, а то опять мне гадостей наговорит, дык я его уж и посильнее заколдую!

— Спасибо, бабушка!

А вы, добры молодцы да красны девицы, зашифруйте да расшифруйте оба послания.

Решение.

Возьмем первый текст:

F962B3161A036588C344E1A0286282A60689658D1B9A0E9E686E0

Переведем его в двоичный код:

```
1111 1001 0110 0010 1011 0011 0001 0110 0001 1010
0000 0011 0110 0101 1000 1000 1100 0011 0100 0100
1110 0001 1010 0000 0010 1000 0110 0010 1000 0010
1010 0110 0000 0110 1000 1001 0110 0101 1000 1101
0001 1011 1001 1010 0000 1110 1001 1110 0110 1000
0110 1110 0000
```

Запишем без пробелов:

```
11111001011000101011001100010110000110100000001101100
101100010001100001101000100111000011010000000101000
0110001010000010101001100000011010001001011001011000
11010001101110011010000011101001111001101000011011100
```

Построим коды символов русского алфавита (берем все подряд идущие двузначные числа и вычеркиваем те, в записи которых на конце три нуля или в середине два нуля, и приписываем к каждому оставшемуся числу справа два нуля):

а	б	в	г	д	е	ё
100	1000	1100	10000	10100	11000	11100
ж	з	и	й	к	л	м
101000	101100	110000	110100	111000	111100	1010000
н	о	п	р	с	т	у
1010100	1011000	1011100	1101000	1101100	1110000	1110100
ф	х	ц	ч	ш	щ	ъ
1111000	1111100	10101000	10101100	10110000	10110100	10111000
ы	ь	э	ю	я	пробел	
10111100	11010000	11010100	11011000	11011100	000	

Разделим наш код на группы символов, которые мы можем однозначно декодировать:

```
1111100 1011000 10101100 11000 10110000 11010000 000
1101100 1011000 1000 110000 1101000 100 1110000 11010000
000 1010000 11000 10100 000 1010100 11000 000 1101000 100
101100 1011000 1101000 11011100 110100 000 1110100 111100
11010000 11011100
```

Переведем:

Хочешь собирать мед не разоряй улья

Переведем второй текст:

A9810B19634602CB9C056628741D2B30F366E0B2E701042C658D180B2E70159C2C0E96549E0

Запишем его в двоичном коде:

```
1010 1001 1000 0001 0000 1011 0001 1001 0110 0011
0100 0110 0000 0010 1100 1011 1001 1100 0000 0101
0110 0110 0010 1000 0111 0100 0001 1101 0010 1011
0011 0000 1111 0011 0110 0110 1110 0000 1011 0010
1110 0111 0000 0001 0000 0100 0010 1100 0110 0101
1000 1101 0001 1000 0000 1011 0010 1110 0111 0000
0001 0101 1001 1100 0010 1100 0000 1110 1001 0110
0101 0100 1001 1110 0000
```

Уберем пробелы:

```
1010100110000001000010110001100101100011010001100000
0010110010111001110000000101011001100010100001110100
00011101001010110011000011110011011001101110000010110
0101110011100000001000001000010110001100101100011010
0011000000010110010111001110000000101011001110000101
100000011101001011001010100100111100000
```

Разделим на кодовые комбинации:

1010100 11000 000 10000 1011000 1100 1011000 1101000
110000 000 101100 1011100 1110000 000 10101100 11000
1010000 1110100 000 1110100 10101100 110000 111100 1101100
11011100 000 101100 1011100 1110000 000 100 000 10000
1011000 1100 1011000 1101000 110000 000 101100 1011100
1110000 000 10101100 1110000 1011000 000 1110100 101100
1010100 10011100 000

Переведем:

Не говори зпт чему учился зпт а говори зпт что узнал

Задача 9

Как-то раз правители Тридесятого царства Тридевятого государства и Тридевятого царства Тридесятого государства, за рюмкой чая обсуждая «полихтическую ситуацию», решили не отставать от модных веяний и придумали создать у себя Шпиенский приказ. Воеводами назначили двух самых больших пройдох своих царств-государств. В Тридевятиом царстве главным шпионом назначили Колобка. Он три года в Янкленде жил, иностранный лоск приобрел. На имя Колобок не откликается, только на Даунтнет какой-то. Иностранные слова говорит разные: «Окей», «веригут», «О, май гад». Про гада мы поняли, а вот при чем тут «май» — не очень. Ну, может, оно и на пользу делу будет. В Тридесятом же царстве выбрали Жихарку. Ухватист, пронырлив, ловок, в любую дырку залезет, любого уговорит, эскимосам снег зимою продаст. Велели Жихарке и Колобку работать отдельно, но информацией обмениваться регулярно. Работа началась.... Жихарка по своему царству-государству носится, шпиёнов ловит, а Колобок, Даунтнет который, по своему. Переловили полцарства. Решили обменяться информацией, как велено было. Послали друг к другу посыльных с письмами тайными.

«Мистер Жихарка, нашел я на вверенной мне территории шпиёнов иностранных немерено. Чтoб никто не понял, сколько, пишу шифром тайным, с подсказкой маленькой: в десятке не 10 единиц. $\frac{163 \cdot 11}{5} + 391$. Всегда Ваш, Даунтнет».

Получил Жихарка письмо сие. Долго думал и решил отомстить: «Я ниче не понял, ну и тебе того же».

«Дорогой Колобок или как там тебя... У меня такая же фигня на территории. Кишмя кишит наше государство шпиёнами всех мастей и диверсантами. Пишу, сколько их у меня: $\frac{454 \cdot 15 - 26}{5} + 2633$. В десятке тоже не 10 единиц.

Остаюсь твоим заклятым другом, Жихарка».

«Отомстил, — подумал Колобок, — я ведь не знаю, как по-другому-то считать, — как меня научили, я так и считаю».

Сидят над письмами друг друга оба и репу чешут. Делать нечего, надо «налаживать дипломатические контакты». Собрались вместе «на нейтральной территории» (в кабаке «Вдали от жен») и за рюмкой чая выяснили, что поймали-то они равное количество людей, в десятке же одного в два раза больше людей, чем в десятке другого!

А вы, люди добрые, определите, сколько людей поймали да сколько у них в десятке единиц-то было?

Решение.

Решать можно двумя способами.

Способ 1.

Классическим методом подбора получим, что Колобок представил информацию в четырнадцатеричной системе счисления, а Жихарка — в семеричной.

Способ 2.

Применив математику, переведем выражение Колобка из системы счисления с основанием x в десятичную систему счисления:

$$\frac{(x^2 + 6x + 3)(x + 1)}{5} + 3x^2 + 9x + 1.$$

Переведем выражение Жихарки из системы счисления с основанием y в десятичную систему счисления:

$$\frac{(4y^2 + 5y + 4)(y + 5) - 2y - 6}{5} + 2y^3 + 6y^2 + 3y + 3.$$

Умножим оба выражения на 5. Получим:

$$(x^2 + 6x + 3)(x + 1) + (3x^2 + 9x + 1) \cdot 5 = x^3 + 22x^2 + 54x + 8;$$

$$(4y^2 + 5y + 4)(y + 5) - 2y - 6 + (2y^3 + 6y^2 + 3y + 3) \cdot 5 = 14y^3 + 55y^2 + 42y + 29.$$

Очевидно, что $x = 2y$. Приравняв два выражения и подставив в них равенство $x = 2y$, получим:

$$x^3 + 22x^2 + 54x + 8 = 14y^3 + 55y^2 + 42y + 29;$$

$$8y^3 + 88y^2 + 108y + 8 = 14y^3 + 55y^2 + 42y + 29;$$

$$6y^3 - 33y^2 - 66y + 21 = 0;$$

$$2y^3 - 11y^2 - 6y + 7 = 0.$$

Корни находятся среди делителей свободного члена. Проверив 7, получим, что это единственное натуральное число, которое является решением данного уравнения. Тогда получим, что $x = 14$, а $y = 7$. Количество людей: $1564_{10} = 7DA_{14} = 4363_7$.

Задача 10

Вздумалось же советнику по стратегическим вопросам Бабы-яги Елене Устиновой учить Василину (просто Бабы-ёгина помощница) считать в разных системах счисления! Больше всего понравилась Василине двоичная система счисления — меньше всего цифр запоминать надо! Собрала Василина ватагу ребят и заставила рассчитывать.

— Первый!

— Не первый, а один!

— Второй!

— Нельзя «второй», надо «один-ноль»! Понятно?

— Да! Один-один!

— Один-ноль-ноль!

— Один-ноль-один!

— Один-один-ноль!

— Один-один-один!

— Один-ноль-ноль-ноль!

— Один-ноль-ноль-один!

— Один-ноль-один-ноль!

— Один-ноль-один-один!

— Блиииииииииииин! Я запутался... Один-ноль или ноль-один?

— Может, можно попроще что-нибудь?

— Может, и можно! Но мы не ищем легких путей!

— Понятно, — вздохнули ребята и про себя прикинули, что всего их человек 40, так сколь долго они еще «один-нолькать» будут?

— Итак, усложняем задание! Сегодня после вечерней дойки вы все с родителями, братьями, сестрами и всеми своими родственниками приходите на большую поляну! Будем выяснять, кто у нас самый умный и сообразительный!

— А если они не пойдут? — робко пропищал кто-то из девчонок.

— А тогда им отключим газ! — как-то непонятно пошутила Василина. — А вот мы и узнаем, кто у нас самый сообразительный! По домам!

То ли организаторские способности у Василины были ого-го, то ли дети на редкость сообразительными стали, то ли взрослые — любопытными, только вечером на поляне было не продохнуть. Как на весенней ярмарке народу!

Стали они друг друга пересчитывать...

— Мама, он меня сто сорок вторым обозвал!

— А как надо-то, деточка?

— Один четыре два!

— Ах ты пакостник, не можешь, что ли, по-человечески говорить! А то ребенка обидел!

...

— Сам ты два два пять! Мы же в пятеричной системе считаем! Ну и что, что тебе цифра пять нравится!

— Сам дурак! Мы уже в семеричной считаем!

...

— А ты... А ты... А ты... Один ноль ноль ноль ноль ноль ноль ноль ноль ноль ноль...

— А ты... А ты... А ты... Два ноль ноль ноль ноль ноль ноль ноль ноль...

— Сам ты такой!!!

И понеслось! Бабы визжат, дети орут, мужики да парни молодые морды друг другу бьют! Типа, драку заказывали? Получите! Тридцать минут друг друга по поляне гоняли, потом помирились, мировую выпили и довольные отправились по домам. Баба Матрена да тетка Настасья к Василине подошли, сказали, что шибко интересно было да поучительно и чтоб на следующую субботу опять всех собрала, что, мол, они уж придут готовенькие, семечками да орешками заряженные! Мужики пришли, сказали, что в следующий раз придут «со своим», т. е. с палками да досками, чтоб, значит, во время драки из соседских заборов колья не драть. Купцы же обещали по такому поводу ряды торговые соорудить... семечками, орешками да пряниками торговать. Да... Мероприятие удалось... Надо итоги подводить... Села Василина в уголок с бумажкой, кликнула Елену Устинову, и стали они считать да прикидывать, сколько народу-то было.

— Так... В самом начале сколько людей было — не помню, но помню, что, когда считали в четырех последовательных системах счисления, числа получались очень интересные. Если эти числа рассматривать как записанные в десятичной системе счисления, то получатся точные квадраты подряд идущих четырех чисел, одно из которых совпадает с наименьшим основанием систем счисления.

— По такому свойству даже подбором можно найти!

— Я и нашла... Потом еще люди подошли...

— И там какие закономерности выявились?

— Там смешнее всё. Да пошутил тут у нас один. Считали, считали, досчитали до половины, как раз ему эта половина и досталась... Он взял и поменял первую и последнюю цифры местами. Никто не заметил, и про-

должили считать дальше... В результате получили число, большее на 154 в той же системе счисления. А получить должны были 532.

— Ну а дальше?

— А дальше все просто. Прибежали еще люди. Считали мы их, считали и получили следующую зависимость:

$$ROOROO_p$$

$$OSOS_{p+4}$$

$$RRTT_{p+8}$$

— И что?

— Ничего... Считайте, ребята! Должно получиться три числа в десятичной системе счисления. Для первого условия — четыре числа в четырех системах счисления, которые в десятичной системе счисления равны. Для второй задачи надо найти основание системы счисления и три числа в найденной системе счисления. Для третьей — решить математические загадки: надо найти p , R , O , S и T .

Решение.

Первая задача. Необходимо подобрать четыре последовательных системы счисления, наименьшее основание которых будет наибольшим числом, квадрат которого является наибольшим числом, десятичная запись которого является полным квадратом основания системы счисления.

Понятно, что человек было больше 40, так как первоначально только детей было 40, поэтому и квадраты необходимо искать среди чисел, квадраты которых больше 40. Значит, начинаем с 7.

Пусть основание системы счисления — 7. $7^2 = 49$. Не подходит, так как в семеричной системе счисления нет цифры 9.

Пусть основание системы счисления — 8. $8^2 = 64$. Тогда число в десятичной системе счисления $64_8 = 52_{10}$. Не является полным квадратом никакого числа, поэтому наименьшим основанием не может быть 8.

Пусть основание системы счисления — 9. $9^2 = 81$. Переведем в десятичную систему счисления: $81_9 = 73_{10}$. Опять не полный квадрат.

Пусть основание системы счисления — 10. $10^2 = 100$. Переведем в одиннадцатеричную систему счисления. $100_{10} = 91_{11}$. 91 не является полным квадратом никакого числа.

Пусть основание системы счисления — 11. $11^2 = 121$. Переведем в десятичную систему счисления $121_{11} = 144_{10}$. Переведем 144 в двенадцатеричную систему счисления. $144_{10} = 100_{12}$. 100 в десятичной системе счисления является полным квадратом числа 10. Рассмотрим систему счисления по основанию 13. $144 = B1_{13}$. $B1$ в десятичной системе счисления даже не число. Значит, 11 не подходит.

Пусть основание системы счисления — 12. $12^2 = 144$. Переведем в десятичную систему счисления: $144_{12} = 196_{10}$. Переведем в систему счисления по основанию 13: $196_{10} = 121_{13}$. $121 = 11^2$. Рассмотрим систему счисления по основанию 14: $196_{10} = 100_{13}$. Система счисления по основанию 15: $196_{10} = D1_{15}$. Опять не угадали.

Пусть основание системы счисления — 13. $13^2 = 169$. Переводим в десятичную систему счисления $169_{13} = 256_{10}$. Рассмотрим систему счисления по основанию 14: $256_{10} = 144_{14}$, $144 = 12^2$. Рассмотрим систему счисления по основанию 15: $256_{10} = 121_{15}$, $121 = 11^2$. Рассмотрим

систему счисления по основанию 16: $256_{10} = 100_{16}$. И так, результаты $256_{10} = 169_{13} = 144_{14} = 121_{15} = 100_{16}$.

Ответ. 256.

Вторая задача. Можно решать подбором, но лучше составим уравнение, где x — это основание системы счисления:

$$532_x = 5x^2 + 3x + 2.$$

Основная проблема — не зная основания системы счисления, не потеряв значений, поменять местами первую и последнюю цифру числа.

Половина числа:

$$\frac{5}{2}x^2 + \frac{3}{2}x + 1.$$

Проанализируем, что может быть «цифрами», которые будут обменены. Понятно, что из дроби $\frac{5}{2}$ останется целая часть 2, $\frac{1}{2}x^2$ перейдет «в центр числа». С последней цифрой могут быть следующие варианты: либо это 1, либо так же, как и в предыдущем случае, к 1 прибавится $\frac{1}{2}x$. Возьмем более сложный вариант. Составим уравнение с учетом того, что полученное число будет больше исходного на 154_x :

$$\begin{aligned} 5x^2 + 3x + 2 &= \\ = \frac{5}{2}x^2 + \frac{3}{2}x + 1 + \left(\frac{x}{2} + 1\right)x^2 + \left(\frac{x}{2} + 1\right)x + 2 - x^2 - 5x - 4. \end{aligned}$$

Раскроем скобки и приведем подобные:

$$x^3 - 4x^2 - 11x - 6 = 0.$$

Если есть целочисленное решение, то оно является делителем 6. Нам необходимо проверить только 6, так как остальные числа не могут быть основанием системы счисления, для которой можно записать число 532. Поделителем многочлен на $(x - 6)$.

$$x^3 - 4x^2 - 11x - 6 = (x - 6)(x^2 + 2x + 1).$$

Если искомое основание системы счисления 6, то получаем:

$$532_6 = 200.$$

Тогда

$$100 = 244_6,$$

$$532_6 = 244_6 + 442_6 - 154_6.$$

Второй вариант (когда последняя цифра — 1):

$$5x^2 + 3x + 2 = \frac{5}{2}x^2 + \frac{3}{2}x + 1 + x^2 + \frac{x^2}{2} + \frac{3}{2}x + 2 - x^2 - 5x - 4.$$

Раскроем скобки и приведем подобные:

$$3x^2 + 5x + 3 = 0.$$

Данное уравнение корней не имеет.

Ответ. 200.

Третья задача. Решить математический ребус:

ROORO_p

OSOS_{p+4}

RRTT_{p+8}

Решать только методом подбора. Это одно число в разных системах счисления. Очевидно, что R по значению меньше O и O не может равняться 0. Следовательно, p не может быть меньше 3.

Если $p = 3$, то очевидно, что R = 1, O = 2. Тогда $122122_3 = 476_{10}$. Переведем это число в систему счисления по основанию 7: $476_{10} = 1250_7$. Не подходит по схеме.

Пусть $p = 4$. Тогда R = 1, O = 2 или O = 3. Первый вариант:

$$122122_4 = 1690_{10},$$

$$1690_{10} = 3232_8.$$

По схеме не подходит.

Пусть O = 3. Тогда:

$$133133_4 = 2015_{10},$$

$$2015_{10} = 3737_8,$$

$$2015_{10} = 11BB_{12}.$$

Значит, $p = 4$, R = 1, O = 3, S = 7, T = B.

Ответ. 2015.

Окончание следует

НОВОСТИ

Компьютерный генератор игровых сценариев справляется почти как человек

Систему искусственного интеллекта, автоматически генерирующую интерактивные истории для видеоигр, разработали в Технологическом институте Джорджии.

Систему Scheherazade IF обучают по принципу краудсорсинга на любительских продолжениях популярных произведений. На ее основе можно было бы, к примеру,

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

создать игру из серии «Звездные войны», которая генерировала бы бесчисленные варианты развития сюжета. Проводилось сравнение со сценариями, написанными человеком, и сгенерированными случайным образом. Искусственный интеллект выдавал в среднем по три ошибки на сценарий, как и люди, а случайный генератор — по 12,5 ошибки.



О. В. Александрова,
дипломант конкурса «Изучаем информатику с героями любимых книг»,
детский сад № 50, г. Североморск, Мурманская область

ИНТЕГРИРОВАННОЕ ЗАНЯТИЕ-ВИКТОРИНА «УВЛЕКАТЕЛЬНОЕ ПУТЕШЕСТВИЕ В ВОЛШЕБНЫЙ МИР СКАЗОК»*

Аннотация

В статье описан опыт использования компьютерных интерактивных дидактических игр на занятиях с детьми старшего дошкольного возраста в рамках компьютерного кружка.

Ключевые слова: информационные технологии, дошкольная педагогика, компьютерные дидактические игры, интерактивная доска, нарушения речи.

Контактная информация

Александрова Оксана Васильевна,
воспитатель детского сада № 50, г. Североморск, Мурманская область; *адрес:* 184606, Мурманская область, г. Североморск, ул. Чабаненко, д. 7а; *телефон:* (815-37) 5-02-24; *e-mail:* kcana07@mail.ru

O. V. Aleksandrova,
Kindergarten 50, Severomorsk, Murmansk Region

INTEGRATED LESSON-QUIZ "AN EXCITING JOURNEY INTO THE MAGIC WORLD OF FAIRY TALES"

Abstract

The article describes the experience of using computer interactive didactic games with children of preschool age in the computer club.

Keywords: information technologies, preschool pedagogy, computer didactic games, interactive whiteboard, speech disorders.

Тема занятия: Увлекательное путешествие в волшебный мир сказок.

Цели занятия:

- способствовать возникновению у детей интереса к русским народным сказкам с помощью интерактивной дидактической игры;
- учить расставлять события в правильной последовательности.

Задачи занятия:

образовательные:

- обобщить и закрепить знания детей о русских народных сказках, их персонажах;
- упражнять детей узнавать сказку или сказочный предмет по заданию;
- упражнять детей в подборе рифм к персонажам сказок;
- закреплять умение расставлять сказочные события в правильной последовательности;

развивающие:

- развивать мышление, речь, внимание;
- развивать любознательность и читательскую память;
- развивать творческое воображение и фантазию;

воспитательные:

- воспитывать у детей интерес и любовь к русским народным сказкам, к чтению;
- создать у детей эмоционально приподнятое настроение, увлечь их интерактивной компьютерной игрой;

оздоровительные:

- снятие зрительного напряжения (гимнастика для глаз);
- снятие мышечного напряжения (физкультминутки).

Интеграция образовательных областей:

- «Познавательное развитие (информатика)»;
- «Речевое развитие»;
- «Ознакомление с художественной литературой».

Методы обучения: игровой, словесно-логический, использование ИКТ, самостоятельный.

Приемы обучения: дидактическая игра-викторина, художественное слово (загадки, стихи), пояснение, поощрение, пальчиковая гимнастика, гимнастика для глаз, физкультминутка, самостоятельная деятельность детей.

Формы работы на занятии: индивидуальная, коллективная.

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_09-2015/

Оснащение занятия:*оборудование:*

- ноутбук воспитателя;
- мультимедийный проектор;
- интерактивная доска;
- телескопическая указка;
- компьютеры для воспитанников;

программное обеспечение:

- операционная система Windows 8;
- программа презентаций Microsoft PowerPoint 2013;
- графический редактор Paint.net;
- Macromedia Flash MX;
- интерактивная компьютерная игра «Увлекательное путешествие в волшебный мир сказок» (авторская разработка, созданная в Microsoft PowerPoint 2013);
- Flash-игра «Колобок» (авторская разработка, созданная в Macromedia Flash MX);

материалы для занятия:

- ростовая кукла Сорока;
- поощрительные фишки — монетки;
- тренажеры для зрительной гимнастики;
- карточки-смайлики для этапа рефлексии.

Литературные и интернет-источники, использованные при подготовке занятия.

1. Габдуллина З. М. Развитие навыков работы с компьютером у детей 4–7 лет. Волгоград, 2010.
2. Наглядно-дидактические пособия (рассказы по картинкам) «Колобок». М.: Мозаика-синтез, 2010.
3. Наглядно-дидактические пособия (рассказы по картинкам) «Теремок». М.: Мозаика-синтез, 2010.
4. Сказки (моя любимая книга). Тула: Родничок; М.: АСТ. Астрель, 2008.
5. <http://allforchildren.ru/kidfun/schitalki01.php> — считалка.
6. <http://www.stihi.ru/2012/06/02/4622> — стихотворение про Незнайку.

План занятия.

1. Организационный этап.
2. Повторение правил техники безопасности при работе в компьютерном кабинете.
3. Интерактивная игра-викторина.
4. Зрительная гимнастика — «Отдыхалочка».
5. Динамическая пауза.
6. Практическая работа на компьютере.
7. Зрительная гимнастика.
8. Подведение итогов занятия. Рефлексия.

Ход занятия**1. Организационный этап**

Центр компьютерной грамоты украшен разноцветными шарами. Звучит веселая детская музыка, дети заходят в кабинет, встают в круг.

Воспитатель.

На занятия хожу
(дети держатся за руки и размахивают ими —
в круг, из круга)
И с компьютером дружу.
(пожимают ладони)

Здесь играю
(изображают движениями пальцев работу
на клавиатуре)
И рисую —
(рисуют пальцем в воздухе)
Никогда не устаю.
(размахивают указательным пальцем
вправо-влево)

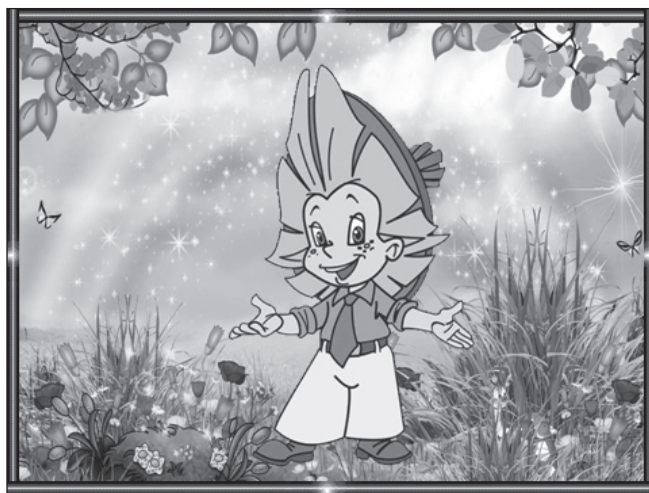
Дети садятся на стульчики.

Воспитатель. Ребята, к нам на занятие кто-то спешит. А чтобы узнать, кто это, отгадайте загадку:

У кого большая шляпа?
Кто бездельник и растяпа?
Кто хвостун, болтун, зазнайка?
Знают все: это — Незнайка.

(Стихотворение Веры Аношиной.)

На интерактивной доске демонстрируется слайд с изображением Незнайки.

**2. Повторение правил техники безопасности при работе в компьютерном кабинете**

Воспитатель. Ребята, давайте вспомним и расскажем Незнайке правила поведения и техники безопасности в компьютерном кабинете.

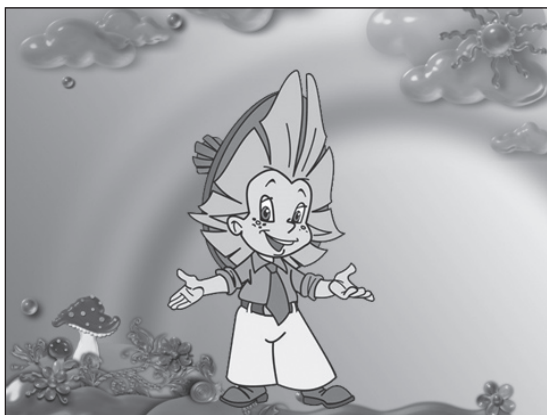
Дети по очереди называют правила:

- Не включать без разрешения компьютер.
- Не трогать провода и розетки руками.
- Не трогать руками части компьютера, не относящиеся к содержанию занятия, и не нажимать неизвестные клавиши.
- Не сидеть слишком близко к экрану монитора — расстояние должно быть не менее длины вытянутой руки. Спина должна быть прямая, ровная.
- Не смотреть в линзу проектора.

3. Интерактивная игра-викторина

Все игры викторины демонстрируются на слайдах мультимедийной презентации, подготовленной в Microsoft PowerPoint 2013.

Незнайка здоровается и предлагает детям отправиться в увлекательное путешествие в волшебный мир сказок:



Открывается карта игры:



Все картинки на карте игры являются гиперссылками на другие отдельные игры.

Незнайка предлагает детям выбрать задание и нажать на картинку.

За каждый правильный ответ в игре воспитатель выдает ребенку поощрительную фишку-монетку.

Воспитатель берет ростовую куклу Сороку и выбирает водящего считалкой:

Раз, два, три, четыре, пять,
Мы собрались поиграть.
К нам сорока прилетела
И тебе водить велела.

Ребенок подходит и выбирает игру указкой — «волшебной палочкой».

Игра «Кто за кем появлялся в сказке “Репка”?».

Если ребенок нажмет на репку, то откроется игра «Кто за кем появлялся в сказке “Репка”?»



Цели игры:

- ребята должны вспомнить содержание сказки и последовательность выхода героев;
- учить ориентироваться в пространстве;
- развивать наглядное мышление;
- знакомить детей с предлогами: за, перед, до, после, между;
- развивать элементарные математические представления: сначала, потом, первый, последний.

Дети наблюдают рост репки, а затем вспоминают последовательность выхода персонажей сказки.

Игра «Рифма».

Если ребенок нажмет на колобка, то откроется игра «Рифма».



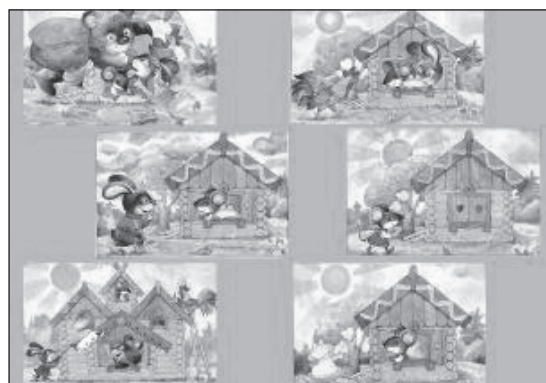
Цели игры:

- знакомство детей с рифмой;
- развитие у детей чувства ритма.

Колобок катится по тропинке, а навстречу ему — сказочный сундук, который произносит такие слова: «Вспомнишь сказочные выражения — пропущу». Воспитатель просит ребят помочь ему справиться с этим заданием.

Игра «Теремок».

Если ребенок нажмет на домик, то откроется игра «Теремок».

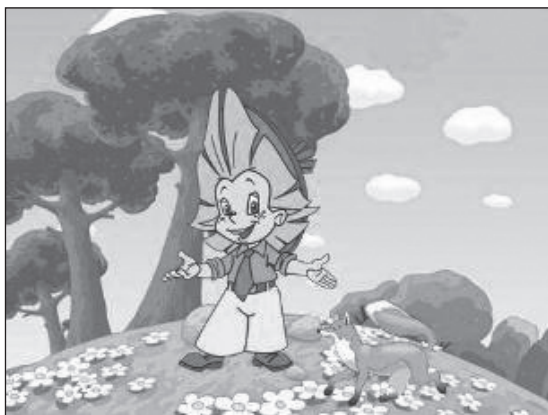


Цель игры: учить расставлять события в правильной последовательности.

Дети «волшебной палочкой» или пальчиком расставляют картинки в правильной последовательности.

Игра «Назовите сказки, в которых главным героем была лиса».

Если ребенок нажмет на лисичку, то откроется игра «Назовите сказки, в которых главным героем была лиса».

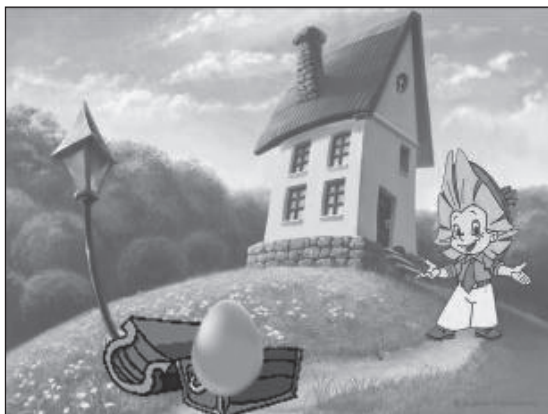


Цель игры: учить детей вспоминать названия сказок по художественным иллюстрациям, в которых главным героем была лиса.

На экране всплывают иллюстрации из книг, дети должны вспомнить название сказки.

Игра «Из какой сказки предмет?»

Если ребенок нажмет на сундук, то откроется игра «Из какой сказки предмет?»



Цели игры:

- учить детей отгадывать название сказки по сказочному предмету;
- развивать связную речь, внимание, мышление.

Из сундука появляются сказочные предметы. Дети должны вспомнить, из какой сказки предмет.

4. Зрительная гимнастика — «Отдыхалочка»

Зрительная гимнастика проводится после того, как у доски побывают два ребенка; после еще двух детей проводится динамическая пауза.

Воспитатель.
 Наши глазки так устали,
 Мы дадим им отдохнуть,
 Их закроем на чуть-чуть.
 А теперь их открываем,
 Быстро-быстро поморгаем.

5. Динамическая пауза

Воспитатель.
 Наш Незнайка потянулся,
 Раз — нагнулся, разогнулся.

Два — в ладоши три хлопка,
 Головою три кивка,
 На четыре — ноги шире.
 Пять — на месте пошагать.
 Шесть — на стульчик тихо сесть.

Затем к доске выходят по очереди оставшиеся дети. Игра продолжается.

В конце игры Незнайка прощается с ребятами.

6. Практическая работа на компьютере

Воспитатель. Еще раз вспомним, как нужно правильно работать за компьютером:

- пододвинуть стул к компьютеру;
- не сидеть слишком близко к экрану монитора — расстояние должно быть не менее длины вытянутой руки;
- спина у вас должна быть прямая, ровная:

Быть сутулой не должна
 За компьютером спина.

Задание 1. Флеш-игра по сказке «Колобок». Расставьте картинки в правильной последовательности.

Воспитатель включает детям игру по сказке «Колобок». Каждый ребенок должен расставить картинки по сказке в правильной последовательности.

Задание 2. Воспитатель предлагает детям нарисовать сказочный предмет в графическом редакторе Paint.

7. Зрительная гимнастика

«Бабочка». Воспитатель с помощью тренажера «бабочка» проводит зрительную гимнастику. Дети должны следить глазами за передвижениями бабочки в воздухе.

«Раскрасим мир вокруг». Воспитатель предлагает детям выбрать предмет в окружающем пространстве. Дети держат голову неподвижно и «раскрашивают» взглядом предмет, который находится в кабинете.

8. Подведение итогов занятия. Рефлексия

Воспитатель подсчитывает количество монеток и определяет победителя.

Всем детям вручаются медальки.

Дети вместе с воспитателем встают в круг и произносят слова:

Мы за руки возьмемся,
 Возьмемся, возьмемся,
 Друг другу улыбнемся,
 Улыбнемся, улыбнемся.
 По кругу мы пойдем,
 Итог занятия подведем.

Воспитатель. Ребята, чему вы научились на занятии? Что вам больше всего понравилось на занятии?

Ребята, Незнайка прислал нам волшебную шкатулку. Давайте посмотрим, что в ней.

Дети. Смайлики.

Воспитатель. Чем они отличаются?

Дети. Настроением — одни выражают радость, другие — грусть.

Воспитатель. Если вам сегодня понравилось играть с Незнайкой, возьмите радостный смайлик, если не понравилась игра — возьмите грустный смайлик.



В. Д. Кильдишов,

Западно-Подмосковный институт туризма (филиал Российской международной академии туризма)

ИСПОЛЬЗОВАНИЕ MS EXCEL ДЛЯ ИЗУЧЕНИЯ ОСНОВ СТЕГАНОГРАФИИ

Аннотация

В статье представлены модели скрытой передачи данных с использованием табулированных значений функций и цветов заливки ячеек MS Excel.

Ключевые слова: стеганография, передача данных, MS Excel.

Контактная информация

Кильдишов Вячеслав Дмитриевич,
канд. тех. наук, доцент, профессор Российской международной академии туризма, профессор кафедры управления персоналом Западно-Подмосковного института туризма; адрес: 143050, Московская область, Одинцовский р-н, р.п. Большие Вяземы, ул. Институт, вл. 5, стр. 1; телефон: (495) 598-26-98; e-mail: kildishov47@rambler.ru

V. D. Kildishov,

Western-Moscow Institute of Tourism (branch of Russian International Academy for Tourism)

USING MS EXCEL FOR STUDYING THE BASICS OF STEGANOGRAPHY

Abstract

The article describes the models of hidden data transmission using tabulated values of functions and colors of filling of the MS Excel cells.

Keywords: steganography, transmission of data, MS Excel.

Для скрытия факта передачи сообщений в стеганографии используются так называемые контейнеры: графические и звуковые файлы, тексты и т. д. Запись сообщений производят в последние разряды кодов графических и звуковых файлов с помощью специальных программ.

Рассмотрим формирование сообщений в контейнере табулированных значений функции и в контейнере цветовой мозаики, созданной на основе цветового формата заливки ячеек MS Excel. Считаем, что у адресата имеются эталонные контейнеры для извлечения сообщений.

Используем в качестве контейнера табулированные значения функции [1]. Функцию можно взять любую, в нашем примере мы остановимся на стандартной функции $\cos x$, табулированные значения которой занесем в массив (C4:R4) с заданным шагом (C1) и начальным значением аргумента $x = 0$ (рис. 1).

Длина таблицы определяется возможным количеством передаваемых символов в сообщении. В нашем случае нужно передать слово «Стеганография».

Под каждым значением табулированной функции заносим в ячейки по букве (символу) из слова «Стеганография» (C9:O9), начиная с первого значения таблицы. Далее определяем коды символов (C11:O11) с использованием функции КОДСИМВ() (рис. 2).

Для передачи сообщения будем использовать три последних разряда значений табулированной функции. Делим значения кодов символов на нормирующий коэффициент К (находится в ячейке F1) и производим сложение нормированных кодов букв слова «Стеганография» (находятся в ячейках C13:O13) с соответствующими табулированными значениями функции, суммы помещаем в ячейки C15:O15. Теперь можно передавать файл со скрытым сообщением в ячейках C15:O15.

После приема файла производится чтение сообщения. Как уже указывалось, у адресата должна быть исходная таблица табулированной функции или данные для ее генерации (вид функции, начальное значение аргумента и шаг изменения аргумента).

При наличии таблицы значений функции вычисляем значения нормированных кодов. Для этого из принятых значений табулированной функции вычитаем ее истинные значения (C22:O22) (рис. 3).

Затем полученные значения умножаем на нормирующий коэффициент К и получаем значения кодов символов сообщения (C24:O24). Применяя функцию СИМВОЛ(), прочитываем принятое сообщение (C26:O26).

С целью исследования предлагаемого метода стеганографии можно построить графики табулированной функции и таблицы с сообщением. Выделяем диапазоны C3:O4 и C15:O15 и обращаемся к диаграмме *Точечная с гладкими кривыми и маркерами*. В нашем случае на графике две кривые будут совпадать. То есть факт передачи сообщения практически нельзя обнаружить. Если изменить нормирую-

1	ΔX	0,2		К	1000000													
2																		
3	X	0	0,2	0,4	0,6	0,8	1	1,2	1,4	1,6	1,8	2	2,2	2,4	2,6	2,8		
4	COS(X)	1,000000	0,980067	0,921061	0,825336	0,696707	0,540302	0,362358	0,169967	-0,029200	-0,227202	-0,416147	-0,588501	-0,737394	-0,856889	-0,942222	-0,989	

Рис. 1

7	Передающая сторона																	
8																		
9	Сообщение	С	т	е	г	а	н	о	г	р	а	ф	и	я				
10			=КОДСИМВ(C9)															
11	Код символа	209	242	229	227	224	237	238	227	240	224	244	232	255				
12						=F11/\$F\$1												
13	Нормированный код	0,000209	0,000242	0,000229	0,000227	0,000224	0,000237	0,000238	0,000227	0,000240	0,000224	0,000244	0,000232	0,000255				
14			=C4+C13															
15	Письмо отправили	1,000209	0,980309	0,921290	0,825563	0,696931	0,540539	0,362596	0,170194	-0,028960	-0,226978	-0,415903	-0,588269	-0,737139				

Рис. 2

ший коэффициент так, чтобы коды символов совпадали со старшим разрядом после запятой табулированных значений (K = 1000), то увидим несовпадение графиков. Изменяя значение коэффициента, можно оценить возможности по скрытию сообщения.

Кроме значений табулированной функции в качестве контейнера можно использовать коды цветовой заливки ячеек, которые образуют мозаику. Количество ячеек в мозаике должно определяться объемом сообщения.

Рассмотрим возможность передачи слова «страна» с помощью кодов цветов заливки ячеек. Создадим контейнер из шести ячеек (B5:G5) (рис. 4). Через контекстное меню выходим на пункт *Формат ячеек* и производим заливку различными цветами. Затем над ячейками записываем значения кодов основных цветов заливки, которые соответствуют цветовой модели RGB (B2:G4).

Для этого через контекстное меню каждой ячейки выходим на *Другие цвета* и в окне *Цвета* — на закладку *Спектр*, где переписываем значения кодов основных цветов модели. Коды основных цветов модели RGB изменяются от 0 до 255. Поэтому для передачи сообщения выбираем основной цвет, значения кодов которого обеспечивают передачу всех букв нашего алфавита. В нашем случае это зеленый цвет.

Записываем в ячейки B7:G7 слово для передачи и определяем коды символов с помощью функции КОДСИМВ(). Нормируем коды символов путем вычитания из них значения 224, которое обеспечивает символу «а» нормированное значение, равное 1.

Осталось упаковать значения нормированных кодов символов в контейнер путем прибавления их к коду выбранного основного цвета заливки. Через закладку *Разработчик* и группу *Элементы управления* создаем

20	Приемная сторона																	
21																		
22	1 этап распаковки	0,000209	0,000242	0,000229	0,000227	0,000224	0,000237	0,000238	0,000227	0,000240	0,000224	0,000244	0,000232	0,000255				
23			=C22*\$F\$1															
24	2 этап распаковки	209	242	229	227	224	237	238	227	240	224	244	232	255				
25						=СИМВОЛ(F24)												
26	Письмо прочли	С	т	е	г	а	н	о	г	р	а	ф	и	я				
27																		

Рис. 3

1	A	B	C	D	E	F	G	H	I	J	
2		R	83	225	250	0	255	228	Очистка данных		
3		G	142	217	192	176	153	109			
4		B	213	185	144	80	255	10			
5	Контейнер								Пакуем слово в контейнер		
6		=КОДСИМВ(B7)			=B8-224						
7	Слово для передачи	с	т	р	а	н	а				
8	Код буквы	241	242	240	224	237	224		Распаковываем контейнер		
9	Нормированный код	17	18	16	0	13	0				
10	Контейнер со словом										
11		R	83	225	250	0	255	228			
12		G	159	235	208	176	166	109			
13		B	213	185	144	80	255	10			
14	Расшифровка слова	с	т	р	а	н	а				

Рис. 4

Кнопку под названием «Пакуем слово в контейнер» и записываем программу упаковки:

```
Private Sub CommandButton1_Click() 'Пакуем слово
в контейнер
For i=0 To 5 'Формируем цвет фона ячеек контей-
нера
Cells(10, 2+i).Interior.Color=rgb(Cells(2,
2+i), Cells(3, 2+i)+Cells(9, 2+i), Cells(4, 2+i))
Next i
End Sub.
```

Формируем кнопку в режиме *Конструктор*. Как видно, к кодам основного зеленого цвета прибавляем нормированные значения кодов символов переданного слова: $Cells(9, 2 + i)$. Процедура `Interior.Color` обеспечивает заливку ячеек B10:G10, в которых скрыто сообщение. Эти ячейки с «мозаикой» и предназначены для передачи.

После приема сообщения его нужно распаковать с помощью кнопки «Распаковываем контейнер». Программа кнопки определяет коды основных цветов заливки, выделяет коды символов сообщения и записывает символы принятого сообщения:

```
Private Sub CommandButton2_Click() 'Распаковыва-
ем контейнер со словом
For i=0 To 5
Cells(11, 2+i)=(Cells(10, 2+i).Interior.Color)
Mod 256 'Код красного цвета
Cells(12, 2+i)=((Cells(10, 2+i).Interior.
Color)\256) Mod 256 'Код зеленого цвета
Cells(13, 2+i)=(Cells(10, 2+i).Interior.
Color)\256^2 'Код синего цвета
Cells(14, 2+i)=Chr(Cells(12, 2+i)-Cells(3,
2+i)+224) 'Код буквы слова
Next i
End Sub.
```

Код цвета K_{RGB} модели RGB определяется выраже-
нием:

$$K_{RGB} = K_R + K_G \cdot 256 + K_B \cdot 256^2,$$

где K_R, K_G, K_B — коды основных цветов модели. Таким образом, зная код K_{RGB} , просто вычисляем коды основных цветов. Используя код зеленого цвета принятой «мозаики» и имеющегося образца, определяем код символа, по которому с помощью функции `Chr()` записываем принятое слово в ячейки B14:G14, где и прочитываем.

Программа кнопки «Очистка данных» обеспечивает очистку области ячеек, формируемых для передачи и прочтения сообщения:

```
Private Sub CommandButton3_Click() 'Очистка дан-
ных
Range(Cells(10, 2), Cells(14, 7)).ClearContents
'Очистка содержимого ячеек
Range(Cells(10, 2), Cells(10, 7)).ClearFormats
'Очистка формата контейнера со словом
End Sub.
```

Данную модель можно использовать для передачи скрытого сообщения, состоящего из шести букв. При этом целесообразно сначала щелкнуть на кнопке «Очистка данных», затем записать слово для передачи, щелкнуть на кнопке «Пакуем слово в контейнер» и на кнопке «Распаковываем контейнер».

Представленные модели скрытой передачи сообщений можно модифицировать. Например, использовать «полный» код цвета заливки, а не код основных цветов.

Литература

1. Кильдишев В. Д. Использование приложения MS Excel для моделирования различных задач. М.: СОЛОН-Пресс, 2015.

НОВОСТИ

Компьютер научили различать слова и звуки любых языков

Систему, способную обучиться различать в потоке речи слова, слоги и звуки, создали в Массачусетском технологическом институте. В отличие от традиционных систем распознавания, она действует по принципу самообучения, руководствуясь лишь стандартными свойствами языков, например знанием о том, что большинство слов в языке встречается относительно редко, а меньшинство — очень часто. Система полагается на теоремы Шеннона, рассматривая аудиосигнал как

последовательность идеально произнесенных звуков, отправленную по каналу с шумами. Задача системы — выучить статистические корреляции между «полученным на выходе» звуком и соответствующей фонемой. Авторы уверены, что их работа поможет в придании системам распознавания речи способности действовать независимо от индивидуальных особенностей произношения. Наряду с этим работа может помочь разобраться в том, как человек обучается говорить.

IDC: настольные ПК остаются главным инструментом офисных сотрудников

Опрос 600 западноевропейских ИТ-руководителей, проведенный IDC, показывает, что, несмотря на рост популярности портативных устройств, в корпоративной среде настольные ПК остаются главным рабочим инструментом. 84 % опрошенных заявили, что готовы рассмотреть возможность покупки таких компьютеров для своей компании. Самыми важными преимуществами традиционных моделей опрошенные считают

производительность, надежность и более долгий срок службы, а также дешевизну. ПК служат в среднем на год дольше ноутбуков, отмечают аналитики. Устройства малого формфактора становятся популярнее, так как компании все чаще стремятся экономить энергию и место на столах сотрудников. Такие устройства планируют использовать также в цифровых вывесках и интернет-киосках.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

В. Ф. Очков, Е. П. Богомолова,
Национальный исследовательский университет «МЭИ», Москва

НЕМНОГО О РЕКУРСИИ, ЧИСЛОВЫХ РЯДАХ И ФРАКТАЛАХ

Аннотация

В статье рассмотрены проблемы использования рекурсии для генерации рядов целых чисел и их анализа с помощью Интернета, а также для решения головоломок.

Ключевые слова: целочисленный ряд, рекурсия, факториал, числа Фибоначчи, головоломка «Ханойские башни».

Контактная информация

Очков Валерий Федорович, доктор тех. наук, профессор, Национальный исследовательский университет «МЭИ», Москва; адрес: 111250, г. Москва, Красноказарменная ул., д. 14; телефон: (495) 362-71-71; e-mail: ochkov@twm.mpei.ac.ru

V. F. Ochkov, E. P. Bogomolova,
National Research University MPEI, Moscow

A LITTLE ABOUT RECURSION, NUMBER SERIES AND FRACTALS

Abstract

The article deals with the problem of using recursion for generating series of integers and for their analysis by the tools of Internet, as well as using recursion to solve puzzles.

Keywords: integers, recursion, factorial, Fibonacci numbers, puzzle "Towers of Hanoi".

Германн сошел с ума. Он сидит в Обуховской больнице в 17-м номере, не отвечает ни на какие вопросы и бормочет необыкновенно скоро: «Тройка, семерка, туз! Тройка, семерка, дама!..»

А. С. Пушкин, «Пиковая дама»

Почти все знают, что такое факториал числа. Это произведение всех натуральных чисел от 1 до n включительно. Для факториала введено обозначение $n!$. Так, например:

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120.$$

Такое классическое определение факториала не охватывает нуль. Для удобства общей записи формул, в которых $0!$ обычно входит в виде множителя, математики договорились считать $0! = 1$.

Замечательно то, что факториал нуля легко подсчитать (не определить, а именно подсчитать) с помощью рекурсии (от *лат.* *recursio* — возвращение). На рисунке 1 помещена программа-функция Mathcad Prime для подсчета факториала с использованием рекурсии.

Рекурсивная функция — это функция, которая при вычислении вызывает сама себя. Она обычно зависит от параметра, при изменении которого происходит рекурсивный спуск или подъем.

Вот как определяет рекурсию один юмористический справочник по информатике: «Рекурсия — см. рекурсия».

Проще всего проиллюстрировать рекурсию геометрически: многократно (даже бесконечное число раз) повторить какие-то геометрические элементы (возможно, с изменением масштаба), т. е. получить самоподобные геометрические объекты. Классическим примером бесконечной рекурсии являются изображения в двух поставленных друг напротив друга зеркалах: в них образуются два коридора из уменьшающихся и заступающих отражений зеркал.

Если вычислительный алгоритм основан на принципе разбиения основной задачи на подзадачи, каждая из которых повторяет основную, то мы получаем **рекурсивный алгоритм**. Такой алгоритм решает задачу путем ее сведения к решению одной или нескольких таких же задач, но в более простом варианте*. Пример такого алгоритма представлен на рисунке 1. Вычисление факториала начинается со значения $n = 5$, а затем идет или рекурсивный подъем ($n > 5$), или рекурсивный спуск ($n < 5$). Заметим, что можно было бы начать с любого натурального числа n , чей факториал нам известен. Например, взять $n = 10$ и $10! = 3\,628\,800$ или взять $n = 1$ и $n! = 1$ и использовать только рекурсивный подъем.

$\text{Factorial}(n) := \begin{cases} \text{if } n = 5 & \text{return } 120 \\ \text{if } n > 5 & \text{return } (\text{Factorial}(n-1) \cdot n) \\ \text{if } n < 5 & \text{return } \frac{\text{Factorial}(n+1)}{n+1} \end{cases}$	
$\text{Factorial}(4) = 24$	$\text{Factorial}(6) = 720$
$\text{Factorial}(5) = 120$	$\text{Factorial}(0) = 1$

Рис. 1. Рекурсивная функция «Факториал»

* Как вскипятить воду в чайнике? Нужно налить в него воды и поставить чайник на плиту. А если в чайнике уже есть вода? Нужно ее вылить и... свести задачу к предыдущей ;-).

Словесное описание используемой функции *Factorial* такое:

- если n равно 5, то $n!$ равен 120 (см. две первые строки программы на рисунке 1);
- если n больше 5, то факториал такого целого числа равен $(n - 1)! \cdot n$. Пример: $6! = 5! \cdot 6 = 720$;
- если n меньше 5, то $n!$ равен $(n + 1)! / (n + 1)$. Примеры: $4! = 5! / 5 = 24$; $3! = 4! / 4 = 6$; $2! = 3! / 3 = 2$; $1! = 2! / 2 = 1$ и, наконец, $0! = 1! / 1 = 1$, что и требовалось получить.

При рекурсии очень легко математически посчитать размерность геометрического объекта. Математики заметили, что некоторые рекурсивные объекты имеют дробную размерность. Такие объекты назвали **фракталами**. Вспомним, что точка имеет размерность 0, линии

(кривые и прямые) — 1 («длину»); плоскость и поверхности — 2 («площадь»); трехмерное пространство и тела — 3 («объем»). Чтобы представить себе геометрический объект дробной размерности, достаточно найти в Интернете любой сайт с изображениями фракталов. На авторском сайте: <http://twf.mpei.ac.ru/MCS/Worksheets/Levy-curve.htm> можно поиграть с так называемой кривой Леви — с фракталом, предложенным французским математиком П. Леви. Этот фрактал получается, если взять отрезок прямой, переломить его пополам, затем новые отрезки переломить пополам и т. д. Повторяя эту операцию бесконечное число раз, мы получим кривую Леви, у которой дробная размерность (рис. 2).

В Интернете есть очень интересный сайт — «Открытая энциклопедия целочисленных последователь-

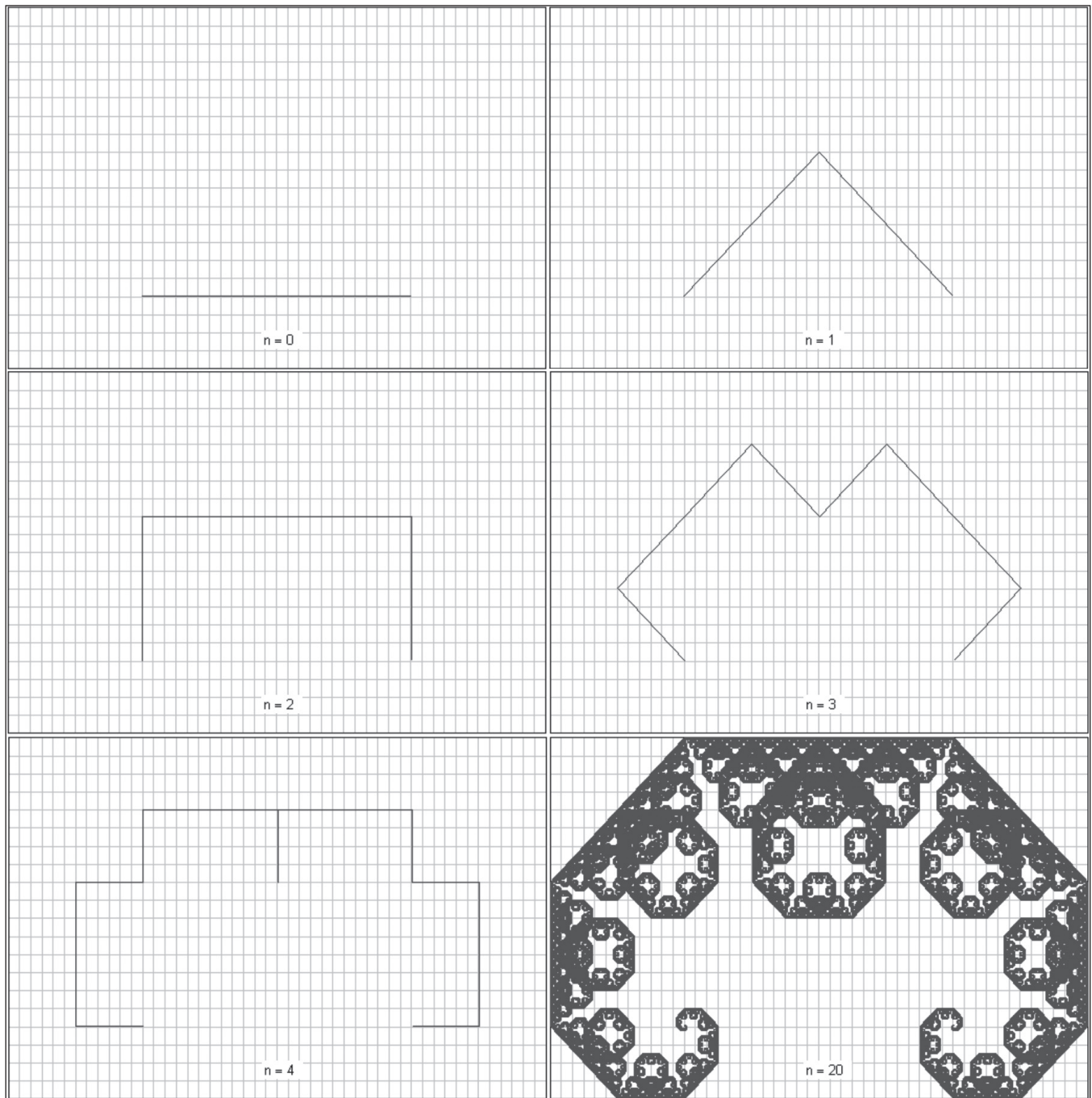


Рис. 2. Фрактал Леви

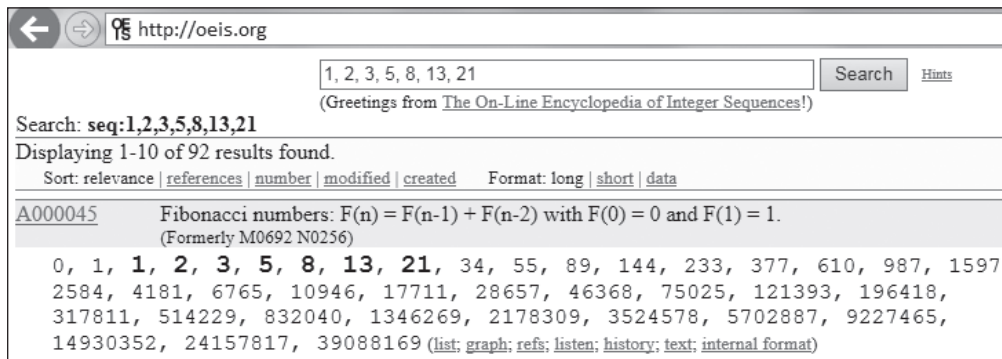


Рис. 3. Сайт определения закономерностей в целочисленных последовательностях

ностей»: <http://oeis.org>. Если в окошко этого сайта ввести последовательность чисел 1, 1, 2, 6, 24, 120, 720 (см. выше) и нажать кнопку Search (Искать), то сайт выдаст ответ, что это факториал, и даст ссылки на интересные материалы об этом математическом операторе.

Очень забавно вводить на этом сайте различные последовательности целых чисел и узнавать, какой закономерности они подчиняются. Так, например, если ввести числа 1, 2, 3, 5, 8, 13, 21, то будет получен ответ, что это числа Фибоначчи — числовой ряд, где каждое очередное число — это сумма двух предыдущих чисел (рис. 3).

Интересно найти закономерность той или иной числовой последовательности самому или с помощью сайта, показанного на рисунке 3. Но еще интересней «утереть нос» этому сайту: ввести числовую последовательность с известной закономерностью и увидеть, что сайт дал слабинку и не нашел ответа. После этого можно будет зарегистрироваться на этом сайте и ввести в него новую информацию. Попробуем это сделать.

Что такое числа Фибоначчи (см. выше), знают многие. А вот что такое **прекрасные числа Фибоначчи**, мало кто знает. Это числа 1, 3 и 7. Если их сложить, то получится еще одно «симпатичное» число — 11. Вспомним: «Тройка, семерка, туз (11)!» На рисунке 4 показана рекурсивная функция, возвращающая прекрасные числа Фибоначчи не только при положительном, но и при отрицательном аргументе. В этой более полной последовательности чисел появилась и пятерка, которой также нельзя отказать в «изящности», если вспомнить школьные и институтские оценки знаний.

Так вот, если в окошко упомянутого сайта (рис. 3) ввести числа 1, 3, 7, 11, 21, 39, то он «раскусит» эту последовательность, отметив, что каждое очередное число — это сумма трех (а не двух — см. рис. 3) предыдущих. Но если этому сайту «оказать медвежью услугу» — расширить ряд до $-1, 3, 1, 3, 7, 11, 21, 39$ (приписать в начале последовательности чисел минус 1 и тройку), то ответа не последует — сайт можно ввести в ступор не только недостатком информации, но и ее избытком.

<pre> Fibonacci(n) ::= if n = 1 return 3 if n = 2 return 7 if n = 3 return 11 if n > 3 return (Fibonacci(n-3) + Fibonacci(n-2) + Fibonacci(n-1)) if n < 1 return (Fibonacci(n+3) - Fibonacci(n+2) - Fibonacci(n+1)) </pre>																																
$n := -7..7$	$n =$	$Fibonacci(n) =$																														
	<table border="1"> <tr><td>-7</td></tr> <tr><td>-6</td></tr> <tr><td>-5</td></tr> <tr><td>-4</td></tr> <tr><td>-3</td></tr> <tr><td>-2</td></tr> <tr><td>-1</td></tr> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> <tr><td>6</td></tr> <tr><td>7</td></tr> </table>	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	<table border="1"> <tr><td>11</td></tr> <tr><td>-1</td></tr> <tr><td>-5</td></tr> <tr><td>5</td></tr> <tr><td>-1</td></tr> <tr><td>-1</td></tr> <tr><td>3</td></tr> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>7</td></tr> <tr><td>11</td></tr> <tr><td>21</td></tr> <tr><td>39</td></tr> <tr><td>71</td></tr> <tr><td>131</td></tr> </table>	11	-1	-5	5	-1	-1	3	1	3	7	11	21	39	71	131
-7																																
-6																																
-5																																
-4																																
-3																																
-2																																
-1																																
0																																
1																																
2																																
3																																
4																																
5																																
6																																
7																																
11																																
-1																																
-5																																
5																																
-1																																
-1																																
3																																
1																																
3																																
7																																
11																																
21																																
39																																
71																																
131																																

Рис. 4. Прекрасные числа Фибоначчи

```

n := 11      p := 2^n - 1 = 2047      V_0 := "begin"      V_{p+1} := "end"

HT(n, V, x, y, z) := || p ← 2^n - 1
                    || i ← 1
                    || if n = 1
                    ||   || while V_i ≠ 0
                    ||   ||   || i ← i + 1
                    ||   ||   || V_i ← concat(x, z)
                    ||   || else
                    ||   ||   || V ← HT(n - 1, V, x, z, y)
                    ||   ||   || while V_i ≠ 0
                    ||   ||   ||   || i ← i + 1
                    ||   ||   ||   || V_i ← concat(x, z)
                    ||   ||   ||   || V ← HT(n - 1, V, y, x, z)
                    ||   ||   || V
                    || V

HT(n, V, "A", "B", "C")^T = [ "begin" "AC" "AB" "CB" "AC" "BA" ... ]

```

Рис. 5. Ханойские башни с рекурсией

Предлагаем читателям «поиграть» с сайтом oeis.org, вводя в него числовые последовательности, созданные рекурсивным программированием (см. рис. 1 и 4) или с опорой на свою собственную смекалку. Сайт oeis.org, к примеру, знает, что 1, 2, 5, 10, 20, 50 и 100 — это номиналы банкнот США, а 5, 10, 20, 50, 100, 200, 500 — это банкноты евро, и прогнозирует будущие банкноты в случае гиперинфляции. Но этот сайт не знает, что 1, 3, 5, 10, 25, 50, 100 — это номиналы банкнот бывшего СССР*. Неизвестно сайту также и то, что 7, 9, 11, 13, 15, 19, 22, 28, 48 — это стоимость (в копейках) мороженого в том же бывшем Советском Союзе. В программе, показанной на рисунке 4, можно поменять опорные числа «тройка, семерка, туз» на «тройка, семерка, дама» (еще одна тройка — см. эпиграф), получить новую последовательность целых чисел и пропустить ее через сито сайта oeis.org.

Но вернемся к нашей основной теме — к рекурсии.

Расчет факториала (рис. 1) и чисел Фибоначчи (обычных или прекрасных) можно вести и без рекурсии — с помощью, например, такого алгоритма: знаешь предыдущие числа — высчитываешь очередное, очередное становится одним из предыдущих, а сама операция продолжается до тех пор, пока не доберемся до искомого числа. Такой *рекуррентный*, а не рекурсивный счет ведется быстрее и не требует значительных ресурсов памяти компьютера. Но рекурсия очень упрощает написание программы. Классический пример — программа, решающая головоломку «Ханойские башни» (рис. 5).

* Здесь везде по семь банкнот. Считается, что устойчивая банковская система должна базироваться на «красивом» числе банкнот. В СССР, кстати, до денежной реформы 1961 года и монет было семь — 1, 2, 3, 5, 10, 15 и 20 копеек. Оптимальное число банкнот и монет, а также гирек для весов (разновесов) — это еще одна интересная математическая задача.

Суть головоломки состоит в следующем. Имеется три стержня, на первый из которых (у нас в программе на рисунке 5 он маркирован как А) нанизаны диски на манер детской пирамидки: самый большой диск — внизу, самый маленький — наверху. Предлагается переложить эти диски на стержень С, беря их по одному и не кладя больший диск на меньший. Для временного складирования разрешается использовать третий стержень В. В программе на рисунке 5 достаточно сообщить только число дисков в пирамиде n . После запуска программа будет возвращать порядок перекладывания дисков:

- $n = 1$: AC (один ход)
- $n = 2$: AB, AC и BC (три хода)
- $n = 3$: AC, AB, CB, AC, BA, BC и, наконец, AC (семь ходов)
- $n = 4$: ... (15 ходов) и т. д.

Число перестановок p в общем случае равно $2^n - 1$. Задача с n дисками легко сводится к задаче с $n - 1$ диском, а задача с $n - 1$ диском легко сводится к задаче с $n - 2$ дисками и т. д. до задачи с одним диском, которая решается просто — диск со стержня А перекладываем на диск С (см. на рисунке 5 фрагмент программы: `if n = 1...`). Отсюда и рекурсия в программе на рисунке 5.

По древней легенде, тибетские монахи уже несколько тысячелетий перекладывают 64 золотых диска, нанизывая их на алмазные стержни. Когда головоломка будет решена и на стержне С окажутся все диски, наступит конец света. Спасает нас лишь то, что при 64 дисках для решения головоломки потребуется свыше... триллиона лет, если на каждый ход тратить по секунде. Это без учета ложных ходов и времени, необходимого на ручной или компьютерный расчет порядка перестановки дисков. В Интернете, кстати, есть виртуальные игры «Ханойские башни». Откройте такую игру и поиграйте с 5–10 дисками.

Е. Д. Зацепина, А. С. Мишина,
Московский педагогический государственный университет

СИСТЕМЫ СЧИСЛЕНИЯ: ЭТО МОЖЕТ БЫТЬ ИНТЕРЕСНО!*

Аннотация

В статье рассмотрены примеры заданий, которые можно использовать на уроках информатики для закрепления темы «Системы счисления».

Ключевые слова: урок информатики, системы счисления, мотивация.

Контактная информация

Зацепина Екатерина Дмитриевна, студентка Московского педагогического государственного университета; *адрес:* 119571, г. Москва, пр. Вернадского, д. 88, комн. 410; *телефон:* (495) 438-79-77; *e-mail:* dryusik15@gmail.com

E. D. Zatsepina, A. S. Mishina,
Moscow State Pedagogical University

NUMBER SYSTEMS: IT CAN BE FUN!

Abstract

The article describes examples of tasks that can be used on informatics lessons to consolidate the theme "Number systems".

Keywords: informatics lesson, number systems, motivation.

Знакомство школьников с системами счисления, формирование у них устойчивых умений по переводу чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную, а также из двоичной в восьмеричную и шестнадцатеричную и наоборот — одна из задач курса информатики основной и старшей школы. Соответствующий теоретический материал включен в школьные учебники информатики [2]. Задачи «на системы счисления» занимают прочное место в ОГЭ и ЕГЭ по информатике. В учебном пособии [1] есть ряд заданий, позволяющих повысить мотивацию школьников к изучению этой темы и сделать работу по переводу чисел из одной системы счисления в другую разнообразной, захватывающей и интересной.

Как известно, интересных заданий никогда не бывает много. Важно, чтобы у учителя в резерве всегда был их запас.

Ниже представлены два типа заданий по системам счисления.

Первый тип — это задания на перевод двоичных, восьмеричных и шестнадцатеричных чисел в десятичную систему счисления.

Ученикам предлагается построить изображение по координатам, заданным в указанных системах счисления.

Порядок выполнения заданий таков: ученики:

- 1) записывают координаты всех точек в десятичной системе счисления;
- 2) на листочке в клетку строят декартову прямоугольную систему координат, самостоятельно определяя размер единичного отрезка;
- 3) отмечают на координатной плоскости каждую из точек, при этом рядом с каждой точкой рекомендуется проставлять ее номер;
- 4) соединяют точки в заданной последовательности.

Приведем примеры таких заданий.

Задание 1.1.

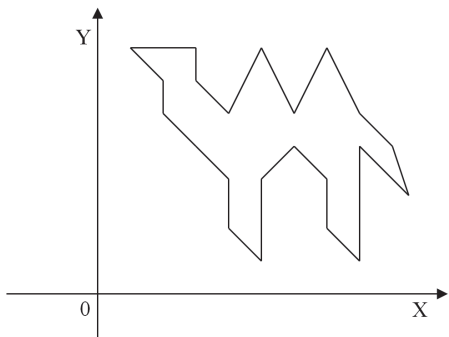
Постройте на координатной плоскости рисунок, отметив и последовательно соединив точки:

№ п/п	X _p	Y _p	X ₁₀	Y ₁₀	№ п/п	X _p	Y _p	X ₁₀	Y ₁₀
1	16 ₈	100 ₂			15	110 ₂	15 ₈		
2	10000 ₂	2 ₈			16	6 ₈	F ₁₆		
3	10000 ₂	1001 ₂			17	2 ₈	17 ₈		
4	10011 ₂	110 ₂			18	100 ₂	D ₁₆		
5	10010 ₂	11 ₈			19	4 ₈	B ₁₆		
6	10000 ₂	B ₁₆			20	110 ₂	1001 ₂		
7	F ₁₆	D ₁₆			21	8 ₁₆	111 ₂		
8	E ₁₆	1111 ₂			22	1000 ₂	4 ₁₆		
9	D ₁₆	15 ₈			23	A ₁₆	2 ₈		
10	C ₁₆	B ₁₆			24	12 ₈	111 ₂		
11	1011 ₂	1101 ₂			25	C ₁₆	11 ₈		
12	A ₁₆	F ₁₆			26	E ₁₆	111 ₂		
13	1001 ₂	1101 ₂			27	16 ₈	100 ₂		
14	1000 ₂	1011 ₂							

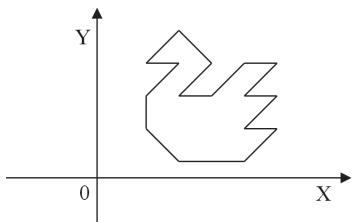
* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_09-2015/

Ответы.

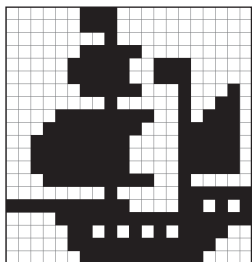
1.1.



1.2.



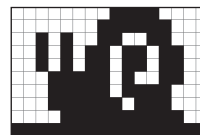
2.1.



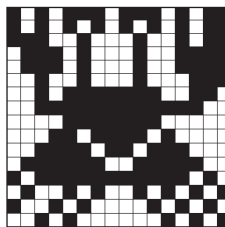
2.2.



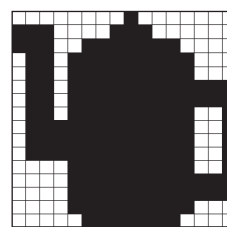
2.3.



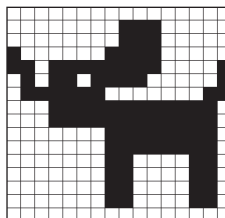
2.4.



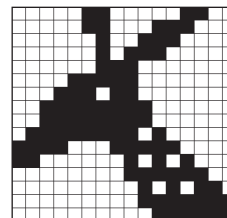
2.5.



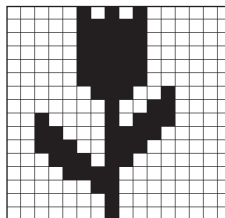
2.6.



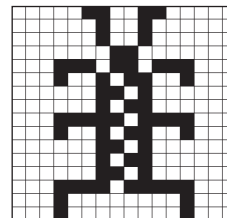
2.7.



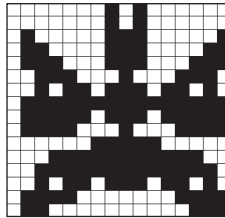
2.8.



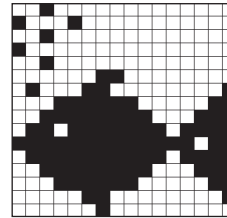
2.9.



2.10.



2.11.



Литература

1. Босова Л. Л., Босова А. Ю. Информатика: рабочая тетрадь для 8 класса. М.: БИНОМ. Лаборатория знаний, 2015.
2. Босова Л. Л., Босова А. Ю. Информатика: учебник для 8 класса. М.: БИНОМ. Лаборатория знаний, 2013.

НОВОСТИ

ИТ в руках фрилансеров

По данным опроса, проведенного в 2014 году по заказу Союза фрилансеров США и компании Elance-oDesk, в США фрилансерской работой хотя бы частично занимается 53 млн человек, т. е. 34 % трудоспособного населения. В консультационной фирме Deloitte Consulting называют новую экономику «экономикой открытых кадров», по аналогии с экономикой программ с открытым исходным кодом. В этой экономике

наличие доступа к кадровым ресурсам важнее, чем обладание ими. ИТ-службы часто первыми в компании замечают переход к фрилансерской модели и пытаются использовать ее, отмечают в Deloitte. Но при этом возникает множество вопросов: как планировать работу, как оплачивать ее, должен ли работодатель обучать фрилансера и т. д. Традиционную модель управления кадрами приходится менять.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

ПРАВИЛА ДЛЯ АВТОРОВ

Общие положения

Все присланные статьи рецензируются. Публикация статей возможна только при наличии положительного отзыва рецензентов.

Поскольку рецензирование и предпечатная подготовка материалов занимают не менее трех месяцев, статьи следует присылать в редакцию заблаговременно.

Редакция не берет платы за публикацию рукописей аспирантов.

Требования к файлам рукописи

- Текст статьи должен быть представлен в формате текстового редактора Microsoft Word (*.doc, *.rtf):
 - формат листа — А4;
 - все поля по 2 см;
 - шрифт — Times New Roman, кегль — 12 пт, расстояние между строками — 1,5 (полтора) интервала;
 - графические материалы вставлены в текст.
- Файл со статьей должен содержать следующие данные для публикации, **необходимо строго придерживаться указанной ниже последовательности** (пожалуйста, проверяйте оформление по образцу статьи, представленному на сайте ИНФО):
 - И. О. Фамилия** автора(ов) на русском языке.
 - Место работы** автора(ов) на русском языке. Необходимо указать место работы **каждого** автора. Если из названия организации не следует принадлежность к населенному пункту, через запятую надо указать название населенного пункта.
 - Название статьи** на русском языке.
 - Аннотация** на русском языке (3–5 строк в указанном выше формате).
 - Ключевые слова** на русском языке (не более 10, через запятую).
 - Подробная информация об авторах** — для каждого из авторов:
 - фамилия, имя, отчество (полностью);
 - ученая степень;
 - ученое звание;
 - должность;
 - место работы;
 - адрес места работы (обязательно с индексом);
 - рабочий телефон (обязательно с кодом города);
 - адрес электронной почты (e-mail).
 - И. О. Фамилия** автора(ов) на английском языке.
 - Место работы** автора(ов) на английском языке.
 - Название статьи** на английском языке.
 - Аннотация** на английском языке.
 - Ключевые слова** на английском языке (через запятую).
 - Текст статьи** в указанном выше формате.
 - Список литературных и интернет-источников**, упорядоченный в алфавитном порядке.
- При отправке статьи в редакцию в полях электронной формы необходимо указать подробные сведения об авторе:
 - фамилия, имя, отчество (полностью);
 - домашний почтовый адрес (с индексом);
 - домашний телефон (обязательно с кодом города);
 - мобильный телефон;
 - адрес электронной почты (e-mail).

Данные сведения необходимы для оперативной связи с автором статьи и пересылки авторского экземпляра журнала и **не подлежат публикации**.

Если авторов несколько, необходимо представить указанные сведения **обо всех авторах**.

4. При необходимости статья может сопровождаться дополнительным материалом в электронном виде (презентации, листинги программ, книги Excel, примеры выполнения работ и др.), который будет размещен на сайте ИНФО.

5. Иллюстрации следует представлять в виде отдельных графических файлов (даже при их наличии в документе Word) в формате TIFF или JPG, разрешение — не менее 300 пикселей на дюйм.

Уважаемые коллеги!

С 1 октября 2015 года статьи для публикации в журналах «Информатика и образование» и «Информатика в школе» должны отправляться в редакцию **только через электронную форму на сайте ИНФО (раздел «Авторам → Отправка статьи»):**

<http://infojournal.ru/authors/send-article/>

Обращаем ваше внимание, что для отправки статьи необходимо предварительно зарегистрироваться на сайте ИНФО (или авторизоваться — для зарегистрированных пользователей).

Требования к оформлению представляемых для публикации материалов остаются прежними, с ними можно ознакомиться на сайте ИНФО в разделе **«Авторам»:**

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

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

e-mail: readinfo@infojournal.ru

телефон: (495) 364-95-97

Журнал «Информатика в школе»

Индексы подписки (агентство «Роспечать»)
на 1-е полугодие 2016 года

- 81407 — для индивидуальных подписчиков
- 81408 — для организаций

Периодичность выхода: 5 номеров в полугодие (в январе не выходит)

Редакционная стоимость:
индивидуальная подписка — 150 руб.
подписка для организаций — 300 руб.



Федеральное государственное унитарное предприятие "Почта России" Ф СП - 1											
Бланк заказа периодических изданий											
АБОНЕМЕНТ На газету Информатика в школе журнал <small>(наименование издания)</small>	<input style="width: 100%;" type="text"/> <small>(индекс издания)</small>										
Количество комплектов	<input style="width: 100%;" type="text"/>										
На 2016 год по месяцам											
1	2	3	4	5	6	7	8	9	10	11	12
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Куда <input style="width: 40%;" type="text"/>											
<small>(почтовый индекс)</small> <small>(адрес)</small>											
Кому <input style="width: 100%;" type="text"/>											
----- Линия отреза -----											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> <td style="width: 33%;"><input style="width: 100%;" type="text"/></td> </tr> <tr> <td style="text-align: center;">ПВ</td> <td style="text-align: center;">место</td> <td style="text-align: center;">литер</td> </tr> </table>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	ПВ	место	литер	ДОСТАВОЧНАЯ <input style="width: 100%;" type="text"/> КАРТОЧКА <small>(индекс издания)</small>				
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>									
ПВ	место	литер									
На газету журнал Информатика в школе <small>(наименование издания)</small>											
Стоимость	подписки руб.		Количество комплектов								
	каталожная руб.										
	переадресовки руб.										
На 2016 год по месяцам											
1	2	3	4	5	6	7	8	9	10	11	12
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
почтовый индекс		Город		село		область		Район		улица	
код улицы		дом		корпус		квартира		Фамилия И.О.			

Электронная подписка

Оформив электронную подписку, вы получаете уникальную возможность получать журналы не выходя из дома сразу же после их выпуска издательством, экономя при этом свои деньги.

Вы можете оформить электронную подписку 2015 года на наши издания

«ИНФОРМАТИКА И ОБРАЗОВАНИЕ»

Издается с 1986 года ♦ 64 страницы ♦ Выходит 10 раз в год

- Ежемесячные тематические выпуски по практике информатизации образования.
- Обзоры школьной методической литературы по информатике.
- Образовательные стандарты и примерные программы по информатике.
- Материальная база школ: оснащение программным и аппаратным обеспечением.
- Организация сетевого взаимодействия участников образовательного процесса.
- Подготовка и повышение квалификации педагогических кадров.
- Актуальные вопросы информатизации образования в России.
- Информатизация процесса управления образованием.
- Обзоры программных продуктов и практика их применения.



«ИНФОРМАТИКА В ШКОЛЕ»

Издается с 2002 года ♦ 64 страницы ♦ Выходит 10 раз в год

- Методические разработки уроков.
- Сценарии конкурсов, викторин, деловых игр по информатике.
- Проектная деятельность в школьном курсе информатики.
- Формирование УУД на основе ФГОС второго поколения.
- Рекомендации для подготовке к ЕГЭ и ГИА.
- Документы по вопросам аттестации учителей информатики.
- Дидактические материалы по информатике.
- Задачи по информатике с решениями.
- Разбор олимпиадных задач по информатике.
- Использование ИКТ в начальной школе.



Подробную информацию об электронной подписке вы можете найти на нашем сайте: www.infojournal.ru



Моделируя, учимся и учим

Московский педагогический государственный университет и фирма «1С» сообщают о проведении педагогического конкурса «Креативная математика». Цель конкурса — выявление и поддержка лучших примеров использования сред динамической математики в обучении школьников и студентов педагогических специальностей. Девиз конкурса: «Моделируя, учимся и учим».

К участию в конкурсе приглашаются учителя математики, преподаватели математических дисциплин и студенты вузов педагогической направленности. На конкурс принимаются:

- разработки конспекта урока или задания для организации проектной деятельности с использованием интерактивной модели динамической математики из коллекции моделей среды «1С:Математический конструктор»;
- разработки интерактивных моделей динамической математики по собственному оригинальному сценарию.

Заявки на участие в конкурсе принимаются до конца января, а сами конкурсные работы — в феврале 2016 г.

В состав экспертного совета конкурса вошли известные представители системы математического образования и информатизации образования в России. Конкурс поддержали лаборатория проблем информатизации образования ИПИ ФИЦ ИУ РАН, ГБОУ «Центр педагогического мастерства», проект «Школа новых технологий» Департамента образования и Департамента информационных технологий г. Москвы; СУНЦ им. А.Н. Колмогорова; МГУ им. М.В. Ломоносова; ГБОУ ВО Московской области «Академия социального управления»; кафедра методики преподавания математики Института математики, информационных и космических технологий САФУ им. М.В. Ломоносова; кафедра информатики и методики преподавания математики и информатики Пензенского государственного университета, газета «Математика» ИД «Первое сентября», журнал «Информатика и образование».

Участие в конкурсе «Креативная математика» бесплатное. Участникам предоставляются свидетельства. Победители и лауреаты конкурса получают грамоты, возможность публикации в журнале «Информатика и образование».

Призовой фонд конкурса составляет 120 тысяч рублей.

Контактный адрес электронной почты оргкомитета Конкурса: obr@1c.ru.

Подробнее об условиях участия, составе организаторов и экспертного совета, порядке регистрации и подачи работ на конкурс, датах проведения 1-го заочного этапа и 2-го этапа — очного научно-практического семинара см. в Положении о конкурсе «Креативная математика» на его официальном сайте: <http://obr.1c.ru/mpgu>.