

ИНФОРМАТИКА В ШКОЛЕ

№ 4 (117) май 2016

Редакционный совет

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

Редакция

Босова Л. Л.
главный редактор
Кириченко И. Б.
*заместитель
главного редактора*
Губкин В. А.
Коптева С. А.
Кузнецова Е. 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 23.05.2016.
Формат 60×90¹/₈. Усл. печ. л. 8,0.
Тираж 2000 экз. Заказ № 094.

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

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

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

© «Образование и Информатика», 2016

Содержание

От редакции 2

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Босова Л. Л., Босова А. Ю. Изучаем ИКТ. Третий класс 3

УРОКИ ИНФОРМАТИКИ

Гусева Л. А. Урок на тему «Техника безопасности и правила поведения
в компьютерном классе» 16

Бабичев А. В. Проектно-исследовательская деятельность на тему
«Дом, в котором мы живем» 20

Лейман Е. В. Урок на тему «Построение фигур с центральной
симметрией на языке программирования Лого» 24

Сабитова Г. М. Урок на тему «Исполнитель Чертежник.
Команда вектор (x, y)» 28

ИНТЕГРИРОВАННЫЕ УРОКИ

Иванова С. В. ИКТ в проектной деятельности на уроках геометрии 34

МЕТОДИЧЕСКАЯ КОПИЛКА

Зенцова Л. В. Опыт проектирования образовательной деятельности
на основе системно-деятельностного подхода 39

Мухаметзянов Р. Р. Об изучении объектно-ориентированного
программирования в средней общеобразовательной школе на примере
языка программирования С# 44

ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Лалетин Н. В., Лалетин Д. Н. Школьный этап всероссийской
олимпиады по информатике в Красноярском крае 51

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

Очков В. Ф., Соколов А. В. Как вы корабль рассчитаете и назовете,
так он и поплывет 56

ЗАДАЧИ

Мазничевская Л. И. Использование процедур и функций в языке
программирования Паскаль 60

КОНКУРСЫ 63



Дорогие друзья!

Наступил последний весенний месяц — май, близится завершение очередного учебного года. Это одновременно и радостная, и ответственная пора. Выпускники усиленно готовятся к экзаменационным испытаниям, остальные ученики с нетерпением ждут начала летних каникул. А для педагогов это время, когда подводятся итоги работы, оцениваются достижения, анализируются недочеты, чтобы в новом учебном году работать более успешно и эффективно.

Редакция журнала «Информатика в школе» тоже подводит итоги и намечает новые планы. Одним из основных направлений нашей работы стало совершенствование взаимодействия с читателями посредством сайта ИНФО. В значительной степени был обновлен функционал сайта, а также появились новые разделы, в том числе форум, где можно обсудить самые разные проблемы и задать вопросы сотрудникам редакции.

Впервые в этом году в конкурсах, проводимых издательством «Образование и Информатика», появилась номинация «Методическая копилка», победителей которой выбирают не только члены жюри, но и вы, уважаемые коллеги, путем онлайн-голосования на нашем сайте. Мы планируем сделать эту номинацию традиционной и в XIII Всероссийском конкурсе ИНФО-2016 предложить в ней новые интересные темы. Приглашаем всех к участию в конкурсе и к голосованию за работы конкурсантов на сайте ИНФО!

Май отмечен еще одной новинкой на сайте ИНФО — вышел первый номер интернет-журнала для учащихся «Мир информатики»: <http://infojournal.ru/mir-info/>. В выпусках журнала будут представлены учебные материалы по различным вопросам школьного курса информатики. Пожалуйста, знакомьте своих учеников с материалами журнала, приглашайте их к участию в конкурсах и выполнению заданий. Победители конкурсов и авторы лучших ответов будут награждаться дипломами. И конечно, мы рассмотрим возможность публикации в журнале «Мир информатики» ваших материалов, полезных и интересных учащимся.

Благодаря новым сервисам сайта ИНФО вы можете в онлайн-режиме оформить годовую подписку на журналы «Информатика в школе» и «Информатика и образование» и приобрести их отдельные выпуски.

Напоминаем вам, что на странице каждого номера журнала «Информатика в школе» на сайте ИНФО вы можете скачать дополнительные материалы к статьям, в том числе к статьям данного выпуска журнала.

Впереди у нас еще много новых проектов, в которых, мы надеемся, вы, уважаемые читатели, примите активное участие. Благодарим всех авторов и читателей за сотрудничество и желаем успешного окончания учебного года!

*Редакция журнала
«Информатика в школе»*



Л. Л. Босова,

Федеральный институт развития образования, Москва,

А. Ю. Босова,

Учебно-методический центр Истринского района Московской области

ИЗУЧАЕМ ИКТ. ТРЕТИЙ КЛАСС*

Аннотация

Умения, связанные с использованием информационных и коммуникационных технологий, должны быть сформированы у каждого выпускника начальной школы. В статье представлены материалы, направленные на формирование таких умений у учащихся третьего класса. Ими можно воспользоваться как в рамках самостоятельного курса, так и в процессе изучения русского языка, окружающего мира, технологии и изобразительного искусства.

Ключевые слова: начальная школа, информационные и коммуникационные технологии, компьютер.

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

Босова Анна Юрьевна, директор Учебно-методического центра Истринского района Московской области; адрес: 143500, Московская область, г. Истра, пл. Революции, д. 7-2; телефон: (498) 315-18-61; e-mail: moyumc@gmail.com

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

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

WE STUDY ICT. THIRD CLASS

Abstract

Skills related to the use of information and communication technologies should be formed in each graduate of primary school. In the article materials aimed at the creation of such skills at the third grade pupils are presented. They can be used either as a stand-alone course and as in the process of studying Russian language, natural science, technology and art.

Keywords: primary school, information and communication technologies, computer.

§ 11. Текстовая информация и текстовый редактор

Письменность — важнейший для человека способ представления информации. С помощью письменности люди передают свои знания потомкам. Много полезной информации можно получить из школьных учебников, книг, журналов, газет и других печатных изданий.

Информация, представленная письменно, называется **текстовой информацией**.

На протяжении тысячелетий люди записывали информацию. В течение этого времени менялось и то, на чём записывали информацию (камень, глина, дерево, папирус, пергамент, бумага), и то, с помощью чего это делали (острый камень, костяная палочка, птичье перо, перьевые ручки, авторучки, с конца XIX века для выполнения письменных работ стала применяться пишущая машинка). Но не менялось главное: чтобы внести изменения в текст, его надо было заново переписать. А это очень длительный и трудоёмкий процесс.

Появление компьютеров коренным образом изменило технологию письма. С помощью специальных компьютерных программ можно набрать любой текст, при необходимости внести в него изменения, записать текст в память компьютера для длительного хранения, распечатать на принтере какое угодно количество копий текста без его повторного ввода или отправить текст с помощью электронной почты на другие компьютеры.

На каждом компьютере установлены специальные программы, предназначенные для создания текстов, — текстовые редакторы. Любой текст, созданный с помощью текстового редактора, вместе с включёнными в него нетекстовыми материалами называют **документом**.

Мы с вами будем работать с текстовым редактором WordPad. Он поставляется вместе с операционной системой Windows. Программа WordPad запускается из меню **Стандартные** щелчком на её названии.

Окно программы WordPad имеет стандартный вид. Это значит, что оно выглядит примерно так же, как и ранее рассмотренные нами окна. В этом окне вы можете найти уже известные вам элементы. Примерно так же будут выглядеть и окна других программ.

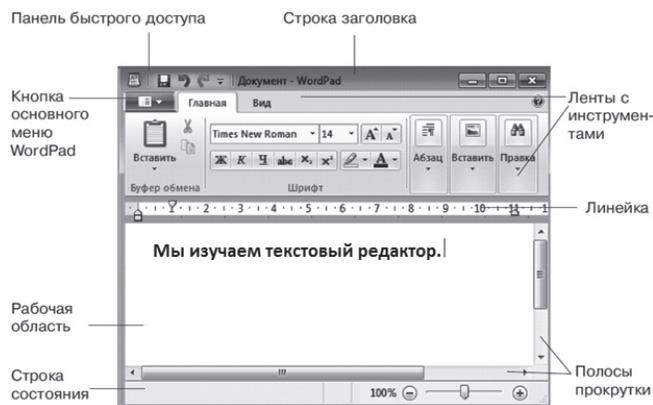
Самый верхний край окна программы называется **строкой заголовка**. В этой строке указано название программы и имя файла, над которым ведётся работа. Так как мы ещё не успели присвоить имя файлу, то компьютер дал ему временное имя — **Документ**.

Под строкой заголовка находятся **ленты с инструментами**. С ними мы познакомимся чуть позже.

Большую часть окна текстового редактора занимает белое поле. Оно называется **рабочим листом**.

* Продолжение. Начало см.: Информатика в школе. 2016. № 3.

В левом верхнем углу рабочего листа видна вертикальная мигающая черта — **текстовый курсор**. Он указывает место, в котором при вводе появится текст. Символ, который вы набираете, появляется перед курсором, а сам курсор смещается вправо.



Попробуйте ввести предложение:

Мы изучаем текстовый редактор.



Самое главное

Информация, представленная письменно, называется текстовой информацией.

Для обработки текстовой информации предназначены специальные программы — текстовые редакторы.

Любой текст, созданный с помощью текстового редактора, вместе с включёнными в него нетекстовыми материалами называют документом.



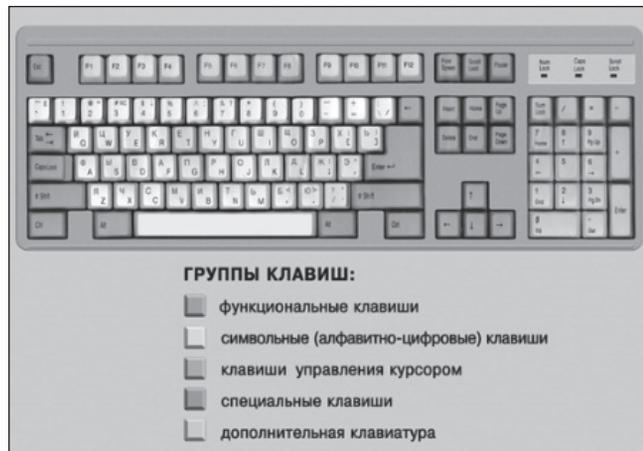
Вопросы и задания

1. Расскажите о текстовой форме представления информации. Какие другие формы представления информации вы знаете? Расскажите о преимуществах или недостатках представления информации в виде текста по сравнению с описанной вами формой представления информации.
2. Как называются компьютерные программы, предназначенные для создания текстов?
3. С какой целью вы создаёте тексты? Приведите два-три примера.
4. Приведите примеры текстов, отличающихся по размеру; по оформлению; по назначению.
5. Какие принципиальные изменения в процесс создания текста внёс компьютер?
6. Перечислите основные элементы окна программы WordPad.
7. Почему окно программы считается стандартным?
8. Укажите общие элементы окна программ Блокнот и WordPad. Какая из программ, по вашему мнению, обладает большими возможностями? Обоснуйте свой ответ.

§ 12. Клавиатура — основное устройство ввода текстовой информации

Очень важным устройством ввода информации в память компьютера является клавиатура. Клавиатура

настольного компьютера содержит более 100 клавиш. Чтобы быстрее их освоить, клавиши можно разделить на группы. На рисунке эти группы выделены разными цветами.



Группа функциональных клавиш F1—F12 размещена в верхней части клавиатуры. Действия, выполняемые в результате нажатия на эти клавиши, определяются работающей в конкретный момент программой. Некоторым функциональным клавишам в *разных* программах могут соответствовать *разные* действия. За другими клавишами практически всегда закреплено одно и то же действие. Так, очень часто клавиша F1 служит для вызова справки.

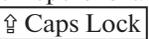
В центре клавиатуры расположены **символьные клавиши**. На них нанесены цифры, специальные символы (!, :, * и т. д.), буквы русского алфавита и буквы английского алфавита. В нижней части клавиатуры находится большая белая клавиша — пробел. Пробел используется для отделения слов и выражений друг от друга.

С помощью символьных клавиш в текстовом редакторе вы можете набирать всевозможные тексты, арифметические выражения, записывать свои программы. Но прежде посмотрите, на какой из двух алфавитов настроена ваша клавиатура. Эта информация находится на панели задач справа.



Переключение с русского алфавита на английский и наоборот можно производить так:

- 1) найдите значок алфавита на панели задач;
- 2) наведите на него указатель мыши;
- 3) выполните щелчок основной кнопкой мыши;
- 4) в появившемся меню выберите указателем мыши нужный алфавит;
- 5) выполните щелчок основной кнопкой мыши.

Буквы каждого алфавита могут быть прописными (большими) или строчными (маленькими). Переключают режим ввода букв нажатием клавиши , («кэпс лок», сокращение от английского словосочетания «фиксация прописных букв»). Этой клавишей стоит воспользоваться, если прописными буквами нужно печатать слово или целую фразу.

Вводить одну или несколько прописных букв удобно с помощью комбинации клавиш  + {буква}.



Выполнить эту и другие клавиатурные комбинации можно так: нажать первую из указанных клавиш и, удерживая её нажатой, нажать вторую клавишу; затем обе клавиши отпустить.

Символьная клавиатура условно делится на две части — для правой и левой рук. Каждый палец «закреплён» за определёнными клавишами. На рисунке разными геометрическими фигурами обозначены зоны «ответственности» для каждого пальца.

Чтобы научиться работать на клавиатуре, используйте специальную компьютерную программу — **клавиатурный тренажёр**. При этом соблюдайте следующие правила:

1. Применяйте удар-толчок — чёткий, отрывистый и лёгкий «прыжок» пальца к клавише. При тяжёлых, прижатых ударах по клавишам может развиться заболевание суставов пальцев.
2. Удары по всем клавишам компьютера, независимо от их расположения, должны быть равномерными и одинаковой силы.
3. После удара по клавише неосновной позиции палец должен возвращаться на своё основное место.
4. При ударе по клавише палец не должен прогибаться, при этом другие пальцы не должны подниматься со своих мест.
5. Если вы почувствовали напряжение или усталость, то можно на несколько секунд прервать работу, закрыть глаза, откинуться на спинку стула, расслабить кисти рук и вытянуть ноги.



Самое главное

С помощью символьных клавиш набирают всевозможные тексты, арифметические выражения, записывают программы.

Желательно научиться работать на клавиатуре вслепую.



Вопросы и задания

1. Расскажите об основной позиции пальцев на клавиатуре.
2. Какие правила следует соблюдать при работе на клавиатуре?
3. Запустите имеющийся в вашем распоряжении клавиатурный тренажёр и начните выполнять в нём тренировочные упражнения. Старайтесь делать это ежедневно.

§ 14. Хранение информации

Каждый человек хранит определённую информацию в собственной памяти — «в уме». Вы помните свой домашний адрес, имена, адреса и дни рождения близких родственников и друзей. В вашей памяти хранятся таблицы сложения и умножения, основные орфограммы и другие знания, полученные в школе. Собственную (внутреннюю) память человека можно назвать **оперативной**, потому что содержащаяся в ней информация воспроизводится достаточно быстро. Но так уж устроен человек, что он не может долго хранить большие объёмы информации в собственной памяти: информация, которой человек не пользуется, со временем забывается. Чтобы избежать этого, люди используют записные книжки, справочники, энциклопедии и другие внешние хранилища информации — **внешнюю память**. Такую память можно назвать **долговременной**.

Память — важная часть любого компьютера. В ней в файлах хранятся компьютерные программы, а также

документы, созданные или обработанные с помощью этих программ. Подобно памяти человека различают оперативную и долговременную компьютерную память.

Когда вы открываете окно компьютерной программы, то вся информация, необходимая для работы компьютера, помещается в его **оперативную память**. Как только вы закроете окно программы и выключите компьютер, содержимое оперативной памяти компьютера будет утеряно.

Чтобы созданный документ не исчез бесследно, его надо сохранить в файле в **долговременной памяти** компьютера.

Каждый файл имеет своё имя. По именам файлы различаются между собой. **Имя файла** состоит из двух частей, разделённых точкой. Например, *Стихи.doc* или *Рисунок.bmp*. Левая часть (до точки) — это **собственно имя файла**. Придумывает его создатель файла. Записать его можно на русском или на английском языке. Правая часть (после точки) называется **расширением имени файла**. Расширение обычно записывается тремя английскими буквами. Оно присваивается автоматически той программой, с помощью которой создан этот файл.

Правила записи имён файлов зависят от операционной системы, с которой работает компьютер. Например, в операционной системе Windows:

- имя может (но не обязательно должно) быть длиной до 255 знаков;
- в имени можно использовать любые знаки, кроме * ? < > \ / |;
- имя должно отражать содержание документа (это не обязательно, но желательное условие в любой операционной системе).

Файлов в компьютерной памяти хранится очень много. Чтобы не запутаться в них и быстро находить нужный файл, отдельные файлы принято объединять в **папки**.



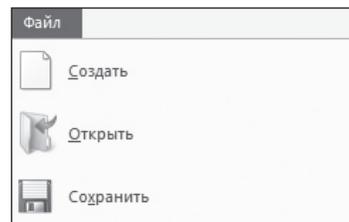
Запустите программу WordPad. Наберите фразу:

Мы изучаем текстовый редактор WordPad.

Пока вы не закрыли окно программы WordPad, набранный текст хранится в оперативной памяти.

Чтобы компьютер при следующем запуске не «забыл» введённый вами текст, этот текст нужно сохранить — записать в долговременную память компьютера.

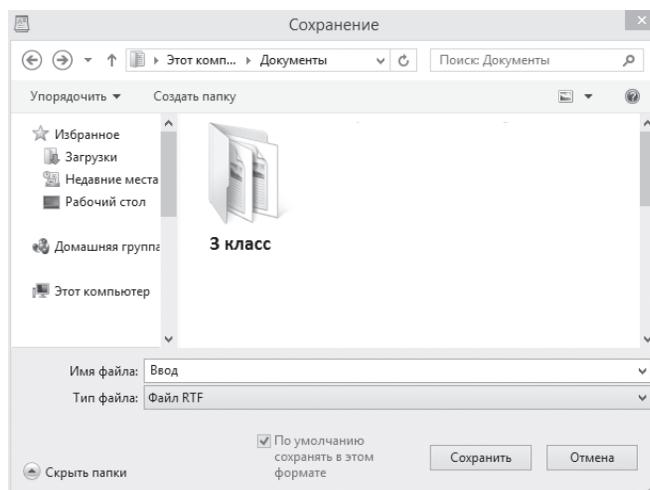
Вызовите основное меню WordPad и выберите в нём команду **Сохранить**.



В открывшемся окне **Сохранение** надо указать, куда и под каким именем следует разместить наш файл.

В рабочей области окна **Сохранение** найдите папку *3 класс* и откройте её двойным щелчком.

В папке *3 класс* найдите и откройте свою личную папку. Введите имя файла **Ввод** и щёлкните на кнопке **Сохранить**.



Завершите работу с текстовым редактором.

i Чтобы файлы не терялись, надо думать о том, куда и под каким именем их следует сохранять.

i **Самое главное**

Различают оперативную и долговременную компьютерную память.

Чтобы созданный на компьютере документ не исчез бесследно, его надо сохранить в файле в долговременной памяти компьютера.

Каждый файл имеет своё имя. Имя файла состоит из двух частей, разделённых точкой.

? **Вопросы и задания**

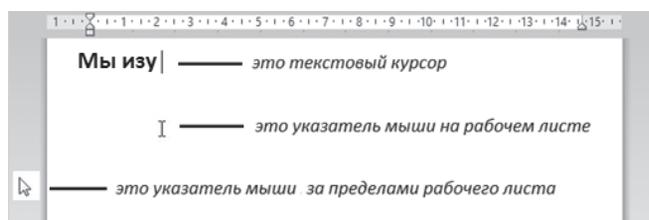
1. Для чего предназначена оперативная память человека и компьютера?
2. Для чего предназначена долговременная память человека и компьютера?
3. Что вам известно о правилах именования файлов в операционной системе Windows?
4. Можно ли окно папки *Сохранение* считать стандартным? Обоснуйте свой ответ.

§ 15. Ввод текста

Когда человек пишет карандашом или ручкой, буквы появляются там, где инструмент касается бумаги. А где же появляются буквы, когда вы набираете текст на клавиатуре? В каком месте экрана?

Вы могли обратить внимание, что при работе в текстовом редакторе место для ввода очередного символа указывается на экране с помощью мигающей вертикальной черты — текстового курсора.

Указатель мыши на рабочем листе текстового редактора становится похож на английскую букву I. За пределами рабочего листа он принимает форму стрелки.



Когда мы пишем ручкой на бумаге, то начинаем работу «с красной строки» — отступаем примерно на 2 см вправо. «С красной строки» принято начинать каждый новый абзац. При этом надо внимательно следить за тем, чтобы все абзацные отступы были одинаковыми.

В письменной речи абзац — это единица членения текста, его законченная смысловая часть, состоящая из одного или нескольких предложений.

Для компьютера **абзац** — произвольная последовательность символов, ограниченная специальными символами конца абзаца. Символ конца абзаца вводится нажатием клавиши Enter.

В текстовом редакторе есть специальный инструмент для установки абзацного отступа. Это линейка с маркерами:



Для того чтобы установить абзацный отступ:

- 1) установите указатель мыши на верхний маркер;
- 2) нажмите левую кнопку мыши и, не отпуская кнопку, перетащите маркер на 1–1,25 см вправо;
- 3) отпустите кнопку мыши — абзацный отступ установлен.

Ввод текста желательно начинать с установки абзацного отступа. Кроме того, следует придерживаться и других правил ввода:

- Там, где это нужно, используйте прописные буквы.
- Все знаки препинания, кроме тире, ставьте сразу же за последней буквой слова; после любого знака препинания нажимайте клавишу пробела. Тире выделяйте пробелами с двух сторон.
- Избегайте ввода подряд двух и более пробелов; не используйте пробел для выравнивания границ абзаца.
- Не следите за концом строки: как только он будет достигнут, курсор автоматически перейдёт на начало следующей строки.
- Для того чтобы перейти к вводу нового абзаца (или строки стихотворения), нажимайте клавишу Enter.

Не всякий текстовый документ удаётся ввести за один раз. Иногда удаётся ввести и сохранить лишь часть документа. В следующий раз документ нужно открыть и продолжить работу по вводу текста.

При открытии документа текстовый курсор всегда находится в начале первой строки. Чтобы перевести текстовый курсор в то место, куда будет вводиться текст, можно:

- 1) подвести указатель мыши к тому месту, куда следует поместить текстовый курсор;
- 2) щёлкнуть кнопкой мыши.

Перевести текстовый курсор в нужное место можно также и с помощью клавиш с курсорными стрелками (↑, ↓, ←, →). Если нужно перейти вниз, в ту часть рабочего листа, где ещё нет текста, необходимо клавишей Enter создать новые строки — пустые абзацы.

i **Самое главное**

Чтобы научиться создавать тексты на компьютере, нужно много тренироваться. При вводе текста нужно придерживаться ряда правил.

? Вопросы и задания

1. Какие правила необходимо соблюдать при наборе (вводе) текста?
2. Откройте в программе WordPad документ *Ввод*. Дополните текст и сохраните файл под тем же именем.

Мы изучаем текстовый редактор WordPad. Текстовый редактор нужен для создания и оформления рассказов, докладов, статей для школьной газеты и других документов.

3. Создайте в текстовом редакторе новый документ. Введите следующий текст и сохраните его в личной папке под именем *Медвежонок*:

На поляне с брусничным настоем стоял медвежонок. Он был один и делал то, что хотел. А делал медвежонок что-то странное. То вдруг взмахивал головой, а лапками и носом тыкался в землю. То переваливался через тощий задок и скреб коготками землю. Медвежонок явно что-то ловил и никак не мог поймать. Я вдруг понял: медвежонок ловил свою тень!

4. Напечатайте текст стихотворения Агнии Барто «Мячик»:

Наша Таня громко плачет:
Уронила в речку мячик.
— Тише, Танечка, не плачь:
Не утонет в речке мяч.

Сохраните документ в личной папке. Имя файлу придумайте самостоятельно.

§ 16. Исправление ошибок

Какие досадные ошибки третьеклассники допускают при письме в тетрадке?

Согласитесь, очень часто это ошибки следующих типов:

- пропущенная буква;
- лишняя буква;
- неверная буква.

Если в написанном слове пропущена буква (пусть даже совершенно случайно!), исправить в тетрадке допущенную оплошность незаметным образом не получится. Если же пропущена буква в тексте, который вы набираете в текстовом редакторе, то все легко поправимо.

Для вставки пропущенного символа нужно:

- 1) с помощью курсорных стрелок подвести текстовый курсор к тому месту, где должен находиться пропущенный символ (установить курсор за последним правильным символом);
- 2) нажать клавишу с правильным символом.

Как можно установить курсор в место вставки пропущенного символа без помощи курсорных стрелок?

Ошибку, допущенную при письме на бумаге, иногда можно исправить с помощью ластика или «замазки». При работе в текстовом редакторе такие инструменты нам не помогут. Для исправления ошибок набора на клавиатуре имеются две особые клавиши. Это клавиши *Backspace* и *Delete*.

Удалять последний набранный символ удобно клавишей *Backspace* (читается: «бэк спэйс», в переводе с английского языка это означает «назад пространство»). Вот так она может выглядеть на разных клавиатурах:



При нажатии клавиши *Backspace* удаляется символ, стоящий слева от текстового курсора. Сам текстовый курсор после этого перемещается влево.

Как вы считаете, почему клавишу *Backspace* называют обратным пробелом?

Клавиша *Delete* (читается: «делит», в переводе с английского это означает «удалить») имеет вид:



При нажатии клавиши *Delete* удаляется символ, стоящий справа от текстового курсора. Сам текстовый курсор остаётся на прежнем месте, влево перемещается символ, следующий за курсором.

Объясните, почему говорят, что клавиша *Delete* удаляет «на месте».

Для того чтобы удалить лишний символ, можно:

- 1) установить курсор перед лишним символом;
- 2) нажать клавишу *Delete*.

Предложите другой способ удаления лишнего символа.

Для того чтобы заменить ошибочно введённый (неверный) символ, можно:

- 1) установить курсор за неверным символом;
- 2) удалить неверный символ, нажав клавишу *Backspace*;
- 3) нажать клавишу с верным символом.

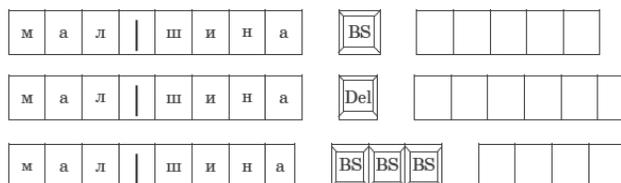
Предложите другой способ замены ошибочно введённого символа.

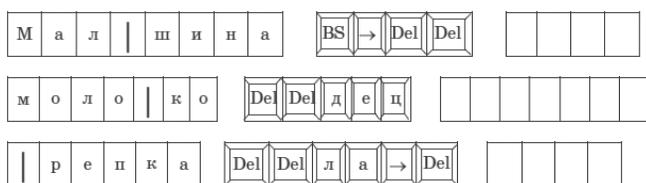
! Самое главное

Ошибку, допущенную при вводе текста, можно легко исправить. Символ, расположенный слева от курсора, удаляется клавишей *Backspace*. Символ, расположенный справа от курсора, удаляется клавишей *Delete*.

? Вопросы и задания

1. Ошибки каких типов легко исправить при вводе текста в текстовом редакторе?
2. Для чего предназначена клавиша *Backspace*?
3. Для чего предназначена клавиша *Delete*?
4. Что получится из исходного слова после нажатия указанных клавиш?





§ 17. Ввод текстов, содержащих повторы

Перед вами стихотворение Сергея Михалкова «Овощи»:

*Хозяйка однажды с базара пришла,
Хозяйка с базара домой принесла:
Картошку,
Капусту,
Морковку,
Горох,
Петрушку и свеклу.
Ох!..*

*Вот овощи спор завели на столе —
Кто лучше, вкусней и нужней на Земле:
Картошка?
Капуста?
Морковка?
Горох?
Петрушка или свекла?
Ох!..*

*Хозяйка тем временем ножик взяла
И ножиком этим крошить начала:
Картошку,
Капусту,
Морковку,
Горох,
Петрушку и свеклу.
Ох!..*

*Накрытые крышкою, в душном горшке
Кипели, кипели в крутом кипятке:
Картошка,
Капуста,
Морковка,
Горох,
Петрушка и свекла.
Ох!..
И суп овощной оказался неплох!*

Сколько времени вам потребуется для того, чтобы ввести этот текст? На самом деле, не так уж и много! Дело в том, что в тексте этого стихотворения несколько раз повторяется одна и та же группа строк. Иначе говоря, в тексте есть повторяющийся фрагмент.

Повторяющийся фрагмент достаточно набрать на клавиатуре только один (первый) раз. Затем его можно запомнить и сколько угодно раз вставить в любое другое место нашего документа.

Как это сделать — разберемся подробнее!

Итак, вы набрали первые восемь строк стихотворения, шесть из них будут повторяться ещё три раза (правда, с небольшими изменениями). Выделим эти строки. Для этого можно:

- 1) переместить текстовый курсор в начало первой строки фрагмента;
- 2) нажать и удерживать клавишу Shift;

- 3) нажать курсорную стрелку  и удерживать её до тех пор, пока не выделятся все нужные нам строки.

Выделить фрагмент можно и с помощью мыши. Для этого следует выполнить следующие действия:

- 1) подвести указатель мыши к началу фрагмента;
- 2) удерживая нажатой левую кнопку мыши, передвигать её до тех пор, пока не выделится требуемый фрагмент текста;
- 3) отпустить кнопку мыши.

Текст выделяется контрастным цветом

 Выбор фрагмента можно отменить щелчком в произвольном месте рабочей области окна.

Теперь надо скопировать выделенный фрагмент. Для этого можно щёлкнуть на кнопке



расположенной на панели инструментов. Никаких изменений с текстом, находящимся на экране, не произойдет. Текст останется там, где и был, а вот его копия будет отправлена в специальную область памяти компьютера, называемую **буфером обмена**.

После этого вы продолжаете набирать текст и, дойдя до того места, где должен быть повторяющийся фрагмент, вставляете его из буфера — щёлкаете на кнопке



Эту процедуру можно повторять сколько угодно раз.

 Операции копирования и вставки гораздо быстрее выполняются с помощью комбинаций следующих клавиш:
Ctrl + C — копирование текста в буфер обмена;
Ctrl + V — вставка текста из буфера обмена.

Ну вот, кажется, текст стихотворения введён! Сравните свой текст с тем, что напечатан в учебнике. Исправьте неточности, имеющиеся в вашем тексте.

Сохраните текст стихотворения в личной папке в файле с именем *Овощи*.

 **Самое главное**

Повторяющийся фрагмент набирается только один раз, затем его точная копия помещается в буфер обмена. Хранящийся в буфере фрагмент можно вставить в любое место документа сколько угодно раз.

 **Вопросы и задания**

1. Каким способом — рукописным или компьютерным удобнее создавать тексты, содержащие повторяющиеся фрагменты? Почему?
2. Как можно выделить текстовый фрагмент?
3. Какими способами можно скопировать выделенный фрагмент? Опишите их.
4. Какими способами можно вставить фрагмент из буфера обмена? Опишите их.
5. Каким образом осуществить копирование и вставку фрагмента с помощью контекстного меню?
6. Найдите текст стихотворения Самуила Маршак «Багаж». Как можно быстро набрать текст этого стихотворения на компьютере?

§ 18. Приёмы редактирования

Редактирование — следующий этап подготовки текста на компьютере. При редактировании вы внимательно просматриваете весь набранный текст, чтобы убедиться, что всё правильно. Вы можете не только исправить обнаруженные ошибки (например, в правописании), но и внести в текст куда более серьёзные изменения. Вот некоторые из них:

- удалить из текста отдельные слова, предложения или абзацы;
- перенести предложение или абзац из одной части текста в другую;
- разбить текст на абзацы или соединить несколько абзацев в один.

Если текст большой, то на экране будет видна только его часть, а весь он будет храниться в памяти компьютера. Поэтому важно освоить разные способы «перемещения» по тексту.

В пределах экрана текстовый курсор можно устанавливать с помощью указателя мыши или перемещать с помощью стрелок. Дело пойдёт быстрее, если для перемещения курсора использовать специальные клавиши или комбинации клавиш:

| | |
|-------------|-------------------|
| Home | — в начало строки |
| End | — в конец строки |
| Ctrl + → | — на слово вправо |
| Ctrl + ← | — на слово влево |
| Ctrl + Home | — в начало текста |
| Ctrl + End | — в конец текста |

Кроме того, окно текстового редактора имеет полосы прокрутки: горизонтальную и вертикальную. С их помощью можно быстро вывести на экран части текста, находившиеся за его пределами.

При редактировании можно работать не только с отдельными символами, но и с целыми фрагментами текста. Предварительно фрагмент должен быть выделен.



Вспомните, как выделяется фрагмент.

Выделенный фрагмент можно удалить из текста и «забыть» о нём. Для этого достаточно нажать клавиши Backspace или Delete.

Если вырезать выделенный фрагмент (щелчок на кнопке ) , то он исчезнет с экрана компьютера, но будет сохранён в буфере обмена. Из буфера обмена этот фрагмент можно будет вставить в другое место вашего текста. Это очень удобно, если вы захотите поменять местами некоторые строки или абзацы в своём тексте. Если же вы захотите перенести в другое место хотя бы одно предложение в рукописном тексте, то, скорее всего, вам придётся заново переписать весь текст!



Переместить фрагмент текста в другое место можно и без буфера обмена. Достаточно:

- 1) выделить фрагмент;
- 2) нажав и удерживая нажатой правую кнопку мыши, перемещать указатель мыши к новому месту расположения фрагмента;
- 3) когда указатель мыши достигнет нового места расположения фрагмента, отпустить правую кнопку мыши.

Важную роль в текстовом редакторе играет клавиша Enter. Она вставляет в текст невидимый символ конца абзаца.

Клавишу Enter нажимают, чтобы начать печатать текст с нового абзаца или с новой строки. С её помощью можно разбить текст на абзацы, а один абзац — на несколько.

Если нажать клавишу Enter в середине строки, то строка будет «разрезана» на две.

Если удалить символ конца абзаца (клавиши Backspace или Delete), два абзаца «склеятся».



Самое главное

При редактировании вы можете не только исправить обнаруженные ошибки (например, в правописании), но и внести в текст следующие изменения:

- удалить из текста отдельные слова, предложения или абзацы;
- перенести предложение или абзац из одной части текста в другую;
- разбить текст на абзацы или соединить несколько абзацев в один.



Вопросы и задания

1. Как можно преобразовать текст на этапе его редактирования?
2. Какие способы «перемещения» по большому текстовому документу вам известны?
3. Какой символ вставляет в текст клавиша Enter?
 - а) Конец слова
 - б) Конец абзаца
 - в) Конец страницы
 - г) Конец предложения
 - д) Конец главы
 - е) Конец текста
4. Всем знакомо четверостишие Агнии Барто «Мишка». Открыв в текстовом редакторе файл с этим стихотворением, вы увидели на экране слегка «повреждённый» текст стихотворения.

Уронили
мышку на пол,
Оторвали
мышке лапу. Все равно
его не брошу —
Потому
что он хороший.

- а) Какие повреждения произошли с текстом стихотворения?
- б) С помощью каких клавиш можно отредактировать текст стихотворения?
- в) Опишите последовательность своих действий.

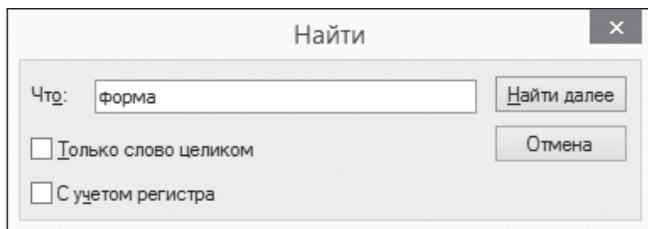
§ 19. Поиск и замена слова

Текстовый редактор WordPad имеет инструмент для поиска слов в документе. Чтобы вызвать этот инструмент, щёлкните на кнопке  Поиск .

В открывшемся диалоговом окне *Найти* надо ввести слово, которое разыскивается.

Дополнительные флажки позволяют более точно задать условия поиска:

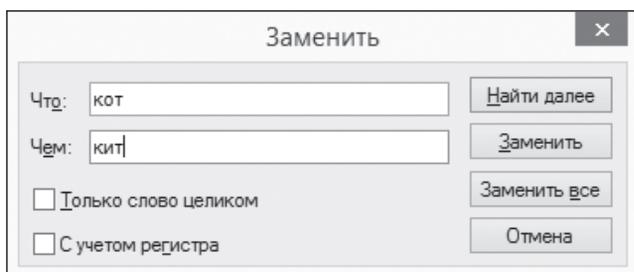
- установка флажка *Только слово целиком* отменяет поиск слова «внутри» более длинных слов;
- установка флажка *С учетом регистра* позволит найти слова с полным совпадением как строчных, так и прописных букв.



Начинается поиск после щелчка на кнопке *Найти далее*. Программа находит первый подходящий фрагмент и выделяет его цветом.

Возможность автоматического поиска слова в большом тексте позволяет человеку быстро находить требуемую информацию, а значит, быстро отвечать на вопросы и принимать решения.

Слово можно не только найти, но и автоматически заменить его во всём тексте другим словом. Для этой цели служит инструмент .



В поле *Что* вводится заменяемое слово — то слово, которое вы хотите заменить. В поле *Чем* вводится заменяющее слово — то слово, которое должно появиться в тексте вместо заменяемого слова. Замена первого подходящего слова будет сделана после щелчка на кнопке *Заменить*. Для автоматической замены слова во всём документе нужно щёлкнуть на кнопке *Заменить все*.

Самое главное

В текстовых редакторах есть инструменты, позволяющие автоматически найти слово, а также автоматически заменить его во всём тексте на другое слово.

Вопросы и задания

1. Витя и Маша с помощью инструмента *Поиск* подсчитали, сколько раз фрагмент КОЛ встречается во фразе ОКОЛО КОЛОКОЛА КОЛ. У Вити получилось, что фрагмент КОЛ встречается в указанной фразе четыре раза, а у Маши — только один раз. Как вы можете объяснить такое расхождение в ответах ребят?
2. В какой ситуации вам может пригодиться быстрый поиск конкретного слова в текстовом документе?
3. Дан текст:

Он, словно слон в посудной лавке, неповоротлив и смешон и, окончательно смутившись, нам всем представился: «Антон».

Сколько замен будет сделано в этом тексте при выполнении автоматической замены фрагмента «он» на «ОН» с учетом и без учета регистра?

4. Как вы думаете, что произойдёт в результате выполнения команды *Заменить все*, если в поле *Что* диалогового окна *Заменить* указать заменяемое слово (например, «лето»), а поле *Чем* оставить пустым?

5. Выполните следующее задание в текстовом редакторе WordPad:

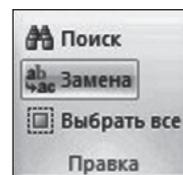
Откройте документ *Медвежонок.rtf*, находящийся в вашей личной папке.

Последовательно выделяйте в тексте каждое слово «медвежонок» и вводите вместо него слово «волчонок» (с учётом строчных и прописных букв).

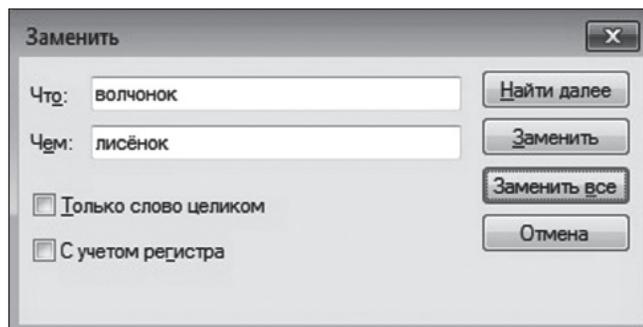
Сохраните изменённый документ под именем *Волчонок.rtf* в личной папке.

Установите курсор в начало первой строки.

Выполните команду *Правка* → *Замена*.



В поле *Найти* введите слово «волчонок». В поле *Заменить на* введите слово «лисёнок». Щёлкните на кнопке *Заменить все*. Закройте окно *Найти и заменить*. Убедитесь, что в тексте вместо слова «волчонок» везде стоит слово «лисёнок».



Сохраните изменённый документ под именем *Лисёнок* в личной папке.

§ 20. Форматирование текста

Когда текст набран и отредактирован, самое время приступить к его оформлению. Этот этап подготовки текстового документа называют *форматированием*.

Редактор WordPad имеет специальную *панель форматирования*, с помощью которой очень просто придать документу требуемый вид. Все инструменты собраны на этой панели в две группы: *Шрифт* и *Абзац*. Рассмотрим их подробнее.



В группу *Шрифт* включены элементы, позволяющие выбрать тот или иной шрифт и его цвет, установить размер букв и их начертание.

Сегодня существует несколько тысяч различных компьютерных шрифтов. Все их можно разделить на четыре основные группы: шрифты с засечками, рубленые, каллиграфические (рукописные) и декоративные.

Если внимательно рассмотреть текст этого параграфа, то можно заметить, что на концах букв имеются маленькие чёрточки — засечки. При чтении за них «цепляется» глаз. Засечки облегчают зрительное восприятие букв, а значит — процесс чтения. Шрифты с засечками применяются в книгах, учебниках и других изданиях с большой длиной строки.

В журналах и газетах, где текст располагается в несколько колонок (короткие строки), обычно применяют рубленые шрифты.

Для оформления обложек, открыток, рекламных проспектов используются декоративные и каллиграфические шрифты. Каллиграфические шрифты имитируют почерк человека. Декоративные весьма причудливы и абсолютно не похожи друг на друга.

Важно помнить, что увлекаться употреблением шрифтов двух последних групп не следует — эти шрифты трудно читаются, особенно когда ими набраны целые абзацы текста.

Чтобы посмотреть перечень доступных вам шрифтов, нужно щёлкнуть мышью на небольшом треугольнике справа от поля выбора *Семейства шрифтов*. Откроется список, из которого можно выбрать шрифт по своему вкусу.

Размер (высота) шрифта измеряется в пунктах. Один **пункт** — это очень маленькая единица. В большинстве книг чаще всего используется шрифт размером 12 пунктов.

Все текстовые редакторы позволяют изменять начертание символов.

Нормальный шрифт ничем не выделяется (нормальное начертание).

Ж Полужирный шрифт темнее, он хорошо заметен (полужирное начертание).

К Курсивный шрифт имеет наклон (курсивное начертание).

Ц А так выглядит подчёркнутый шрифт (подчёркнутое начертание).

И Чтобы изменить шрифт, размер, цвет или начертание отдельного слова, предложения, абзаца или всего текста, соответствующий фрагмент нужно выделить. После этого к выделенному фрагменту следует применить те или иные инструменты форматирования.

Изменить расположение всего текста или его отдельных абзацев на странице можно с помощью инструментов группы *Абзац*.

Прежде всего, можно произвести выравнивание текста.

| | |
|---|---|
|  | Когда текст выровнен влево, левая граница абзаца образует прямую линию. При этом все строки имеют одинаковые отступы от края страницы. Данный абзац выровнен влево. |
|  | Когда текст выровнен вправо, правая граница абзаца образует прямую линию. Каждая строка абзаца заканчивается на одном и том же расстоянии от края страницы. Данный абзац выровнен вправо. |

| | |
|---|--|
|  | Выровненный по центру, или центрированный, текст располагается так: с обеих сторон каждой строки ширина свободного пространства одинакова. С обеих сторон края абзаца получаются неровными. Данный абзац выровнен по центру. |
|  | Текст, выровненный по ширине, равномерно распределён между левым и правым краями. Край такого документа ровные и чёткие. Данный абзац выровнен по ширине. |

И Самое главное

В текстовых редакторах есть инструменты, позволяющие придать документу привлекательный внешний вид.

? Вопросы и задания

1. Для чего необходимо осуществлять форматирование текста? Как можно преобразовать текст на этом этапе?
2. Дайте характеристику каждому тексту, указав для него использованную группу шрифта; начертание символов; выравнивание абзацев.
 - а)

*Дорогой Иван Иванович!
Сердечно поздравляем Вас с Днем рождения!
Желаем здоровья, счастья,
успехов в Ваших начинаниях
и исполнения желаний!*

б)

Черепаша
Тортилла не указала дороги из страны дураков.
Буратино бежал куда глаза глядят. За черными деревьями блестели звезды. Над дорогой свешивались скалы. В ущелье лежало облако тумана.

в)

Глядя на экран монитора, вы на самом деле видите тысячи крошечных световых квадратиков. Для наглядности представьте себе экран в виде листа миллиметровки.

3. Сколькими разными способами можно отформатировать текст, если можно использовать два разных шрифта, четыре способа начертания и один вариант выравнивания?

§ 21. Ищем картинку в сети Интернет

Итак, вы научились создавать на компьютере текстовые документы — можете вводить текст, вносить в него изменения и даже красиво его оформлять. Документы можно сделать ещё привлекательнее, если научиться вставлять в них картинки.

Картинки можно создавать самостоятельно, а можно использовать готовые. Множество картинок на разные темы хранится в открытом доступе в сети Интернет.

И **Открытый доступ** — это бесплатный доступ к информационным ресурсам без каких-либо ограничений, ориентированный на любого пользователя сети Интернет.

Каждый рисунок или другой информационный ресурс кем-то создан, он кому-то принадлежит. Некоторые владельцы ресурсов размещают их в Интернете для всеобщего обозрения, но ставят защиту от копирования. Другие — позволяют всем желающим копировать ресурсы и использовать их в тех или иных целях.

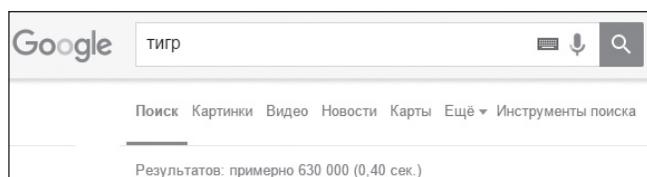
Найденный в сети Интернет рисунок или другой информационный ресурс вы можете использовать в своих личных целях. Например, вы можете распечатать этот рисунок, повесить над своим рабочим столом и любоваться им. А вот говорить друзьям, что этот рисунок нарисован вами, нехорошо, это обман. Если вы, например, будете готовить доклад на какую-то тему по окружающему миру, то обязательно надо будет указать список всех использованных в нём источников — книг, журналов, ресурсов сети Интернет. Это необходимо делать для того, чтобы не нарушать авторские права.

Запустите браузер Chrome. В рабочей зоне браузера вы увидите следующее изображение:



Google — это название компьютерной системы, предназначенной для поиска информации (поисковой системы).

Введите в строку поиска запрос: тигр (можете воспользоваться голосовым вводом запроса). Обратите внимание на результаты поиска:



И **Поисковый запрос** — это слово или набор слов, по которым система будет искать информацию.

Перейдите к картинкам, найденным по этому запросу. Вы увидите множество небольших изображений — это картинки в режиме предварительного просмотра. Среди них будут и рисованные изображения, и фотографии.

Уточните запрос, указав в строке поиска: тигр рисунок. Проследите за изменениями в результатах поиска.

Выберите по своему усмотрению одну из картинок с изображением тигра. Щёлкните на выбранной картинке левой кнопкой мыши, чтобы увидеть её в реальном размере. Сохраните выбранное изображение в своей личной папке:

- 1) Подведите указатель мыши к картинке.
- 2) Нажмите правую кнопку мыши и выберите в контекстном меню команду *Сохранить рисунок как*.

- 3) Сохраните графический файл в нужную папку под именем *Тигр*.

И **Самое главное**

Открытый доступ — это бесплатный доступ к информационным ресурсам без каких-либо ограничений, ориентированный на любого пользователя сети Интернет.

Поисковая система — это компьютерная программа, предназначенная для поиска информации в сети Интернет.

Поисковый запрос — это слово или набор слов, по которым система будет искать информацию.

? **Вопросы и задания**

1. Что такое поисковая система? Какие поисковые системы вам известны?
2. Что такое поисковый запрос? С помощью какого поискового запроса можно искать рисунки с изображением черепахи? С помощью какого поискового запроса можно искать фотографии с изображением кенгуру?
3. Найдите в сети Интернет небольшие рисунки с изображениями рыбы, льва, черепахи, птицы, осы, медведя, зайца, совы, удава и муравья. Сохраните их в личной папке в файлах с именами, соответствующими изображениям.

§ 22. Печать документа

 Создайте в текстовом редакторе WordPad следующий документ:

В начале освоения текстового редактора я боялся, как заяц, и был неповоротлив, как медведь. Задания выполнял медленно, как черепаха.

На протяжении многих дней я был упрям, как осёл, и трудолюбив, как муравей.

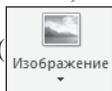
Теперь, когда я сижу за компьютером, я ощущаю себя свободно, как птица в небе. Я ориентируюсь в клавишах, как рыба в воде. Я смел в выборе пунктов меню, как лев.

Сегодня я спокоен, как удав, и мудр, как сова.

Сохраните файл в личной папке под именем *Рассказ*.

Отредактируйте документ, удалив из него слова: заяц, медведь, черепаха, осёл, муравей, птица, рыба, лев, удав, сова и вставьте вместо них соответствующие картинки, которые вы нашли в Интернете и сохранили в своей личной папке.

Для этого:

- 1) выделите слово «заяц» (это можно сделать двойным щелчком на слове);
- 2) удалите выделенное слово клавишей Delete;
- 3) щёлкните на кнопке *Изображение* ();

в открывшемся меню выберите команду *Изображение*;

- 4) перейдите в свою личную папку и выберите в ней файл с именем *Заяц*;

- 5) щелчком на кнопке *Открыть* вставьте картинку на место соответствующего слова;
- 6) щёлкните на появившейся картинке — вокруг неё появится рамка с маркерами изменения размера;



- 7) с помощью протягивания маркеров установите нужные размеры картинке;
- 8) повторите операции 1–7 для всех остальных картинок.

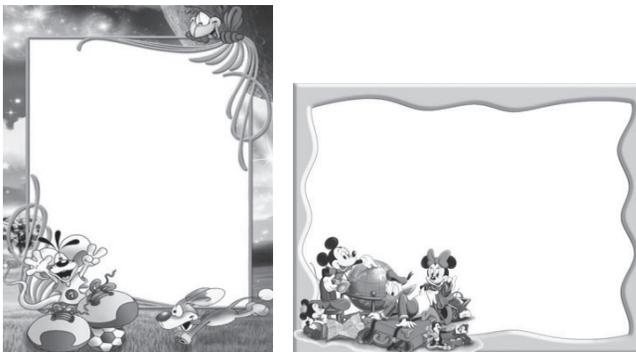
Итак, мы создали красивый документ, который хотелось бы не просто сохранить на компьютере (кстати, не забудьте сохранить сделанные изменения), но и распечатать на бумаге.

Для вывода документа на бумагу к компьютеру подключается печатающее устройство — **принтер**. Как правило, принтеры печатают на стандартных листах бумаги размером 21 см × 30 см. Такой размер называют форматом А4.

Перед печатью важно понять, как наш документ будет расположен на листе бумаги. Для этого можно воспользоваться возможностью предварительного просмотра (*Файл → Печать → Предварительный просмотр*).

Скорее всего, вам не понравится то, что вы увидите. Вы можете вернуться, закрыв окно предварительного просмотра, и попробовать изменить ситуацию.

Прежде всего воспользуйтесь диалоговым окном *Параметры страницы* (*Файл → Параметры страницы*). В нём вы можете установить *Книжную* или *Альбомную* ориентацию страницы.



Расстояния между краями страницы и текстом называют **полями**. Для настройки их размеров служит группа элементов управления *Поля*.

Вы можете по своему вкусу изменить параметры страницы, размер шрифта и выравнивание абзацев текста и ещё раз воспользоваться возможностью предварительного просмотра документа.

Для того чтобы напечатать документ на бумаге, надо убедиться, что компьютер подключён к принтеру, а принтер включён. Теперь можно отдать команду *Файл → Печать*.

i Самое главное

Для вывода документа на бумагу к компьютеру подключается печатающее устройство — принтер.

Для того чтобы понять, как наш документ будет расположен на листе бумаги, пользуются возможностью предварительного просмотра.

В окне *Параметры страницы* можно выбрать ориентацию страницы и настроить размеры полей.

? Вопросы и задания

1. Как называется устройство для вывода документа на бумагу?
2. Для чего следует перед печатью выполнять предварительный просмотр документа?
3. Какие параметры страницы вы можете изменять?
4. Какой способ создания текста — компьютерный или рукописный — вам нравится больше? Свой выбор обоснуйте.
5. Одно из значений слова «редактор» — лицо, исправляющее рукопись с согласия автора. Попытайтесь на основе этой информации объяснить, почему компьютерные программы для создания текстов называли текстовыми редакторами.

§ 23. Звуковая информация и устройства для работы с ней

i Всё, что мы слышим — речь, музыка, пение птиц, шелест листьев, сигналы машин и многое другое, — это **звуковая информация**.

Звуковую информацию можно представить разными способами. Например, созданную мелодию композитор записывает на нотном стане с помощью особых знаков — нот. Вариации в изображении ноты, а также её сочетания с другими символами позволяют задать высоту и длительность звука, порядок его исполнения по отношению к другим звукам.

ПУСТЬ ВСЕГДА БУДЕТ СОЛНЦЕ!

Слова Л. ОЩАНИНА

Музыка А. ОСТРОВСКОГО

Спокойно. В умеренном темпе



Для того чтобы понимать нотные знаки и пользоваться ими, надо владеть нотной грамотой. Как правило, нотной грамотой владеют те, кто учится в музыкальной школе.

Без специальной подготовки воспользоваться звуковой информацией, представленной с помощью нот, невозможно.

Но каждый современный человек может воспользоваться звуковой информацией, представленной в форме **аудиозаписи**.

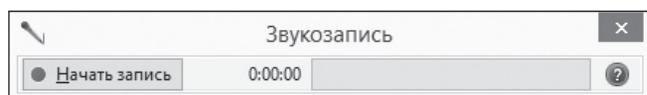
Для работы с аудиозаписями на компьютере используются устройства:

- микрофон для ввода (записи) звуков в память компьютера;
- акустические колонки или наушники для вывода звука и его прослушивания;
- гарнитура (микрофон с наушниками) может использоваться для записи и прослушивания звуковой информации, но особенно удобна для осуществления переговоров.

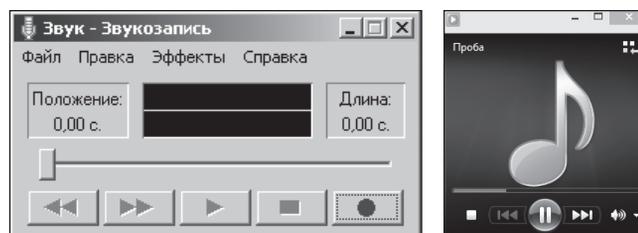


Скорее всего, вы уже встречались с **аудиокнигами** — аудиозаписями текстов литературных произведений, которые читают известные артисты. Такие аудиокниги можно найти в сети Интернет. Многие из них доступны для бесплатного прослушивания в режиме онлайн, т. е. при подключении к сети Интернет и непосредственном запуске с того сайта, где они размещены. Также в сети Интернет можно найти аудиозаписи музыкальных произведений — классической музыки или современных мелодий. Их можно просто слушать, а можно использовать при подготовке собственных выступлений.

На каждом компьютере обязательно есть программа звукозаписи. Она может быть совсем простой, позволяющей только записывать звук. Окно такой программы перед вами:



На вашем компьютере может быть установлена другая программа звукозаписи, позволяющая не только записать, но и обработать звуковую информацию — обрезать звукозапись, соединить несколько звукозаписей, можно замедлить или ускорить звучание аудиозаписи, повернуть её «задом наперёд». С этими и другими возможностями вы сможете познакомиться, изучив соответствующие меню в строке меню.



Возьмите микрофон, убедитесь, что он подключён к компьютеру. Запустите имеющуюся в вашем распоряжении программу звукозаписи, щёлкните на кнопке **Начать запись** и начинайте говорить в микрофон. Можете рассказать что-нибудь о себе или прочесть небольшое стихотворение. Закончив речь, нажмите на кнопку **Остановить запись**. Выполнив звукозапись, сохраните её в памяти компьютера.

На любом компьютере можно воспроизвести (прослушать) звуковой файл. После выполнения двойного щелчка на значке звукового файла начинается его воспроизведение.



Самое главное

Всё, что мы слышим — речь, музыка, пение птиц, шелест листьев, сигналы машин и многое другое, — это звуковая информация.

Звуковая информация может быть представлена в форме аудиозаписей. Для работы с аудиозаписями на компьютере используются микрофон, колонки или наушники, а также гарнитура. Специальные программы позволяют создавать, обрабатывать и воспроизводить аудиозаписи на компьютере.



Вопросы и задания

1. Какая информация называется звуковой?
2. Как может быть представлена звуковая информация?
3. Приведите примеры устройств, в которых используется звуковая информация.
4. Расскажите, как можно записать свою речь на компьютере.
5. На сайте <http://audiohrestomatiya.ru/> найдите и прослушайте аудиозапись басни И. А. Крылова «Слон и Моська».

Окончание следует



Л. А. Гусева,
лицей № 82, Нижний Новгород

УРОК НА ТЕМУ «ТЕХНИКА БЕЗОПАСНОСТИ И ПРАВИЛА ПОВЕДЕНИЯ В КОМПЬЮТЕРНОМ КЛАССЕ»*

Аннотация

В статье представлена методическая разработка урока информатики, который построен таким образом, что учащиеся самостоятельно называют известные им правила техники безопасности, а затем на основе актуализированных знаний в ходе совместной работы создают бабл-комикс (презентацию в Google Презентациях), рассказывающий о правилах поведения в компьютерном классе.

Ключевые слова: техника безопасности, бабл-комикс, правила поведения в компьютерном классе, творчество, Google Презентации.

Контактная информация

Гусева Людмила Александровна,
учитель информатики лицея № 82, Нижний Новгород; адрес: 603003, г. Нижний Новгород, ул. Культуры, д. 1; телефон: (831) 225-42-26; e-mail: lguseva87@gmail.com

L. A. Guseva,
Lyceum 82, Nizhny Novgorod

LESSON ON THE THEME "SAFETY RULES AND RULES OF CONDUCT IN A COMPUTER CLASS"

Abstract

The article describes the informatics lesson on which students call the safety rules and then on the basis of updated knowledge together create a bubble-comics (the presentation in Google Presentations), telling about the rules of conduct in a computer class.

Keywords: safety rules, bubble-comics, rules of conduct in computer class, creativity, Google Presentations.

Урок на тему «Техника безопасности и правила поведения в компьютерном классе» построен таким образом, что учащиеся самостоятельно называют известные им правила, а затем на основе актуализированных знаний в ходе совместной работы создают бабл-комикс, рассказывающий о правилах поведения в компьютерном классе.

Тема урока: Техника безопасности и правила поведения в компьютерном классе.

Цель урока: закрепить знания о технике безопасности и правилах поведения в компьютерном классе.

Задачи урока:

образовательные:

- повторение и закрепление знаний о технике безопасности и правилах поведения в компьютерном классе;
- формирование умения обобщать информацию, представлять ее в различной форме с использованием ИКТ;

развивающие:

- формирование ответственного и бережного отношения к технике;
- формирование навыка ответственного отношения к собственному здоровью и безопасности;
- формирование умения планировать свою деятельность;
- формирование умения ставить цели своей деятельности;
- формирование навыков рационального использования времени;
- развитие творческих способностей;
- развитие коммуникативных навыков;
- развитие способности к совместному творчеству;
- развитие мыслительных навыков;

воспитательная:

- воспитание навыков самооценки, самоконтроля.

УМК: Босова Л. Л., Босова А. Ю. Информатика. 7 класс («БИНОМ. Лаборатория знаний»).

Тип урока: урок повторения и закрепления.

Вид урока: урок проектной деятельности.

Класс: седьмой.

Планируемые результаты обучения: учащиеся:

- будут знать правила техники безопасности и правила поведения в компьютерном классе, осознавать важность их соблюдения;
- приобретут навыки планирования собственной деятельности, совместной работы.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

Опорные знания:

- знание правил техники безопасности и правил поведения в компьютерном классе;
- умение создавать презентации в Microsoft PowerPoint и Google Презентациях.

Литература, использованная при подготовке урока.

Босова Л. Л., Босова А. Ю. Информатика: учебник для 7 класса. М.: БИНОМ. Лаборатория знаний, 2015.

Источники изображений, использованных в презентации «Физкультминутка».

- Баблы. <http://nyaski.ru/pages/speech-bubbles>
- Карандаш. <https://lh3.googleusercontent.com/8ILMebx2WjncTm8iNQZbvogFRWv9ZYvxDIDukEIMoheTE5Za7USUIWXOUkN0jmkdZoKF=s85>
- Бабочка. https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcS1XoasMpb_JB9nPaX_gRbTkaLY80GEJ9DG7j-VThuizeTnkf_OBuZOxg
- Футбольный мяч. https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRnbg4a9ndHhFBmdwUhY__SEZWedkcp27t7ILX3BxcYV8dDs8gM
- Бабочка на цветке. <http://animashki.kak2z.org/pic/7/babochkia-13.gif>

Оснащение урока:

- компьютеры с доступом в Интернет;
- фотоаппарат или телефон с функцией фотокамеры;
- кабели для подключения фотоаппарата или телефона к компьютеру;
- мультимедийный проектор;
- маркеры;
- заранее заготовленные баблы.

Предварительная подготовка к уроку.

Для работы над совместной презентацией необходимо создать учетные записи Google — по количеству компьютеров в классе, включая учительский. До урока необходимо войти с каждого компьютера в соответствующую учетную запись. С учительского компьютера в Google Диск надо создать пустую презентацию, оформить ее титульный слайд. При необходимости можно работать локально — с презентацией MS PowerPoint.

Для создания бабл-комикса необходимо заготовить баблы — «пузыри» (размером в лист бумаги формата А4), в которых пишутся слова и мысли героев комиксов:



Баблы можно распечатать на плотной бумаге или нарисовать самим, вырезать и прикрепить к держателю (например, к палочке или трубочке для коктейля). В кадре бабл помещают возле головы героя — либо закрепляют, либо держат.

**План урока.**

1. Мотивация к учебной деятельности — 1 мин.
2. Актуализация знаний — 8 мин.
3. Планирование деятельности — 5 мин.
4. Работа в группах — 15 мин.
5. Физкультминутка — 1 мин.
6. Просмотр готовой презентации — 10 мин.
7. Рефлексия учебной деятельности — 4 мин.
8. Домашнее задание — 1 мин.

Ход урока**1. Мотивация к учебной деятельности**

Учитель. Здравствуйте! Я рада снова встретиться с вами после долгого перерыва. Как вы провели лето? (*Ответы учеников.*)

Никаких контрольных работ, домашних заданий, оценок, можно спать и гулять сколько угодно... Недаром американский писатель О. Генри, автор замечательных юмористических новелл, заметил однажды, что лето — это пора безответственности. Но наступила осень, и настало время снова вернуться в школу — чтобы становиться умнее, меняться к лучшему, открывать новые горизонты.

2. Актуализация знаний

Учитель. Тема нашего первого урока традиционно связана с ответственностью. (*Ребята вспоминают, что первый урок курса информатики традиционно посвящается технике безопасности.*)

Как же нужно вести себя в компьютерном классе? (*Учащиеся вспоминают правила техники безопасности. Учитель записывает соответствующие правила в презентации на интерактивной доске — рисунок 1.*)

- Входить в компьютерный класс можно только с разрешения учителя.
- Нельзя входить в компьютерный класс в верхней одежде и без сменной обуви.
- Спину держим прямо, расстояние от глаз до компьютера — вытянутая рука.
- Нельзя самостоятельно пытаться устранить неисправность. При появлении запаха гари сообщить учителю.
- Нельзя трогать провода.
- Нельзя прикасаться руками к экрану, тыльной части монитора и системного блока.
- Не нужно сильно стучать по клавишам.
- Нельзя есть и пить за компьютером.
- Необходимо выполнять все указания учителя.
- Продолжительность работы за компьютером в седьмом классе — 20–25 минут.

Вы уже сами умеете делать презентации. Скажите, вам понравилась презентация, которая у нас получилась? (*Ученики отмечают достоинства получившейся презентации, высказывают замечания.*)

Вы правильно отметили, что можно было бы добавить изображения на слайды — и правила сразу стали бы более наглядными.

Можно было бы добавить картинки, фотографии, иллюстрирующие, как надо и как не надо поступать, находясь в компьютерном классе.

Можно было бы нарисовать целую историю в комиксе!

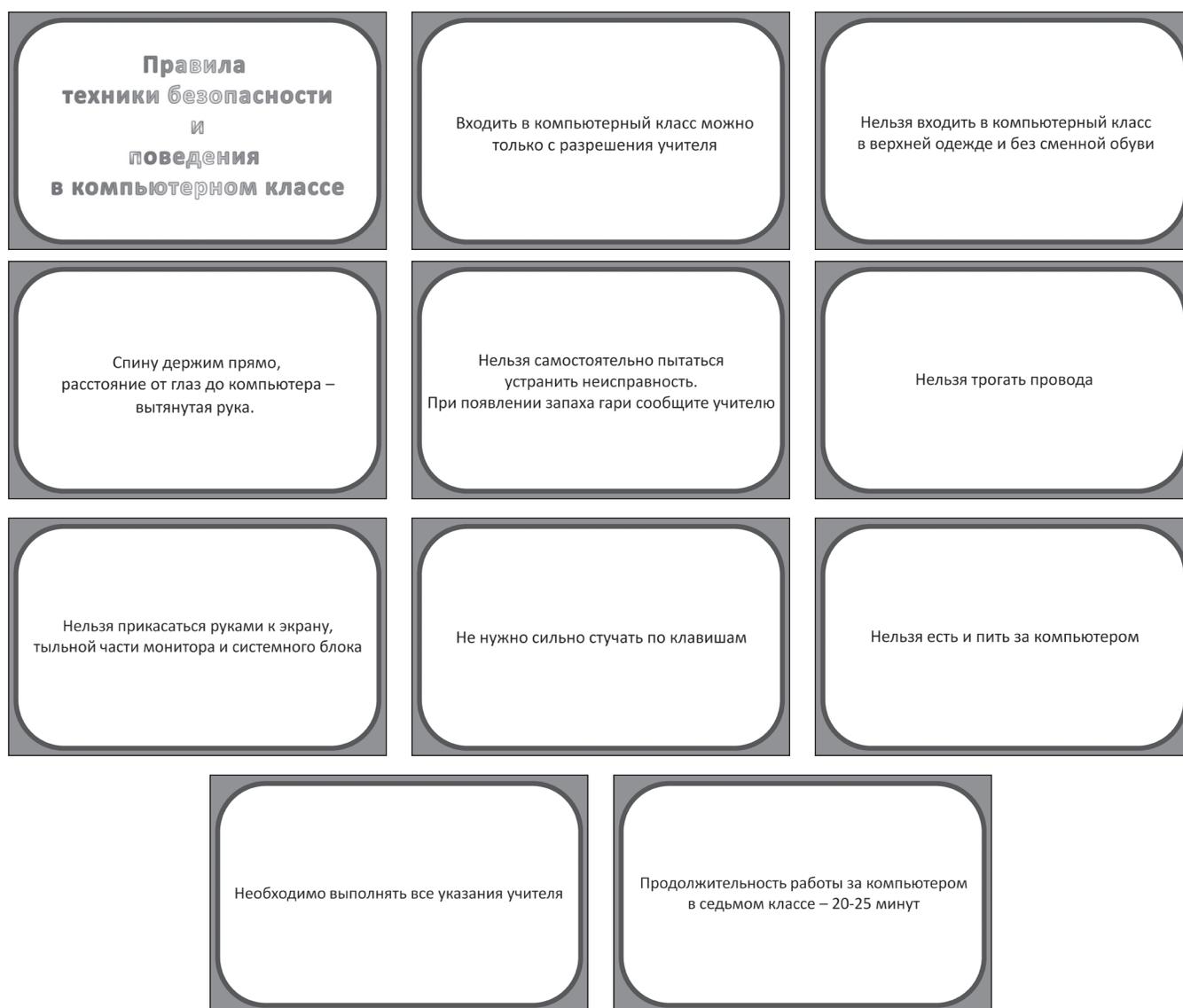


Рис. 1. Пример получившейся презентации «Техника безопасности»

Я уверена, что у вас все может получиться гораздо лучше, чем у меня, и в следующий раз мы будем смотреть не скучную презентацию, а захватывающий ролик — может быть, с вами в главной роли!

3. Планирование деятельности

Учитель. Итак, мы сегодня будем создавать фото-бабл-комикс.

Как мы сегодня будем работать: индивидуально или в группах? (*Учащиеся делятся на группы по два-три человека.*)

План наших действий таков. Правила техники безопасности вы вспомнили — у нас получилось ... правил. Теперь **ваша задача — создать иллюстрированную историю о том, как нужно вести себя в компьютерном классе.**

Каждая группа получает по три правила, которые она должна проиллюстрировать с помощью фотографий. Важное условие — на фотографии должен присутствовать бабл, на котором вы пишете правило, мысли героя или от автора: например, слово «Нельзя!» или «Даже не думай!» (*Учитель демонстрирует примеры баблов. Каждая группа получает набор баблов и маркеры.*)

Вы можете использовать любые имеющиеся у нас в классе средства — компьютеры, камеры на телефонах,

все, что угодно. Не стесняйтесь обращаться за помощью к ребятам из других групп и ко мне.

После того как материал будет собран, вы должны:

- добавить свои фотографии на соответствующие слайды нашей презентации;
- убрать ненужный текст;
- оформить слайд.

(*Для такой работы используются учетные записи Google, созданные учителем для каждого компьютера.*)

4. Работа в группах

Учащиеся в группах работают по следующему плану:

1. Разработка сценария — как именно должно быть проиллюстрировано конкретное правило. Распределение ролей внутри групп.
2. Съёмка.
3. Работа над презентацией.

5. Физкультминутка

Учитель. Сейчас самое время вспомнить о том, что при работе за компьютером наши глаза устают и нужно время от времени делать для них специальную гимнастику. Запустите презентацию и выполняйте упражнения.

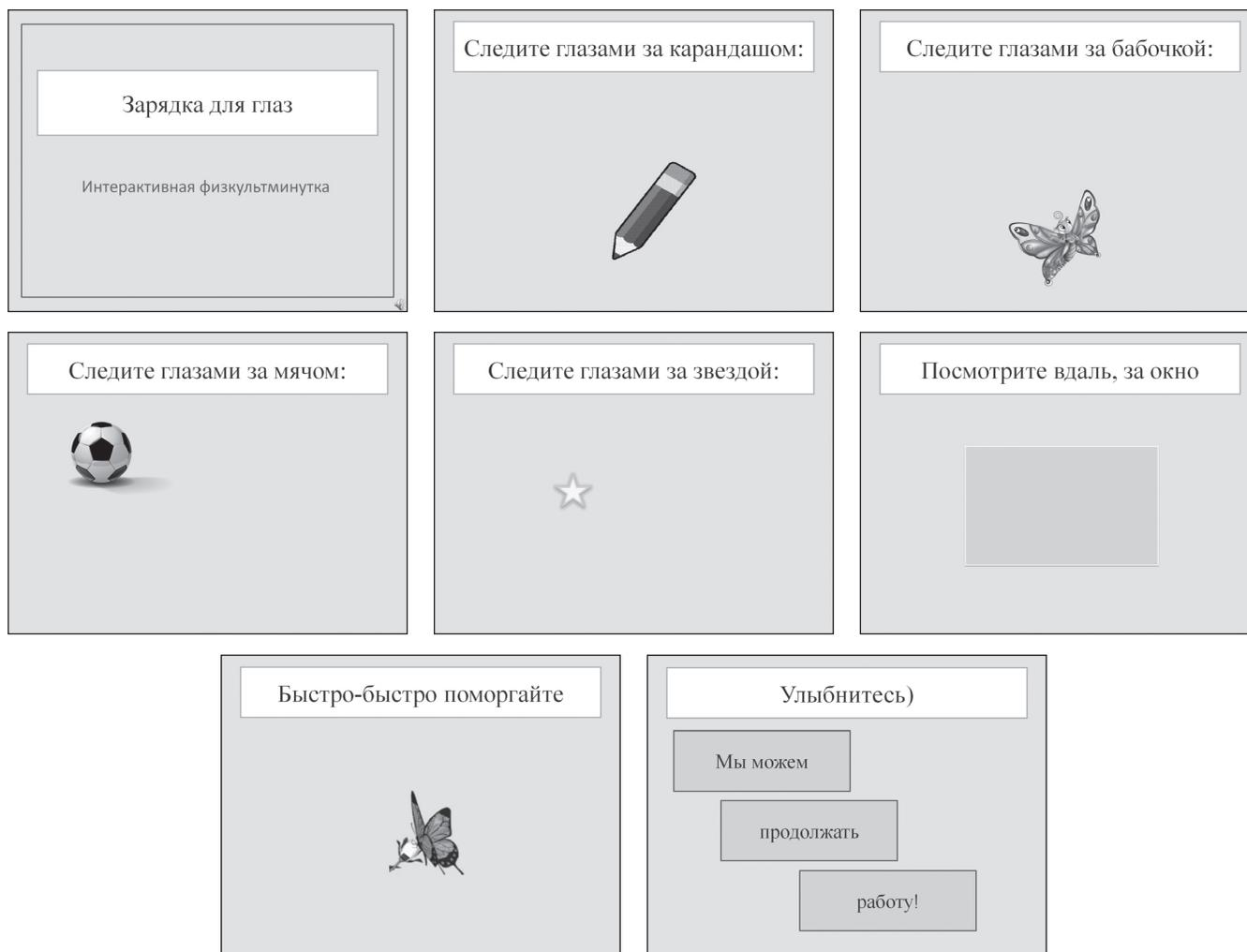


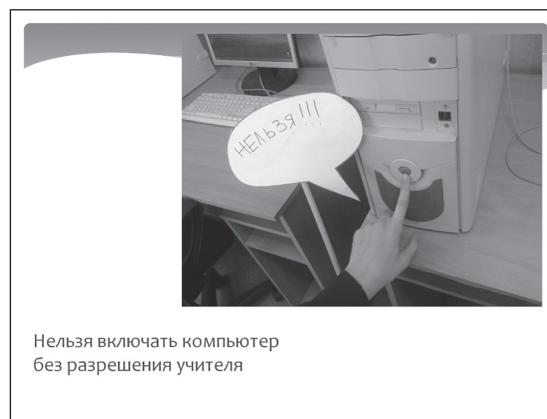
Рис. 2. Интерактивная презентация «Физкультминутка»

Учащиеся запускают на своих компьютерах презентацию для выполнения физкультминутки, на которой им надо глазами отслеживать движение карандаша, бабочки, мяча, звездочки и т. д. (рис. 2).

6. Просмотр готовой презентации

Учащиеся возвращаются за парты, учитель демонстрирует результат работы — полученную общими усилиями презентацию о технике безопасности. Проводится обсуждение полученной презентации.

Примеры слайдов презентации, созданных учащимися:



7. Рефлексия учебной деятельности

Учитель.

- Что нового вы узнали на уроке?
- Что показалось интересным?
- Что вы бы сделали по-другому?
- Если бы вам дали возможность провести такой урок, что бы мы сегодня делали?

8. Домашнее задание

Задание. Повторить правила техники безопасности.

А. В. Бабичев,
школа № 626 имени Н. И. Сац, г. Москва

ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКАЯ ДЕЯТЕЛЬНОСТЬ НА ТЕМУ «ДОМ, В КОТОРОМ МЫ ЖИВЕМ»*

Аннотация

В статье представлена методическая разработка урока информатики, на котором в ходе реализации проекта учащиеся знакомятся с деятельностью ЖКХ, вопросами водо- и энергосбережения, выполняют задания в группах, а затем представляют результаты своей работы с помощью презентаций, подготовленных в Microsoft PowerPoint.

Ключевые слова: проектно-исследовательская деятельность, проект, презентация, Microsoft PowerPoint, ЖКХ.

Контактная информация

Бабичев Александр Владимирович,
учитель информатики школы № 626 имени Н. И. Сац, г. Москва; адрес: 117186, г. Москва, ул. Нагорная, д. 22, корп. 1; телефоны: (499) 127-66-14; (499) 123-31-51; e-mail: babicheff.alexander@yandex.ru

A. V. Babichev,
School 626, Moscow

PROJECT AND RESEARCH ACTIVITY ON THE THEME "THE HOUSE IN WHICH WE LIVE"

Abstract

The article describes the informatics lesson at which during the developing of the project students get acquainted with the activities of housing and communal services, water and energy conservation issues, carry out tasks in groups and then present the results of their work in the presentation prepared in Microsoft PowerPoint.

Keywords: project and research activity, project, presentation, Microsoft PowerPoint, housing and communal services.

Тема урока: Создание презентации «Дом, в котором мы живем».

Цель урока: развитие умений и навыков применения ИКТ в повседневной жизни.

Задачи урока:

образовательные:

- закрепление навыков работы с программой презентационной графики Microsoft PowerPoint;
- выработка умений и навыков создания и оформления компьютерной презентации;
- освоение навыков представления своей проектной деятельности;
- формирование навыков ориентирования в финансовой математике и экономической политике;

развивающие:

- развитие навыков проектно-исследовательской деятельности;
- развитие информационной культуры;
- развитие логического мышления;
- развитие познавательного интереса;
- расширение кругозора;

воспитательные:

- воспитание эстетических навыков в оформлении презентации;
- воспитание чувства коллективизма при работе в группах;
- воспитание умения отстаивать свою точку зрения;
- формирование социально активной позиции;
- формирование бережливости в использовании водных и энергетических ресурсов.

Тип урока: урок закрепления изученного материала и выработки практических умений и навыков.

Вид урока: урок-практикум.

Возраст учащихся: V—VII классы.

Методы: проблемный, поисковый, метод творческих заданий.

Формы организации познавательной деятельности: групповая — проектно-исследовательская деятельность с использованием ИКТ.

Оснащение урока:

- персональные компьютеры с ОС Microsoft Windows и выходом в Интернет;
- Microsoft Office;
- мультимедийный проектор;
- интерактивная доска SMART;
- файл с коллекцией ссылок на сайты по теме урока;
- папка с коллекцией фотографий и рисунков для создания презентации и файл со ссылками на источники этих изображений;
- раздаточный материал:
 - текстовый материал по теме урока (файл *Приложение 1.doc*);
 - задания для групповой работы (файл *Приложение 2.doc*);
 - материалы сайта Департамента экономической политики и развития города Москвы: <http://depr.mos.ru/> (файл *Приложение 3.doc*);
 - практико-ориентированное задание по созданию презентации (файл *Приложение 4.doc*).

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

Литературные и интернет-источники, использованные при подготовке урока.

1. Босова Л. Л., Босова А. Ю. Информатика: учебник для 5 класса. М.: БИНОМ. Лаборатория знаний, 2013.
2. Босова Л. Л., Босова А. Ю. Информатика: учебник для 6 класса. М.: БИНОМ. Лаборатория знаний, 2013.
3. Босова Л. Л., Босова А. Ю. Уроки информатики в 5–7 классах: методическое пособие. М.: БИНОМ. Лаборатория знаний, 2012.
4. Официальный сайт Департамента экономической политики и развития города Москвы. <http://depr.mos.ru/>

План урока.

1. Организационный момент — 3 мин.
2. Формулировка темы и целей урока — 3 мин.
3. Практическая групповая работа — выполнение заданий — 4 мин.
4. Творческая групповая работа — создание презентации — 15 мин.
5. Защита проектов — 15 мин.
6. Подведение итогов урока. Рефлексия — 5 мин.

Ход урока

1. Организационный момент

Формируемые УУД:

- **личностные:** психологическая готовность к уроку;
- **коммуникативные:** планирование учебного сотрудничества со сверстниками.

Учитель приветствует учеников, проверяет их готовность к уроку. Учащиеся оценивают собственную готовность к уроку.

2. Формулировка темы и целей урока

Формируемые УУД:

- **личностные:** способность увязать учебное содержание с собственным жизненным опытом;
- **познавательные:** выделение и формулирование познавательной цели;

- **коммуникативные:** умение слушать собеседника; умение выражать свои мысли;
- **регулятивные:** целеполагание.

Каждый ученик получает лист с текстовым материалом, содержащим сведения об управлении многоквартирным домом и структуре платы за жилищно-коммунальные услуги (*Приложение 1*), знакомится с текстом.

Педагог обращает особое внимание на раздел, касающийся платы за коммунальные услуги (водоснабжение, электроснабжение, водоотведение, отопление, газоснабжение), объясняет, что вопросы оплаты коммунальных услуг касаются всех без исключения. Поэтому каждый должен уметь рассчитать размер платы за коммунальные услуги.

В ходе обсуждения прочитанного учащиеся отвечают на вопросы учителя, участвуют в целеполагании. Педагог вместе со школьниками формулирует тему урока и его цели.

3. Практическая групповая работа — выполнение заданий

Формируемые УУД:

- **познавательные:** структурирование знаний; рефлексия способов и условий действий; контроль и оценка процесса и результатов деятельности;
- **коммуникативные:** ориентация на партнера по общению; умение слушать собеседника; умение аргументировать свое мнение, убеждать и уступать;
- **регулятивные:** формирование собственных целей при создании проекта и выборе темы.

Для выполнения проекта класс делится на две группы. Каждая группа получает рабочий лист с заданиями (файл *Приложение 2.doc*), выполняет их и записывает на листе результаты выполнения. В качестве справочного материала учащиеся получают распечатки с материалами сайта Департамента экономической политики и развития города Москвы: <http://depr.mos.ru/>, а именно, копии экрана работы с калькулятором коммунальных платежей (файл *Приложение 3.doc*).

DEПАРТАМЕНТ ЭКОНОМИЧЕСКОЙ ПОЛИТИКИ И РАЗВИТИЯ ГОРОДА МОСКВЫ

Калькулятор коммунальных платежей

Информация предоставлена органами исполнительной власти субъектов РФ в сфере государственного регулирования тарифов

Субъект РФ г. Москва
Муниципальный район Москва
Муниципальное образование Москва (ОКТМО: 45000000)
Период Апрель 2016

ВЫБРАТЬ ДРУГОЕ

Условия проживания

Общая площадь, кв. м. 52

МНОГOKВАРТИРНЫЙ ДОМ ЧАСТНЫЙ ДОМ

Количество проживающих (зарегистрированных) граждан 3

Размер платежа граждан за каждую коммунальную услугу складывается из тарифа, утверждаемого на коммунальный ресурс для конкретной ресурсоснабжающей организации и объемов потребления, определяемых посредством приборов учета, а при отсутствии приборов учета, исходя из нормативов потребления.

Коммунальные услуги

Водоотведение

город начинается здесь

Рис. 1. Калькулятор коммунальных платежей на сайте Департамента экономической политики и развития города Москвы

Задания для группы 1.

1. Рассчитайте стоимость услуг ЖКХ (горячего и холодного водоснабжение) с помощью калькулятора коммунальных платежей на сайте Департамента экономической политики и развития города Москвы: <http://depr.mos.ru/>

Для расчета используйте следующие данные:

| | |
|---|---|
| Год | 2016 |
| Месяц | Апрель |
| Общая площадь, кв. м | 52 |
| Кол-во проживающих граждан | 3 |
| Метод расчета | Счетчик |
| Ваш поставщик услуг | АО «Мосводоканал», ОАО «Мосэнерго» |
| Расход по прибору учета за период (месяц) | 7 куб. м холодной воды на 1 чел. 5 куб. м горячей воды на 1 чел. |

2. Сформулируйте правила и нормы, которые помогают экономить электроэнергию дома и в школе.

Задания для группы 2.

1. Рассчитайте стоимость услуг ЖКХ (электроснабжения и водоотведения) с помощью калькулятора коммунальных платежей на сайте Департамента экономической политики и развития города Москвы: <http://depr.mos.ru/>

Для расчета используйте следующие данные:

| | |
|---|---|
| Год | 2016 |
| Месяц | Апрель |
| Общая площадь, кв. м | 52 |
| Кол-во проживающих граждан | 3 |
| Метод расчета | Счетчик |
| Ваш поставщик услуг | АО «Мосводоканал», ОАО «Мосэнерго» |
| Расход по прибору учета за период (месяц) | Водоотведение — 12 куб. м на 1 чел. Электроснабжение — 200 кВт на 1 чел. |
| Ваш вид благоустройства | Электроснабжение в домах с газовыми плитами или электроснабжение в домах с электрическими плитами |
| Тариф (для электроснабжения) | Однотарифный |

2. Сформулируйте правила и нормы, которые помогают экономить воду дома и в школе.

4. Творческая групповая работа — создание презентации**Формируемые УУД:**

- *личностные*: умение организовать свое рабочее время, распределить силы;

- *познавательные*: выбор наиболее эффективных способов выполнения задания; оценивание начальных данных и планируемого результата; умение оформить результаты своей деятельности, представить их на современном уровне;
- *регулятивные*: формирование собственных целей при реализации проекта; умение принимать решение, брать ответственность на себя (например, быть лидером группового проекта, принимать решение в случае нестандартной ситуации).

Группам предлагается следующее задание:

Люди, которые работают в ЖКХ, — настоящие герои нашего времени. Ведь на них возлагается огромная ответственность. Они должны сделать так, чтобы каждый житель, приходя домой, чувствовал себя комфортно и был удовлетворен своим бытом. Изо дня в день в теплой квартире мы включаем телевизор и много бытовой техники и с удовольствием просматриваем любимые программы, но редко кто из нас задумывается, благодаря кому и чему мы всем этим пользуемся.

Что означают цифры на счетчиках? Как проверить правильность выставленных коммунальных счетов? Это лишь некоторые из нюансов основ ЖКХ, которые мы, современные люди, должны знать.

Свет, тепло, воду и газ поставляют в дом коммунальные предприятия: водоканал, электросети, ТЭЦ, котельная. За жилищные услуги в ответе управляющая компания. Ежемесячно большинство семей вынуждены отдавать весомую сумму из семейного бюджета за услуги ЖКХ.

А теперь подумайте: **как за счет современных технологий можно сэкономить на платежах, и что для этого нужно сделать?**

Ответ представьте в виде презентации «Дом, в котором мы живем».

Материалы для презентации учащиеся находят самостоятельно, используя поисковые сервисы Google и «Яндекс». Также они могут воспользоваться материалами, подготовленными учителем: коллекцией ссылок на соответствующие сайты и коллекцией фотографий и рисунков.

Учитель организует практическую работу за компьютерами по реализации проекта, контролирует работу с информацией из Интернета.

По завершении работы все ученики размещают свои файлы в указанном месте (учительский компьютер, собственное рабочее место, облако).

5. Защита проектов**Формируемые УУД:**

- *личностные*: завоевание авторитета в глазах одноклассников с помощью уникальных результатов своей деятельности;
- *коммуникативные*: умение слушать и вступать в диалог, формулировать и аргументировать свое мнение; владение языковой, лингвистической компетенцией;
- *регулятивные*: контроль и оценка своей деятельности в рамках урока.

Каждая группа представляет подготовленную презентацию, участники группы отвечают на вопросы учителя и одноклассников.

Учащиеся оценивают свою презентацию и презентацию, подготовленную другой группой, в соответствии с заранее выработанными критериями.

«Дом, в котором мы живем»

Развитие технологий не стоит на месте и на смену лампам накаливания пришли компактные люминесцентные лампы, то есть лампы дневного света энергосберегающие. На рисунке приведены данные соответствия мощности энергосберегающих ламп лампам накаливания.

ВАЖНО Современная энергосберегающая лампа служит в десять, а то и в пятнадцать раз дольше, чем ее предшественница. Также, отличаются они по светоотдаче, превышающей традиционную в пять раз. Для сравнения: 100 Вт обычной лампы накаливания равноценны 20 Вт энергосберегающей.

Рис. 2. Пример слайда презентации, созданной учащимися

6. Подведение итогов урока. Рефлексия

Формируемые УУД:

- **личностные:** рефлексия способов и условий действия, контроль и оценка процесса и результатов деятельности;
- **познавательные:** построение речевого высказывания в устной форме.

Учащиеся дают самооценку своей деятельности и ее результатов.

Учитель подводит итоги работы учащихся в группах и класса в целом, благодарит учеников за работу.

Приложение 1

Выдержки из нормативных документов

Многоквартирный дом — отдельное жилое здание, состоящее из помещений для постоянного проживания граждан, отвечающих установленным санитарным и техническим правилам и нормам, а также помещений вспомогательного использования, предназначенных для удовлетворения гражданами бытовых и иных нужд, связанных с их проживанием в таком здании, законченное строительством и принятое в экс-

плуатацию в соответствии с установленным законом порядком.

Виды помещений в многоквартирном доме.

Жилым помещением признается изолированное помещение, которое является недвижимым имуществом и пригодно для постоянного проживания гражданина (отвечает установленным санитарным и техническим правилам и нормам, иным требованиям законодательства). То есть квартира, часть квартиры либо комната относятся к жилым помещениям.

Нежилыми помещениями являются помещения, ставшие таковыми в результате их перевода в установленном порядке из жилых либо изначально (при строительстве здания) спланированные и построенные для использования их в административных и других, не связанных с проживанием граждан, целях.

У помещений в любом здании имеется хозяин — собственник, которым может быть и гражданин, и муниципалитет, и государство. Зачастую в многоквартирном доме имеются в наличии разные виды собственников помещений.

Способ управления многоквартирным домом.

Закон обязывает собственников жилья выбрать способ управления многоквартирным домом. Многоквартирный дом не может оставаться без управления. Поэтому если сами собственники не выполняют обязанности по выбору способа управления, то за них такой выбор делает муниципалитет в установленном порядке. Жилищный кодекс РФ (ч. 2 ст. 161) устанавливает следующие способы:

- непосредственное управление собственниками помещений в многоквартирном доме;
- управление товариществом собственников жилья либо жилищным кооперативом или иным специализированным потребительским кооперативом;
- управление управляющей организацией.

Структура платы за жилищно-коммунальные услуги.

Жилищный кодекс РФ разграничивает плату за жилищно-коммунальные услуги на плату:

- за жилое помещение;
- за коммунальные услуги (водоснабжение, электроснабжение, водоотведение, отопление, газоснабжение);
- взнос на капитальный ремонт.

НОВОСТИ

Обучаемые «мобильники»

В компании Qualcomm, обещая снабдить мобильные устройства «кремниевым мозгом», предлагают инструмент, позволяющий обучать смартфоны распознавать людей, предметы, жесты и эмоции. Компания выпустила комплект разработчика Snapdragon Neural Processing Engine для аппаратов на процессоре Snapdragon 820, позволяющий придавать им способности к глубинному обучению.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

SDK также рассчитан на разработку ПО самоуправляемых автомобилей, дронов и роботов, сообщают в Qualcomm. Кроме того, комплект можно будет использовать в средах безопасности, например для организации отслеживания перемещений конкретного человека с помощью камер видеонаблюдения. Возможно применение SDK в медицинских и страховых компаниях, добавляют в Qualcomm.

Е. В. Лейман,
гимназия № 1, г. Новосибирск

УРОК НА ТЕМУ «ПОСТРОЕНИЕ ФИГУР С ЦЕНТРАЛЬНОЙ СИММЕТРИЕЙ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ЛОГО»*

Аннотация

В статье приведена методическая разработка урока для учащихся шестого класса, в котором изучается построение фигур с центральной симметрией на языке программирования Лого на примере создания снежинок. По плану урок проходит в середине декабря и создает праздничную атмосферу в преддверии новогодних праздников.

Ключевые слова: язык программирования Лого, программирование, центральная симметрия, правильные многоугольники, снежинки, цикл.

Контактная информация

Лейман Екатерина Владимировна,
учитель информатики гимназии № 1, г. Новосибирск; адрес: 630091, г. Новосибирск, Красный пр-т, д. 48; телефон: (383) 222-25-62; e-mail: katm@ngs.ru

E. V. Leyman,
Gymnasium 1, Novosibirsk

LESSON ON THE THEME "CONSTRUCTION OF FIGURES WITH CENTRAL SYMMETRY ON LOGO PROGRAMMING LANGUAGE"

Abstract

The article describes the informatics lesson for sixth-grade students on which construction of figures with central symmetry on Logo programming language by creating snowflakes is studied. According to the plan the lesson takes place in mid-December and creates a festive atmosphere on the eve of New Year holidays.

Keywords: Logo programming language, programming, central symmetry, regular polygons, snowflakes, loop.

Урок по теме «Построение фигур с центральной симметрией на языке программирования Лого» является одним из интереснейших в разделе «Программирование в среде Лого». В соответствии с календарно-тематическим планированием урок проходит в середине декабря и создает праздничную атмосферу в преддверии новогодних праздников.

Тема урока: Построение фигур с центральной симметрией на языке программирования Лого.

Цели урока:

- знакомство учащихся с понятием «фигура с центральной симметрией» с точки зрения программирования на языке Лого;
- отработка навыков создания фигур с центральной симметрией;
- развитие творческой активности, познавательного интереса;
- развитие интереса к информатике;
- воспитание самостоятельности в выполнении практических заданий на компьютере.

Тип урока: комбинированный урок.

Класс: шестой.

Оснащение урока:

- компьютер учителя;
- компьютеры учащихся с программой «ЛогоМиры»;
- интерактивная доска;
- презентация по теме урока для интерактивной доски, подготовленная в SMART Notebook;
- презентация «Физкультминутка» (подготовлена в MS PowerPoint);
- раздаточный материал:
 - карточка для этапа повторения пройденного материала;
 - карточка с заданием для этапа объяснения нового материала.

Литература, использованная при подготовке урока.

1. Челак А. А. Практикум по информатике в среде ЛогоМиры. Новосибирск: Изд-во НИПКиПРО, 2007.
2. Юдина А. Г. Практикум по информатике в среде Лого: пособие для учащихся общеобразовательных учреждений. М.: Мнемозина, 2003.

План урока.

1. Организационный момент — 1 мин.
2. Повторение пройденного материала — 5 мин.
3. Объяснение нового материала — 10 мин.
4. Закрепление изученного материала — 5 мин.
5. Физкультминутка — 1 мин.
6. Закрепление изученного материала (продолжение) — 15 мин.
7. Подведение итогов урока. Домашнее задание — 3 мин.

Ход урока

1. Организационный момент

Учитель приветствует учащихся, проверяет их готовность к уроку.

Учитель. Здравствуйте, садитесь. Приготовьтесь, пожалуйста, к уроку. Достаньте карандаш, ручку, транспортир и тетрадь.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

2. Повторение пройденного материала

Урок сопровождается демонстрацией слайдов презентации на интерактивной доске.

Учитель. На предыдущем занятии мы с вами познакомились с понятием «правильный многоугольник» и способом построения такого многоугольника. Итак, что такое правильный многоугольник?

Учащиеся. Геометрическая фигура, у которой все стороны и все углы равны.

Учитель. Давайте разделимся на четыре группы.

Учащиеся рассаживаются по группам.

На интерактивной доске демонстрируется **задание на повторение:**

| Повторение по теме «Правильные многоугольники» | |
|--|-----------------------------|
| 1) Повтори 4 [вп 10 пр 4] | • Правильный треугольник |
| 2) Повтори 5 [вп 50 пр 360 / 5] | • Квадрат |
| 3) Повтори 3 [вп 50 пр 45] | |
| 4) Повтори 4 [вп 30 пр 90] | • Правильный пятиугольник |
| 5) Повтори 7 [вп 50 пр 360 / 7] | • Правильный шестиугольник |
| 6) Повтори 3 [вп 100 пр 120] | • Правильный семиугольник |
| 7) Повтори 8 [вп 50 пр 45] | |
| 8) Повтори 6 [вп 30 пр 60] | • Правильный восьмиугольник |

Каждая группа получает карточку для выполнения задания.

Учитель. Подумайте в группах и найдите соответствие между правой и левой колонками записей. Заполните карточку. На обсуждение и заполнение карточки вам дается две минуты.

| Многоугольник | № ответа | Фамилия, имя участников группы |
|---------------------------|----------|--------------------------------|
| Правильный треугольник | | |
| Квадрат | | |
| Правильный пятиугольник | | |
| Правильный шестиугольник | | |
| Правильный семиугольник | | |
| Правильный восьмиугольник | | |

По истечении двух минут команды сдают заполненные карточки учителю. Представители от команд по очереди выходят к доске и вписывают ответы в таблицу. Правильный ответ демонстрируется на слайде.

Ответ.

| Многоугольник | № ответа |
|---------------------------|----------|
| Правильный треугольник | 6 |
| Квадрат | 4 |
| Правильный пятиугольник | 2 |
| Правильный шестиугольник | 8 |
| Правильный семиугольник | 5 |
| Правильный восьмиугольник | 7 |

Повторение по теме «Правильные многоугольники»

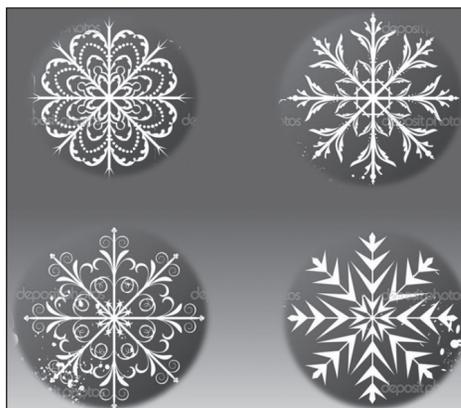
- 1) Повтори 4 [вп 10 пр 4]
 - Правильный треугольник
- 6) Повтори 3 [вп 100 пр 120]
 - Квадрат
- 3) Повтори 3 [вп 50 пр 45]
 - Правильный пятиугольник
- 4) Повтори 4 [вп 30 пр 90]
 - Правильный шестиугольник
- 2) Повтори 5 [вп 50 пр 360 / 5]
 - Правильный семиугольник
- 8) Повтори 6 [вп 30 пр 60]
 - Правильный восьмиугольник
- 5) Повтори 7 [вп 50 пр 360 / 7]
 - Правильный восьмиугольник
- 7) Повтори 8 [вп 50 пр 45]
 - Правильный восьмиугольник

Учитель. Все ли циклы, которые остались не-востребованными, являются циклами для построения правильных многоугольников?

Учащиеся. Нет, потому что неправильно указаны угол или количество повторений.

3. Объяснение нового материала

Учитель. Ребята, домашним заданием у вас было вырезать снежинку и принести ее на урок. Надеюсь, у вас новогоднее настроение? После урока мы украсим вашими снежинками наш кабинет. Достаньте свои снежинки, внимательно посмотрите на них и на снежинки на доске. Что у них общего?



Учащиеся. Одинаковое число лучиков.

Учитель. Есть ли повторяющиеся элементы?

Учащиеся. Есть.

Учитель. Можно использовать повторение?

Учащиеся. Можно.

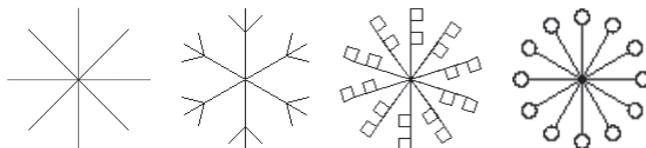
Учитель. Обратите внимание, что все лучики выходят из одной точки.

Если есть наличие определенного порядка, закономерности в расположении частей относительно одной точки, такое явление называют центральной симметрией.

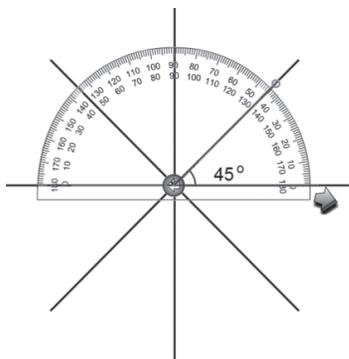
Центральносимметричные фигуры рисуют из центра.

(Раздает ученикам карточки с заданием.)

Карточка-задание.



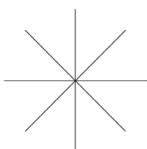
Учитель. Давайте рассмотрим снежинку. У нее восемь лучей. Давайте измерим угол между лучами снежинки. Возьмите транспортир, будем измерять. Прикладываем, совмещаем. Чему равен угол?



Учащиеся. 45 градусам.

Учитель. Мы выяснили, что у нас восемь одинаковых лучиков, выходящих из центра. Давайте напишем процедуру рисования этой фигуры.

Внимательно рассмотрим снежинку. Угол 90° у нас разделяется на два одинаковых: $90 / 2 = 45$.



это сн1
по повтори 8 [вп 50 нд 50 пр 360 / 8]
конец

Учитель. Черепашка, рисуя снежинку, возвращается в исходную точку, совершая при этом полный оборот, как и при рисовании правильного многоугольника.

Как вы думаете, может ли в снежинке быть другое количество лучей?

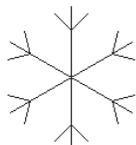
Как найти при этом угол поворота?

Учащиеся. 360° разделить на количество лучей.

Учитель. Откройте тетрадь, нарисуйте снежинку и запишите процедуру для ее рисования.

4. Закрепление изученного материала

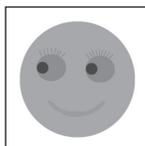
Учитель. Теперь давайте запустим программу «ЛогоМиры» и нарисуем более сложную снежинку. Принцип ее построения такой же, но количество лучей другое.



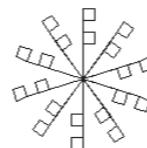
это сн2
по повтори 6 [вп 30 пр 45 вп 20 нд 20 лв 45 вп 20 нд 20 лв 45 вп 20 нд 20 пр 45 нд 30 пр 360 / 6]
конец

5. Физкультминутка

Ребята выполняют упражнения для глаз, следуя за движениями глаз личика на кадрах мультимедийной презентации.



6. Закрепление изученного материала (продолжение)



Учитель. А сколько лучиков у этой снежинки?

Учащиеся. Десять.

Учитель. Какую вспомогательную процедуру нам необходимо написать?

Учащиеся. Процедуру для построения квадрата.

Учитель. Продиктуйте ее мне.

Учащиеся по очереди диктуют строки программы:

```
это кв
по повтори 4 [вп 10 пр 90]
конец
это сн3
по повтори 10 [вп 30 кв вп 20 кв нд 50 пр 360 / 10]
конец
```

Учитель. Попробуйте самостоятельно нарисовать следующую снежинку:



Учащиеся самостоятельно выполняют задание.

```
это окр
по повтори 360 [вп 1 / 15 пр 1]
конец
это сн4
по повтори 12 [вп 30 лв 90 окр пр 90 нд 30 пр 360 / 12]
конец
```

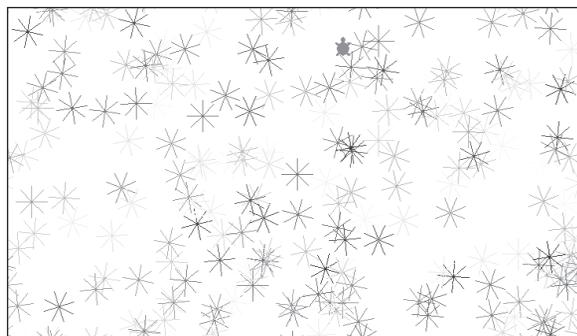
Учитель. А сейчас давайте нарисуем снегопад, т. е. много снежинок на листе.

В качестве основной снежинки будем использовать новую снежинку с длиной луча 10 шагов.

Но для начала давайте рассмотрим команду установки цвета, она выглядит так: нц <число>. Число задает конкретный цвет.

А еще нам понадобится такое понятие, как «датчик случайных чисел», — чтобы на экране все снежинки были разного цвета.

Команда будет выглядеть следующим образом: нц сл 140. То есть Черепашка будет менять цвет пера случайным образом в диапазоне от 0 до 140 (самое большое число цвета).



это снм
по повтори 8 [вп 10 нд 10 пр 360 / 8]
конец
это снегопад
по повтори 200 [нд сл 140 снм пп вп 300 пр сл 360]
конец

7. Подведение итогов урока.

Домашнее задание

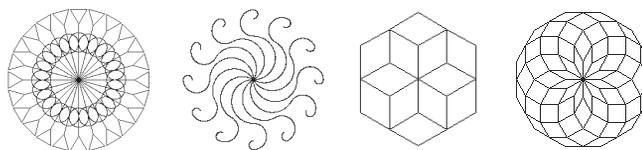
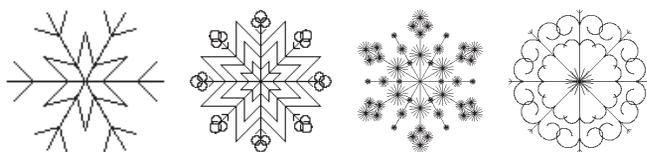
Учитель. Итак, что нового вы сегодня узнали на уроке?

Учащиеся. Поняли алгоритм построения снежинок, т. е. фигур с центральной симметрией.

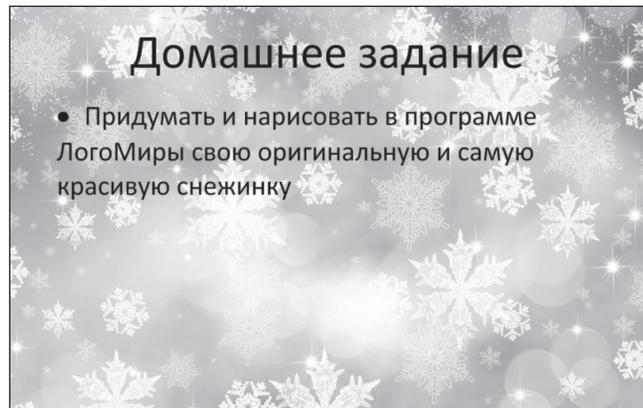
Узнали новую команду: нц <число>.

Построили четыре снежинки и целый снегопад.

Учитель. В дальнейшем мы будем строить более сложные снежинки и другие фигуры с центральной симметрией. Например, такие:



На доске вы видите домашнее задание, запишите его.



Спасибо за урок!

НОВОСТИ

«Умный» баллончик сам рисует граффити за художника

В рамках совместного проекта Disney Research, Швейцарской высшей технической школы Цюриха и еще ряда научных учреждений разработана система распыления аэрозольной краски, позволяющая перенести на стену цифровое изображение в увеличенном виде. При этом достаточно водить баллончиком вдоль стены, а тот сам будет автоматически в нужных местах выпускать краску. Прототип получился легким и быстро реагирующим на команды управляющего компьютера — «умный» баллончик оснащен двумя вебкамерами для слежения

и небольшим приводным устройством для нажатия на пульверизатор, которое прикреплено к нему с помощью конструкции, распечатанной на 3D-принтере. На управляющем компьютере работает алгоритм, который в реальном времени распознает положение баллончика относительно стены и определяет оптимальное количество краски, которое нужно выдавить в конкретном месте. Результат — изображение постепенно проявляется на стене по «мановению руки» оператора, причем ему даже не обязательно знать, что именно он «рисует».

Нестандартная улыбка



Особенности воспроизведения эмодзи на различных платформах могут сильно влиять на воспринимаемое значение сообщения, выяснили в Университете Миннесоты. В частности, в рамках исследования изучалось воспринимаемое значение символа «лицо с широкой улыбкой и улыбающимися глазами» в зависимости от платформы, на которой он отображается. «Эталонного» шрифта

эмодзи не существует, и на устройствах различных производителей и в разных сервисах смайлы отображаются по-своему. Согласно результатам исследования, варианты изображения упомянутого смайлика на устройствах Microsoft, Samsung, LG и Google вызывают однозначно положительную эмоциональную реакцию, тогда как вариант Apple воспринимается «слегка негативно».

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

Г. М. Сабитова,
лицей № 68, г. Уфа, Республика Башкортостан

УРОК НА ТЕМУ «ИСПОЛНИТЕЛЬ ЧЕРТЕЖНИК. КОМАНДА ВЕКТОР (X, Y)»*

Аннотация

В статье представлена методическая разработка урока открытия нового знания, на котором особое внимание уделяется навыкам актуализации знаний, самостоятельной и групповой работы, самопроверки.

Ключевые слова: исполнитель Чертежник, вектор, команда, программа, алгоритм.

Контактная информация

Сабитова Гузель Мухарямовна, учитель информатики лицея № 68, г. Уфа, Республика Башкортостан; *адрес:* 450043, Республика Башкортостан, г. Уфа, бул. Тухвата Янаби, д. 49/4; *телефон:* (347) 238-57-83; *e-mail:* guzelsab@yandex.ru

G. M. Sabitova,
Lyceum 68, Ufa, The Republic of Bashkortostan

LESSON ON THE THEME "PERFORMER DRAWMAN. THE COMMAND VECTOR (X, Y)"

Abstract

The article describes the lesson of discovery of new knowledge, which focuses on the skills of actualization of knowledge, independent and group work, self-examination.

Keywords: performer Drawman, vector, command, program, algorithm.

Раздел программы: Исполнители и алгоритмы.

Тема урока: Исполнитель Чертежник. Команда вектор (x, y).

Цель урока: сформировать:

- представление о команде вектор (x, y);
- умение работать по алгоритму;
- умение составлять алгоритмы для исполнителя Чертежник.

Планируемые образовательные результаты:

личностные:

- формирование навыков самоорганизации;

метапредметные:

- умение самостоятельно планировать пути достижения цели;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности;
- умение корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение осознанно и произвольно строить речевое высказывание в устной форме;

предметные:

- умение разрабатывать алгоритмы для управления исполнителем.

Формируемые универсальные учебные действия:

личностные:

- способность увязать учебное содержание с собственным жизненным опытом;
- способность понять значение алгоритмического мышления для современного человека;

познавательные:

- развитие представлений о типах исполнителей алгоритмов;
- умение использовать систему команд исполнителя Чертежник;

коммуникативные:

- развитие диалогической речи;

регулятивные:

- умение организовывать свою учебную деятельность.

Тип урока: урок открытия нового знания.

Класс: шестой.

Организационные формы работы на уроке: фронтальная, парная, индивидуальная.

Технология: урок разработан в рамках технологии деятельностного метода.

Оснащение урока:

- компьютер учителя;
- компьютеры учащихся;
- интерактивная доска;
- проектор;
- программа robwin Чертежник: <http://kpolyakov.spb.ru/school/robots/download.htm>
- презентация к уроку, подготовленная в Microsoft PowerPoint;

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

- раздаточный материал:
 - лист с заданиями 1–4 (файл *Приложение 1.doc*);
 - лист с теоретическим материалом по теме урока (файл *Приложение 2.doc*);
 - лист с таблицей для заполнения (файл *Приложение 3.doc*);
 - лист с самостоятельной работой (файл *Приложение 4.doc*);
- звездочки красного, желтого и зеленого цветов для проведения рефлексии.

Интернет-источники, использованные при подготовке урока.

1. *Гаврюкова Г. А.* Уроки по теме «Управление исполнителем Чертежник». <http://kpolyakov.spb.ru/school/robots/download.htm>

2. *Поляков К. Ю.* Система «Исполнители», версия 2.5. <http://kpolyakov.spb.ru>

План урока.

1. Организационный момент.
2. Актуализация знаний и фиксация затруднений в пробном действии.
3. Выявление места и причины затруднений.
4. Построение проекта выхода из затруднения.
5. Реализация построенного проекта.
6. Первичное закрепление во внешней речи.
7. Самостоятельная работа с самопроверкой по эталону.
8. Включение в систему знаний и повторение.
9. Рефлексия учебной деятельности на уроке.
10. Домашнее задание.

Ход урока

1. Организационный момент

Урок сопровождается демонстрацией слайдов мультимедийной презентации.

Учащиеся рассаживаются по местам, проверяют наличие принадлежностей.

Учитель приветствует ребят.

Учитель. Несколько уроков мы изучаем одну большую тему. Какое понятие на наших уроках мы упоминаем чаще всего?

Учащиеся. Алгоритм.

Что такое алгоритм?

- Алгоритм – точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за конечное число шагов.

Учитель. Кто или что может выполнить алгоритм?

Учащиеся. Исполнитель: человек, группа людей, животное, техническое устройство.

Кто или что может выполнить алгоритм?

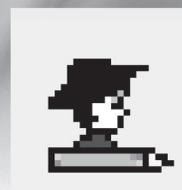
- человек, группа людей, животное, техническое устройство

Учитель. С каким исполнителем мы с вами познакомились?

Учащиеся. С исполнителем Чертежник.

С каким исполнителем мы с вами познакомились?

Чертежник



Учитель. Что входит в систему команд исполнителя Чертежник?

Учащиеся. Чертежником можно управлять с помощью специальных команд, входящих в его СКИ. Чертежник работает в прямоугольной системе координат, начало которой, точка (0, 0), задается в окне настройки.

Команды Чертежника:

покажись; — после выполнения этой команды можно будет видеть Чертежника на поле и наблюдать за его действиями;

скройся; — после выполнения этой команды Чертежник одевает шапку-невидимку и его не будет видно, хотя весь рисунок сохранится;

подними_перо; — после выполнения этой команды Чертежник перемещается по полю, не оставляя следа;

опусти_перо; — теперь Чертежник при любом перемещении по полю рисует за собой линию;

в_точку (x, y); — Чертежник перемещается в точку с координатами (x, y).

Какие команды Чертежника вы знаете?

- **покажись;**
после выполнения этой команды можно будет видеть Чертежника на поле и наблюдать за его действиями
- **скройся;**
после выполнения этой команды Чертежник одевает шапку-невидимку и его не будет видно, хотя весь рисунок сохранится
- **подними_перо;**
после выполнения этой команды Чертежник перемещается по полю, не оставляя следа
- **опусти_перо;**
теперь Чертежник при любом перемещении по полю рисует за собой линию
- **в_точку (x, y);**
чертежник перемещается в точку с координатами (x, y)

Учитель. Выполним задание 1, работаем карандашом.

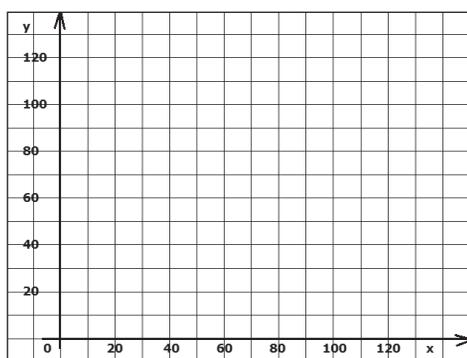
Ученики индивидуально выполняют задание.

Задание 1.

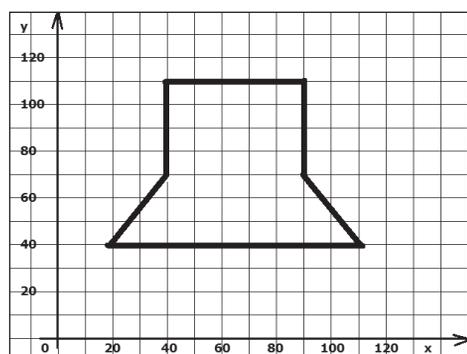
Что будет нарисовано после выполнения Чертежником алгоритма?

Программа:

```
{
подними_перо;
в_точку (20, 40);
опусти_перо;
в_точку (40, 70);
в_точку (40, 110);
в_точку (90, 110);
в_точку (90, 70);
в_точку (110, 40);
в_точку (20, 40);
подними_перо;
}
```



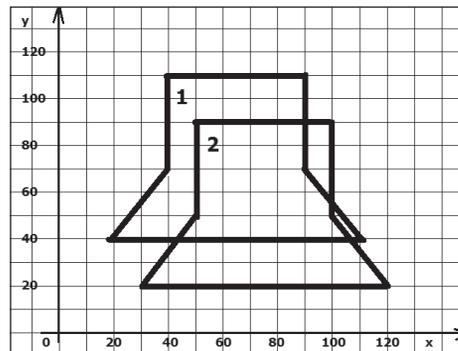
У учащихся получится следующая фигура:



2. Актуализация знаний и фиксация затруднений в пробном действии

Учитель. Попробуйте нарисовать точно такую же фигуру, сместив координаты точек по оси X на 10 единиц, по оси Y — на -20 единиц.

Учащиеся выполняют задание:

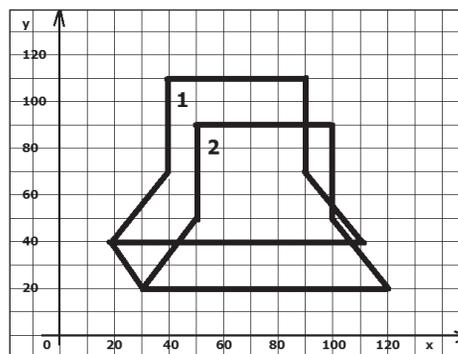


Учитель. Что мы сделали с изображением?

Учащиеся. Мы его сместили на две клетки вниз и на одну клетку вправо.

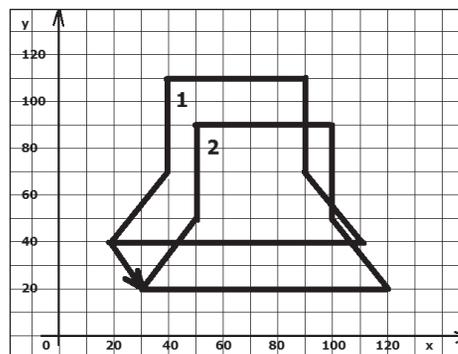
Учитель. Давайте нарисуем отрезок, который соединит нижние точки наших фигур.

Учащиеся рисуют отрезок:



Учитель. А как показать, что мы сместили одну фигуру относительно другой?

Учащиеся рисуют стрелку:



Учитель. Таким образом, что будет показывать наш отрезок?

Учащиеся. Направленный отрезок.

Учитель. В математике направленные отрезки называют **вектором**. В информатике мы будем работать с командой **вектор**.

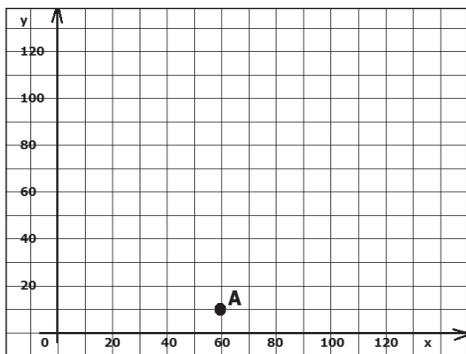
Пробное действие.

Учитель. Выполните следующее задание.

Задание 2.

Укажите координаты точки, в которой окажется Чертежник после выполнения команды:

вектор (40, 70)



Учащиеся работают индивидуально, пробуя выполнить задание.

3. Выявление места и причины затруднений

Учитель. Проверим, что у вас получилось.

Кто не смог выполнить задание, почему?

Вопрос к тем, у кого получилось: почему у многих разные координаты?

Почему вы не справились с заданием?

Учащиеся. Пока не знаем правила, по которым работает команда вектор (x, y).

4. Построение проекта выхода из затруднения

Учитель. Сформулируйте цель нашей дальнейшей деятельности.

Учащиеся. Узнать принцип работы команды вектор (x, y).

Учитель. Как вы назвали бы тему нашего урока?

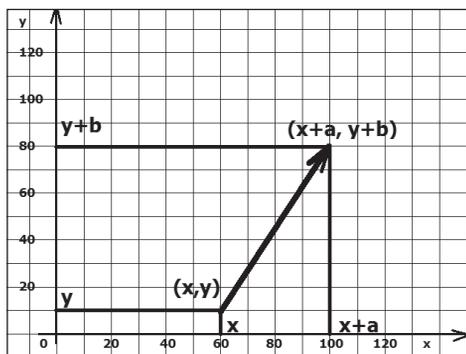
Учащиеся. Команда вектор (x, y).

Учитель. Далее вы будете работать в парах — вам надо будет ответить на вопросы (Приложение 3), используя имеющиеся у вас листы с теоретическим материалом (Приложение 2).

Работа в парах.

Теоретический материал.

По команде вектор (a, b) Чертежник отсчитывает на a единиц вправо вдоль горизонтальной оси, на b единиц вверх вдоль вертикальной оси. Если текущие координаты были (x, y), то новое положение будет: (x + a, y + b).



Если a < 0, b < 0 и текущие координаты были (x, y), то новое положение будет: (x - a, y - b).

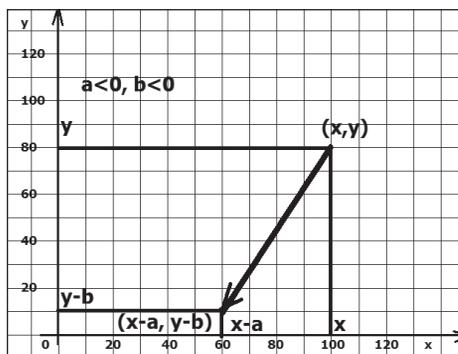


Таблица для заполнения.

| № п/п | Вопрос | Ответ |
|-------|--|---|
| 1 | Как называют команду смещения на отрезок? | |
| 2 | Дополните предложение | По команде вектор (a, b) Чертежник отсчитывает на _____ вдоль горизонтальной оси, на _____ вдоль вертикальной оси. Если текущие координаты были (x, y), то новое положение будет (_____, _____) |
| 3 | Если чертежник находится в точке с координатами (10, 20), то в какой точке он окажется после выполнения команды вектор (30, 40)? | В точке с координатами (____, ____) |

5. Реализация построенного проекта

Учитель. Проверим результаты вашей работы.

Путем обсуждения учащиеся находят верный ответ.

Ответ.

| № п/п | Вопрос | Ответ |
|-------|--|--|
| 1 | Как называют команду смещения на отрезок? | Вектор (x, y) |
| 2 | Дополните предложение | По команде вектор (a, b) Чертежник отсчитывает на a единиц вправо вдоль горизонтальной оси, на b единиц вверх вдоль вертикальной оси. Если текущие координаты были (x, y), то новое положение будет (x + a, y + b) |
| 3 | Если чертежник находится в точке с координатами (10, 20), то в какой точке он окажется после выполнения команды вектор (30, 40)? | В точке с координатами (40, 60) |

6. Первичное закрепление во внешней речи

Учитель. Вернемся к заданию 2.

Один из учеников выполняет задание у доски с проговариванием, остальные учащиеся решают его на листочках:

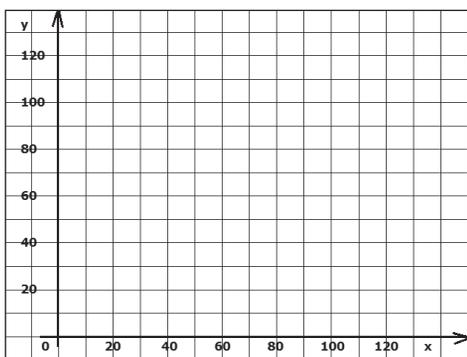
$$(60 + 40, 10 + 70) = (100, 80).$$

Задание 3.

Что будет нарисовано после выполнения Чертежником алгоритма?

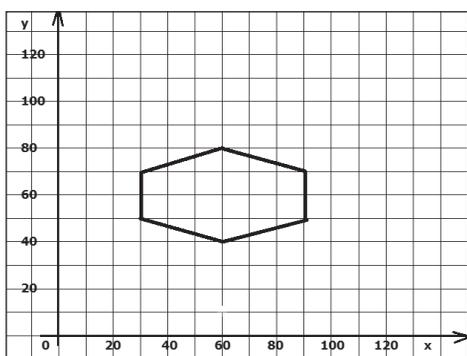
Программа:

```
{
подними_перо;
вектор (60, 40);
опусти_перо;
вектор (30, 10);
вектор (0, 20);
вектор (-30, 10);
вектор (-30, -10);
вектор (0, -20);
вектор (30, -10);
подними_перо;
}
```



Задание выполняется на доске с проговариванием.

У учащихся получится следующая фигура:



7. Самостоятельная работа с самопроверкой по эталону

Учитель. А теперь проверим, как вы поняли работу команды **вектор (x, y)**.

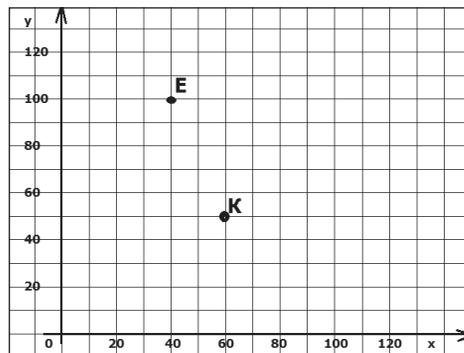
Выполните самостоятельно задания 1–2, задание 3 — дополнительное.

Критерии оценивания самостоятельной работы:

- верно выполнено одно задание — оценка «4»;
- верно выполнено два или три задания — оценка «5».

Задание 1 самостоятельной работы.

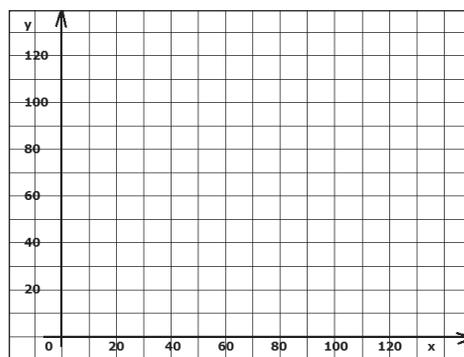
Напишите координаты точек, в которых окажется перо Чертежника после выполнения команды **вектор (30, -20)**.



Задание 2 самостоятельной работы.

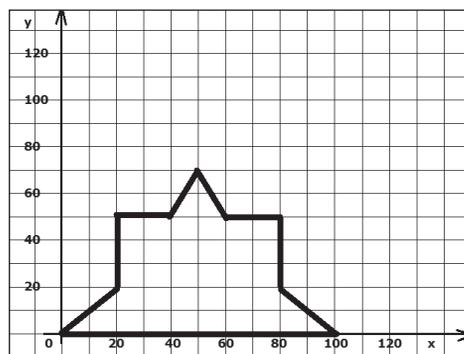
Что будет нарисовано после выполнения Чертежником приведенной ниже программы?

```
{
подними_перо;
вектор (0, 20);
опусти_перо;
вектор (0, 40);
вектор (60, 0);
вектор (0, 20);
вектор (60, -40);
вектор (-60, -40);
вектор (0, 20);
вектор (-60, 0);
подними_перо;
}
```



Задание 3 самостоятельной работы.

Составьте программу рисования приведенного ниже изображения, используя команду **вектор (x, y)**.



Ответы учащиеся проверяют в ходе самопроверки по слайду презентации.

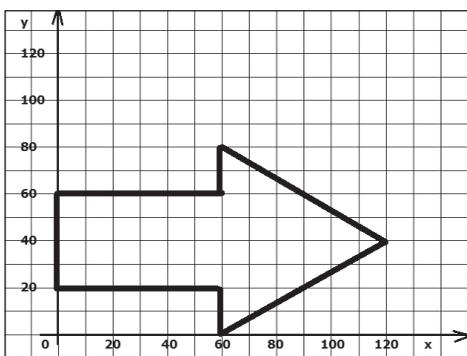
Ответы.

1.

K(60, 50)
вектор (30, -20)
K(90, 30)

E(40, 100)
вектор (30, -20)
E(70, 80)

2.



3.

```
{
опусти_перо;
вектор (20, 20);
вектор (0, 30);
вектор (20, 0);
вектор (10, 20);
вектор (10, -20);
вектор (20, 0);
вектор (0, -30);
вектор (20, -20);
вектор (-100, 0);
подними_перо;
}
```

Проверка самостоятельной работы

Задание 1
K(60, 50)
вектор (30, -20)
K(90, 30)
E(40, 100)
вектор (30, -20)
E(70, 80)

Задание 3.
{
опусти_перо;
вектор (20, 20);
вектор (0, 30);
вектор (20, 0);
вектор (10, 20);
вектор (10, -20);
вектор (20, 0);
вектор (0, -30);
вектор (20, -20);
вектор (-100, 0);
подними_перо;
}

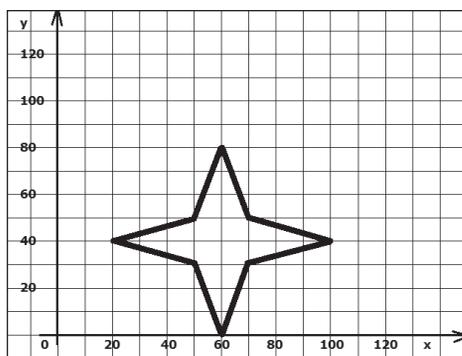
Задание 2.

8. Включение в систему знаний и повторение

Для закрепления полученных знаний ребята выполняют задание 4 из раздаточного материала.

Задание 4.

Составьте программу создания фигуры с помощью команды **вектор (x, y)**.



Ответ.

```
{
подними_перо;
вектор (20, 40);
опусти_перо;
вектор (30, 10);
вектор (10, 30);
вектор (10, -30);
вектор (30, -10);
вектор (-30, -10);
вектор (-10, -30);
вектор (-10, 30);
вектор (-30, 10);
подними_перо;
}
```

9. Рефлексия учебной деятельности на уроке

Учитель. Какова была цель вашей деятельности? (Ответы учащихся.)

Учитель раздает всем звездочки красного, желтого и зеленого цветов.

Учитель. Выполните одно из трех действий, чтобы я увидела результат работы на уроке:

- те, кто считает, что не достиг цели, — поднимает красную звездочку;
- те, у кого не все получилось, — желтую;
- те, кто считает, что он выполнил цель, поставленную на уроке, — зеленую.

Учащиеся отвечают на вопрос, поднимая звезду определенного цвета.

10. Домашнее задание

Учащиеся записывают домашнее задание, которое демонстрируется на слайде:

Домашнее задание

В программе Чертежник нарисовать фигуру



С. В. Иванова,

Октябрьская средняя общеобразовательная школа, Удмуртская Республика

ИКТ В ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ НА УРОКАХ ГЕОМЕТРИИ*

Аннотация

В статье представлен опыт формирования у учащихся навыков проектной деятельности при изучении геометрии с использованием ИКТ.

Ключевые слова: технология Stop Motion, ИКТ, киностудия Windows Live, Paint, виртуальный конструктор «Живая Математика», критерии оценивания.

Контактная информация

Иванова Светлана Владимировна,
зам. директора по УВР, учитель математики
Октябрьской средней общеобразовательной
школы, Удмуртская Республика; адрес: 427006,
Удмуртская Республика, Завьяловский район,
село Октябрьский, д. 39; телефон: (341-2)
69-00-28; e-mail: oksosh@mail.ru

S. V. Ivanova,
Octyabr'skaya Secondary School, The
Udmurt Republic

ICT IN PROJECT ACTIVITY ON GEOMETRY LESSONS

Abstract

The article describes the experience of the formation of students' skills of project activity in the study of geometry using ICT.

Keywords: Stop Motion technology, ICT, Windows Live Movie Maker, Paint, Geometer's Sketchpad, evaluation criteria.

Требования к выпускнику средней школы сегодня кардинально изменились — теперь школа должна не только давать знания, но и воспитывать качества личности, важные для жизни в новых условиях открытого общества, развивать творческие способности учащихся, самостоятельность мышления, чувство личной ответственности за результат своей работы. Проектная деятельность — это один из методов, направленный на самостоятельную выработку умений, способствующий развитию творческих способностей и логического мышления, объединяющий знания, полученные в разных предметах школьной программы, и приобщающий к конкретным жизненно важным проблемам.

Под проектом подразумевается специально организованный учителем и самостоятельно выполняемый учащимися комплекс действий, завершающихся созданием продукта, состоящего из объекта труда, изготовленного в процессе проектирования, и его представления в рамках устной или письменной презентации.

Актуальность использования метода проектов обусловлена, прежде всего, необходимостью для современного человека самостоятельно ставить профессиональные цели и задачи, продумывать способы их осуществления. И не случайно один из параметров нового качества образования — способность проектировать.

Отличительная черта проектной методики — особая форма организации деятельности. Организуя работу над проектом, необходимо соблюдать основные этапы проектной деятельности. Каждый проект соотносится с определенной темой и разрабатывается в течение определенного времени.

Более увлекательным процесс создания проекта становится при использовании ИКТ. Люди самых разных профессий применяют в своей работе компьютерную графику: художники, конструкторы, дизайнеры, медики, разработчики рекламной продукции, фотографы, модельеры и т. д. Все более широкое распространение в сети Интернет получает технология создания мультфильма Stop Motion. Большое количество баннеров, сайтов, игр и других приложений делается с помощью программ Macromedia Flash, DVD-студия Windows, «Киностудия Windows Live» и др. Эти программы легки в освоении и позволяют дать учащимся понятие о новых способах работы с графической и текстовой информацией.

В ходе выполнения проектной работы учащимися было предложено использовать технологию **Stop Motion** для создания собственного анимационного фильма по геометрии. Проект позволил учащимся окунуться в увлекательный мир мультипликации и попробовать себя в роли сценариста, художника-мультипликатора, режиссера мультфильма. Во время выполнения проекта обучающиеся изучили программный материал по геометрии, а также получили возможность самостоятельно планировать свою деятельность, развивать навыки

* Материалы к статье можно скачать на сайте ИНФО:
http://infojournal.ru/journals/school/school_04-2016/

коммуникации, умения работать в группе. Проектные занятия были проведены в седьмом классе при изучении тем «Правильные многогранники», «Движение плоскости».

Цель занятий: формирование навыков проектной деятельности при изучении геометрии с использованием ИКТ.

Планируемые результаты обучения: учащиеся:

- расширят и углубят знания по геометрии;
- разовьют навыки использования ИКТ;
- научатся выступать с подготовленным сообщением, грамотно отвечать на вопросы;
- научатся реализовывать проекты;
- научатся адекватно оценивать собственные знания и навыки.

План занятий:

- **Урок 1.** Подготовительный этап. Планирование работы. Построение чертежей.
- **Урок 2.** Подготовка фотоснимков. Решение промежуточных задач.
- **Урок 3.** Создание проекта в программе «Киностудия Windows Live». Оценка процесса и результатов работы.
- **Урок 4.** Представление готового продукта. Презентация проекта. Оценка процесса и результатов работы.

Этапы проектной деятельности

1. Этап подготовки к осуществлению проекта

Метод, преобладающий на данном этапе, — беседа. Учащиеся делятся на малые группы по два человека и знакомятся с необходимым теоретическим материалом «Технология проектной деятельности» на сайте «Об-

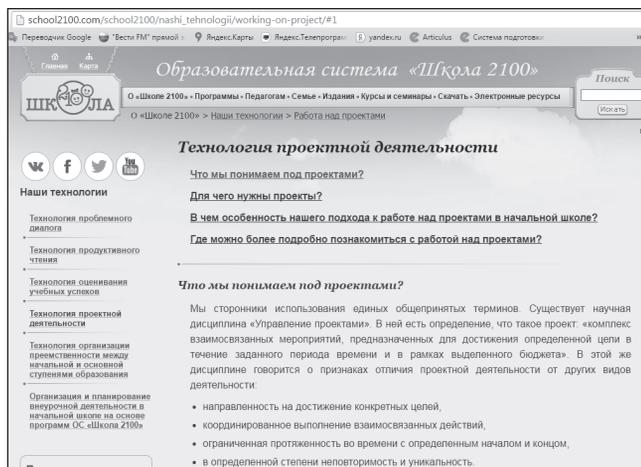


Рис. 1

разовательная система «Школа 2100»: <http://goo.gl/cdYpIA> (рис. 1).

После знакомства с теорией ребята отвечают на **вопросы** учителя:

- Что такое проекты? Принимали ли вы когда-нибудь в них участие? Приведите примеры.
- Как вы думаете, как организовывается работа над проектом?
- Какие виды проектов бывают?

Отвечая на вопросы, учащиеся уточняют информацию. Учитель мотивирует школьников, объясняет цели проекта, приглашает в творческую мастерскую «Веселые встывшки»: <http://goo.gl/d9jfbp>. В презентации дается определение технологии Stop Motion, рассказывается о ее истории, современных тенденциях, оборудовании, которое используется в этой технологии, этапах создания мультфильма и др. (рис. 2).



Рис. 2

Ребята обсуждают презентацию и делают вывод, что им предстоит работа над творческим проектом по созданию фильма по геометрии. Они определяют свои роли и объединяются в соответствии с ними в группы.

Затем учащиеся знакомятся с примерами мультфильмов, созданных в технологии Stop Motion: <https://cloud.mail.ru/public/JfkJ/c1oUejWPK>, в частности с мультфильмом «Мой павлин» (рис. 3).

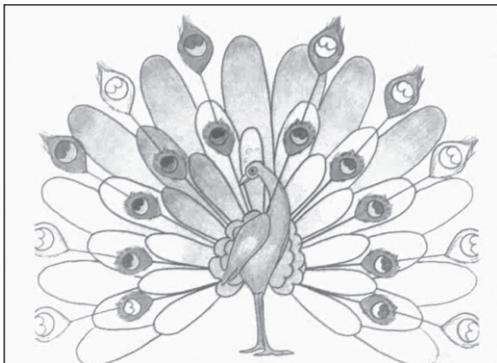
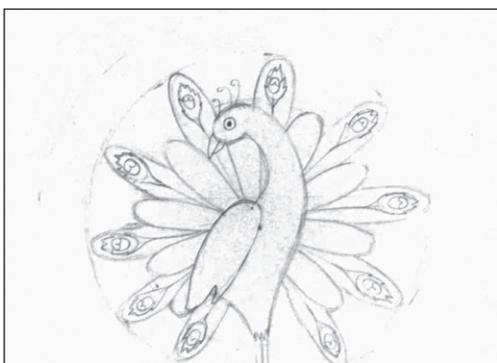
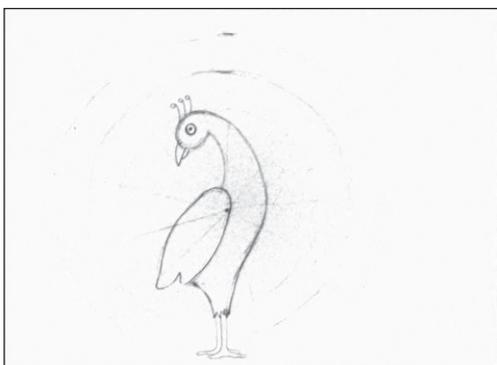
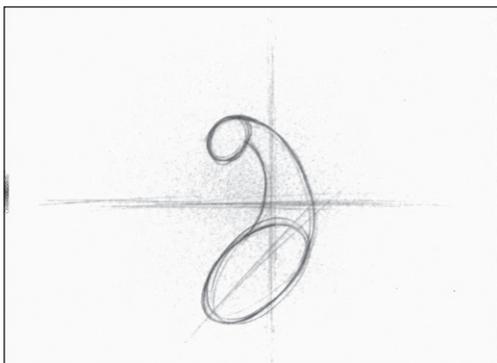


Рис. 3

Учитель наблюдает за деятельностью школьников и при необходимости помогает проводить организационную работу.

2. Этап планирования

На этом этапе учащиеся:

- формулируют задачи проекта;
- уточняют информацию о правильных многогранниках в учебнике геометрии;
- выбирают и обосновывают свои критерии успеха.

Учитель помогает в формулировании задач проекта.



Фото 1

3. Этап принятия решения

На этом этапе учащиеся:

- работают с информацией;
- строят развертки и делают чертежи;
- проводят синтез и анализ идей.

Учитель наблюдает, консультирует, координирует работу групп.



Фото 2

4. Этап выполнения

Учащиеся выполняют построения в программах Paint и «Живая Математика» (рис. 4), а затем оформляют проект.

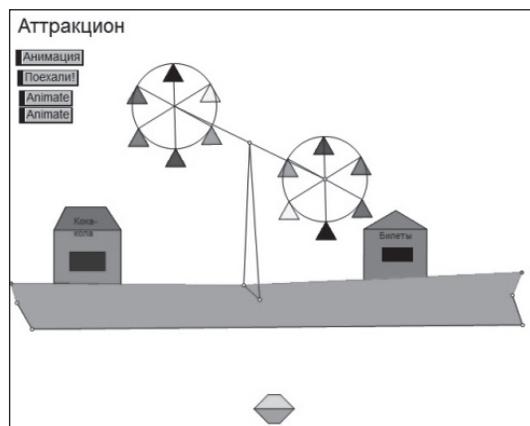


Рис. 4

Учитель наблюдает, консультирует, советует.

5. Этап защиты проекта (презентации)

Учащиеся защищают свои проекты — демонстрируют ролики, рассказывают о результатах своей работы, отвечают на вопросы одноклассников и учителя, а затем участвуют в коллективной оценке результатов проекта.

Учитель участвует в коллективном анализе и оценке результатов проекта.

Фрагменты некоторых работ учащихся представлены на рисунках 5–7:

- «Движение плоскости»: <https://cloud.mail.ru/public/4Sg2/zdLawMe2L> (рис. 5);
- «Живая геометрия»: <https://cloud.mail.ru/public/7DAS/EbVka3fYu> (рис. 6);
- «Парад фигур»: <https://cloud.mail.ru/public/AcXr/yGYe39FcJ> (рис. 7).

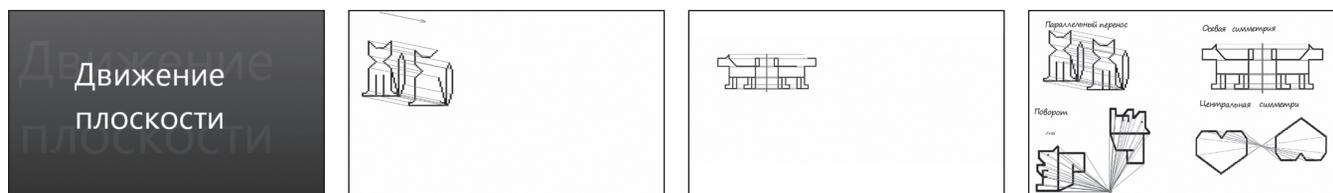


Рис. 5

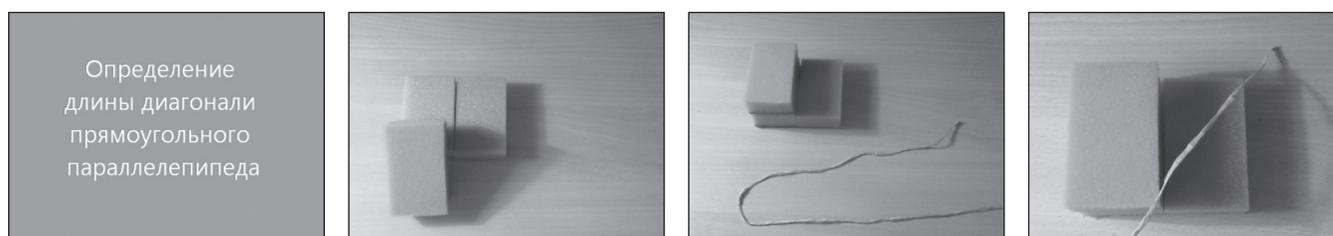


Рис. 6

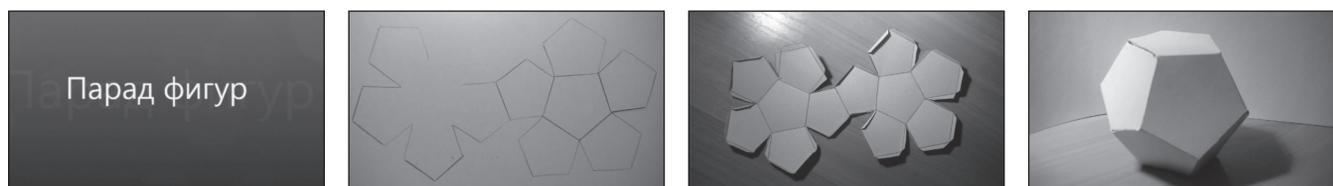


Рис. 7

6. Этап рефлексии

На этом этапе проводятся коллективный самоанализ проекта и его самооценка. Учащиеся выявляют трудности, с которыми столкнулись, анализируют, как они были преодолены.

Максимальное количество баллов, которое можно получить за проект, — 100. Оцениваются содержание фильма, оформление работы, устное выступление. Бланк с критериями оценивания доступен всем для изучения — он вывешивается в кабинете для ознакомления.

Критерии оценивания результатов проекта

| № п/п | Критерий | Максимальное количество баллов |
|-------|--|--------------------------------|
| 1 | Содержание | |
| 1.1 | Идея, оригинальность, уникальность сюжета мультфильма | 5 |
| 1.2 | Логичность, последовательность, завершенность сюжета мультфильма | 5 |
| 1.3 | Смысловое содержание сюжета мультфильма | 5 |
| 1.4 | Указание авторства материала | 5 |
| 1.5 | Наличие последнего кадра, содержащего надпись «Конец фильма» | 5 |
| 2 | Эргономические требования | |
| 2.1 | Грамотное композиционное решение экрана | 5 |
| 2.2 | Грамотно подобранная цветовая гамма фоновых композиций | 5 |
| 2.3 | Наличие звукового сопровождения | 5 |

Окончание таблицы

| № п/п | Критерий | Максимальное количество баллов |
|-------|--|--------------------------------|
| 2.4 | Качество графических объектов | 5 |
| 3 | Использование возможностей анимации | |
| | Передача движения, наличие различных видов анимации: | |
| 3.1 | • движение Motion | 5 |
| 3.2 | • движение по траектории | 5 |
| 3.3 | Использование разнообразных приемов, улучшающих качество анимации (масок, преобразований текста в графику с дальнейшей его обработкой, преобразование линий в заливки и др.) | 5 |
| 3.4 | Качество анимации (реалистичность ролика) | 5 |
| 4 | Дополнительные баллы | |
| 4.1 | Творческий подход и оригинальность оформления | 5 |
| 5 | Презентация и защита работы | |
| 5.1 | Качество доклада: композиция, полнота представления работы, подходов, результатов; аргументированность, убедительность и убежденность | 5 |
| 5.2 | Объем и глубина знаний по геометрии, эрудиция, опора на межпредметные связи | 5 |
| 5.3 | Культура речи, использование наглядных средств, манера выступления, чувство времени (3–5 мин), импровизационное начало, удержание внимания аудитории | 5 |
| 5.4 | Ответы на вопросы: полнота, аргументированность, убедительность и убежденность, дружелюбность, стремление использовать ответы для успешного раскрытия темы и сильных сторон работы | 10 |
| 5.5 | Деловые и волевые качества докладчика: ответственное решение, стремление к достижению высоких результатов, доброжелательность, контактность | 5 |
| | Суммарная оценка работы и защиты: • 90–100 баллов — «отлично»; • 40–89 баллов — «хорошо»; • менее 40 баллов — «удовлетворительно» | 100 |

Небольшие проекты на несколько уроков, которые учащиеся реализуют в ходе совместной деятельности, — это прекрасная возможность дать каждому школьнику шанс проявить свою творческую индивидуальность. Проекты предполагают активизацию деятельности учащихся, так как ребята должны знакомиться с учебной

литературой, справочниками, подбирать рисунки и чертить чертежи. Учащиеся с разным уровнем подготовки могут участвовать в проектной работе в соответствии со своими возможностями. А возникший интерес и чувство радости от выполненного задания у школьника и есть критерий успешной работы.

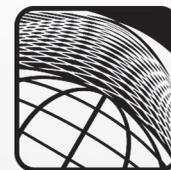
НОВОСТИ

Электрокожа для роботов растягивается, светится и меняет цвета

Эластичную «кожу» на основе светоизлучающих конденсаторов изобрели специалисты Корнельского университета. Она растягивается вшестеро и может менять цвет. Кожа изготовлена из слоев эластомерного материала, чередующихся со слоями гидрогелевых электродов. Эластомер способен сохранять заряд при деформациях и светиться. Из шести таких конденсаторов исследователи изготовили мягкого робота. Поочередно

то наполняя его камеры воздухом, то сдувая их, робота заставляли перемещаться. Разработчики выражают уверенность, что их изобретение имеет большое значение для медицинской отрасли, транспортировок и связи. В эластичную кожу можно будет, например, одеть домашнего робота-сиделку, который мог бы отображать жизненные показатели пациента и менять цвет, считают исследователи.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)



Л. В. Зенцова,

лицей № 36 открытого акционерного общества «Российские железные дороги», г. Иркутск

ОПЫТ ПРОЕКТИРОВАНИЯ ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ СИСТЕМНО-ДЕЯТЕЛЬНОСТНОГО ПОДХОДА

Аннотация

Интенсивное развитие современного образовательного пространства, реализация федеральных государственных образовательных стандартов предполагают активное применение системно-деятельностного подхода в обучении. В статье приводятся примеры реализации системно-деятельностного подхода на уроках информатики, предлагаются варианты проведения мотивационного этапа урока, а также варианты проектных задач и учебных проектов.

Ключевые слова: системно-деятельностный подход, ценностные ориентации поколения Z, личностные, метапредметные и предметные результаты, мотивационный этап, урок информатики, проектный метод, моделирование, реляционная база данных, критерии оценивания.

Контактная информация

Зенцова Людмила Владимировна, канд. тех. наук, учитель информатики лицея № 36 открытого акционерного общества «Российские железные дороги», г. Иркутск; адрес: 664005, г. Иркутск, ул. Профсоюзная, д. 3; телефон: (395-2) 63-25-80; e-mail: lyus2004@mail.ru

L. V. Zentsova,
Lyceum 36 of JSC Russian Railways, Irkutsk

EXPERIENCE OF DESIGNING EDUCATIONAL ACTIVITY BASED ON SYSTEM ACTIVITY APPROACH

Abstract

Intensive development of modern educational space, the implementation of the federal state educational standards assume active use of system activity approach to learning. The article gives examples of system activity approach to informatics lessons, available options for the motivational stage of the lesson, as well as options for project tasks and training projects.

Keywords: system activity approach, value orientations of Z generation, personal, meta-subject and subject results, motivational stage, informatics lesson, project method, modeling, relational database, evaluation criteria.

Интенсивное развитие современного образовательного пространства, реализация федеральных государственных образовательных стандартов предполагают активное применение системно-деятельностного подхода в обучении. Одно из его преимуществ — органичное сочетание данного подхода с различными современными образовательными технологиями: ИКТ, игровыми технологиями, технологиями критического мышления, исследовательской, проектной деятельностью, а также формирование у учащихся личностных, метапредметных и предметных результатов.

На уроках информатики системно-деятельностный подход кроме проектирования и конструирования развивающей образовательной среды позволяет формировать активную учебно-познавательную деятельность школьников.

За последние тридцать лет изменился не только урок информатики — изменился ученик и изменился учитель. Как известно (замечу, что по этому поводу у социологов и демографов нет единого мнения), современные школьники (родившиеся в 2000-х) являются представителями поколения Z, или поколения МеМеМе (ЯЯЯ). В цифровом мире они — аборигены. Их учителя, родители, старшие братья и сестры, принадлежащие к поколениям X и Y, и уж тем более их бабушки и дедушки, принадлежащие к поколению беби-бумеров, — цифровые иммигранты.

Учителей поколения X (родившихся в 1965–1982 годах) характеризуют готовность к изменениям, глобальная информированность, техническая грамотность, индивидуализм, стремление учиться в течение всей жизни, неформальность взглядов, поиск эмоций, прагматизм, надежда на себя, равноправие полов.

Ценности учителей поколения Y (родившихся в 1983–1999 годах) продолжают формироваться и сейчас. Но в них уже включены понятия «гражданский долг», «мораль», «ответственность». Психологи отмечают их наивность, умение подчиняться и выдвигать на первый план немедленное вознаграждение.

Как современному учителю вести урок информатики для детей, представителей поколения Z, коренных жителей цифрового мира, чей цифровой образ жизни в режиме многозадачности органично переплетен с гаджетами и информационно-коммуникационными технологиями?

Современный цифровой мир меняет роль самого учителя: учитель детей поколения Z в большей степени становится тьютором. Учитель информатики помогает своим ученикам ориентироваться в информационно-цифровом пространстве. А при всем многообразии образовательных проектов и их форматов, разносторонности увлечений нынешних учеников, их быстрой адаптации к различным средам, желании попробовать себя в разных областях знаний одна из важных задач учителя — заинтересовать и направить ребенка.

Учитель сегодня — предметный эксперт, но в большей степени — эксперт по учебной деятельности. Он должен

научить ребенка учиться самостоятельно, учить не столько знаниям, сколько способам обращения со знаниями, их применения, поиска, отбора. Учить компетентности. И постоянно учиться сам.

На уроке информатики важен каждый этап. Трудно переоценить значение **мотивационного этапа**, когда учитель устанавливает в классе межличностный контакт и включает учеников в активную интеллектуальную деятельность. Предлагаю несколько авторских вариантов этого этапа урока.

Урок по теме «Устройства ввода», X класс (программа К. Ю. Полякова, Е. А. Еремина, углубленный уровень).

Учитель не сообщает в начале урока его тему, а ошеломляет учащихся сведениями, которые кажутся невероятными для школьного кабинета, — задает вопрос:

— Сейчас в нашем компьютерном классе (для читателей замечу, что в классе находятся 15 учеников, один учитель, 15 ученических ноутбуков и один учительский компьютер) имеется 48 видеокамер. Укажите, где они находятся.



Фото. Компьютерный класс — 55 видеокамер

После нескольких секунд замешательства ученики быстро называют 15 видеокамер на ноутбуках + одну видеокамеру на учительском компьютере, вспоминают, что у каждого с собой сотовый телефон, а это еще 16 видеокамер. Насчитали 32. Где остальные? На некоторых сотовых телефонах по две видеокамеры, но таких телефонов в классе семь. Тогда учитель уточняет и просит указать находящиеся в кабинете 55 (48 + 7) видеокамер. Итак, нам осталось предъявить 16 устройств. Мало кто из десятиклассников дает правильный ответ.

Оказывается, миниатюрные видеокамеры есть в оптических мышках, которые лежат у каждого на компьютерном столе.

Далее учитель задает закономерный в таком случае вопрос о теме урока. Школьники называют тему урока и активно участвуют в определении его учебных целей. И хотя материал об устройствах ввода информации в компьютер не является новым для десятиклассников, важным и полезным для них становится его содержательное обобщение, побуждающим фактором которого стал первый мотивационный этап урока.

Урок по теме «Способы познания окружающего мира», VI класс (программа для основной школы Л. Л. Босовой, А. Ю. Босовой).

С самого начала урока учитель задает высокий темп, с помощью смыслового чтения включая учащихся в активную интеллектуальную деятельность, дает им задание бегло просмотреть § 7 учебника (Босова Л. Л., Босова А. Ю. Информатика: учебник для 6 класса. М.: БИНОМ. Лаборатория знаний, 2015).

Текст параграфа вместе с вопросами составляет пять страниц. За три-четыре отведенные минуты шестиклассники не в состоянии вдумчиво прочитать текст. Они учатся быстрому чтению, учатся определять основную и второстепенную информацию. Учитель подсказывает им, как быстро увидеть текст целиком: следует выбирать основные идеи, ключевые слова, определения. Авторы учебника помогают в этом, применяя выделения рамочкой, цветом, шрифтом. Попутно у школьников происходит осмысление цели чтения, они учатся выбирать вид чтения в зависимости от цели.

Далее перед шестиклассниками стоит проблема: тема урока на доске не записана, но они видят ее в названии параграфа: «Как мы познаем окружающий мир». Так чего же от них потребует учитель? Учитель подводит учащихся к самостоятельному формулированию определяющего вопроса. Он просит учеников побыть в роли учителя, придумать и задать такой вопрос, чтобы уже их ученики сами догадались о теме урока и определились с его целями. Прочитанный с помощью смыслового чтения текст параграфа помогает ребятам в этом.

Шестиклассникам нравится такой подход, они творчески подходят к решению поставленной задачи. Приведу лишь один ученический ответ по названию темы урока: «Какие у человека есть “устройства” ввода информации?» Напоминаю, что параграф учебника назывался «Как мы познаем окружающий мир».

Урок по теме «Кодирование информации», XI класс (программа Н. Д. Угриновича, профильный курс).

На этапе мотивации к учебной деятельности учитель ставит ученикам задачу: расшифровать имена ученых, внесших значительный вклад в развитие информатики. Правильное решение задач на декодирование информации позволит ученикам расшифровать имена ученых. Следующее задание — узнать биографию каждого ученого и его научные достижения, ответить на вопросы: какую роль сыграло изобретение этого ученого в нашей жизни и можно ли считать это изобретение великим?

Хочется привести диалог ученика и учителя на таком уроке:

— А это не основатель сети Facebook?

— Нет, я не использовала для шифрования имена ученых, внесших значительный вклад в развитие информатики и ИКТ в последнее десятилетие. Такие имена на слуху, задание для вас было бы легким, вы по первым буквам можете догадаться.

— К сожалению, об основателе сети Facebook я знаю только фамилию, а хотелось бы узнать больше...

— А почему ты сам о нем не прочитаешь?

— ???

Оказывается, даже имея открытый и свободный доступ к этой информации, ученики не удосуживаются самостоятельно ее получить. Стало понятно, что при внешнем сопротивлении всему «навязанному» пред-

ставители поколения Z осознают, что учитель их «ведет», дает направление в современном информационном мире.

Изучая историю науки, в частности историю развития вычислительной техники, старшеклассники — представители поколения Z — не просто изучают последовательность фактов, а стараются «примерить» биографии ученых, их поступки на себя. Ищут ответ на вопрос: как выбрать свою дорогу? Отбирают образцы поведения и выстраивают систему ценностей. И учитель, используя специально спроектированные учебные задачи, ориентированные на проектную деятельность в течение одного или нескольких уроков, достигает личностных результатов.

Реализация системно-деятельностного подхода посредством **проектного метода обучения** позволяет создать условия для формирования личностных, метапредметных и предметных результатов. На своих уроках я использую краткосрочные (один—три урока) индивидуальные и малогрупповые (два-три человека) проекты.

Рассмотрим некоторые варианты проектных задач и учебных проектов и продукты проектной деятельности.

Построение информационной модели ученика, VII класс.

Учащимся предлагаются задания:

- 1) в Microsoft PowerPoint создайте информационную модель «Успеваемость». Для этого создайте слайд с таблицей личных параметров, отражающих успеваемость по предметам за предыдущую четверть, подсчитайте средний балл;
- 2) на следующем слайде создайте информационную модель «Мой распорядок дня». По данным таблицы постройте диаграмму вашего распорядка дня.

| | Сон | Работа по дому | Занятия в школе | Кружки (секции) | Домашние задания | Свободное время |
|---------------------------|-----|----------------|-----------------|-----------------|------------------|-----------------|
| Действительный распорядок | | | | | | |
| Желаемый распорядок | | | | | | |

В строке «Желаемый распорядок» внесите данные с учетом того, что бы вы хотели изменить в своем распорядке дня;

- 3) на третьем слайде создайте информационную модель «Характер». Используйте схему, отражающую самоанализ личности (рис. 1). Для построения схемы применяйте элементы векторной графики.

Выполняя эту проектную задачу, учащиеся получают возможность взглянуть на себя со стороны, учатся адекватной самооценке и самостоятельному принятию решений (рис. 2, 3).

Интересными стали результаты (рис. 4–6) выполнения проектов, над которыми задумались и ученики, и учитель.

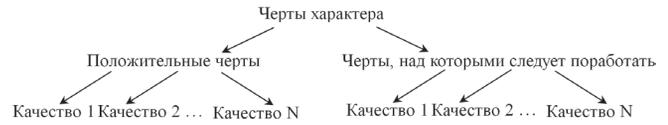


Рис. 1. Схема «Черты характера»

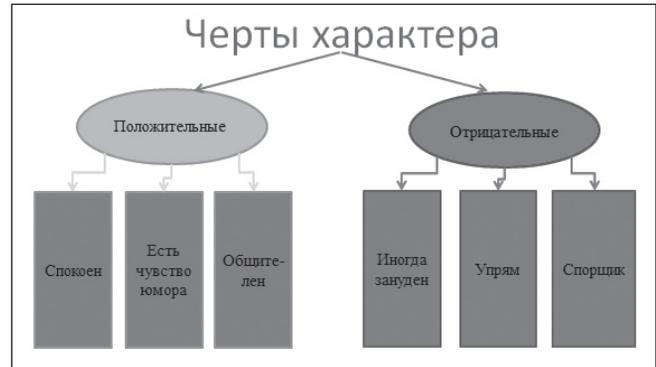


Рис. 2. Модель «Черты характера 1»



Рис. 3. Модель «Черты характера 2»

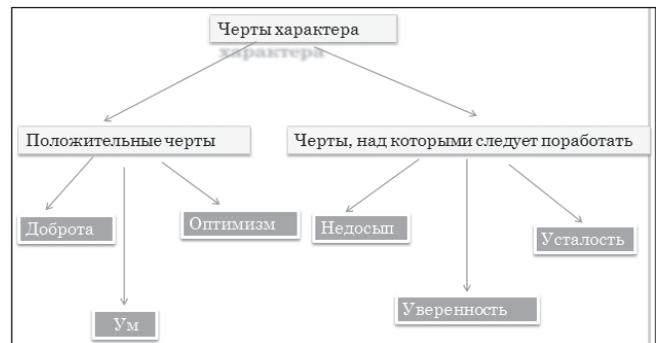


Рис. 4. Модель «Черты характера 3»

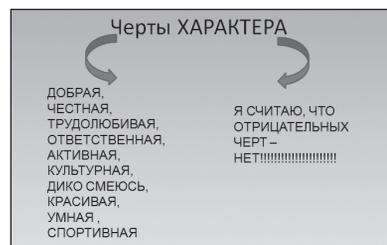


Рис. 5. Модель «Черты характера 4»



Рис. 6. Модель «Черты характера 5»

Тема «Оптимизационное моделирование», XI класс.

При изучении этой темы проектная деятельность очень эффективна. Старшеклассники самостоятельно разрабатывают задачу, которая решается с помощью оптимизационного моделирования, и представляют ее решение на уроке.

Тема «Технология хранения, поиска и сортировки информации», X класс.

Эта тема дает возможность интересной работы над проектом «Проектирование реляционных баз данных».

Предварительно старшеклассники получают домашнее задание: создать структуру реляционной базы данных об объектах реального мира, состоящей не менее чем из трех таблиц.

Работая над этим проектом, школьники осознают сложность проектирования базы данных: необходимость абстрактного представления реальных объектов и связей между ними в терминах таблиц, полей, атрибутов, записей. И самое трудное — представление системы объектов реального мира в виде реляционной модели.

Работа с реляционными базами данных требует от учителя большого профессионализма и мастерства: нарушение целостности данных не позволяет создать связи между таблицами, а самостоятельно найти такие логические ошибки могут не все учащиеся. Краткосрочность проекта требует от учителя не только предметных знаний в области разработки реляционных баз данных, но и быстроты реакции в оказании квалифицированной помощи по устранению ошибок в структуре модели.

Выбор темы моделирования школьники осуществляют согласно своим интересам:

- «Альбомы рок-групп»,
- «Экранизации художественных произведений»,
- «Музыкальные альбомы»,
- «Планшеты»,
- «Компьютеры»,
- «Карандаши»,
- «Части речи»,
- «Породы собак»,
- «Компьютерные игры»,
- «Спортивные игры»,
- «Олимпиады» и др.

На рисунках 7–8 представлены схема данных и одна из трех связанных таблиц реляционной базы данных «Части речи». Создание такой модели требует демонстрации предметных результатов, специфичных для предметных областей «Филология» и «Математика и информатика»:

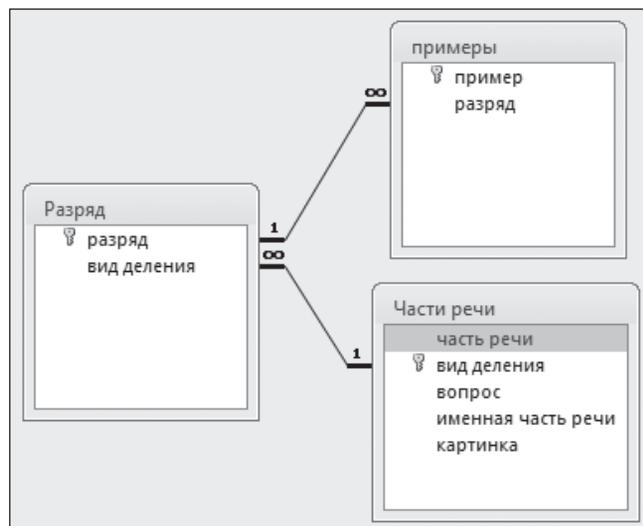


Рис. 7. Схема данных реляционной базы данных «Части речи»

| разряд | вид деления |
|-----------------|--------------------|
| возвратное | разряд местоимения |
| вопросительные | разряд местоимения |
| личные | разряд местоимения |
| неопределенные | разряд местоимения |
| определительные | разряд местоимения |
| относительные | разряд местоимения |
| отрицательные | разряд местоимения |
| притяжательные | разряд местоимения |
| указательные | разряд местоимения |
| вопрос. | разряд частиц |
| пример | Щелкните |
| ли (ль) | |
| неужели | |
| разве | |
| что ли | |
| * | |
| восклицательные | разряд частиц |
| ограничительные | разряд частиц |
| отриц. | разряд частиц |
| сомнение | разряд частиц |
| указательн. | разряд частиц |
| усилительные | разряд частиц |
| уточняющие | разряд частиц |
| формообразующие | разряд частиц |
| * | |

Рис. 8. Таблица «Разряд» реляционной базы данных «Части речи»

Непременным условием проектной деятельности являются афиширование промежуточных результатов проекта (как рефлексия проектной деятельности), защита проекта, его оценка экспертной группой (для краткосрочных проектов экспертная группа может состоять из одноклассников автора проекта). Такая **организация оценивания** способствует формированию позитивной Я-концепции обучающегося, развитию информационной компетентности, адекватной самооценки, воспитывает открытость и тактичность.

Критерии оценки учебного проекта должны быть известны обучающимся до выполнения проекта, могут составляться учителем вместе с обучающимися и, как правило, включают следующие показатели:

- актуальность;
- практическая значимость;
- технологическая сложность;
- оригинальность идеи;
- дизайн;
- завершенность;
- интерактивность.

* * *

В чем-то учителю информатики стало проще работать с детьми поколения Z: они — как самообучающиеся нейронные сети. Если вдруг в классе оказывается ученик, у которого что-то не получается в компьютерном практикуме, «ближайшие к нему узлы сети» с удовольствием

помогут соседу, и уже хитро горят глаза у всего класса: задание выполнено.

И все же, какой он, современный урок информатики? Стал ли он другим тридцать лет спустя? Стали ли другими дети и учителя? И как современный урок информатики помогает познать окружающий мир и самих себя?

Литературные и интернет-источники

1. *Зенцова Л. В.* Ценностные ориентации поколения Z и результат долгосрочного проекта «Обучение предмету» на уроках информатики // Актуальные аспекты устойчивого развития образовательных организаций в условиях введения ФГОС нового поколения: сборник тезисов XVIII Регионального научно-педагогического симпозиума (27 марта, 2015 г.). Иркутск: Изд-во ИРНИТУ, 2015.

2. *Фомина С.* Евгения Шамис: «Шпаргалки поколения X: как воспитывать новое поколение?» // Letidor — онлайн-проект для современных и умных родителей. http://letidor.ru/article/evgeniya_shamis_shpargalki_pok_117053/

НОВОСТИ

Самоуправляемые автомобили Ford научились ездить в полной темноте

В Ford объявили, что благодаря новому лазерному дальномеру самоуправляемый седан Fusion Hybrid может ездить даже в абсолютной темноте. Такие дальномеры работают по принципу испускания направленного луча света и оценки характеристик отраженного луча. При этом может использоваться свет как видимого, так и не видимого глазу человека диапазона. В Ford подчеркивают, что новый дальномер оказался достаточно надежным, чтобы автомобиль, управляемый созданной

в компании программной системой «виртуальный водитель», мог уверенно передвигаться по извилистым дорогам без использования видеокamer; машине не нужен дневной свет и средства распознавания дорожной разметки. В Ford также сообщили, что в этом году собираются увеличить парк самоуправляемых седанов Fusion Hybrid втрое, до 30 машин, которые будут испытываться на дорогах Калифорнии, Аризоны и Мичигана.

Первое ДТП, виновником которого стал беспилотный автомобиль Google

В феврале произошло первое ДТП, виновником которого стал беспилотный автомобиль Google, ехавший в полностью автономном режиме. Аварий с участием машин-роботов Google было уже немало, но все они, как утверждалось, становились следствием ошибки человека. Согласно отчету полиции, внедорожник Lexus RX450h, управляемый искусственным интеллектом, ехал в правом ряду широкого проспекта, по три полосы в каждом на-

правлении. Перед перекрестком Lexus дал сигнал правого поворота, но путь ему преградили мешки с песком, уложенные вокруг ливневого стока. Начав объезжать их, робот въехал в левый бок автобуса, который в это время двигался по центральному ряду. И водитель, находившийся для подстраховки в Lexus, и искусственный интеллект понадеялись, что автобус притормозит и пропустит работа.

Интернет рубашек и кроссовок: предметы одежды и обуви получают онлайн-профили

Evrything, компания, занимающаяся технологиями Интернета вещей, и производитель упаковки Avery Dennison будут снабжать одежду и обувь популярных марок, часто становящуюся объектом подделки, идентификационными радиометками либо штрихкодами с информацией, по которой покупатели и продавцы смогут устанавливать происхождение товара. Новшество может помочь бороться с мошенническими возвратами товаров в магазины: Janela будет хранить информацию

о том, где и когда было куплено изделие. Будет польза от платформы и для потребителей: приложения для смартфона, прочитав ярлык, смогут рассказать о том, где и из каких материалов изготовлен товар и как его отправить на вторичную переработку по окончании срока службы. Через приложение можно будет получать скидки и дополнительные сервисы, а также заказывать новые экземпляры самого товара или другие, имеющие к нему отношение.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

Р. Р. Мухаметзянов,

Набережночелнинский институт социально-педагогических технологий и ресурсов, Республика Татарстан

ОБ ИЗУЧЕНИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ НА ПРИМЕРЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ C#*

Аннотация

Одной из главных задач учителя информатики при изучении программирования согласно ФГОС является развитие алгоритмического мышления учащихся, необходимого для профессиональной деятельности в современном обществе. Данная задача реализуется, прежде всего, за счет изучения раздела «Алгоритмизация и программирование» и конкретного языка программирования. Одним из вариантов повышения мотивации и качества подготовки учащихся в области программирования является переход на изучение современного объектно-ориентированного стиля программирования. В статье рассматривается обучение объектно-ориентированному программированию на основе языка C#.

Ключевые слова: алгоритмизация, программирование, объектно-ориентированное программирование, язык программирования C#, класс, наследование классов, методы класса, свойства класса, конструкторы класса.

Контактная информация

Мухаметзянов Рамиль Рафаилович, канд. пед. наук, доцент кафедры информатики и вычислительной математики, декан факультета математики и информатики Набережночелнинского института социально-педагогических технологий и ресурсов, Республика Татарстан; *адрес:* 423838, Республика Татарстан, г. Набережные Челны, пер. Железнодорожников, д. 9А; *телефон:* (855-2) 46-71-15; *e-mail:* mrr-nisptr@mail.ru

R. R. Mukhametzyanov, Naberezhnye Chelny Institute of Social-Pedagogical Tech, The Republic of Tatarstan

ABOUT STUDYING OBJECT ORIENTED PROGRAMMING IN SECONDARY SCHOOL ON THE EXAMPLE OF C# PROGRAMMING LANGUAGE

Abstract

One of the main tasks of the teacher in the teaching programming in accordance with the FSES is the development of algorithmic thinking of students required for the professional activity in modern society. This task is realized, first of all, by studying the section "Algorithmization and programming" and a specific programming language. One of the options for increasing the motivation and the quality of training in the field of programming is to move students to the study of modern object oriented programming style. The article discusses the training of object oriented programming based on C# language.

Keywords: algorithmization, programming, object oriented programming, C# programming language, class, class inheritance, methods of class, properties of class, constructors of class.

С момента появления школьного курса «Основы информатики и вычислительной техники» в преподавании информатики в средней общеобразовательной школе изучению программирования отводилась особая роль. Не было ни одного поколения государственных стандартов и учебников в области информатики, где бы совсем не было программирования. Федеральные государственные образовательные стандарты (ФГОС) основного общего образования и среднего общего образования в этом плане нас тоже не разочаровали. Вопреки мнению многих скептиков, которые категорически против изучения программирования на уроках информатики всеми учащимися, данные стандарты четко предписывают изучение программирования в средней школе. Главными задачами учителя информатики при изучении программирования согласно ФГОС являются: развитие алгоритмического мышления учащихся, необходимого для профессиональной деятельности в современном обществе; развитие у учащихся умений составить и записать алгоритм для конкретного исполнителя; формирование у учащихся знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство учащихся с одним из языков программирования и основными алгоритмическими структурами — линейной, условной и циклической [6]. Традиционным для системы общего образования нашей страны является изучение императивного программирования с использованием таких языков, как Pascal, Basic, Python, C и др. Именно эти языки программирования плюс алгоритмический включены в ЕГЭ по информатике 2016 года.

Выбор языка программирования для обучения школьников полностью отдается на откуп самому учителю информатики. Именно он решает, какой язык лучше использовать для развития алгоритмического мышления и изучения основных алгоритмических конструкций программирования. Какими же критериями должен он руководствоваться при этом?

Прежде всего, это, конечно, профессиональные знания, умения и навыки самого педагога: для обучения учитель выберет тот язык программирования, который сам хорошо знает и на котором умеет программировать.

Другим критерием для многих учителей информатики сегодня является существование учебников, методических пособий и разработок по данному языку программирования. Несомненным лидером по этим двум критериям является язык Pascal.

Встречаются в школах нашей страны учителя информатики, которые в первую очередь обращают внимание на актуальность языка программирования. Сегодня в мире очень много популярных языков программирования. Если посмотреть статистику на сайте www.tiobe.com, можно обнаружить картину, представленную на рисунке 1 [5]. Этот неполный перечень языков говорит о том, что сегодня самыми популярными среди разработчиков являются такие языки программирования, как Java, C, C++, C#, Python. В первой пятерке языков четыре (за исключением языка Python) являются так называемыми С-подобными, три языка программирования относятся к категории объектно-ориентированных (Java, C++, C#). Язык программирования Pascal в этом рейтинге занимает далеко не первую позицию.

Таким образом, **можно выделить три основных критерия, которые используются учителями информатики нашей страны при выборе языка программирования в школе:**

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

| Aug 2015 | Aug 2014 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 2 | ▲ | Java | 19.274% | +4.29% |
| 2 | 1 | ▼ | C | 14.732% | -1.67% |
| 3 | 4 | ▲ | C++ | 7.735% | +3.04% |
| 4 | 6 | ▲ | C# | 4.837% | +1.43% |
| 5 | 7 | ▲ | Python | 4.066% | +0.95% |
| 6 | 3 | ▼ | Objective-C | 3.195% | -6.36% |
| 7 | 8 | ▲ | PHP | 2.729% | -0.14% |
| 8 | 12 | ▲▲ | Visual Basic .NET | 2.708% | +1.40% |
| 9 | 10 | ▲ | JavaScript | 2.162% | -0.01% |
| 10 | 9 | ▼ | Perl | 2.118% | -0.10% |
| 11 | 11 | | Visual Basic | 1.781% | -0.23% |
| 12 | 24 | ▲▲ | Assembly language | 1.760% | +1.11% |
| 13 | 13 | | Ruby | 1.416% | +0.17% |
| 14 | 18 | ▲▲ | Delphi/Object Pascal | 1.407% | +0.49% |

Рис. 1. Статистика сайта www.tiobe.com по рейтингу языков программирования в мире

- личные профессиональные знания, умения и навыки учителя информатики;
- наличие учебников и методических пособий по конкретному языку программирования;
- актуальность (современность, востребованность) языка программирования.

С одной стороны, кажется, что не имеет значения, на каком языке программировать в школе. Многие учителя информатики так и думают. Однако, если учесть, что, согласно распоряжению Правительства Российской Федерации от 6 января 2015 года № 7-р, большое количество специальностей и направлений подготовки по информационным технологиям попали в список приоритетных направлений модернизации и технологического развития российской экономики, и то, что учитель информатики должен вести профориентационную работу на данные специальности (об этом говорят ФГОС высшего образования по направлениям подготовки 44.03.01 и 44.03.05 «Педагогическое образование», компетенция ПК-5), реализуя тем самым преемственность в подготовке ИТ-специалистов и будущих учителей информатики, вопрос выбора языка программирования для школьного курса информатики становится отнюдь не праздным.

В вопросе выбора языка программирования нужно еще исходить из того, что сегодня существуют несколько основных парадигм программирования и, соответственно, языков программирования, реализующих эти парадигмы. **Парадигма программирования** — это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию). Это способ концептуализации, определяющий

организацию вычислений и структурирование работы, выполняемой компьютером [3].

Императивное программирование — это парадигма программирования, которая описывает процесс вычисления в виде инструкций, изменяющих состояние данных. Императивная программа очень похожа на приказы, выражаемые повелительным наклонением в естественных языках, т. е. это последовательность команд, которые должен выполнить компьютер. К императивным языкам программирования можно отнести Pascal, Basic, C и др.

Функциональное программирование — это раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании). К наиболее известным функциональным языкам программирования сегодня относятся Haskell, Ocaml, Scheme, Erlang, F# и др.

Логическое программирование — парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода. Самым известным языком логического программирования является Prolog.

Объектно-ориентированное программирование (ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов. Яркими представителями данной категории языков программирования являются Java, C#, Visual

Basic .Net. Только не стоит путать последний с языком Visual Basic, который также используется учителями информатики. Его мы не считаем классическим объектно-ориентированным, так же как и столь популярный в последние годы Delphi. В Delphi, конечно, можно создавать классы, но в нем не реализованы все возможности классического ООП. Кроме того, создание класса в этом языке, например, после Java или C# настолько неудобно и непонятно, что мы бы не рекомендовали его для изучения ООП. Создание приложений с графическим интерфейсом пользователя в Visual Basic и Delphi с использованием компонентов типа Button, Edit, Label ни в коем случае нельзя называть объектно-ориентированным программированием. Это скорее визуальное программирование.

Ничуть не умаляя возможностей императивного программирования, хотелось бы рассказать о **возможностях объектно-ориентированного программирования на примере языка C#**. Словосочетание «объектно-ориентированное программирование» для многих учителей информатики звучит каким-то волшебным, далеким и сложным словосочетанием, тем более если они сами не изучали ООП в педагогическом вузе при получении профессионального образования. Это отталкивает многих педагогов от этой парадигмы и соответствующих языков программирования. Не стоит пугаться. Наоборот, объектно-ориентированный стиль программирования настолько близок к человеческому стилю мышления, что программировать на объектно-ориентированных языках — одно удовольствие.

Прежде чем начать рассмотрение вопросов, связанных с созданием классов и объектов на языке C#, а именно эти вопросы мы считаем главными при изучении объектно-ориентированного программирования, необходимо выделить **несколько главных особенностей данного языка программирования**:

- Во-первых, язык является регистрочувствительным в отличие от языка Pascal.
- Во-вторых, один оператор от другого отделяется символом «точка с запятой», что делает этот язык похожим на популярный язык программирования Pascal.
- В-третьих, при введении пользовательских элементов программы (таких как переменные, классы, методы и др.) в названии рекомендуется использовать нотацию «camel case», т. е., если, например, название класса состоит из двух или более слов, мы их пишем слитно, каждое новое слово начинаем с заглавной буквы. Таким образом, получаем «верблюда» (camel) из одного, двух и т. д. «горбов».
- Четвертая особенность языка заключается в его строгой типизации, так же как и в знакомом нам всем языке программирования Pascal.

Язык C# и связанную с ним среду .NET Framework можно без преувеличения назвать самой значительной из предлагаемых в настоящее время технологий для разработчиков. C# — это язык, основанный на современной объектно-ориентированной методологии проектирования, при разработке которого специалисты из компании Microsoft опирались на опыт создания подобных языков, построенных в соответствии с предложенными около 20 лет назад объектно-ориентированными принципами. Язык C# был разработан корпорацией Microsoft в конце 90-х годов прошлого столетия как часть общей стратегии платформы .NET.

Впервые он был выпущен в виде альфа-версии в середине 2000 года. Главным разработчиком языка C# является Андерс Хейльсберг — датский ученый в области информатики, один из ведущих в мире специалистов по языкам программирования. Язык C# непосредственно связан с языками C, C++ и Java. Благодаря тому что C# построен на столь прочном и понятном основании, перейти на этот язык с C, C++ или Java не представляет особого труда. Язык программирования C уже несколько лет как прочно вошел в школьный курс информатики и включен в перечень языков программирования для решения заданий ЕГЭ по информатике. Поэтому изучение языка программирования C# будет являться логическим продолжением программирования на C.

Описание языка мы решили построить, не акцентируя внимания на его синтаксисе, так как последний фактически унаследован от языка C. При рассмотрении объектно-ориентированного программирования самым важным, на наш взгляд, является **процесс разработки пользовательских классов**. При описании процесса создания классов на языке C# мы решили сосредоточить внимание на нескольких основных моментах:

- создание классов (поля, методы класса);
- наследование классов;
- конструкторы класса;
- создание объектов класса.

Как говорят программисты, *объектно-ориентированное программирование держится на трех «ки-тах»: наследование, инкапсуляция и полиморфизм*. Не вникнув в чуть этих понятий и не научившись их использовать на практике, человек не может утверждать, что он обладает объектно-ориентированным стилем программирования.

Типичной ошибкой многих педагогов является то, что, если они пишут программный код на языке C#, они считают, что уже знают и применяют объектно-ориентированный стиль программирования. Уверяем вас, что это далеко не так. Прежде чем приступить к изложению материала по классам, хотелось бы сказать о том, что мы будем рассматривать только создание консольных приложений и рекомендуем изучение ООП начинать именно с них. Разработка приложений типа windows forms сильно отвлекает от сути изложения данной темы, т. е. от создания пользовательских классов на языке C#.

Еще один момент, на котором хочется заострить внимание читателя, — это **выбор среды разработки**. Здесь долго думать и выбирать не приходится, так как существует замечательная среда разработки Visual Studio от компании Microsoft. Версия MS Visual Studio Express Edition является совершенно бесплатной для образовательных целей, так что любые школа, колледж или вуз могут его скачать в сети Интернет и установить в компьютерных классах или дома у обучающихся. Самая свежая версия на момент написания данной статьи MS Visual Studio 2015, хотя ничто не мешает использовать и версии MS Visual Studio 2008, 2010, 2012, 2013. Для нас это не принципиально.

Создание классов в C#

Класс определяется с помощью ключевого слова class:

```
[модификатор доступа] class Название_класса
{
    тело класса
}
```

В качестве тела класса выступают поля, методы, конструкторы, свойства и др.

Рассмотрим небольшой пример. Заметим, что здесь самым главным является приведение хорошего примера из обыденной жизни, чтобы учащиеся могли всё понять. Возьмем пример создания класса Dog (собака). У всех собак есть порода, цвет, возраст и кличка. Поэтому можно определить класс с такими полями. Мы говорим именно «поля класса», так как это принято в языке C#, хотя в обыденной жизни мы, скорее всего, сказали бы «свойства класса». Но свойства класса в C# имеют немного другой смысл и предназначение. О них мы будем говорить далее.

Итак, для начала определим наш класс Dog следующим образом:

```
class Dog
{
    public string nickname; //кличка
    public string breed; //порода
    public string color; //цвет
    public int age; //возраст
    public double weight; //вес
}
```

При определении полей класса кроме названия обязательно нужно указать тип переменной (string — строковый, int — целочисленный, double — вещественный типы соответственно). Перед типами полей класса указан модификатор доступа public, который говорит о том, что все поля открытые, т. е. к ним можно получить прямой доступ из любого класса программы. Можно, в принципе, ограничиться полями класса, но на практике часто используют классы с методами, которые как раз и позволяют ограничить доступ к некоторым полям класса, реализуя так называемый *принцип инкапсуляции*.

Продолжим разработку нашего класса. Добавим в наш класс простой метод — Say («говорить», или, правильнее будет, «издавать звук»). Почему мы не употребляем метод «лаять», вы поймете в дальнейшем при изложении механизма наследования классов. Итак, любая собака может издавать звук, поэтому наш класс изменится следующим образом:

```
class Dog
{
    string nickname; //кличка
    string breed; //порода
    string color; //цвет
    int age; //возраст
    double weight; //вес
    void Say()
    {
        Console.WriteLine("Гав-гав!!!");
    }
}
```

Методы класса описывают поведение *объектов класса*. Реализация нашего метода Say в классе Dog достаточно примитивна, так как в жизни лай — это издавание звука. Лай в нашем классе будет всего лишь выводить на экран сообщение «Гав-гав!!!». Как мы знаем, все собаки могут лаять, еще они могут бегать, есть, спать, прыгать и т. д. С точки зрения программирования метод класса — это, по сути, есть не что иное, как *функция*. В объектно-ориентированном программировании функции могут существовать сами по себе, отдельно от класса, поэтому говорят именно о методах класса. Ключевое слово «void» — это пустой тип. Его используют, если метод ничего не возвращает (аналог процедуры из

языка Pascal). Кстати говоря, в языке программирования функцию, которая ничего не возвращает, называют *процедурой*. В языке программирования C# понятие процедуры не используется и заменяется функцией, которая возвращает значение void. Тело метода класса заключается в фигурные скобки. Console — это встроенный класс языка C#, который помогает при создании именно консольных приложений. В нашем случае метод класса WriteLine используется для вывода информации на консоль с последующим переводом курсора на начало новой строки. Согласитесь, напоминает оператор writeln языка Pascal. Теперь можно сказать, что мы получили класс Dog с полями и методом. Но класс сам по себе редко находит применение в программах. В практике программирования, так же как в обыденной жизни, мы имеем дело с конкретными объектами класса. Ведь мы не включаем класс «компьютер», а запускаем конкретный компьютер, который стоит на компьютерном столе. Мы не кормим класс «собака», а балуем косточкой или мясом свою любимую Жучку или Шарика. Поэтому следующим шагом при изучении ООП для нас является создание объектов, или, как говорят зачастую, *экземпляров класса*.

Создание объектов класса в C#

Для создания объекта класса используется следующая запись:

```
Название_класса Название_объекта =
new Конструктор_класса;
```

Пока мы не знаем, что такое конструктор класса, поэтому скажем лишь, что запись простого (пустого) конструктора представляет из себя название «класс» и пустые скобки. Исходя из вышесказанного, создание объекта класса Dog можем записать следующим образом:

```
Dog d1 = new Dog();
```

где d1 — это имя объекта в памяти компьютера.

Для инициализации полей класса (задание значений полям) используется традиционная точечная нотация, которая характерна для всех объектно-ориентированных языков программирования. Опишем нашего Шарика:

```
d1.nickname = "Шарик";
d1.breed = "овчарка";
d1.color = "черный";
d1.age = 7;
d1.weight = 12.7;
```

Теперь можем даже заставить его лаять:

```
d1.Say();
```

Аналогичным образом мы можем определить еще одну собаку:

```
Dog d2 = new Dog();
d2.nickname = "Тузик";
d2.breed = "доберман";
d2.color = "белый";
d2.age = 5;
d2.weight = 9.4;
d2.Say();
```

Как говорят программисты, класс является шаблоном для создания однотипных объектов и позволяет в разы сократить объем программного кода. Однако такой подход инициализации полей объекта является не совсем удобным, поэтому теперь имеет смысл рассмотреть понятие конструктора класса.

Определение конструкторов класса

Конструктор класса — это специальный метод класса, который позволяет создавать объекты класса. Еще один момент, о котором нужно всегда помнить, — название конструктора класса всегда совпадает с названием самого класса.

Определим для нашего класса Dog для начала первый конструктор:

```
public Dog(string nn, string br, string cl,
int ag, double wgh)
{
    nickname = nn;
    breed = br;
    color = cl;
    age = ag;
    weight = wgh;
}
```

Обращаем внимание на ключевое слово «public» перед названием конструктора класса. Это так называемый **модификатор доступа**. Данный модификатор мы уже использовали при определении полей нашего класса.

Вообще говоря, в языке C# применяются следующие модификаторы доступа:

- **public**: публичный, общедоступный класс или член класса. Такой член класса доступен из любого места в данной программе, а также из других программ и сборок;
- **private**: закрытый класс или член класса. Представляет собой полную противоположность модификатору public. Такой закрытый класс или член класса доступен только из кода в том же классе или контексте;
- **protected**: такой член класса доступен из любого места в текущем классе или в производных классах;
- **internal**: класс и члены класса с подобным модификатором доступны из любого места кода в той же сборке, однако он недоступен для других программ иборок (как в случае с модификатором public);
- **protected internal**: совмещает функционал двух модификаторов. Классы и члены класса с таким модификатором доступны из текущей сборки и из производных классов.

Объявление полей класса без модификатора доступа равнозначно их объявлению с модификатором private. Классы, объявленные без модификатора, по умолчанию имеют доступ internal. Так как предполагается, что конструктор класса будет вызываться из другого класса, мы объявляем его как public. Описание класса в таком виде не совсем удобно, так как вводит новые переменные для аргументов метода. Чаще всего на практике большинство программистов поступают иначе:

```
public Dog(string nickname, string breed,
string color, int age, double weight)
{
    this.nickname = nickname;
    this.breed = breed;
    this.color = color;
    this.age = age;
    this.weight = weight;
}
```

Поскольку имена параметров и имена полей в данном случае совпадают, то мы используем ключевое слово «this». Это ключевое слово представляет ссылку

на текущий класс. Поэтому в выражении «this.breed = breed» первая часть «this.breed» означает, что «breed» — это поле текущего класса, а не название параметра breed. Если бы у нас параметры и поля назывались по-разному, то использовать слово «this» было бы необязательно. Теперь полное определение класса выглядит следующим образом:

```
class Dog
{
    string nickname; //кликча
    string breed; //порода
    string color; //цвет
    int age; //возраст
    double weight; //вес
    public void Say()
    {
        Console.WriteLine("Гав-гав!!! ");
    }
    public Dog(string nickname, string breed,
string color, int age, double weight)
    {
        this.nickname = nickname;
        this.breed = breed;
        this.color = color;
        this.age = age;
        this.weight = weight;
    }
}
```

Теперь создание объектов нашего класса выглядит гораздо проще за счет использования конструктора:

```
Dog d3=new Dog("Мухтар", "бультерьер",
"коричневый", 8, 16.8);
```

Однако тут возникает одна небольшая проблема. Как быть, если нам необходимо создать объект класса Dog, но мы знаем только кличку и породу, а остальные свойства нам пока неизвестны? Разработанный конструктор не позволяет этого сделать. Теперь не получится создать и отдельно проинициализировать поля класса, как в первом случае, так как в языке C# после определения собственного конструктора «пусто» конструктор необходимо определить явно. Пустой конструктор неявно определен в классе только в том случае, если пользователь не создал ни одного конструктора самостоятельно. Выходом из нашей ситуации является определение другого конструктора, который позволил бы нам создать объект класса Dog только по кличке и породе. К счастью, язык C# позволяет пользователю определять неограниченное количество конструкторов класса. Нужно только подключить фантазию.

Добавляем новый конструктор в класс Dog:

```
public Dog(string nickname, string breed)
{
    this.nickname = nickname;
    this.breed = breed;
}
```

Теперь окончательное определение нашего класса выглядит следующим образом:

```
class Dog
{
    string nickname; //кликча
    string breed; //порода
    string color; //цвет
    int age; //возраст
    double weight; //вес
    public void Say()
    {
```

```

    Console.WriteLine("Гав-гав!!! ");
}
public Dog(string nickname, string breed,
string color, int age, double weight)
{
    this.nickname = nickname;
    this.breed = breed;
    this.color = color;
    this.age = age;
    this.weight = weight;
}
public Dog(string nickname, string breed)
{
    this.nickname = nickname;
    this.breed = breed;
}
}

```

Свойства класса

Кроме обычных методов класса в языке C# предусмотрены специальные методы доступа, которые называются *свойствами*. Они обеспечивают простой доступ к полям класса, позволяют узнать их значение или, наоборот, выполнить установку их значения. Свойства позволяют реализовать принцип инкапсуляции. Стандартное определение свойства содержит блоки get и set. В блоке get мы возвращаем значение поля, а в блоке set устанавливаем его. Параметр value представляет передаваемое значение. Поля нашего класса не имеют модификаторов доступа. Это значит, что все они приватные, т. е. доступные только из методов этого же класса. Если нам потребуется изменить значение какого-либо из полей, нам нужно определить свойство для конкретного поля класса.

Добавим свойство для поля Age (возраст) класса Dog:

```

public int Age
{
    get
    {
        return age;
    }
    set
    {
        if (value > 0)
            age = value;
    }
}

```

Помимо того что свойство позволяет нам напрямую установить и считать значение возраста собаки, мы можем реализовать логику присваивания значения возраста собаки. В нашем случае возраст никак не может быть отрицательным числом. Обычно название свойства пишут так же, как название поля, но с заглавной буквы. Это правило, которого придерживаются большинство программистов в мире.

Наследование классов

Большие возможности объектно-ориентированного программирования проявляются за счет использования *принципа наследования*.

Например, нам требуется определить класс «кошка». У кошки, так же как и у собаки, есть кличка, порода, цвет, возраст, вес, но звуки издает она по-другому (она мяукает). Обычно на диаграммах классов наследование изображают в следующем виде (рис. 2) — стрелки направляют от наследника к родительскому классу.



Рис. 2. Диаграмма наследования классов

Примеров наследования в нашей обыденной жизни и окружающем нас мире настолько много, что мы просто не обращаем на них внимания. Например, для классов «книга» и «журнал» родительским может быть класс «печатное издание», для классов «яблоня» и «береза» в качестве суперкласса может быть класс «дерево», и т. д. Класс, от которого идет наследование, в терминологии ООП называют обычно *базовым классом*, *родительским классом*, *классом-предком* или *суперклассом*. Классы, которые наследуются, называют *дочерними*, *классами-потомками*, *производными классами* или *подклассами*. Механизм наследования позволяет сократить объем программного кода и делает его более читабельным.

Реализуем принцип наследования для нашей задачи. Для этого сначала определим класс Pet (домашнее животное), дабы объединить всех собак, кошек, хомячков и т. д.:

```

class Pet
{
    public string nickname; //кличка
    public string breed; //порода
    public string color; //цвет
    public int age; //возраст
    public double weight; //вес
    public void Say()
    {
        //издает какой-то звук, пока не знаем, какой
    }
    public Pet(string nickname, string breed,
string color, int age, double weight)
    {
        this.nickname = nickname;
        this.breed = breed;
        this.color = color;
        this.age = age;
        this.weight = weight;
    }
}

```

Сам по себе этот класс нам не нужен. Нам нужны классы для определения собак и кошек. Определим сначала класс Dog:

```

class Dog: Pet
{
    public override void Say()
    {
        Console.WriteLine("Гав-гав!!! ");
    }
}

```

После двоеточия мы указываем базовый класс для данного класса. Для класса Dog базовым является класс Pet, и поэтому класс Dog наследует все те же свойства, методы, поля, которые есть в классе Pet. Для этого метод, который надо переопределить, в базовом классе помечается модификатором virtual, а при переопреде-

лении в производном классе — модификатором `override`. Поэтому в классе `Pet` меняем метод `Say()`:

```
public virtual void Say()
{
}
```

Благодаря наследованию теперь можем определить и класс `Cat` (кошка):

```
class Cat: Pet
{
    public override void Say()
    {
        Console.WriteLine("Мяу-мяу!!! ");
    }
}
```

В качестве самостоятельных заданий можно предложить учащимся разработку классов на различную тематику. При этом нужно предлагать задания с учетом возможности наследования между классами.

Изучение объектно-ориентированного программирования является, на наш взгляд, сегодня одним из приоритетных направлений при изучении программирования в школьном курсе информатики и ИКТ. От того, насколько правильно будет организована методика изучения этой парадигмы программирования в школе, и от того, насколько правильно будут сформированы у школьников такие понятия, как «класс», «объект», «наследование», «полиморфизм», «инкапсуляция» и др., зависит будущая подготовка ИТ-специалистов в вузах и ссузах Российской Федерации. Язык `C#` является сегодня одним из самых

популярных и лучших объектно-ориентированных языков программирования и продолжает линию `C`-подобных языков программирования. Рейтинг популярности языка растет с каждым годом. Наличие бесплатной учебной версии среды разработки `MS Visual Studio` позволяет всем нам беспрепятственно использовать именно его на уроках информатики при изучении объектно-ориентированного программирования и решать задачи по формированию как предметных, так и метапредметных и даже личностных результатов обучения.

Литературные и интернет-источники

1. Мухаметзянов Р. Р. Высокоуровневые методы информатики и программирования в Delphi: учеб. пособие. Набережные Челны: ФГБОУ ВПО «НИСПТР», 2013

2. Павловская Т. А. `C#`. Программирование на языке высокого уровня: учебник для вузов. СПб.: Питер, 2007.

3. Парадигма программирования. https://ru.wikipedia.org/wiki/Парадигма_программирования

4. Тузовский А. Ф. Высокоуровневые методы информатики и программирования. Томск: Изд-во Томского политехнического университета, 2008.

5. TIOBE Index. <http://www.tiobe.com>

6. Федеральный государственный образовательный стандарт основного общего образования (в ред. Приказа Минобрнауки России от 29.12.2014 № 1644). http://минобрнауки.рф/документы/938/файл/749/приказ_Об_утверждении_1897.pdf

7. Федеральный государственный образовательный стандарт среднего общего образования (в ред. Приказа Минобрнауки России от 29.12.2014 № 1644). http://минобрнауки.рф/документы/2365/файл/736/12.05.17-Приказ_413.pdf

ПРАВИЛА ДЛЯ АВТОРОВ

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



Н. В. Лалетин,

Сибирский институт бизнеса, управления и психологии, г. Красноярск,

Д. Н. Лалетин,

лицей № 7, г. Красноярск

ШКОЛЬНЫЙ ЭТАП ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ В КРАСНОЯРСКОМ КРАЕ*

Аннотация

В статье представлены решения четырех задач, предложенных учащимся Красноярского края на школьном этапе всероссийской олимпиады по информатике. Все представленные в статье задачи присутствуют на сайте: <https://acmp.ru> со встроенной проверяющей системой и доступны для самостоятельного решения.

Ключевые слова: олимпиада школьников, информатика, программирование, олимпиадные задачи по информатике.

Контактная информация

Лалетин Николай Викторович, канд. тех. наук, доцент, зав. кафедрой прикладной математики и информатики Сибирского института бизнеса, управления и психологии, г. Красноярск; *адрес:* 660069, г. Красноярск, ул. Московская, д. 7а; *телефон:* (391) 264-55-29; *e-mail:* LaletinNV@mail.ru

N. V. Laletin,

The Siberian Institute of Business, Management and Psychology, Krasnoyarsk,

D. N. Laletin,

Lyceum 7, Krasnoyarsk

THE SCHOOL STAGE OF THE RUSSIAN COMPUTER SCIENCE OLYMPIAD IN KRASNOYARSK KRAI

Abstract

The article presents the solutions of four tasks proposed for the schoolchildren of Krasnoyarsk krai at the school stage of the Russian Olympiad in informatics. All mentioned tasks are available for individual work at the web-site: <https://acmp.ru> which is provided with built-in test system.

Keywords: Olympiad for schoolchildren, informatics, programming, Olympiad tasks in informatics.

Красноярский край всегда отличался высоким уровнем информационного развития, богатыми традициями и опытом в области информатизации системы образования и развития школьной информатики. С 1987/1988 учебного года ежегодно проводятся районные и краевые олимпиады школьников. В январе 1988 года состоялась первая окружная олимпиада. А с 21 по 25 марта 1989 года в Красноярске прошла I Всероссийская олимпиада школьников по информатике, являвшаяся республиканским этапом Всесоюзной олимпиады. С 2007/2008 учебного года региональная предметно-методическая комиссия (ПМК) по информатике предлагает организаторам школьного и муниципального этапов проведение олимпиады по модели, предлагаемой центральной ПМК для регионального этапа, а именно — по единым заданиям, в один день [1]. Олимпиада проводится в форме интернет-олимпиады с использованием образовательного интернет-ресурса «Школа программиста» [4].

В соответствии с Порядком проведения всероссийской олимпиады школьников [3], школьный этап олимпиады по информатике и ИКТ проводится ежегодно в период с 1 сентября по 15 октября. В Красноярском крае учащимся были предложены два различных комплекта заданий для двух возрастных категорий: VII—VIII и IX—XI классы. Основному туру олимпиады предшествовал пробный тур, целью которого являлось ознакомление участников с программным обеспечением олимпиады. На основном туре учащимся IX—XI классов было предложено решить в течение четырех астрономических часов следующие четыре задачи:

| ID | Задача | Тема | Сложность |
|------|----------------------|--------------------------|-----------|
| 1264 | А. Сороконожка | Задачи для начинающих | 16 % |
| 0969 | В. Сверхстепень | Целочисленная арифметика | 30 % |
| 1265 | С. Флаги | Целочисленная арифметика | 34 % |
| 1054 | Д. Ладья в лабиринте | Теория графов | 54 % |

Под ID (сокр. от identification — идентификация) представлен идентификационный номер задачи на образовательном интернет-ресурсе «Школа программиста».

Средняя сложность задач составила 33 %.

Самой простой задачей олимпиады была задача А «Сороконожка», однако в связи с тем, что данную задачу можно

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_04-2016/

отнести к категории задач с подвохом, только 2 % участников удалось полностью с ней справиться, т. е. написать полное решение на 100 баллов.

Второй по сложности была задача В «Сверхстепень». Для успешного решения данной задачи требовались определенные знания из модулярной арифметики (арифметики остатков). Напомним, что модулярная арифметика заключается в том, что у любого выражения, в котором используются только целые числа, требуется вычислить не значение этого выражения, а остаток от деления этого значения на некоторое целое число m . В итоге 15 % участников олимпиады успешно справились с данной задачей.

Задача С «Флаги» — задача, схожая по сложности с задачей В. Для решения задачи требовалось промоделировать способы украсить витрину магазина для малых N ($N \leq 5$) и увидеть, что решение задачи сводится к нахождению удвоенных чисел Фибоначчи. Другая проблема, которую должны были решить участники, — это нахождение решения для $N > 90$, когда решением является достаточно большое число, не помещающееся в используемые большинством участников восьмидесятибитовые типы данных `long long` для C++ и `Int64` для Free Pascal. Около 5 % участников полностью справились с данной задачей.

Задача D «Ладья в лабиринте» — самая сложная на олимпиаде. Положительного результата при ее решении добились учащиеся, которые смогли реализовать один из алгоритмов обхода графа — поиск в ширину (обход по уровням). Лишь нескольким участникам олимпиады удалось получить максимальный балл по данной задаче.

Перейдем непосредственно к рассмотрению задач и их возможных решений.

Задача А. Сороконожка.

Ограничение по времени: 1 секунда.

Ограничение по памяти: 16 Мб.

У сороконожки 40 левых ножек и 40 правых ножек. Под кроватью у сороконожки a левых тапочек и b правых тапочек. Сороконожка, просыпаясь, надевает тапочки. Для этого она засовывает под кровать первую левую ножку и надевает первый попавшийся тапочек, тратя на это одну секунду. Если тапочек оказывается левым, то она переходит ко второй левой ножке. Если же он оказывается правым, она переодевает его на какую-нибудь необутую правую ножку, тратя еще одну секунду, т. е. всего на такой тапочек уходит две секунды. Если все правые ножки уже обуты, то она снимает тапочек и кидает его в угол комнаты, тратя на это одну секунду, т. е. на такой тапочек сороконожка тратит также две секунды. Процесс продолжается до тех пор, пока все левые ножки не окажутся в левых тапочках. Затем сороконожка аналогичным образом начинает надевать правые тапочки, продолжая до тех пор, пока не будут обуты все правые ножки.

Сегодня сороконожка встала не с той ножки, поэтому она готовится к худшему. Несмотря на это она, как обычно, начинает обуваться с левой ножки. Сколько секунд понадобится сороконожке на утреннее обувание в худшем случае?

Входные данные.

Входной файл содержит целые числа a и b ($40 \leq a, b \leq 10^{18}$).

Выходные данные.

В выходной файл выведите: сколько секунд понадобится в худшем случае сороконожке на утреннее обувание.

Пример.

| № | input.txt | output.txt |
|---|-----------|------------|
| 1 | 40 40 | 120 |

Решение.

Как уже было сказано выше, задачу можно отнести к категории задач с подвохом, т. е. задача требует определенного размышления и математического моделирования. Из условия задачи следует, что сороконожка обувает сначала левые 40 ног, потом правые 40 ног. Соответственно, нужно выбрать наихудший по времени способ обуть сороконожку. Рассмотрим два варианта обувания.

Вариант 1. Сначала «неудачно» пытаемся обуть левые ножки всеми правыми тапочками, в таком случае до левых ножек мы доберемся через b попыток, а поскольку после этого под кроватью останутся только левые тапочки, потратим еще 40 с для их обувания. Итого получаем время $t_1 = b \cdot 2 + 40$.

Вариант 2. Сначала нам попадается 39 правых тапочек. Потом 40 левых, затем мы пытаемся на оставшиеся правые ножки надеть все «неодевшие» левые тапочки и, в конце концов, последний правый. Итого получаем время $t_2 = 39 \cdot 2 + 40 + (a - 40) \cdot 2 + 1 = a \cdot 2 + 39$.

Решением задачи будет максимальное из значений t_1 и t_2 .

Один из возможных вариантов решения данной задачи на языках программирования C++ и Python 3 приведен ниже.

```
// C++
#include <bits/stdc++.h>
int main()
{
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    long long a, b;

    std::cin >> a >> b;
    std::cout << max(2 * a + 39, 2 * b + 40);

    return 0;
}

// Python 3.4.1
inp = open('input.txt', 'r')
out = open('output.txt', 'w')

a, b = map(int, inp.readline().split())
print(max(2 * a + 39, 2 * b + 40), file = out)

inp.close()
out.close()
```

Задача В. Сверхстепень.

Ограничение по времени: 1 секунда.

Ограничение по памяти: 16 Мб.

Назовем значение выражения 2^{2^n} n -й сверхстепенью числа 2. Таким образом, например, третья сверхстепень числа 2 равна $2^{2^3} = 2^8 = 256$.

Ваша задача — вычислить n -ю сверхстепень двойки по модулю m .

Входные данные.

Входной файл содержит целые числа: n ($0 \leq n \leq 10^5$) и m ($0 \leq m \leq 10^4$).

Выходные данные.

В выходной файл выведите одно целое число — ответ на поставленную задачу.

Пример.

| № | input.txt | output.txt |
|---|-----------|------------|
| 1 | 3 1000 | 256 |
| 2 | 10 10 | 6 |

Решение.

В основе решения задачи лежит тождество:

$$(2^{2^n})^2 = 2^{2^n} \cdot 2^{2^n} = 2^{2^n+2^n} = 2^{2^{n+1}}.$$

Представим n -ю сверхстепень числа 2 для малых значений n :

$n = 0$: решение — $2^{2^0} = 2$;

$n = 1$: решение — $2^{2^1} = 2^{2^0} \cdot 2^{2^0} = 2 \cdot 2 = 4$;

$n = 2$: решение — $2^{2^2} = 2^{2^1} \cdot 2^{2^1} = 4 \cdot 4 = 16$;

$n = 3$: решение — $2^{2^3} = 2^{2^2} \cdot 2^{2^2} = 16 \cdot 16 = 256$.

Видно, что для вычисления n -й сверхстепени необходимо начать с числа 2 и выполнить n повторных возведений в квадрат. Так как требуется вычислить остаток от деления результата на число m , то следует воспользоваться формулой из модулярной арифметики (арифметики остатков): $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.

Один из возможных вариантов решения данной задачи на языках программирования C++ и Python 3 приведен ниже.

```
// C++
#include <iostream>
using namespace std;

int main()
{
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int n, m, res;
    cin >> n >> m;
    res = 2 % m;
    for (int i = 1; i <= n; i++)
        res = res * res % m;
    return cout << res, 0;
}
```

```
// Python 3.4.1
inp = open('input.txt', 'r')
out = open('output.txt', 'w')

n, m = map(int, inp.readline().split())
res = 2 % m
for i in range(n):
    res = res * res % m
print(res, file = out)

inp.close()
out.close()
```

Задача С. Флаги.

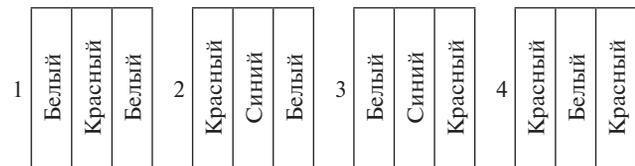
Ограничение по времени: 1 секунда.

Ограничение по памяти: 16 Мб.

В День Государственного флага России владелец магазина решил украсить свою витрину полосками ткани белого, синего и красного цветов. Он хочет, чтобы выполнялись следующие условия:

- полоски одного цвета не должны располагаться рядом друг с другом;
- синяя полоска может быть расположена только между белой и красной или между красной и белой.

Определите количество способов выполнить желание владельца магазина. Например, для $N = 3$ возможны следующие варианты:



Входные данные.

Входной файл содержит натуральное число N — количество полосок ($N \leq 92$).

Выходные данные.

В выходной файл выведите целое число — количество способов украсить витрину магазина.

Пример.

| № | input.txt | output.txt |
|---|-----------|------------|
| 1 | 1 | 2 |
| 2 | 3 | 4 |

Решение.

Для понимания методики решения данной задачи достаточно было просчитать решения для малых N ($N \leq 5$) и заметить, что решение задачи сводится к нахождению удвоенных чисел Фибоначчи. Многим участникам не удалось сдать два последних теста из-за того, что решение не помещалось в используемые большинством участников восьмидесятибитовые целочисленные типы данных `long long` для C++ и `Int64` для Free Pascal. В этом случае выход заключался в применении беззнаковых восьмидесятибитовых целочисленных типов данных `unsigned long long` для C++ и `QWord` для Free Pascal.

Один из возможных вариантов решения данной задачи на языках программирования C++ и Python 3 приведен ниже.

```
// C++
#include <iostream>
int main() {
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    unsigned long long a, b, n;
    a = b = 2;
    std::cin >> n;
    for (int i = 3; i <= n; i++)
        a = a + b,
        b = a - b;
    return std::cout << a, 0;
}
```

```
// Python 3.4.1
inp = open('input.txt', 'r')
out = open('output.txt', 'w')
```

```
n = int(inp.readline())
a = b = 2
for i in range(2,n):
    a = a + b
    b = a - b
print(a,file = out)

inp.close()
out.close()
```

Задача D. Ладья в лабиринте.

Ограничение по времени: 1 секунда.

Ограничение по памяти: 16 Мб.

Ладья — это шахматная фигура, которая за один ход может переместиться на любое количество клеток по горизонтали или вертикали. При этом она не может «перепрыгивать» через стоящие на ее пути фигуры.

Вася недавно соорудил на шахматной доске своеобразный лабиринт, поставив в некоторые клетки доски пешки (самые «слабые» шахматные фигуры). Теперь он хочет знать, за какое минимальное количество ходов ладья может добраться из одной клетки в другую, перемещаясь по свободным клеткам доски.

Он размышляет над этим вопросом уже несколько дней, однако найти ответ не может. Поэтому он решил обратиться за помощью к вам.

Напишите программу, находящую ответ на Васину задачу.

Входные данные.

Первая строка входного файла содержит два натуральных числа: n и m ($1 \leq n, m \leq 500$) — размеры лабиринта.

Каждая из последующих n строк содержит m символов. j -й символ i -й из этих строк соответствует клетке с координатами (i, j) . Он равен «.» (точка), если клетка пуста, «P» — если клетка занята пешкой, «S» — если это начальная клетка для ладьи, и «F» — если это конечная клетка.

Выходные данные.

В выходной файл выведите минимальное количество ходов, требуемое ладье для того, чтобы из начальной клетки попасть в конечную. Если конечная клетка недостижима из начальной, выведите -1 .

Пример.

| № | input.txt | output.txt |
|---|-------------------------------------|------------|
| 1 | 4 4 F.PS .PP. .PP. | 3 |
| 2 | 4 4 F.PS .PP. .PP. .P.. | -1 |

Решение.

Данная задача может быть сведена к нахождению длины кратчайшего пути в невзвешенном неориентированном графе. Лабиринт можно рассматривать в виде графа (доступные локации — вершины, ходы — ребра). Такой тип задач решается с помощью реализации алгоритма обхода графа — поиск в ширину. Данная задача

предлагалась для решения на одиннадцатой олимпиаде (базовый уровень) из цикла интернет-олимпиад для школьников сезона 2006/2007 учебного года [2]. В материалах данной олимпиады можно найти подробный разбор задачи и тесты к ней.

Один из возможных вариантов решения данной задачи на языке программирования Pascal приведен ниже.

```
// Free Pascal 2.6.4
type T_xy = record
    x,y:integer;
end;
const
    dx:array[1..4] of integer = (1, -1, 0, 0);
    dy:array[1..4] of integer = (0, 0, 1,-1);
    INF=500000;
var queue:array[1..250001] of T_xy;
    pl,pr:integer;
    n,m,xS,yS,xF,yF,i,j:integer;
    ch:char;
    a:array[0..501,0..501] of integer;

procedure BFS(x,y:integer);
var i:integer;
begin
    pl:=1;
    pr:=2;
    queue[pl].x:=x;
    queue[pl].y:=y;

    while pl<>pr do
        begin
            for i:=1 to 4 do
                begin
                    if a[queue[pl].y + dy[i],
                        queue[pl].x + dx[i]] = INF then
                        begin
                            j:=1;
                            repeat
                                a[queue[pl].y + dy[i]*j,queue[pl].x +
                                    dx[i]*j]:=a[queue[pl].y,queue[pl].x]+1;
                                queue[pr].y:=queue[pl].y + dy[i]*j;
                                queue[pr].x:=queue[pl].x + dx[i]*j;
                                pr:=pr+1;
                                j:=j+1;
                            until a[queue[pl].y + dy[i]*j,
                                queue[pl].x + dx[i]*j] <> INF
                            end;
                        end;
                    pl:=pl+1
                end;
            end;

Begin
assign(input, 'input.txt'); reset(input);
assign(output, 'output.txt'); rewrite(output);

readln(n,m);

for i:=0 to n+1 do
for j:=0 to m+1 do a[i,j]:=-1;

for i:=1 to n do
begin
for j:=1 to m do
begin
read(ch);
if (ch='P') then a[i,j]:=-1
else a[i,j]:=INF;
if (ch='S') then
begin yS:=i; xS:=j; a[i,j]:=0 end
else
if (ch='F') then
begin yF:=i; xF:=j; a[i,j]:=INF end;
end;
end;
```

```

readln
end;

BFS(xS,yS);

if (a[yF,xF]=INF) then write(-1)
    else write(a[yF,xF]);
End.

```

Попробовать сдать для проверки собственные решения к предложенным выше задачам можно на сайте: <https://acmp.ru/>, в разделе «Курсы → Региональные олимпиады → Школьный этап». Там же можно ознакомиться с комплектом заданий для VII—VIII классов.

Литературные и интернет-источники

1. *Беляев С. Н., Лалетин Н. В.* Ключевые факторы развития олимпиадного движения в Красноярском крае // Вестник Красноярского государственного педагогического университета им. В. П. Астафьева. 2010. № 2.
2. Интернет-олимпиады по информатике. <http://neerc.ifmo.ru/school/io/2006-2007.html>
3. Приказ Минобрнауки России от 18.11.2013 № 1252 (ред. от 17.03.2015) «Об утверждении Порядка проведения всероссийской олимпиады школьников» (Зарегистрировано в Минюсте России 21.01.2014 № 31060). http://www.consultant.ru/document/cons_doc_LAW_158227/
4. Школа программиста. <https://acmp.ru>

НОВОСТИ

Экзакомпьютер: первые наметки

На протяжении десятилетий увеличение вычислительной мощности суперкомпьютеров на три порядка через каждые десять лет, соответствующее закону Мура, в США воспринимали как должное. Но после появления в 2008 году систем петамасштаба постепенно стало ясно, что совершить следующий скачок и создать в тысячу раз более мощную систему будет очень сложно.

Поначалу некоторые считали, что компьютер экзамасштаба будет построен примерно через 10 лет, т. е. к 2018 году. Однако проблем становилось все больше и больше. Такому компьютеру требовалось очень много энергии, а для того чтобы приложения могли эффективно использовать невообразимый ранее уровень параллелизма при наличии сотен миллионов процессорных ядер, нужно было искать новые подходы. Еще одна проблема заключалась в обеспечении устойчивости — возможности продолжать работу в условиях многочисленных аппаратных сбоев, сопровождающих функционирование систем такого масштаба.

Теперь выйти на указанный рубеж предполагается к 2023 году, примерно через 15 лет после появления систем петамасштаба. Один экзафлопс — это производительность, равная квинтиллиону (миллиону триллионов) операций с плавающей запятой в секунду. В тысячу раз быстрее, чем петафлопс.

Другими словами, власти намерены создать систему, которая сможет решать вычислительные задачи в 50 раз быстрее доступных сейчас суперкомпьютеров производительностью 20 PFLOPS.

«Соединенные Штаты столкнулись сегодня с серьезными вызовами в области экономики, защиты окружающей среды и национальной безопасности, требующими срочного решения, — указывается в документе, подготовленном Министерством энергетики США для поставщиков компьютерных систем. — Над нами нависли энергетические и климатические угрозы, а также угрозы, связанные с безопасностью. Для того чтобы противостоять им, стране нужны высокопроизводительные вычислительные системы. Необходимо разрабатывать компьютеры экзамасштаба, которым отводится важнейшая роль в решении этих проблем».

Ожидается, что система такого масштаба будет потреблять 20–30 МВт электроэнергии, т. е. для ее энергообеспечения придется построить электростанцию.

Финансовые затраты тоже будут соответствующими. В 2016 году на разработку системы экзамасштаба правительство потратит почти 300 млн долл., а в 2017 году сумма расходов возрастет еще больше. Согласно планам Министерства энергетики США, общая стоимость создания системы экзамасштаба оценивается примерно в 3 млрд долл.

В деле построения суперкомпьютера с производительностью, измеряемой в экзафлопсах, США ведут соперничество с Китаем, Европой и Японией. По словам Джека Донгарры, профессора из Университета Теннесси и одного из составителей рейтинга Top500, Китай уже сейчас работает над двумя проектами по созданию систем производительностью 100 PFLOPS и в ближайшее время объявит об этом.

Квантовый компьютер стал реальнее

Ученые Университета Гриффита и Квинслендского университета создали квантовый вариант вентиля Фредкина — трехбитового логического элемента для управляемого обмена, он обменивает значение двух бит, когда третий равен единице. Вентиль Фредкина является универсальным — любую логическую или арифметическую операцию можно выполнить, поль-

зуясь системой, которая состоит только из таких вентилях. Ввиду высокого уровня сложности реализации сконструировать квантовый вариант до сих пор не удавалось. Таким образом, ученые приблизили возможность создания полноценного квантового компьютера, в разы уменьшив необходимое для этого количество логических элементов.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)



В. Ф. Очков,

Национальный исследовательский университет «МЭИ», г. Москва,

А. В. Соколов,

лицей № 1502 при Московском энергетическом институте

КАК ВЫ КОРАБЛЬ РАССЧИТАЕТЕ И НАЗОВЕТЕ, ТАК ОН И ПОПЛЫВЕТ

Аннотация

В статье обсуждаются вопросы расчета устойчивого плавания судов в пресной и морской воде на основе рассмотрения сил, действующих на судно. Предложена лабораторная работа, охватывающая физику, математику и информатику.

Ключевые слова: остойчивость судна, центр масс, центр водоизмещения, закон Архимеда, пресная вода, морская вода, плотность воды.

Контактная информация

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

Соколов Алексей Викторович, учитель физики и астрономии лицея № 1502 при Московском энергетическом институте, г. Москва; адрес: 111555, г. Москва, ул. Молостовых, д. 10А; телефон: (495) 307-11-61; e-mail: av_sokol_1502@mail.ru

V. F. Ochkov,
National Research University MPEI, Moscow,

A. V. Sokolov,
Lyceum 1502, Moscow

A SHIP WILL SAIL THE WAY YOU CALCULATED AND NAMED IT

Abstract

The article discusses the calculation of sustainable ships in fresh and seawater based on reviewed the forces acting on the ship. A laboratory work, covering physics, mathematics and informatics is given.

Keywords: ship stability, center of gravity, center of buoyancy, Archimedes' principle, fresh water, seawater, density of water.

Первый автор этой статьи работает в Международной ассоциации по свойствам воды и водяного пара (www.iarpws.org). Ее члены каждый год собираются в различных городах мира, тесно связанных с водой*. В 2014 году, например, такая встреча проходила в Москве**, а заключительный банкет, без которого не обходится ни одно научное мероприятие, состоялся на борту теплохода, курсировавшего под стенами Кремля. В 2015 году такая встреча была в Стокгольме, а финальный банкет прошел в музее легендарного шведского парусного корабля «Ваза», который был спущен на воду в 1628 году и сразу... затонул. В 1961 году корабль был поднят, законсервирован, подвергнут реставрации и в настоящее время экспонируется в специально построенном для него музее (рис. 1).



Рис. 1. Реставрированный корабль «Ваза»

Но встречи в рамках Международной ассоциации по свойствам воды и водяного пара, конечно, состоят не только из банкетов. Приходится и работать — рассматривать и утверждать документы, описывающие порядок расчета свойств этой важной субстанции [3]. Эти формуляции становятся стандартами для всех стран и по ним выпускаются справочники, создаются программы для компьютеров. Российский подобный справочник [1] примечателен тем, что он дополнен сайтами Интернета с открытыми интерактивными сетевыми расчетами свойств воды и водяного пара. На рисунке 2 отображен один из таких интернет-расчетов, по которому можно определить плотность морской воды в зависимости от ее температуры, давления и солености.

* Почти все поселения мира возникали на берегах рек, озер, морей, океанов. Эти водоемы не разделяют людей, а, наоборот, дают им удобные пути сообщения.

** В названии российской столицы, по одной из легенд, присутствует вода: Москва — аКВА.

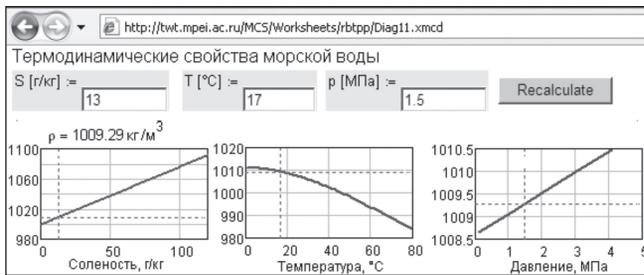


Рис. 2. Сайт, на котором можно оценить плотность морской воды

Интернет-расчет, показанный на рисунке 2, вспомнил автору, когда он сидел на банкете в стокгольмском музее у борта этого незадачливого судна (рис. 1). И вот почему.

По одной из версий*, опрокидывание корабля «Ваза» произошло из-за того, что не было принято во внимание изменение плотности воды при выходе корабля из пресноводного залива в соленое открытое море. Более плотная вода приподняла корабль, порыв ветра накренил его, и вода через открытые пушечные люки хлынула в трюмы...

Так это было или иначе, можно спорить, но то, что тут проявила себя некая инженерная ошибка, связанная с неправильными расчетами, — факт бесспорный.

Давайте посмотрим, как можно применить пакет Mathcad для оценки нашей гипотезы.

В качестве модели судна возьмем деревянный брусок в виде параллелепипеда со сторонами 10 мм (ширина «палубы» — переменная $ш$), 30 мм (высота судна — $в$) и 250 мм (длина — $д$) (рис. 3, а). Это упрощение не так уж далеко от жизни: реальные современные суда имеют почти прямоугольные центральные поперечные сечения (рис. 3, б), и только нос и корма сделаны заостренными для уменьшения сопротивления воды при движении.

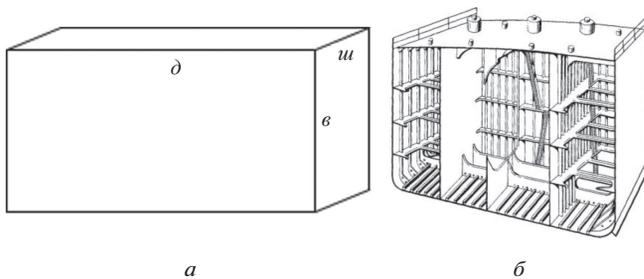


Рис. 3. Модель судна (а) и центральная секция реального судна (б)

Поместим нашу модель судна в воду таким образом, чтобы стороны 10 мм ($ш$) и 250 мм ($д$) образовывали палубу и днище нашего «судна», а сторона 30 мм ($в$) — высоту от днища до палубы. Жизненный опыт показывает, что такое положение деревянного бруска в воде не будет устойчивым (рис. 4, а); сила F_T (сила тяжести, приложенная к центру масс) и сила F_A (выталкивающая

сила Архимеда, приложенная к центру водоизмещения, к центру объема жидкости, вытесненной плавающим телом) развернут наше «судно», и оно ляжет набок.

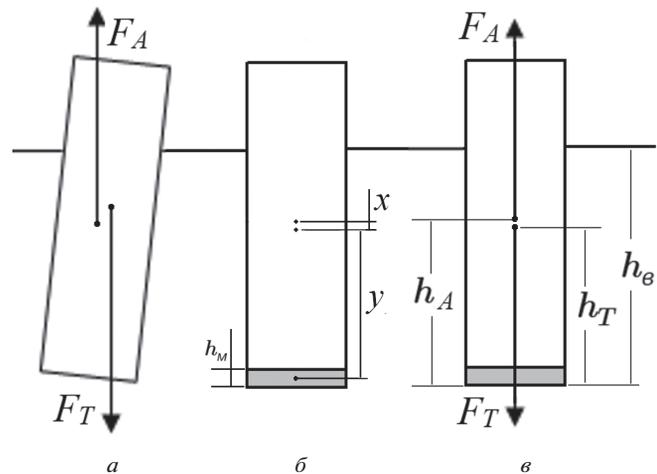


Рис. 4. Силы, действующие на «судно» (а — без балласта, в — с балластом)

Объясняется это тем, что центр масс оказался выше центра водоизмещения, и при любом малом отклонении такого «судна» от положения равновесия оно под действием пары сил перевернется. Но это условие необходимое, но не достаточное, и мы это еще отметим ниже.

Для снижения точки расположения центра масс давайте немного сострогаем рубанком днище «судна» и приклеим туда металлическую полоску толщиной h_m . Это будет тот самый балласт, который должен удержать «судно» от опрокидывания (рис. 4, б, в). Парусные военные корабли строились высокими, чтобы пушки стреляли дальше и сподручнее было брать на бордаж вражеские корабли. Без довольно массивного балласта такой корабль не удержался бы на плаву даже без поднятых парусов.

Проанализируем устойчивость (остойчивость, как говорят профессионалы [2]) нашего «судна» с помощью пакета Mathcad (рис. 5).

Высоту центра водоизмещения (центра объема жидкости, вытесненной плавающим телом) определить несложно через закон Архимеда — см. верхний оператор на рисунке 5, где уравнение (вес вытесненной воды равен весу судна) решается относительно переменной h_B . Для определения центра масс нужно решить уже не одно уравнение, а целых три — см. нижний оператор на рисунке 5. Вот эти уравнения:

- Произведение массы деревянной части судна m_D на расстояние от ее центра до центра масс всего судна (x) равно произведению массы металлической части судна m_M на расстояние от ее центра до центра масс всего судна (y) (рис. 4, б).
- Высота центра масс судна h_T равна половине высоты балласта h_m плюс значение y .
- Высота центра масс деревянной части судна h_D вычисляется двумя способами: $h_m + (в - h_m)/2$ и $h_m/2 + x + y$.

Мы могли бы формулу для h_T вывести вручную, но, имея под рукой пакет Mathcad, можно автоматизировать эту работу, составив балансы крутящих моментов (первое уравнение) и расстояний (два вторых уравнения).

На рисунке 6 показан расчет устойчивости судна с балластом в пресной воде ($\rho = 1 \text{ г/см}^3$) с использованием формулы для h_T , выведенной на рисунке 5. Высота h_A

* Есть «филологическая» версия этого кораблекрушения. Для европейского уха «Ваза» — это в первую очередь не королевская династия и не корабль, названный в ее честь, а... сосуд для хранения чего-то. Одна из разновидностей вазы — это тазик, только сделанный не из металла, а из стекла или фарфора: ваза не для цветов, а для фруктов. Вспомним название статьи и английскую народную песенку в переводе С. Маршака «Три мудреца в одном тазу пустились по морю в грозу...» Истинно сказано: как корабль назовешь, так он и поплывет.

$$\left[\begin{array}{l} \text{«Вес вытесненной воды»} \\ \text{«Объем подводной части»} \cdot \rho_{\text{в}} \cdot g \\ h_{\text{в}} \cdot \omega \cdot \partial \end{array} \right] = \left[\begin{array}{l} \text{«Вес судна»} \\ m \cdot g \end{array} \right] \xrightarrow{\text{solve, } h_{\text{в}}} \frac{m}{\omega \cdot \partial \cdot \rho_{\text{в}}}$$

$$\left[\begin{array}{l} x \cdot m_{\partial} = y \cdot m_{\text{М}} \\ h_{\text{T}} = \frac{h_{\text{М}}}{2} + y \\ h_{\text{М}} + \frac{\text{в} - h_{\text{М}}}{2} = \frac{h_{\text{М}}}{2} + x + y \end{array} \right] \xrightarrow{\text{solve, } \begin{bmatrix} x \\ y \\ h_{\text{T}} \end{bmatrix}} \left[\begin{array}{l} \frac{\text{в} \cdot m_{\text{М}}}{2 \cdot m_{\partial} + 2 \cdot m_{\text{М}}} \quad \frac{\text{в} \cdot m_{\partial}}{2 \cdot m_{\partial} + 2 \cdot m_{\text{М}}} \quad \frac{\text{в} \cdot m_{\partial} + m_{\partial} \cdot h_{\text{М}} + h_{\text{М}} \cdot m_{\text{М}}}{2 \cdot m_{\partial} + 2 \cdot m_{\text{М}}} \end{array} \right]$$

Рис. 5. Вывод формул для расчета высоты центра водоизмещения ($h_{\text{в}}$) и центра масс (h_{T})

(13.55 мм) оказалась больше высоты h_{T} (13.313 мм). Наше «судно» с балластом будет держаться вертикально — пара сил будет возвращать его в исходное положение при небольших отклонениях от вертикали. Но это равновесие неустойчивое (локальная потенциальная яма) — при сильном отклонении судна от вертикали оно все-таки ляжет набок.

Давайте будем постепенно повышать плотность воды (выводить наше «судно» из пресноводной гавани в открытое море) и смотреть, что с ним произойдет. Для этого введем в расчет функцию с именем Δ (разность $h_{\text{в}}$ и h_{T}) и с аргументом $\rho_{\text{в}}$, построим ее график и рассчитаем значение плотности воды, где эта функция поменяет

знак — где силы $h_{\text{в}}$ и h_{T} начнут переворачивать наше «судно» (рис. 7).

Есть легенда о том, как древние русичи спаслись от набега викингов — предков строителей корабля «Ваза». Когда эти морские разбойники поднялись по реке на своих ладьях к поселению русичей и в полном вооружении приготовились к высадке на берег, русичи стали высыпать в воду из мешков самое дорогое, что у них было, — соль. Плотность воды в реке увеличилась, ладьи викингов потеряли устойчивость и перевернулись. Викинги частью потонули, а частью были взяты в плен.

Эту легенду авторы придумали при работе над статьей. Но она с точки зрения физики и математики ничем

$$\begin{array}{l} \omega := 10 \text{ мт} \quad \text{в} := 30 \text{ мт} \quad \partial := 250 \text{ мт} \quad h_{\text{М}} := 0.5 \text{ мт} \\ \rho_{\text{в}} := 1.0 \frac{\text{гм}}{\text{см}^3} \quad \text{Плотность воды} \\ \rho_{\partial} := 0.8 \frac{\text{гм}}{\text{см}^3} \quad \text{Плотность дерева} \\ \rho_{\text{М}} := 7 \frac{\text{гм}}{\text{см}^3} \quad \text{Плотность металла} \\ m_{\text{М}} := \omega \cdot \partial \cdot h_{\text{М}} \cdot \rho_{\text{М}} = 8.75 \text{ гм} \quad m_{\partial} := \omega \cdot \partial \cdot (\text{в} - h_{\text{М}}) \cdot \rho_{\partial} = 59 \text{ гм} \\ m := m_{\text{М}} + m_{\partial} = 67.75 \text{ гм} \\ h_{\text{в}} := \frac{m}{\omega \cdot \partial \cdot \rho_{\text{в}}} = 27.1 \text{ мт} \\ h_{\text{T}} := \frac{\text{в} \cdot m_{\partial} + m_{\partial} \cdot h_{\text{М}} + h_{\text{М}} \cdot m_{\text{М}}}{2 \cdot m_{\partial} + 2 \cdot m_{\text{М}}} = 13.313 \text{ мт} \quad h_{\text{А}} := \frac{h_{\text{в}}}{2} = 13.55 \text{ мт} \end{array}$$

Рис. 6. Расчет остойчивости судна с балластом в пресной воде

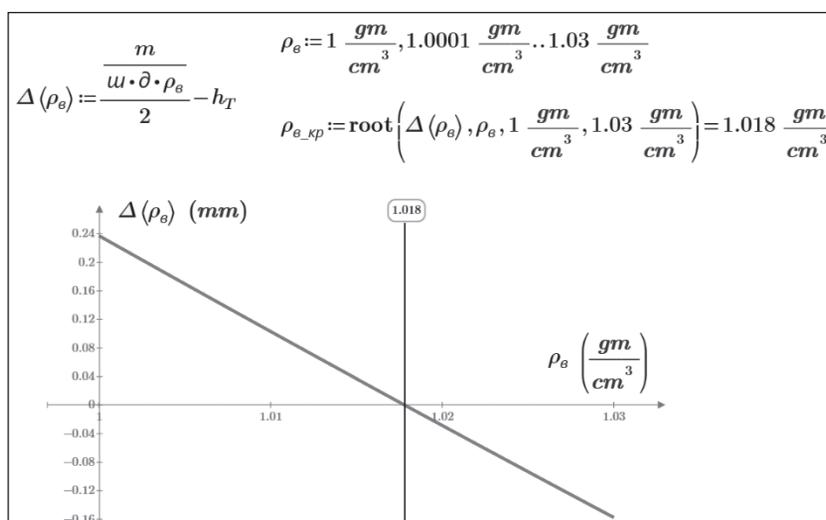


Рис. 7. График остойчивости судна в зависимости от плотности воды

не хуже древнегреческой легенды о том, как был сожжен вражеский флот с помощью зеркал, сфокусировавших солнечные лучи на деревянных кораблях [7].

Но если говорить серьезно, то можно предложить такую **лабораторную работу**.

Изготавливается «судно» в виде линейки, сложенной из деревянной и металлической реек (рис. 4, б, в). Такой «линейный корабль» спускается на воду — опускается в емкость с водой. При этом «судно» спроектировано так (рис. 6), что оно устойчиво держится вертикально в пресной воде. Затем в воду начинают подсыпать поваренную соль, аккуратно перемешивают воду и следят, когда «судно» начнет... заваливаться набок.

Затем можно будет нагревать этот водный раствор и смотреть, как судно будет возвращаться к первоначальному вертикальному положению за счет того, что водный раствор с более высокой температурой имеет меньшую плотность (см. центральный график на рисунке 2). Все это можно дополнить расчетами на компьютере устойчивости судна (решение алгебраических уравнений — см. рисунки 6 и 7) и процессов тепло- и массопереноса (решение дифференциальных уравнений [6]). Форму судна можно делать более сложной, приближенной к реальной и т. д. Это будет хорошей учебной темой, связывающей на школьных занятиях математику, физику, химию и информатику. На уроке можно попытаться создать анимацию переворачивания судна [4] или его раскачивания с рисованием векторов действующих на судно сил [5]...

Конечно, расчет устойчивости реальных кораблей существенно сложнее. Это очень красивая инженерная задача, к решению которой приложили руку почти все великие математики. Нам сегодня остается только восхищаться, как они могли это делать без компьютера. Кстати, на современных судах компьютеры управляют специальными «плавниками» и другими устройствами, снижающими качку даже при сильном волнении моря и предотвращающими опасный крен судна. Удивительно, но у реального корабля центр водоизмещения находится все-таки ниже центра масс, но его устойчивость создается в том числе и сложной формой подводной части корабля, компенсирующей отклонение центра масс смещением места расположения центра водоизмещения. Простейший пример. Если наше судно-брусок без балласта (рис. 4, а) ляжет набок, то несложно подсчитать, что центр масс по-прежнему будет выше центра приложения архимедовых сил. Но при наклоне такого судна (вернее, плота) от бокового ветра, например, центр приложения архимедовых сил (центр водоизмещения) будет

смещаться так, чтобы вернуть судно в горизонтальное положение — в глобальную потенциальную яму.

Кстати, наше судно-линейка — это своеобразный *ареометр*, прибор для измерения плотности жидкости, принцип работы которого основан на законе Архимеда. Ареометр представляет собой стеклянную трубку, нижняя часть которой при калибровке заполняется дробью или ртутью (балластом) для достижения необходимой массы. В верхней части трубки находится шкала, которая проградуирована в значениях плотности раствора или концентрации растворенного вещества. Подобную «шкалу» можно видеть и по бортам судов для контроля его загрузки. Суда класса «река-море» имеют несколько таких шкал. Верхнюю риску на этой шкале называют *грузовой маркой* (load line), которая не должна погружаться в воду при загрузке судна в морском или речном порту. Иначе судно может перевернуться. И таких случаев, увы, было множество.

ПЕРЕФРАЗИРУЯ А. П. Чехова, можно сказать, что в корабле все должно быть прекрасно — и силуэт, и название и... расчеты, по которым он спроектирован. Такой небольшой, но поучительный расчет мы только что провели.

Литературные и интернет-источники

1. Александров А. А., Орлов К. А., Очков В. Ф. Теплофизические свойства рабочих веществ теплоэнергетики: интернет-справочник. М.: Издательский дом МЭИ, 2009. <http://twt.mpei.ac.ru/rbtprr>
2. Основы устойчивости морского судна: учебное пособие / Ф. М. Кацман, Б. П. Коваленко. СПб.: ГМА им. адм. С. О. Макарова, 2003.
3. Очков В. Ф. Вода в научном, культурологическом и религиозном аспектах // Водочистка. Водоподготовка. Водоснабжение. 2009. № 7. <http://twt.mpei.ac.ru/ochkov/WSCR/WaterSCR.pdf>
4. Очков В. Ф. Живые кинематические схемы в Mathcad // Открытое образование. 2013. № 3. <http://twt.mpei.ac.ru/ochkov/Mathcad-15/kinematic.html>
5. Очков В. Ф., Архипцев А. В. Скаляр и вектор в компьютерных вычислениях // Открытое образование. 2013. № 4. <http://twt.mpei.ac.ru/ochkov/Mathcad-15/ScalarVector.pdf>
6. Очков В. Ф., Богомолова Е. П. Это страшное слово «диффуры»... // Информатика в школе. 2015. № 1. <http://twt.mpei.ac.ru/ochkov/ODE.pdf>
7. Очков В. Ф., Калова Я., Соколов А. В., Чудова Ю. В. Литературно-физическая композиция «Истории о зеркале и линзе» // Открытое образование. 2016. № 1. <http://twt.mpei.ac.ru/ochkov/Mirror-Lens.pdf>
8. <https://www.ptcusercommunity.com/message/442440> — сайт статьи.



Л. И. Мазничевская,
средняя общеобразовательная школа № 763, г. Москва

ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ПАСКАЛЬ

Аннотация

В статье представлены материалы, предназначенные для использования учителями информатики при изучении темы «Процедуры и функции в Паскале». Представленные задачи имеют различный уровень сложности. Все программы, написанные на языке программирования Паскаль, содержат комментарии, поясняющие способ решения задачи.

Ключевые слова: информатика, язык программирования, Паскаль, алгоритм, процедура, функция, программа, цикл, массивы.

Контактная информация

Мазничевская Лариса Ивановна, учитель информатики средней общеобразовательной школы № 763, г. Москва; адрес: 129346, г. Москва, ул. Стартовая, д. 27, корп. 3; телефон: (495) 474-90-60; e-mail: mli97@inbox.ru

L. I. Maznichevskaya,
School 763, Moscow

THE USE OF PROCEDURES AND FUNCTIONS IN PASCAL PROGRAMMING LANGUAGE

Abstract

The article presents materials for the use by teachers in studying the theme "Procedures and functions in Pascal". Submitted tasks have different levels of difficulty. All programs written in the programming language Pascal contain comments explaining the method of solving the problem.

Keywords: informatics, programming language, Pascal, algorithm, procedure, function, program, loop, arrays.

Основные теоретические сведения.

Иногда в разных местах программы приходится выполнять практически одни и те же последовательности действий с разными исходными данными. Такие последовательности действий можно оформить в виде так называемых подпрограмм (*англ.* subroutine) — сгруппировать операторы в блок, к которому можно обратиться по имени, причем неоднократно. **Подпрограмма** — автономная часть программы, выполняющая определенный алгоритм и допускающая обращение к ней из различных частей общей программы.

Подпрограммы сокращают текст программ, существенно уменьшают время их исполнения, облегчают жизнь программистам, которые могут создавать программы модульно, т. е. собирая сложную программу из законченных кусочков более простых составляющих. Это позволяет группе программистов создавать большие программы, а группе школьников разрабатывать и реализовывать какие-либо глобальные проекты.

Использование подпрограмм позволяет реализовать один из самых современных методов программирования — структурное программирование.

В языке Паскаль существуют два вида подпрограмм: **процедура** (procedure) и **функция** (function). Процедуры и функции в Паскале объявляются в разделе описания за разделом переменных.

Параметры, записываемые в обращении к подпрограммам, называются **фактическими**; параметры, указанные в описании подпрограмм, — **формальными**. Фактические параметры должны соответствовать формальным по количеству, порядку следования и типу. Параметры, объявленные в основной (главной) программе, действуют в любой подпрограмме и называются **глобальными**. Параметры, объявленные в подпрограмме, действуют только в этой подпрограмме и называются **локальными**.

В данной статье приведены примеры задач, в которых целесообразно использовать процедуры или функции в языке Паскаль.

Задача 1.

Дан массив. Найти количество элементов в нем, которые являются простыми числами.

Решение.

```
program pr1;
uses crt;
const n=5;
var
  a: array[1..n] of integer;
  l, m, p, c: integer;
function fank(m: integer): longint;
{процедура определяет количество делителей числа}
var i, k: integer;
begin
  k:=0;
  for i:=1 to m do
```

```

if m mod i = 0 then k:=k+1;
fank:=k;
end;

begin {основная программа}
cls;
p:=0;
writeln ('Введите ', n, ' чисел');
for l:=1 to n do
  readln(a[l]);
for l:=1 to n do
  begin
    m:=a[l];
    c:=fank(m);
    if c<=2 then p:=p+1;
  end;
writeln('Количество простых чисел в массиве=', p);
end.

```

Задача 2.

Вычислить:

$$Y = \frac{d^a + z^a}{m^b}.$$

Решение.

$$X^N = \underbrace{X \cdot X \cdot X \cdot \dots \cdot X}_{N \text{ раз}}.$$

```

program pr2;
uses crt;
var
  d, q, z, a, m, b: integer;
  y, y1, y2, y3: real;

procedure step(x, n: integer; var w: real);
{процедура находит степень числа}
var i: integer;
begin
  w:=1;
  for i:=1 to n do
    w:=w*x;
  end;

begin {основная программа вычисляет значение
выражения Y}
cls;
read(d, q, z, a, m, b);
step(d, q, y1);
step(z, a, y2);
step(m, b, y3);
y:=(y1+y2)/y3;
writeln('y=', y);
end.

```

Задача 3.

Среди трехзначных чисел найти такие, в которых сумма факториалов цифр равнялась бы самому числу.

Решение.

```

program pr3;
uses crt;
var a, b, c, i, s1, s2, s3: integer;

procedure fak(x: integer; var s: integer);
var i: integer;
begin
  s:=1;
  for i:=1 to x do
    s:=s*i;
  end;

begin
  cls;

```

```

for i:=100 to 999 do
  begin
    a:=i div 100; {выделение 1-й цифры в числе}
    b:=i mod 10; {выделение 3-й цифры в числе}
    c:=(i div 10) mod 10; {выделение 2-й цифры
                           в числе}

    fak(a, s1);
    fak(b, s2);
    fak(c, s3);
    if s1+s2+s3=i then writeln(i);
  end;
end.

```

Функции являются частным случаем процедур и принципиально отличаются от них тем, что:

- результат выполнения функции — одно значение, а процедуры — одно или несколько;
- результат выполнения функции передается в основную программу как значение имени этой функции, а результаты выполнения процедуры — как значения ее параметров.

Задача 4.Даны действительные числа a, b, c . Вычислить:

$$\max(a, a+b) + \max(a, b+c) + \max(a+b \cdot c, 1.5).$$

Решение.

```

program pr4;
uses crt;
var a, b, c, z: real;

function max(x, y: real): real;
var m: real;
begin
  m:=x;
  if y>x then m:=y;
  max:=m;
end;

begin
  cls;
  read(a, b, c);
  z:=a+b;
  z:=max(a, a+b)+max(a, b+c)+max(a+b*c, 1.5);
  writeln(z);
end.

```

Задача 5.

Будем обозначать через $M(a, b, c, \dots, k)$ наименьшее общее кратное, а через $D(a, b, c, \dots, k)$ — наибольший общий делитель целых чисел a, b, c, \dots, k . Написать программу для вычисления следующего выражения:

$$M(a, b, c) \cdot D(a, b) \cdot D(a \cdot b / D(a, b), c).$$

Например, входные данные: 9, 18, 45; выходные данные: 7290.

Решение.

Можно показать, что:

$$M(a, b, c) = M(M(a, b), c),$$

$$M(a, b) = a \cdot b / D(a, b).$$

Допустим, необходимо вычислить наибольший общий делитель двух целых чисел. Результатом решения данной задачи будет одно число, поэтому мы воспользуемся именно функцией.

```

program pr5;
uses crt;

```

```

var a, b, c, dn, d, d1, d2, d3: integer;
function nod_Evk(n, m: integer): integer;
begin
  while n<>m do
    begin
      if m>n
        then m:=m-n
        else n:=n-m;
      end;
      nod_Evk:=n;
    end;
begin
  clrscr;
  write('Введите три числа: ');
  read(a, b, c); {вычисление выражения
  М(a, b, c)·D(a, b)·D(a·b/D(a, b), c),
  где М(a, b)=a·b/D(a, b)}
  d1:=nod_Evk(a, b);
  d2:=(a*b) div d1;
  d3:=nod_Evk(d2, c);
  d:=(d2*c) div d3;
  dn:=d*d1*d3;
  write('Результат= ', dn);
end.

```

Достоинства использования подпрограмм:

- программы, написанные с участием подпрограмм, легче тестировать и отлаживать, у них более четкая логическая структура;
- самостоятельный характер подпрограмм позволяет поручать их составление различным программистам. Так осуществляется разделение работы по программированию и тем самым ускоряется ее завершение;
- использование подпрограмм позволяет экономить память. Память для хранения переменных, используемых в подпрограмме, выделяется только на время ее работы и высвобождается, как только ее выполнение заканчивается.

Литература

1. *Абрамов В. Г., Трифонов Н. П., Трифонова Г. Н.* Введение в язык Паскаль. М.: Наука, 1988.
2. *Березин Б. И., Березин С. Б.* Начальный курс С и С++. М.: ДИАЛОГ-МИФИ, 1996.
3. *Окулов С. М.* Основы программирования. М.: БИНОМ. Лаборатория знаний, 2010.

НОВОСТИ

«Партнер» по парному программированию исправит ошибки

Сервис Kite задуман как искусственный партнер по парному программированию на Python — он следит за набираемым вами кодом, автоматически предлагая варианты продолжения, исправления и документацию. Как объясняют в Kite, сервис позволяет избавиться от постоянного поиска базовой информации при программировании, например сведений о синтаксисе и аргументах функций. За счет этого повышается эффективность труда программиста, которому приходится меньше прерываться при написании кода. При использовании новых библио-

тек Kite может автоматически отображать примеры кода, а кроме того, сервис исправляет опечатки, определяет тип переменных и позволяет быстро выполнять поиск по общедоступным интерфейсам программирования. При работе в терминале Kite выводит документацию по набираемым командам. С ростом числа пользователей Kite сервис обучается, запоминая распространенные ошибки и популярные вызовы функций и пользуясь этой информацией при сортировке возможных вариантов. Kite не привязывает к какому-то конкретному редактору кода.

Портативная фармацевтическая фабрика позволит срочно выпустить лекарство

В МТИ разработали компактную систему производства фармацевтических препаратов, конфигурацию которой можно менять, чтобы выпускать различные лекарства по мере необходимости. Можно, например, срочно изготовить партию средств в случае непредвиденной вспышки эпидемии, выпустить лекарство для клинических исследований или использовать систему как резервную при возникновении проблем на основ-

ной производственной линии. Традиционное производство лекарств может занимать недели или месяцы, которые проходят, пока активные ингредиенты сперва синтезируются на химзаводе, а потом доставляются на другое предприятие, где производятся готовые формы. Система из МТИ действует по принципу поточного производства — все операции выполняются в одном месте.

В Facebook оформили патент на распознавание интернет-жаргона

Неизвестные орфографическим словарям слова в сообщениях пользователей социальных сетей могут в действительности быть специфическим сетевым слэнгом. Компания Facebook получила патент на способ автоматического составления словарей сетевого жаргона с помощью анализа сообщений. Такие словари пригодятся для улучшения качества автокоррекции вводимого текста, выдачи более точных версий, а также составления соци-

альных графов. Слово будет считаться неологизмом, а не ошибкой в том случае, если оно употребляется многими пользователями, особенно близко расположенными в социальном графе, говорящими на одном языке, живущими в одном регионе, возможно принадлежащими к одной демографической категории или имеющими сходные данные профиля. Если частота употребления слова снижается, оно исключается из словаря.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

КОНКУРСЫ

Уважаемые авторы и читатели!

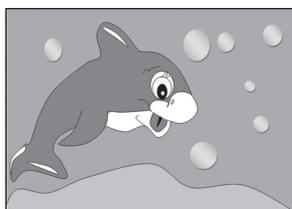
Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» во втором полугодии 2016 года.

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

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



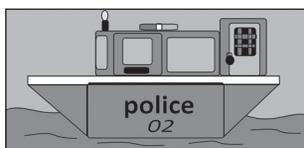
Никита Шевораков,
ученик средней школы № 37, г. Саранск,
Республика Мордовия



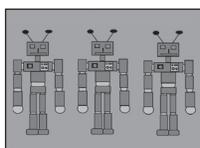
Арина Хлытина,
ученица гимназии № 1, г. Новосибирск



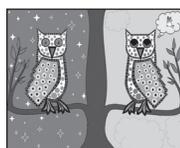
Екатерина Иванистова,
студентка Куйбышевского педагогического колледжа,
г. Куйбышев, Новосибирская область



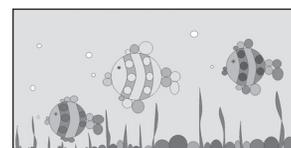
Максим Войнов,
воспитанник Центра психологической помощи и развития «Эмпатия»,
ученик лицея № 36, Нижний Новгород



Святослав Антипин,
ученик гимназии № 1,
г. Новосибирск



Полина Пискленова,
ученица гимназии № 3,
г. Королев, Московская область



Софья Петрова,
ученица средней общеобразовательной школы № 1, г. Зеленокумск,
Ставропольский край



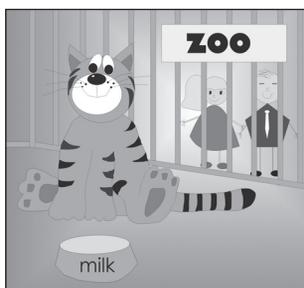
Л. Ю. Черницына,
методист Городского методического центра Департамента образования г. Москвы



Екатерина Пушилина,
ученица Новосибирской классической гимназии № 17



Валерия Андреева,
ученица лицея г. Черногорска,
Республика Хакасия



Полина Глотова,
ученица гимназии № 1, г. Новосибирск



Ксения Здобнова,
ученица Лицея математики и информатики Кировского района г. Саратова



Анна Гринева,
ученица гимназии № 1,
г. Новосибирск

Журнал «Информатика в школе»

Индексы подписки (агентство «Роспечать»)
на 2-е полугодие 2016 года

- 81407 — для индивидуальных подписчиков
- 81408 — для организаций

Периодичность выхода: 5 номеров в полугодие (в июле 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~~ журнал
(индекс издания)

Информатика в школе
(наименование издания)

Количество комплектов

На 20**16** год по месяцам

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| | | | | | | | | | | | |

Куда
(почтовый индекс) (адрес)

Кому

Линия отреза

ДОСТАВОЧНАЯ
КАРТОЧКА (индекс издания)

ПВ место литер

На ~~газету~~ журнал
(наименование издания)

| | | | |
|-----------|---------------|------|-----------------------|
| Стоимость | подписки | руб. | Количество комплектов |
| | каталожная | руб. | |
| | переадресовки | руб. | |

На 20**16** год по месяцам

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| | | | | | | | | | | | |

| | | | | | |
|----------------------|----------------------|----------------------|----------------------|----------------------|-----------------|
| <input type="text"/> | Город |
| <input type="text"/> | село |
| <input type="text"/> | почтовый индекс |
| <input type="text"/> | область |
| <input type="text"/> | Район |
| <input type="text"/> | код улицы |
| <input type="text"/> | улица |
| <input type="text"/> | дом |
| <input type="text"/> | корпус |
| <input type="text"/> | квартира |
| | | | | | Фамилия И.О. |