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
Почтовый адрес:
1119270, г. Москва, а/я 15
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 04.04.2017.
Формат 60×90 1/8. Усл. печ. л. 8,0.
Тираж 2000 экз. Заказ № 62.

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», 2017

ИНФОРМАТИКА В ШКОЛЕ

№ 3 (126) апрель 2017

Содержание

УРОКИ ИНФОРМАТИКИ

Исянбаева З. Р. Урок на тему «Информатика. Информация…» 3

Виноградова Л. С. Урок на тему «Язык программирования Python. Типы данных. Синтаксис. Арифметические операции» 7

ИНТЕГРИРОВАННЫЕ УРОКИ

Самохина Э. В. Занятие по легоконструированию на тему «Робот-лягушка» 13

Шишкова Н. А. Смешанное обучение в профильном курсе информатики 17

МЕТОДИЧЕСКАЯ КОПИЛКА

Карандаева А. В. Увлекательная робототехника для школьников 24

СУПЕРКОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ШКОЛЕ

Плаксин М. А. Пропедевтика параллельных вычислений в школьной информатике. Сетевой график 29

ИТОГОВАЯ АТТЕСТАЦИЯ ПО ИНФОРМАТИКЕ

Кашаев С. М., Шерстнева Л. В., Гладских Д. С. Выполнение сложных заданий ЕГЭ на языке Python 36

ЗАДАЧИ

Федотова Т. В. Использование формул массива в Microsoft Excel 40

ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Слинкин Д. А. Технологические аспекты подготовки школьников-дебютантов к олимпиадам по информатике 46

ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

Каплан А. В. Результаты апробации учебно-методического комплекта «Информатика для всех» в первом классе 52

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

Очков В. Ф., Калинина А. В. По порядку становись! 56

ВСЕРОССИЙСКИЙ КОНКУРС «САМИ ДЕЛАЕМ МУЛЬТФИЛЬМ»

Уважаемые коллеги!

Издательство «Образование и Информатика», редакция журнала «Информатика в школе»
объявляют о проведении конкурса
«Сами делаем мультфильм:
мультипликация (анимация) на уроках информатики»

Конкурс проводится по трем номинациям:

1. Номинация для педагогов: Мультипликация (анимация) на уроках информатики.

В номинации могут быть представлены методические и дидактические материалы для проведения уроков информатики в начальной, средней, старшей школе, а также занятий в учреждениях дополнительного образования, посвященных созданию мультфильмов (анимационных фильмов) разными средствами и в рамках изучения разных тем курса информатики и других дисциплин.

2. Номинация для учащихся: Сами делаем мультфильм.

В номинации могут быть представлены работы учащихся (тексты), рассказывающие о создании мультфильмов (анимационных фильмов) разными средствами.

3. Номинация — презентация проектов: Приглашаем в кинозал.

В номинации могут быть представлены реализованные учащимися проекты по созданию мультфильмов (анимационных фильмов), т. е. готовые к просмотру ролики. Мультфильмы должны сопровождаться обязательной описательной частью — методическими комментариями педагога. То есть работа в данной номинации состоит из двух частей: 1) методического описания, подготовленного педагогом, и 2) анимационного ролика, созданного учащимся. Соответственно, в данной номинации работы представляют авторские коллективы, состоящие из учащегося (или группы учащихся) и педагога. Все работы в данной номинации будут выложены на сайте ИНФО для онлайн-голосования.

Работы на конкурс принимаются с 1 февраля по 15 апреля 2017 года включительно. Работы, присланные позже этой даты, к участию в конкурсе допускаться не будут. Подача работ производится только через заполнение формы заявки на сайте ИНФО (необходима предварительная регистрация на сайте или авторизация для зарегистрированных пользователей).

Голосование на сайте за работы, представленные для онлайн-голосования, будет проходить с 1 по 20 мая 2017 года включительно.

Итоги конкурса будут подведены в № 5-2017 журнала «Информатика в школе», а также опубликованы на сайте издательства «Образование и Информатика».

Лучшие работы в номинациях 1 и 2 будут опубликованы в журнале «Информатика в школе».

Победители конкурса получат:

- диплом от издательства «Образование и Информатика»;
- электронную подписку на журнал «Информатика в школе» на 2017 год;
- печатный экземпляр журнала «Информатика в школе» № 5-2017, в котором будут опубликованы итоги конкурса;
- авторский печатный экземпляр журнала «Информатика в школе» с опубликованной работой.

Подробную информацию
о требованиях к оформлению конкурсной работы и конкурсной заявки,
а также всю дополнительную информацию о конкурсе
вы можете найти на сайте издательства «Образование и Информатика»:
http://infojournal.ru/competition/mult-2017/
а также получить в редакции ИНФО
по адресу: readinfo@infojournal.ru
и по телефону: (495) 364-95-97

З. Р. Исянбаева,
средняя общеобразовательная школа № 1, с. Акъяр, Хайбуллинский район, Республика Башкортостан

УРОК НА ТЕМУ «ИНФОРМАТИКА. ИНФОРМАЦИЯ...»*

Аннотация

В статье представлен урок, на котором в игровой форме повторяются и закрепляются знания, полученные учащимися при изучении курса информатики в основной школе.

Ключевые слова: повторение и закрепление материала, поисково-практическая деятельность учащихся, воспитание коллективизма, урок-игра.

Контактная информация

Исянбаева Зиля Рашитовна, учитель информатики средней общеобразовательной школы № 1, с. Акъяр, Хайбуллинский район, Республика Башкортостан; *адрес:* 453800, Республика Башкортостан, Хайбуллинский район, с. Акъяр, пр-т Салавата Юлаева, д. 10; *телефон:* (34758) 2-13-67; *e-mail:* AZilyaR@mail.ru

Z. R. Isyanbayeva,
School 1, Akjar, Khaybulla District, The Republic of Bashkortostan

LESSON ON THE THEME "INFORMATICS. INFORMATION..."

Abstract

The article presents the lesson in which in the form of a game the knowledge received by students in the informatics course in secondary school are repeated.

Keywords: repetition and consolidation of material, search and practical activity of students, upbringing of collectivism, lesson-game.

Для повторения и закрепления материала по курсу информатики основной школы предлагается урок в форме командной игры. Он также может быть проведен как внеурочное мероприятие, например, в рамках предметной недели.

Перед началом урока выбираются два помощника учителя из учащихся, они контролируют ход игры, выступают в роли жюри и на доске ведут подсчет очков.

Остальные ученики делятся на две команды: одна команда выбирает эмблему с изображением ромба, а другая — с изображением треугольника. На оборотной стороне у них нарисованы: у ромба — 0, у треугольника — 1.

Учитель. Ребята, как вы думаете, почему при делении на команды вы выбирали 1 или 0? Что означают эти цифры в информатике?

Учащиеся. Как правило, все числа в компьютере представляются с помощью нулей и единиц (а не десяти цифр, как это привычно для людей). Иными словами, компьютеры обычно работают в двоичной системе счисления, поскольку при этом устройства для их обработки получаются значительно более простыми.

После этого игроки каждой команды выбирают капитана и придумывают название команды.

Помощники раздают командам опорный конспект мероприятия.

Этап 1. Угадай

В конкурсах этого этапа ребята должны использовать разные органы чувств.

Конкурс 1.1. Чей голос?

Учащийся закрывает глаза и угадывает, кто из одноклассников называет его имя.

Максимальное количество баллов — 2.

Конкурс 1.2. Что это?

Учащийся закрывает глаза, учитель подносит к его носу по очереди два предмета (мандарин, конфету и т. п.; предметы должны быть с уловимым запахом), ученик должен отгадать по запаху, что это.

Максимальное количество баллов — 2.

Конкурс 1.3. Вкусняшка.

Учащийся закрывает глаза, учитель дает ему попробовать два фрукта или овоща (лимон, мандарин и т. д.), ученик должен отгадать по вкусу, что это.

Максимальное количество баллов — 2.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2017/

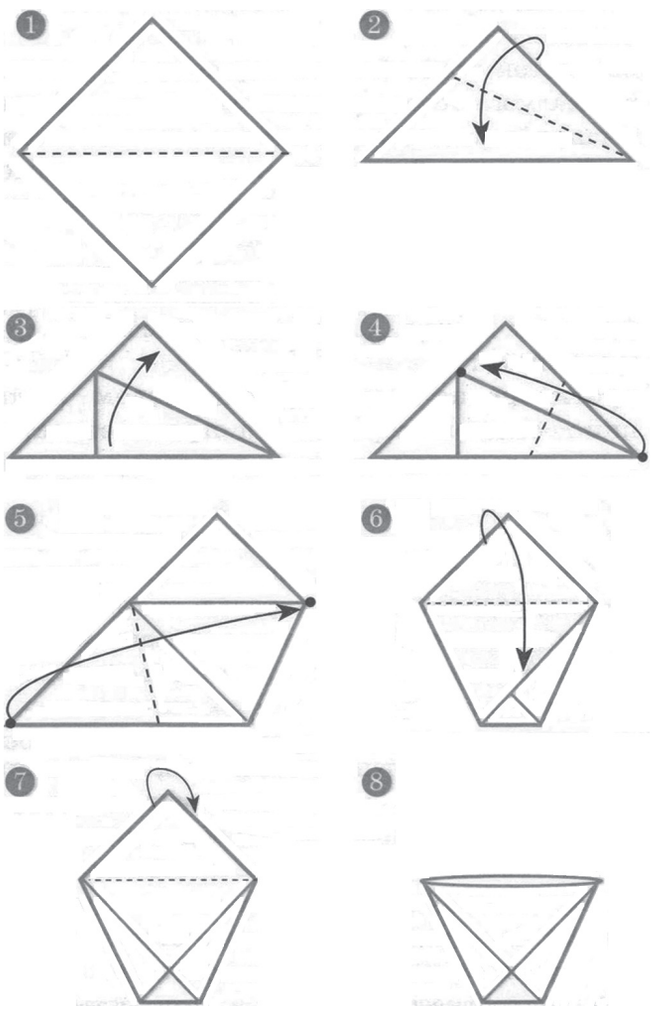
Конкурс 1.4. Найди отличия.

Необходимо найти 10 отличий на картинках:

Максимальное количество баллов — 10.

Конкурс 1.5. Оригами.

По предложенной схеме надо максимально качественно воспроизвести объект (оригами).



1. Складываем листок по диагонали.
2. Верхний край отворачиваем книзу.
3. Отворачиваем край назад.
4. Складываем левый угол так, чтобы отмеченные на рисунке точки совпали.
5. Опять совмещаем отмеченные точки.
6. Верхний слой по пунктиру отворачиваем вниз.
7. Нижний слой загибаем назад.
8. Стаканчик готов.

Максимальное количество баллов — 5.

Учитель. Ребята, при выполнении оригами вы работали по схеме. Как на языке информатики можно назвать эту схему?

Учащиеся. Алгоритм.

Учитель. Что такое алгоритм?

Учащиеся. Алгоритм — это последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу.

Учитель. Итак, сколько конкурсов мы прошли на первом этапе?

Учащиеся. Пять.

Учитель. Да, пять. Мы получили информацию о предметах с помощью различных органов чувств.

Итак, информацию можно получить с помощью:

1) глаз;
2) ушей;
3) носа;
4) языка;
5) рук, ног, тела.

Информация, полученная с помощью тех или иных органов чувств, носит определенное название:

1) зрительная;
2) звуковая;
3) обонятельная;
4) вкусовая;
5) осязательная.

Зрение - 90%, Слух – 9 %

Орган	Вид информации по способу восприятия
Глаза	*Зрительная*
Уши	*Звуковая*
Нос	*Обонятельная*
Язык	*Вкусовая*
Кожа	*Осязательная*

MyShared

Это виды информации *по способу восприятия*.

Этап 2

На этом этапе одновременно работают команды и капитаны.

Конкурс 2.1. Анаграмма (конкурс для команд).

Команды получают по пять слов. В каждом слове необходимо переставить буквы так, чтобы получилось слово, связанное с информатикой и компьютерами. Например, из слова «док» можно получить слово «код».

- пьюромтек (*компьютер*);
- реверс (*сервер*);
- среда (*адрес*);
- стригер (*регистр*);
- иголка (*логика*).

Команды дают ответы по очереди — по одному слову.

На подготовку дается 1 минута.

За каждое правильно составленное слово команда получает 2 балла.

Конкурс 2.2. Волшебная таблица (конкурс для капитанов).

Учитель. Пока команды готовятся, к доске приглашаются капитаны команд. Вам необходимо найти в таблице слова, связанные с информатикой:

Т	И	Л	И	Т	А	К	И	М	С
У	Д	И	Л	Е	Р	Р	О	С	К
Ш	Ь	С	О	Р	А	М	Е	Х	А
Ы	М	С	Е	Ц	О	Р	Р	Е	Н
К	М	О	Н	И	Д	П	Е	Д	О
Л	А	В	И	Т	И	С	М	А	М
П	К	А	А	О	Р	К	Е	Т	К
Л	А	Б	Т	У	Р	Ь	П	М	О
А	Т	Е	Л	Ь	А	Ю	Т	Е	Р

На подготовку отводится 1 минута.

(*Плата, кабель, клавиатура, компьютер, монитор, дискета, модем, мышь, процессор, сканер, микросхема, дилер, утилита*.)

За каждое найденное слово начисляется 1 балл.

Учитель. Итак, сколько конкурсов мы прошли на втором этапе?

Учащиеся. Два конкурса одновременно.

Учитель. Да, правильно, работала команда и работал капитан. На этом этапе мы с вами повторили русский язык и тему курса информатики «Текстовая информация».

Этап 3

Конкурс 3.1. Изобрази и отгадай.

Участник команды получает карточку с некоторым изображением. Нужно с помощью мимики и жестов изобразить то, что представлено на карточке.

Варианты карточек:

- мальчик, играющий в компьютерную игру;
- колонки;
- принтер;
- компьютер завис.

Команда, которая первая угадает изображаемый предмет, получает 3 балла.

Участник, который изображал предмет, может получить максимально 5 баллов (оцениваются творческие способности участника, оригинальность и фантазия).

Учитель. Итак, сколько конкурсов мы прошли на третьем этапе?

Учащиеся. Один.

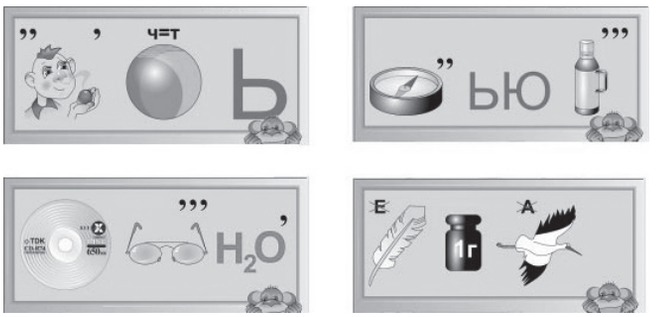
Учитель. Мы стали немыми актерами и повторили тему курса информатики «Кодирование информации».

Этап 4

На этом этапе одновременно работают команды и по одному участнику от каждой команды.

Конкурс 4.1. Разгадай ребус (конкурс для команд).

В течение трех минут команды должны разгадать ребусы.



(*Память, компьютер, дисковод, программист*.)

За каждый правильный ответ команда получает 1 балл.

Конкурс 4.2. Калькулятор (конкурс для представителей команд).

Практическая работа на компьютере.

С помощью калькулятора надо решить примеры и в полученных результатах выявить закономерность.

142/2
528 – 457
50 + 21
25 + 46
4118/58
184458/2598
295 – 224

(*Ответ во всех примерах: 71.*)

Учитель. Какую закономерность вы увидели в этой работе?

Учащиеся. Во всех примерах ответом получилось число 71.

Учитель. Ребята, как вы думаете, что означает это число?

Учащиеся. В 2016 году мы отмечали 71-ю годовщину Победы в Великой Отечественной войне.

Учитель. С какой информацией мы работали на четвертом этапе?

Учащиеся. С графической и числовой.

Учитель. Верно, ребусы — это графическая информация, плюс мы выполнили практическую работу в приложении Калькулятор и поработали с числовой информацией.

Разминка для болельщиков

Конкурс «Эрудит».

1. Ноль или единица в информатике. (*Бит*.)
2. Устройство, служащее для отображения текстовой и графической информации. (*Монитор*.)
3. Специальная программа, выполняющая нежелательные для пользователя действия на компьютере. (*Вирус*.)
4. «Мозг» компьютера. (*Процессор*.)
5. Взломщик компьютерных программ. (*Хакер*.)
6. Состояние, в котором включенный компьютер не реагирует на действия пользователя. (*Зависание*.)
7. Всемирная глобальная сеть. (*Интернет*.)

8. Наименьшая единица количества информации. (*Бит*.)
9. Манипулятор в виде укрепленной на шарнире ручки с кнопкой, используется в основном для компьютерных игр. (*Джойстик*.)
10. Устройство для обмена информацией с другими компьютерами через телефонную сеть. (*Модем*.)
11. Какое действие будет выполняться при перетаскивании файла с помощью левой кнопки мыши с диска C на диск D — копирование или перемещение? (*Копирование*.)
12. Когда появился манипулятор типа «мышь», для него в русском языке некоторое время использовалось название по имени персонажа известной русской сказки. Назовите имя этого персонажа. (*Колобок*.)

Этап 5

Учитель. Ребята, послушайте стихотворение и назовите его автора и название. (*Звучит стихотворение*.)

Учащиеся. «Дедушка Мазай и зайцы» Николая Алексеевича Некрасова.

Учитель. В 2016 году исполнилось 195 лет со дня рождения русского поэта, прозаика, критика, издателя Николая Алексеевича Некрасова (1821–1878). Посмотрим отрывок из мультфильма.

(*Демонстрируется мультфильм «Дедушка Мазай и зайцы» по мотивам стихотворения Н. А. Некрасова: https://www.youtube.com/watch?v=nWOSZrhuuWc*)

Учитель. На пятом этапе мы повторили темы «Мультимедийная информация» и «Звуковая информация».

Итак, на последних этапах урока мы повторили виды информации *по форме представления*.

Этап 6

Учитель ставит перед капитанами некий предмет и читает стихотворение. Услышав слово «три», капитан должен успеть схватить этот предмет. Первое «ложное хватание» штрафуется предупреждением, а второе — снижением баллов.

Однажды щуку мы поймали,
Распотрошили, а вну**три**
Рыбешек мелких увидали,
И не одну, а целых... семь.

Когда стихи запомнить хочешь,
Их не зубри до поздней ночи.
Возьми и на ночь повтори
Разок другой, а лучше... десять.

Мечтает парень закаленный
Стать олимпийским чемпионом.
Смотри, на старте не хи**три**,
А жди команду: раз, два, ... марш!

Однажды поезд на вокзале
Мне **три** часа пришлось прождать...

(*Если капитаны не успевают взять приз, его забирает ведущий*.)

Ну что же вы свой приз не брали,
Когда была возможность брать?

Этап 7

Конкурс 7.1. Логическая задача.

Встретились три подруги: Белова, Краснова, Чернова. Девочка в белом платье говорит Черновой: «Нам надо всем поменяться, а то цвет наших платьев не соответствует фамилиям». Кто в какое платье одет?

На решение отводится 2 минуты.

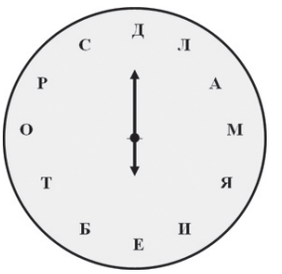
(*Белова — в черное, Краснова — в белое, Чернова — в красное*.)

За правильный ответ команда получает 1 балл.

Если в результате всех конкурсов команды набрали одинаковое количество баллов, для выявления победителя предлагается дополнительный конкурс.

Конкурс 7.2 (финальный). Циферблат-ключ.

Прочтите пословицу.
12, 6, 1, 9
3, 2, 11, 8, 6, 10, 2
7, 9, 5, 8, 11, 4



(*Дело мастера боится*.)

Подведение итогов игры

Слово предоставляется помощникам учителя, которые оглашают команду-победителя и команду-призера.

Учитель совместно с учащимися проводит анализ урока. Выставляются оценки за урок. Учащиеся обязательно делятся впечатлениями об игре.

Л. С. Виноградова,
лицей № 1793 «Жулебино», г. Москва

УРОК НА ТЕМУ «ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON. ТИПЫ ДАННЫХ. СИНТАКСИС. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ»

Аннотация

В статье представлен урок информатики для восьмого класса, особенностью которого является использование игры CodeCombat для обучения и закрепления навыков написания программного кода.

Ключевые слова: язык программирования, синтаксис, данные, Python, программа, арифметические операции.

Контактная информация

Виноградова Людмила Сергеевна, учитель информатики лицея № 1793 «Жулебино», г. Москва; *адрес:* 109431, г. Москва, Жулебинский бул., д. 38; *телефон:* (495) 705-79-68; *e-mail:* lv29@inbox.ru

L. S. Vinogradova,
Liceum 1793 Zhulebino, Moscow

LESSON ON THE THEME "PROGRAMMING LANGUAGE PYTHON. DATA TYPES. SYNTAX. ARITHMETIC OPERATIONS"

Abstract

The article presents the informatics lesson for eighth grade, on which CodeCombat game is used for learning and getting skills of writing programming code.

Keywords: programming language, syntax, data, Python, program, arithmetic operations.

Тема урока: Язык программирования Python. Типы данных. Синтаксис. Арифметические операции.

Цели урока:
- знакомство учащихся с типами данных, синтаксисом, арифметическими операциями в языке программирования Python;
- овладение навыками работы на языке Python — написание программного кода.

Планируемые результаты обучения:

метапредметные:
- развитие алгоритмического мышления;
- получение навыков систематизации информации;

личностные:
- формирование коммуникативной культуры;
- развитие внимательности, аккуратности, дисциплинированности;

предметные:
- умение работать с числами, строками в языке Python;
- умение работать с арифметическими операциями в языке Python;
- умение применять базовые типы данных при работе с программным кодом на языке Python.

Тип урока: урок знакомства с новым материалом.

Класс: восьмой.

Оснащение урока:
- компьютер учителя с доступом в Интернет;
- мультимедийный проектор;
- экран;
- компьютеры учащихся с доступом в Интернет;
- обучающая игра CodeCombat: https://codecombat.com/
- электронный учебник-задачник: http://python-rutour.rhcloud.com/tour/

Список источников, использованных при подготовке урока.
1. Курс «Учим Python». http://informatics.mccme.ru/course/view.php?id=95
2. Самоучитель Python. Python 3 для начинающих и чайников. https://pythonworld.ru/samouchitel-python
3. CodeCombat. https://codecombat.com/
4. Python Learning. Интерактивное обучение языку Python. http://python-rutour.rhcloud.com/tour/

План урока.
1. Организационный момент.
2. Актуализация знаний учащихся. Формулировка темы и целей урока.
3. Работа с электронным учебником-задачником.
4. Практическая работа в CodeCombat.
5. Подведение итогов урока, выставление оценок.
6. Домашнее задание.

Ход урока

1. Организационный момент

2. Актуализация знаний учащихся. Формулировка темы и целей урока

На этапе актуализации знаний учащиеся отвечают на вопросы о том, что такое язык программирования и для чего он нужен, **в ходе обсуждения формулируют тему урока и его цели**.

Возможные **вопросы:**

- Для чего нужны языки программирования?
- Какие современные языки программирования вы знаете?
- Что такое переменная?
- Для чего нужны данные?
- Что такое данные?
- Как компьютер работает с различными видами информации?

В результате учащиеся формулируют тему урока: «Изучение языка программирования Python» и цель урока: создание собственного программного кода, а следовательно, изучение синтаксиса языка, знакомство с типами данных и т. д.

3. Работа с электронным учебником-задачником

Далее учащиеся самостоятельно изучают новую тему при помощи электронного учебника-задачника: http://python-rutour.rhcloud.com/tour/ (рис. 1–4).

Если у класса возникают трудности в восприятии рассматриваемой темы, содержимое страниц учебника

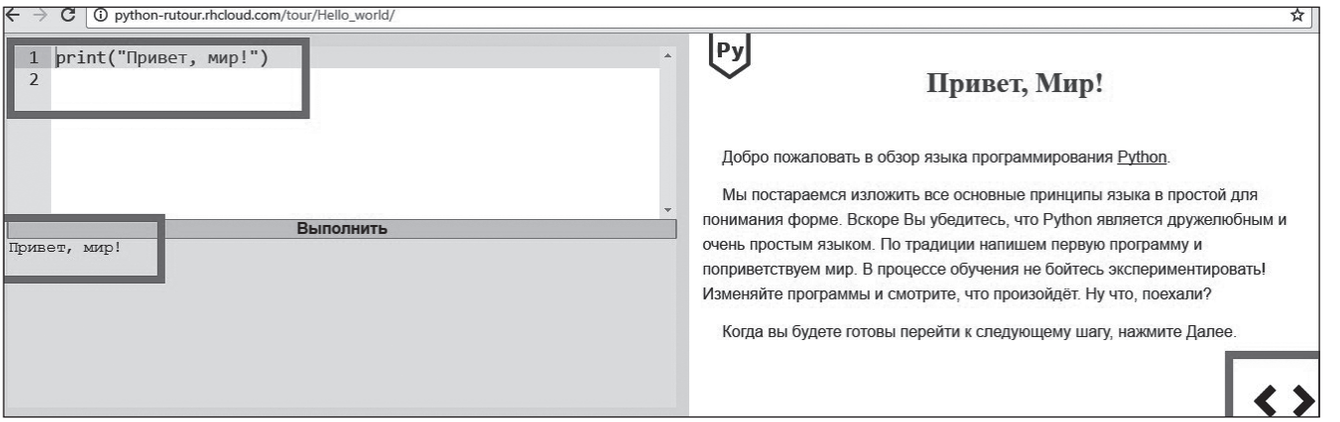


Рис. 1. Работа с электронным учебником-задачником

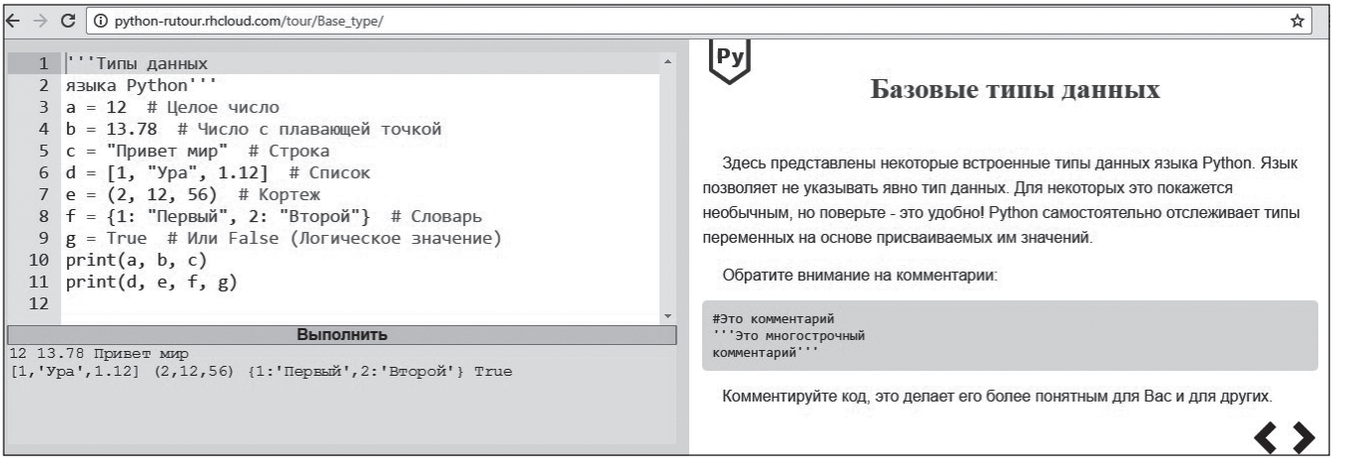


Рис. 2. Работа с электронным учебником-задачником. Типы данных

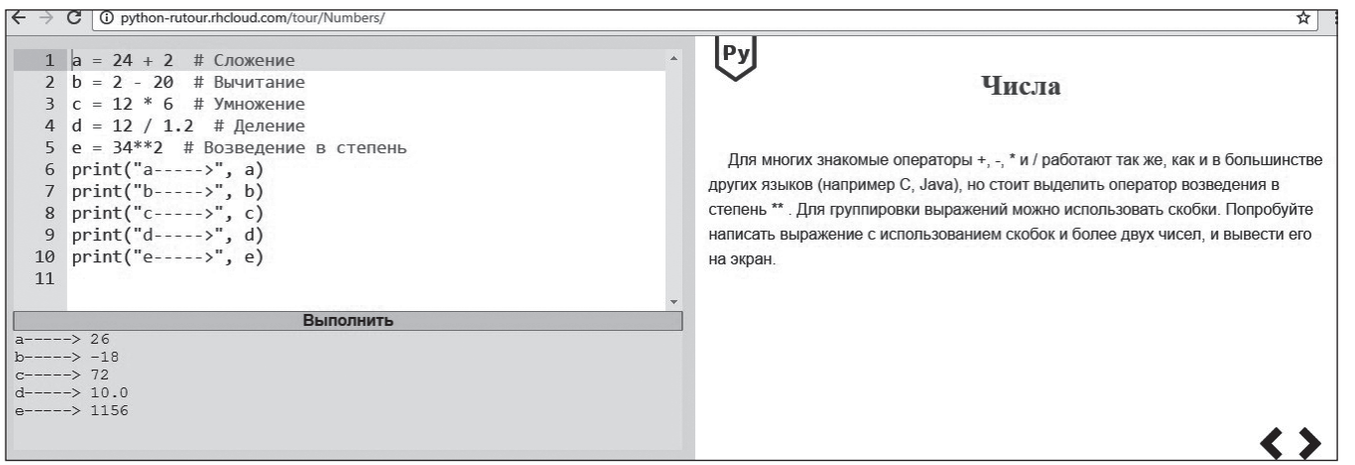


Рис. 3. Работа с электронным учебником-задачником. Числа

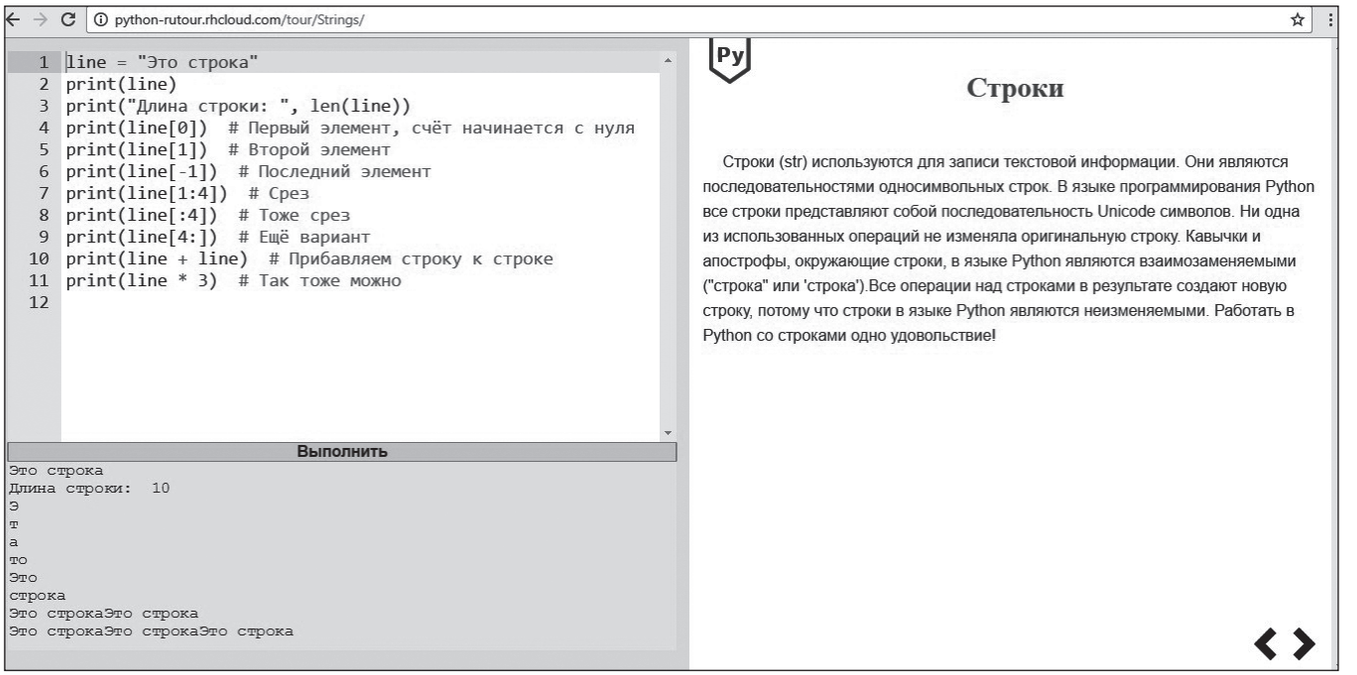


Рис. 4. Работа с электронным учебником-задачником. Строки

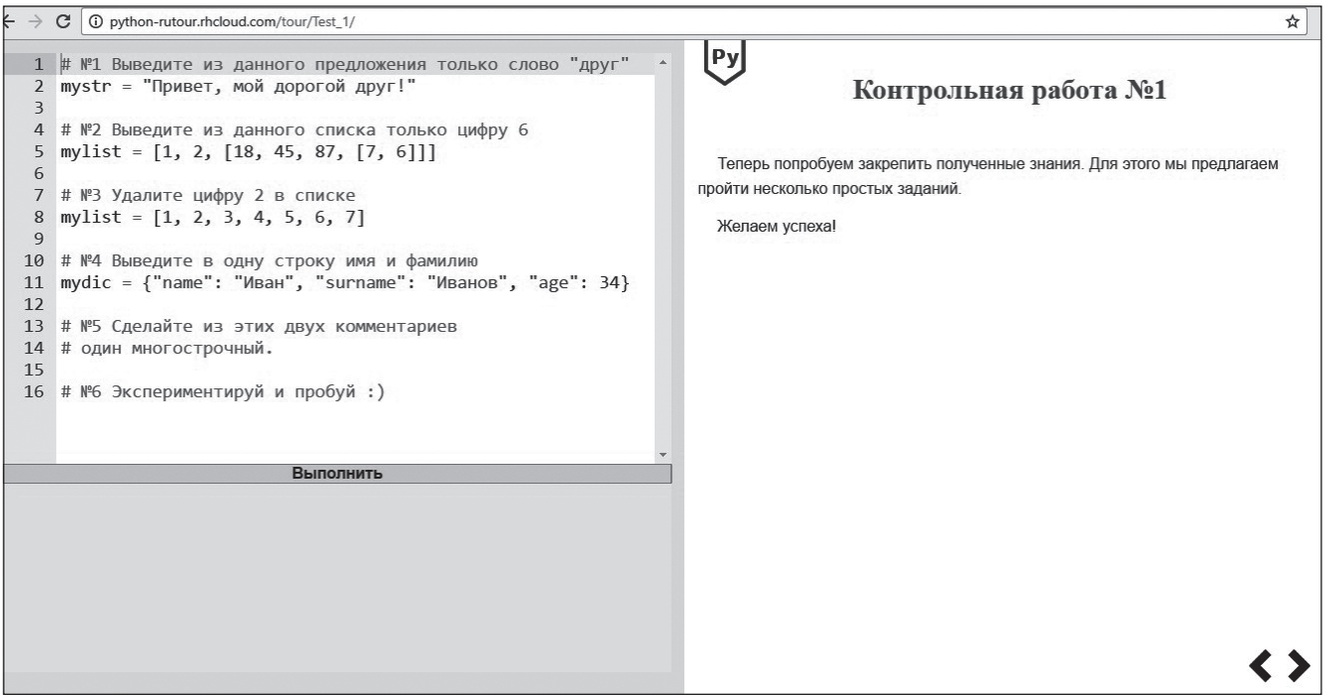


Рис. 5. Работа с электронным учебником-задачником. Контрольная работа

проецируется с помощью проектора на большой экран и учитель объясняет новый материал в виде лекции.

По окончании изучения нового материала (после страницы «Строки») учащиеся выполняют небольшую контрольную работу, предложенную в электронном учебнике-задачнике (рис. 5).

4. Практическая работа в CodeCombat

Далее учащиеся приступают к практической работе в обучающей ролевой игре CodeCombat: https://codecombat.com/. В игровом формате они изучают синтаксис языка и работу со строками (рис. 6–8).

В ходе игры ребята могут пройти несколько этапов, в среднем во время урока они успевают пройти шесть этапов.

5. Подведение итогов урока, выставление оценок

Учитель может оценить прохождение игры каждым учащимся в сводной таблице, предлагаемой программой (рис. 9).

Итогом урока является получение учащимися навыков работы с языком программирования Python. Они

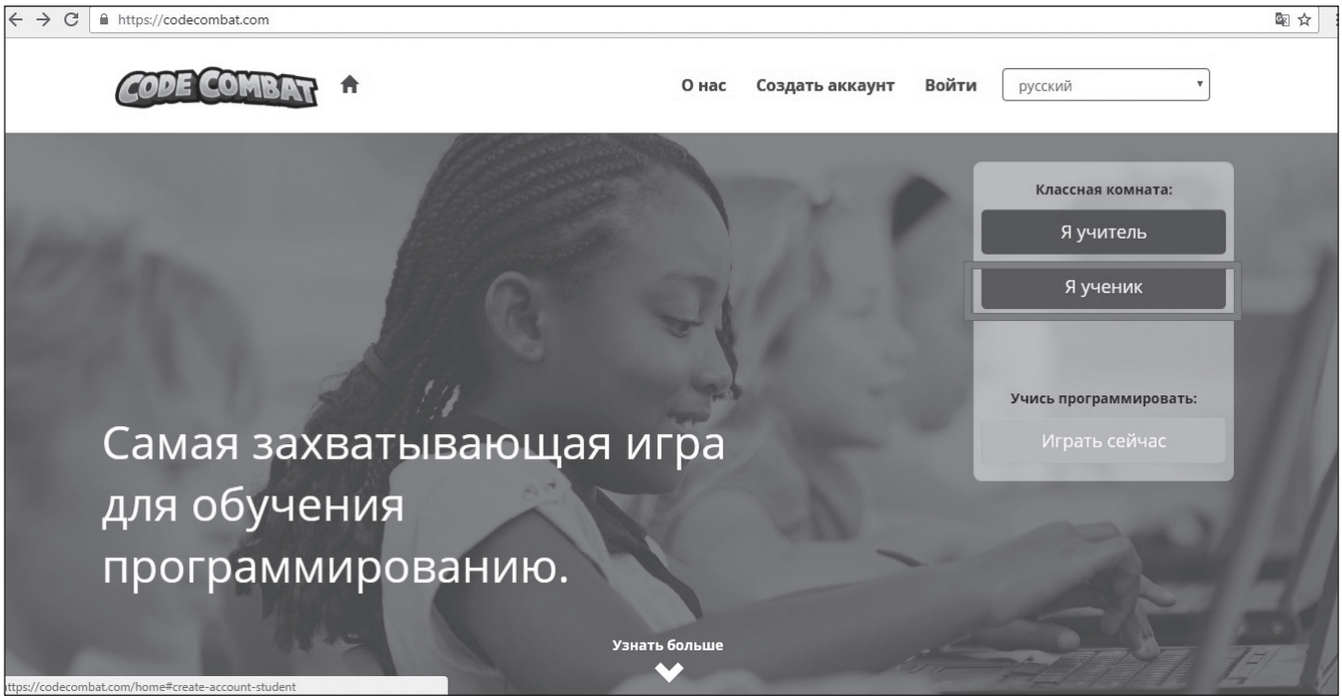


Рис. 6. Стартовая страница игры CodeCombat. Регистрация через классную комнату «Я ученик»

Рис. 7. CodeCombat. Начало. Выбор языка программирования

научатся понимать синтаксис языка, работать с различными типами данных.

В конце урока учитель выставляет оценки по итогам:

1) контрольной работы, выполненной в электронном учебнике-задачнике;
2) прохождения этапов игры CodeCombat.

6. Домашнее задание

В качестве домашнего задания учащимся предлагается самостоятельно пройти уровень 7 в CodeCombat (для тех учащихся, кто не справился с прохождением этого уровня на уроке).

* * *

По результатам проведения данного урока с учащимися восьмого класса московского лицея № 1793 «Жулебино» можно отметить существенный прогресс учащихся в игре CodeCombat: большинство ребят на уроке преодолели шесть уровней игры, программируя на языке Python, а некоторые успели даже дойти до де-

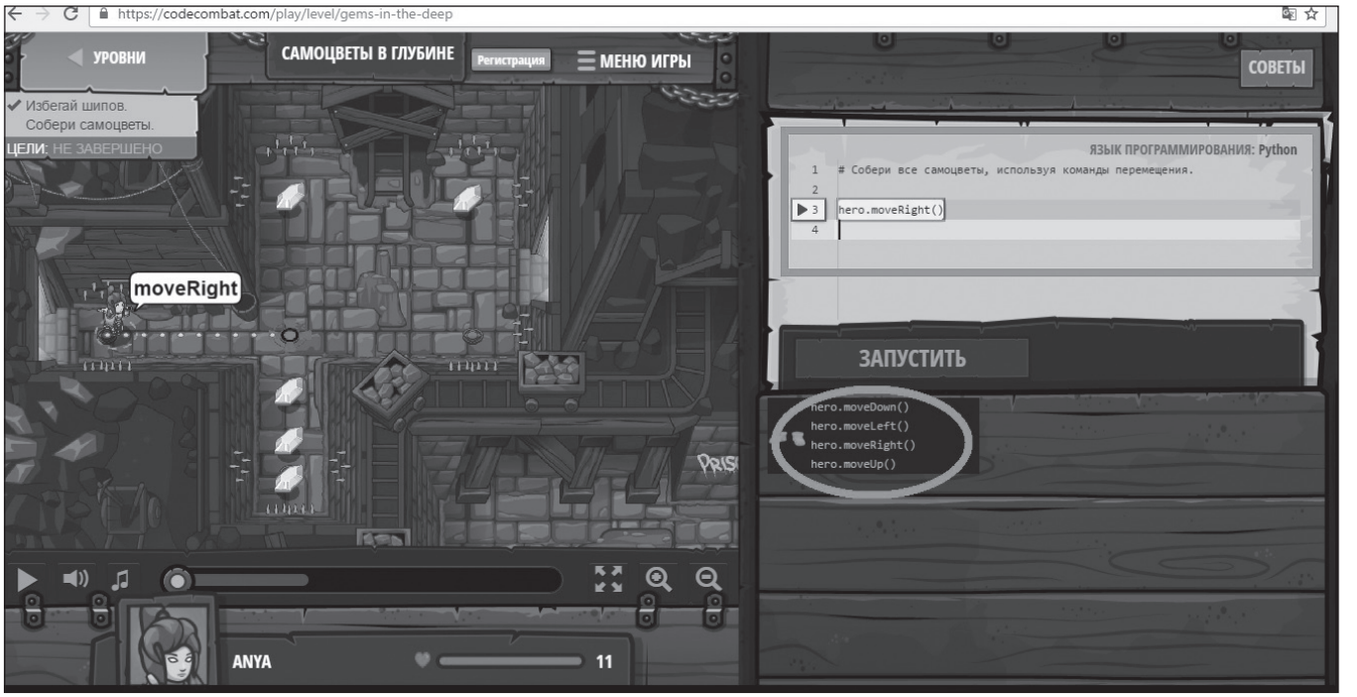


Рис. 8. CodeCombat. Пример работы на уровне 2

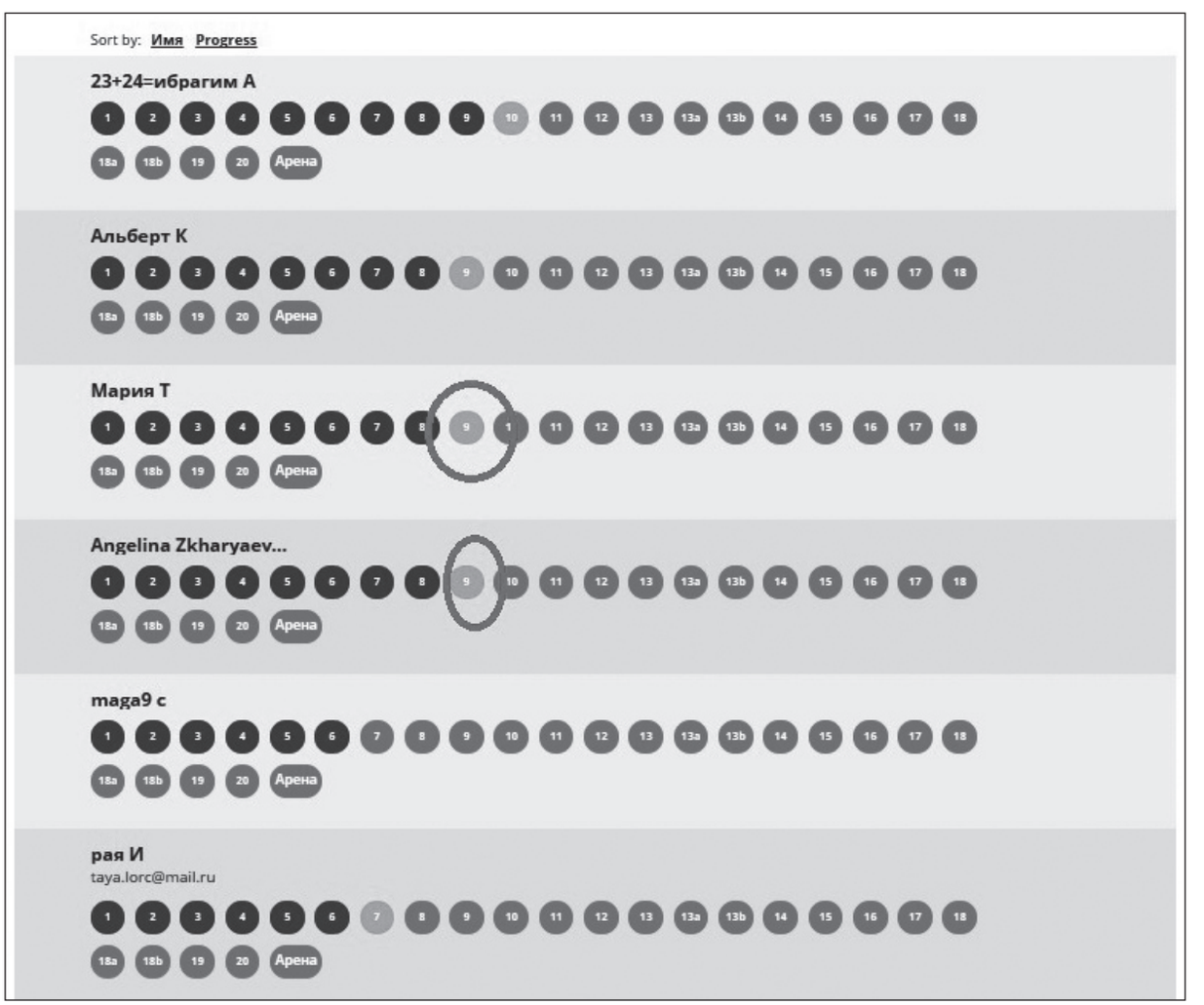


Рис. 9. CodeCombat. Результативность в этапах
(пройденные и непройденные этапы, а также этапы, на которых остановились учащиеся, выделены разным цветом)

Фото. Работа на уроке в CodeCombat

сятого уровня. Также большинство учащихся справились с контрольной работой.

Но самый главный результат данного урока в том, что учащиеся проявили огромный интерес к предложенному способу изучения языка программирования (см. фото). Ведь обычно рассматриваемая тема дается ребятам очень непросто, и трудно мотивировать школьников, не интересующихся программированием. А изучение темы в виде игры проходит легко, новый материал хорошо воспринимается учащимися. Кроме того, ребята меньше боятся совершить ошибку, ведь каждый уровень в игре можно пройти заново.

Безусловно, игровой тип работы при изучении основ программирования наиболее актуален для начального знакомства с языком. Он активизирует мышление, увеличивает интерес и привлекает внимание учащихся.

НОВОСТИ

Intel вступает в гонку автомобильных суперкомпьютеров

Корпорация Intel на выставке CES в Лас-Вегасе представила компьютер Go, до 28 процессоров Xeon которого будут обеспечивать безопасность передвижения робомобилей по улицам. В других вариантах система будет оснащаться чипами Atom следующего поколения и модемом для работы в мобильных сетях пятого поколения (5G).

Первые 40 самоуправляемых автомобилей BMW с Go выйдут на улицы для испытаний уже в этом году.

Робомобилям понадобится огромная вычислительная мощность, чтобы избегать аварий и принимать разумные водительские решения. Эту мощность будут обеспечивать компьютеры вроде Go, для чего они должны работать быстрее, чем геймерские ПК и многие серверы.

Go предстоит соперничать с Drive PX 2, автомобильным суперкомпьютером на водяном охлаждении, который компания Nvidia анонсировала в прошлом году. У него 12 вычислительных ядер и два видеоядра Pascal, но поддержки 5G нет.

Гонка автомобильных компьютеров между Intel и Nvidia начинает напоминать состязание между Intel и AMD времен конца 1990-х — начала 2000-х, когда производители процессоров наперегонки повышали их тактовую частоту. В конечном счете это соперничество сошло на нет, сменившись стремлением к повышению энергоэффективности.

В Intel уверены, что поддержка 5G дает корпорации преимущество перед конкурентами — технология сверхскоростной передачи по сетям мобильной связи данных позволит автомобилям быстро транслировать изображение в облако для анализа и идентификации объектов на улицах.

Go, как и Drive PX 2, сможет самосовершенствоваться. С помощью компьютеров автомобили смогут распознавать пешеходов, различать дорожную разметку и сигналы светофора. При этом решения будут приниматься с помощью алгоритмов, пользующихся информацией с камер, лазерных дальномеров и радиолокаторов.

Intel и Nvidia соревнуются и в скорости вывода робомобилей на улицы. Intel работает над созданием самоуправляемых машин с Mobileye и BMW, а аппаратное обеспечение Nvidia уже используется в электромобилях Tesla с функциями автопилота.

Самый большой плюс робомобилей — это повышенная безопасность, уверена Кэти Винтер, генеральный менеджер подразделения Intel по системам автоматизации вождения, но есть и другие преимущества. В частности, самоуправляемые автомобили могли бы помогать людям рациональнее распоряжаться своим временем, считает она, а также способствовать улучшению экономики работы логистических сетей и служб доставки.

Роботизированные такси в США уже испытывает Uber, экспериментируют с робомобилями и другие компании. Создаются отраслевые альянсы, например, для разработки стандартов обмена сведениями о дорожном движении и погодных условиях.

Помимо чипов Xeon, в аппаратную часть Go входит пара программируемых логических матриц FPGA, которые выполняют в суперкомпьютере те же функции, что и графические процессоры в Nvidia Drive PX 2, — они будут помогать автомобилям мгновенно принимать нужные решения. В Go с процессорами Atom будет только одна FPGA.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

Э. В. Самохина,
Республиканский центр детского творчества, г. Элиста, Республика Калмыкия

ЗАНЯТИЕ ПО ЛЕГОКОНСТРУИРОВАНИЮ НА ТЕМУ «РОБОТ-ЛЯГУШКА»*

Аннотация

В статье представлено занятие кружка «Легоконструирование», в ходе которого на примере робота «Перворобот LEGO WeDo» (LEGO Education WeDo) закрепляется понятие алгоритма как последовательности действий некоторого исполнителя. В программе для робота реализуется циклическая структура алгоритма, благодаря которой удается имитировать поведение лягушки.

Ключевые слова: робот, Перворобот LEGO WeDo, LEGO Education WeDo, зубчатая передача, ременная передача, кулачок, алгоритм, программа, цикл.

Контактная информация

Самохина Элеонора Владимировна, педагог дополнительного образования Республиканского центра детского творчества, г. Элиста, Республика Калмыкия; *адрес:* 358000, Республика Калмыкия, г. Элиста, пр-т О. Бендера, д. 1; *телефон:* (84722) 6-75-90; *e-mail:* esamohina@yandex.ru

E. V. Samokhina,
Republican Center for Children's Creativity, Elista, The Republic of Kalmykia

LESSON ON LEGO CONSTRUCTION ON THE THEME "ROBOT-FROG"

Abstract

The article describes the lesson on LEGO construction, in which the notion of an algorithm as a sequence of actions of a performer is fixed on the example of LEGO Education WeDo robot. In the program for the robot the cyclic structure of the algorithm is implemented, thanks to which it is possible to simulate the behavior of the frog.

Keywords: robot, LEGO Education WeDo, gear, belting, cam, algorithm, program, cycle.

Предлагаемое познавательное занятие разработано для обучения школьников в кружке легоконструирования. Помимо непосредственного конструирования роботов LEGO Education WeDo оно направлено на изучение понятия циклического алгоритма, развитие у школьников любознательности, а также воспитание у них интереса к техническим областям, к работе инженеров и программистов.

Тема занятия: Робот-лягушка.

Образовательные цели:

естественные науки:
- изучение процесса передачи движения и преобразования энергии (в модели);
- понимание того, как при помощи зубчатых колес и кулачков можно изменить направление и скорость движения;
- изучение поведения животных;

технология и проектирование:
- создание, программирование и испытание движущейся модели с целью демонстрации знаний цифровых инструментов и технологических схем и умения работать с ними;

математика:
- измерение времени в секундах с точностью до десятых долей;
- использование чисел для задания звуков и мощности мотора;

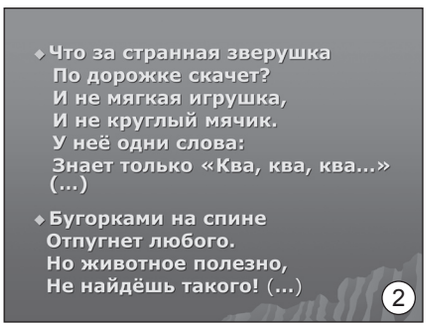
программирование:
- усвоение понятий: алгоритм, исполнитель, свойства алгоритма;
- представление о составлении простейших алгоритмов в среде LEGO Education WeDo.

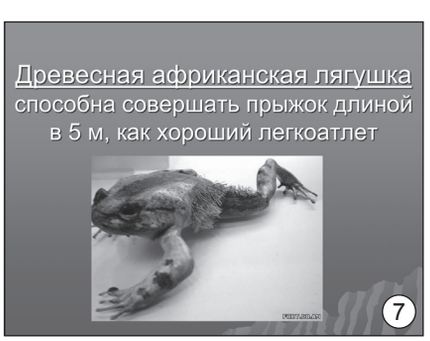
Возраст учащихся: I—IV классы.

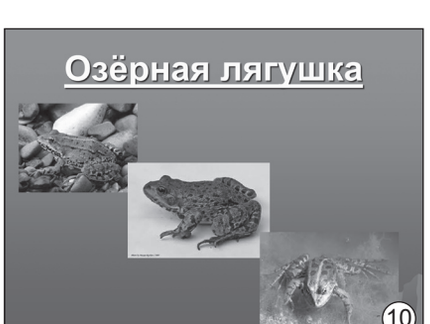
Методические приемы: беседа-диалог, эксперимент, продуктивная деятельность школьников, анализ, подведение итогов.

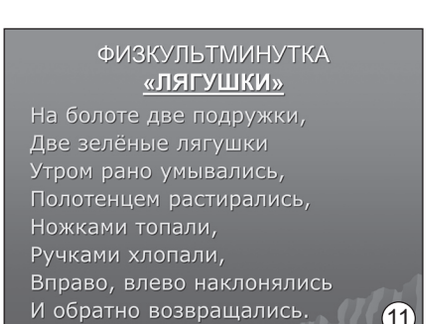
Оснащение занятия:
- компьютеры;
- авторская презентация «О лягушках», созданная в MS PowerPoint;
- инструкция «Сборка лягушки» (http://robotbaza.ru/collection/manuals/lego-wedo);
- программное обеспечение «Перворобот LEGO WeDo» (LEGO Education WeDo Software);
- конструкторы «Перворобот LEGO WeDo» (базовый набор LEGO Education 9580).

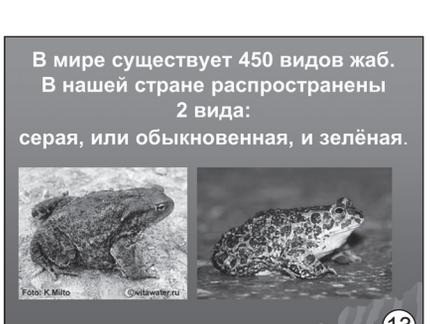
* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2017/

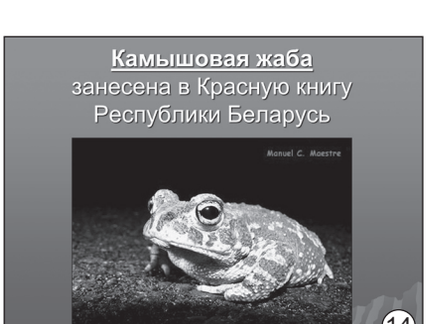












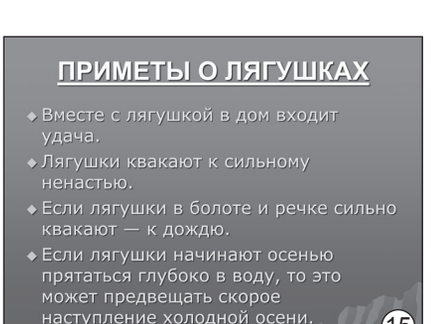


Рис. 1. Слайды презентации «О лягушках»

Список источников, использованных при подготовке урока:

1. Инструкция по сборке робота-лягушки. http://robotbaza.ru/collection/manuals/product/lyagushka
2. *Корягин А. В.* Образовательная робототехника (LegoWeDo): сборник методических рекомендаций и практикумов. М.: ДМК Пресс, 2016.
3. Перворобот LEGO WeDo. Книга для учителя. http://static2.insales.ru/files/1/6403/858371/original/Книга_учителя_Wedo.pdf

Обучение с LEGO Education всегда состоит из четырех этапов:

1) установление взаимосвязей;
2) конструирование;
3) рефлексия;
4) развитие.

Поэтому предлагается следующий **план занятия**:

1. Установление взаимосвязей.
2. Конструирование робота-лягушки.
3. Программирование робота-лягушки.
4. Рефлексия.
5. Развитие модели.
6. Дополнительное задание.
7. Подведение итогов урока.

Ход занятия

1. Установление взаимосвязей

При установлении взаимосвязей учащиеся как бы «накладывают» новые знания на те, которыми они уже обладают, расширяя таким образом свои познания. Для этого предлагается презентация, чтобы проиллюстрировать занятие, заинтересовать учеников, побудить их к обсуждению темы занятия.

Учитель приветствует кружковцев и проверяет их готовность к занятию.

Учитель. Ранее на наших занятиях мы изучали роботы-модели, которые имитируют различных зверей. Сегодня мы продолжим тему животного мира и на примере конструкции робота-лягушки закрепим понятия «зубчатая передача», «ременная передача», а также использование детали конструктора «кулачок». А погрузиться в сегодняшнюю тему нам поможет просмотр презентации «О лягушках»* (рис. 1).

Учитель проводит беседу с ребятами о сложном механизме движения на двух и более конечностях, знакомит с одним из таких видов движения, которым пользуется лягушка.

На этом занятии учащиеся знакомятся с богатой разновидностью животного мира, пополняют свои знания по земноводным (способ передвижения, питание, ареал обитания).

После изучения материала, представленного в презентации, и беседы ребята отвечают на **вопросы:**

- Чем лягушки отличаются от жаб?
- Как ведет себя лягушка?
- Где живет лягушка?
- Как лягушка двигается?
- Какая лягушка самая большая, а какая — самая маленькая?

2. Конструирование робота-лягушки

Учитель. Теперь вы приступаете к построению ходячего робота-лягушки.

Учащиеся приступают к сборке модели, следуя наглядным пошаговым инструкциям в файле *Сборка лягушки.pdf* (рис. 2).

Учитель. Сборка модели шаг за шагом — это алгоритм, который вы выполняете для того, чтобы достичь

* Скачать презентацию можно на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2017/

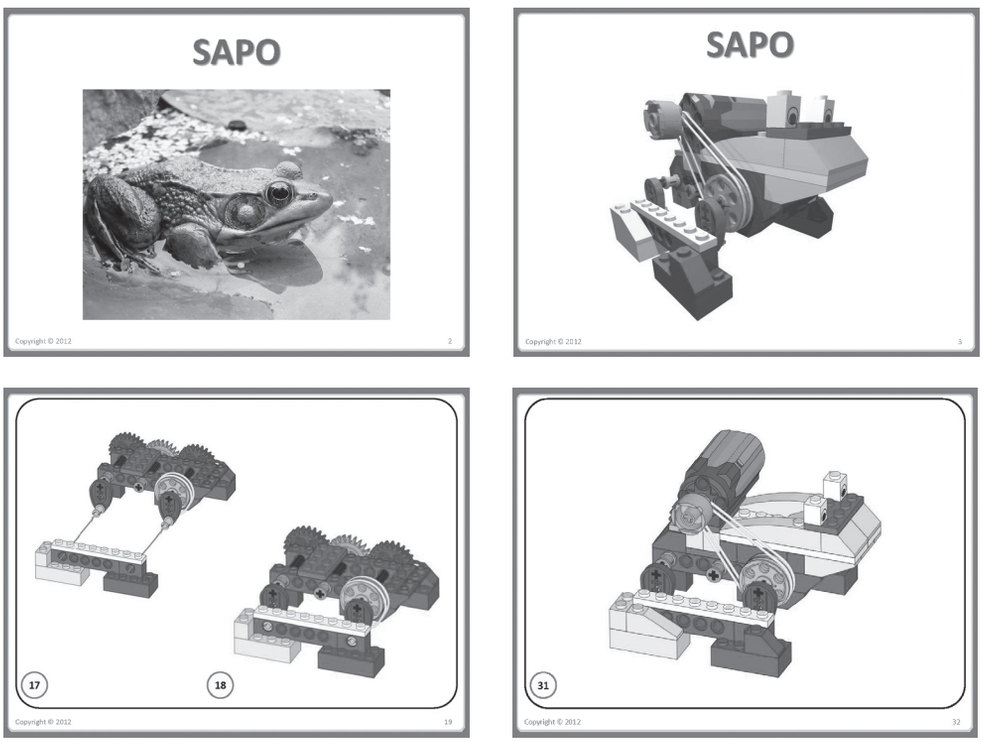


Рис. 2. Фрагмент инструкции по сборке робота-лягушки

конечного результата. Ключевые механизмы, которые приводят данную модель в действие, — это ременная передача, холостая зубчатая передача, кулачковый механизм.

3. Программирование робота-лягушки

Учитель. Переходим к этапу создания программы. Будем писать программу вместе: я — на доске, вы — на компьютере.

Компьютерная программа — это тоже алгоритм. Это набор инструкций, следуя которым компьютер выполняет поставленную задачу.

Моделью лягушки управляет программа, благодаря которой робот имитирует повадки лягушки (рис. 3). Она запускается нажатием блока «Начало». (*Учитель на доске выставляет блок* .)

Чтобы наша лягушка начала двигаться, нужно включить мотор. Подумайте, в какую сторону нужно задать вращение мотора, чтобы лягушка двигалась вперед?

Добавим команды: мощность и направление движения мотора. .

Чтобы модель имела большее сходство с оригиналом, добавим команду звук — кваканье .

Запустим программу и посмотрим работу модели.

В этом случае лягушка двигается вперед (пока мы сами не выключим мотор), квакнув только один раз.

Внесем следующие изменения в программу:

Рис. 3. Программа для управления роботом-лягушкой

В программе реализована циклическая структура алгоритма (цикл бесконечный, работу программы прерывает пользователь).

Энергия передается от компьютера на мотор. От мотора энергия сначала передается шкивам и насаженному с ними на одну ось кулачку, затем с поворотом оси зубчатому колесу, насаженному на одну ось с кулачком, и модель двигается.

Необходимо обеспечить свободное пространство, достаточное для демонстрации движения робота-лягушки.

4. Рефлексия

Учащиеся исследуют, какое влияние на поведение модели оказывает изменение ее конструкции: они заменяют детали, проводят расчеты.

Ученикам предлагается провести **исследование:**

- Изменить положение кулачков и проанализировать характер движения лапок лягушки. Она может передвигать ими по очереди (идти) или одновременно (прыгать).
- Заменить малые шкивы на большие и наоборот.
- Как влияет мощность мотора на движение конструкции?

5. Развитие модели

На этом этапе ученикам предлагаются **идеи по созданию и программированию модели с более сложным поведением:**

- программу можно модифицировать, добавив команду воспроизведения подходящего изображения;
- можно добавить датчик расстояния, чтобы лягушка не сталкивалась с предметами.

6. Дополнительное задание

В качестве дополнительного задания можно предложить учащимся:

- модернизировать конструкцию, например, переделать движущуюся часть;
- устроить соревнование «Чья лягушка быстрее доберется до финиша?»

7. Подведение итогов урока

Учитель. Вы сегодня хорошо поработали. Мы вспомнили и узнали много интересного о животном, живущем рядом с нами, и применили эти знания на практике, построив соответствующую модель. А знания принципа работы зубчатой и ременной передачи позволили «оживить» нашу модель.

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. А. Шишкова,
гимназия № 1576, г. Москва

СМЕШАННОЕ ОБУЧЕНИЕ В ПРОФИЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Аннотация

В статье представлен опыт использования технологии смешанного обучения в преподавании информатики в старших классах общеобразовательной школы на профильном уровне. Рассматриваются методика и практика проведения урока из модуля «Информационное моделирование».

Ключевые слова: смешанное обучение, интерактивные технологии, профильное обучение, интегрированный урок, моделирование движения.

Контактная информация

Шишкова Нелла Антоновна, учитель физики и информатики гимназии № 1576, г. Москва; *адрес:* 127299, г. Москва, ул. Большая Академическая, д. 22А; *телефон:* (495) 450-77-07; *e-mail:* busiky@mail.ru

N. A. Shishkova,
Gymnasium 1576, Moscow

BLENDED LEARNING IN PROFILE TRAINING ON INFORMATICS

Abstract

The article describes the experience of using blended learning technologies in the teaching of informatics in the upper grades of secondary school at the profile level. The methodics and practice of the lesson from the module "Information Modeling" are considered.

Keywords: blended learning, interactive technologies, profile training, integrated lesson, motion simulation.

Перед современным образованием стоят задачи формирования выпускника, владеющего ключевыми компетенциями в различных областях знаний, обладающего креативностью и критическим мышлением, мотивацией на творчество и инновационную деятельность, владеющего основами научных методов познания окружающего мира, нацеленного на образование и самообразование в течение всей жизни. Особенно это относится к учащимся, связывающим свою будущую жизнь с информационными технологиями.

Для достижения этих задач необходимо в образовательном процессе применять новые методики и технологии. Возникает противоречие между классно-урочной системой и необходимостью индивидуализации обучения, формирования критического мышления, ответственности учащегося за свои результаты. Не стоит полностью отбрасывать плюсы классно-урочной системы, такие как возможность общения с преподавателем для более глубокого усвоения материала, организация дискуссий и обсуждений, выполнение совместных групповых заданий с другими учащимися, социализация. В то же время эта система не дает возможности формировать компетенции, связанные с самообразованием, самостоятельным планированием и выполнением индивидуальной работы, поиском и критическим осмыслением информации в различных источниках. Разрешению этого противоречия может способствовать сочетание классно-урочной системы и дистанционного обучающего курса, что становится возможным при использовании технологии смешанного обучения.

Дистанционный курс позволяет учащемуся самостоятельно организовывать свое обучение, учиться планировать, контролировать и анализировать свою учебную деятельность. Это повышает мотивацию к обучению, ответственность за результат, формирует различные компетенции: работы с информацией, поиска информации и ее критического осмысления, применения знаний в различных областях практической деятельности.

В последнее время все больше учащихся выбирают изучение информатики на профильном или углубленном уровне. Это связано с тем, что, с одной стороны, информационные технологии в современной жизни занимают лидирующие позиции с точки зрения получения высокооплачиваемой работы, построения карьеры, признания обществом, а с другой стороны, у общества есть потребность в получении большого количества высокопрофессиональных специалистов в различных областях, связанных с информационными технологиями. По этой причине требования к качеству знаний и уровню компетенций выпускника средней школы становятся все более высокими.

Переход к профильному обучению обусловлен необходимостью создания условий для реализации учащимися своих познавательных способностей, подготовки базы к последующему профессиональному образованию.

У каждого ученика есть свои интеллектуальные и психологические особенности, а также свой диапазон возможного интеллектуального роста. Задача учителя заключается в создании условий для развития познавательной активности ученика за счет применения мотивирующих технологий обучения.

В гимназии № 1576 Москвы созданы все условия для использования технологии смешанного обучения различным предметам [2]. Автор статьи является учителем физики и информатики, а также создателем контента курсов дистанционного сопровождения этой педагогической технологии для физики и информатики на базовом и профильном уровнях. Авторский курс дистанционного сопровождения профильного обучения информатике [1] позволяет педагогу создать условия для реализации требований ФГОС к результатам обучения информатике на углубленном уровне.

Содержание обучения на профильном уровне охватывает следующие темы:

1. Компьютерное и программное обеспечение.
2. Информация и информационные процессы.
3. Теория кодирования информации. Системы счисления.
4. Основы логики и логические основы компьютера.
5. Информационные технологии.
6. Алгоритмизация и программирование.
7. Моделирование и формализация.
8. Компьютерные коммуникации.

На освоение программы по информатике на профильном уровне отводится четыре часа в неделю, 140 часов в X классе и 140 часов в XI классе, всего за период обучения 280 часов. В гимназии № 1576 за основу преподавания профильного курса информатики был взят УМК К. Ю. Полякова и Е. А. Еремина издательства «БИНОМ. Лаборатория знаний».

Курс построен на основе рабочей программы К. Ю. Полякова, Е. А. Еремина [3, 4] и реализован в виртуальной учебной среде Moodle 3.0. Он разбит на темы по неделям, т. е. каждая тема рассчитана на четыре часа учебного времени. Каждая тема содержит материалы для теоретических и практических занятий, а также дополнительный и развивающий материал [1, 2].

Смешанное обучение предполагает сочетание традиционных уроков с учителем и электронного обучения, доступ к которому возможен для ученика в любое время. Именно эту цель и преследует создание электронного контента, без которого невозможно осуществление данной технологии.

Дистанционное сопровождение образовательного процесса содержит элементы для различных типов урока. На теоретических занятиях происходит знакомство учащихся с новым материалом, для которого в курсе представлены презентации, лекции или текстовые файлы. Также в курсе присутствуют задания для совместной работы учащихся на интерактивной доске, для закрепления и отработки полученных навыков. Для проверки усвоения материала созданы автоматически проверяемые элементы курса — тесты. Тесты содержат банки вопросов, разбитые на категории, что позволяет каждому ученику при каждой попытке выполнения получать собственный индивидуальный набор заданий по тем же темам и того же уровня сложности. После выполнения теста ученик может увидеть правильные ответы к своим заданиям, что позволяет ему контролировать правильность выполнения, осуществлять рефлексию.

Самостоятельные и контрольные работы представлены в виде заданий по вариантам, выполняемых в тетради, и в виде автоматически проверяемых тестов. Файлы выполненных практических работ, домашних работ также загружаются в курс для проверки учителем. Все выставленные баллы и комментарии преподавателя доступны учащимся на странице «Оценки».

Для подготовки к практическим занятиям в курсе размещен теоретический материал, используя который учащиеся могут подготовиться к занятиям в любое удобное для них время и в любом месте с подключением к сети Интернет. Задания практических работ содержат подробную инструкцию по выполнению со скриншотами этапов, при необходимости — с дополнительными файлами.

Используя предложенный курс, учитель может построить преподавание в соответствии с обозначенными целями и задачами. Курс содержит все необходимые материалы для проведения уроков различных типов: изучения нового материала, повторения и закрепления, контроля и т. д. Уроки могут проводиться с использованием различных педагогических технологий. Курс дает возможность проводить уроки как в традиционной классно-урочной системе, так и по технологии смешанного обучения, например, по моделям «перевернутый класс» и «ротация станций».

Возможность для ученика в любом месте и в любое время получить доступ к обучающим материалам, самостоятельно планировать свое обучение и отвечать за его результаты, а также получать консультации учителя и составляет основу технологии смешанного обучения.

Рассмотрим уроки 7 и 8 в теме «Модели и моделирование». Эти два урока представляют собой подготовку к практической работе «Моделирование равноускоренного движения» и ее проведение.

Урок «Моделирование равноускоренного движения»

Цель урока: закрепить знания учащихся по теме «Моделирование физических процессов в ЭТ MS Excel».

Оснащение урока:

- интерактивная доска;
- проектор;
- компьютерный класс или планшеты с подключением к Интернету;
- лабораторное оборудование: штатив с закрепленным наклонным желобом, стальной шарик, линейка.

План урока.

1. Актуализация знаний — 10 мин.
2. Проведение физического эксперимента — 15 мин.
3. Подведение итогов эксперимента, обсуждение принципов математического моделирования — 15 мин.
4. Подведение итогов урока, рефлексия — 5 мин.

Ход урока

1. Актуализация знаний

На этапе актуализации знаний материал рассматривается с использованием файла «бесконечной доски» (рис. 1), ссылка на который расположена в курсе [5].

Учитель беседует с учащимися, побуждая их повторить формулы по равноускоренному движению и динамике движения тел в гравитационном поле, которые потребуются для выполнения математического моделирования свободного падения тела без учета сопротивления воздуха и с учетом сопротивления. Целью моделирования является определение условий, при которых сопротивлением воздуха можно пренебречь.

Учитель задает **вопросы**:

- Что такое моделирование? (*Это исследование реальных объектов и явлений на моделях.*)
- Какими свойствами обладает модель? (*Она обладает теми же существенными признаками, что и изучаемый объект.*)
- Для чего может использоваться моделирование равноускоренного движения? (*Для расчета места приземления парашютиста или груза, для расчета попадания в цель.*)

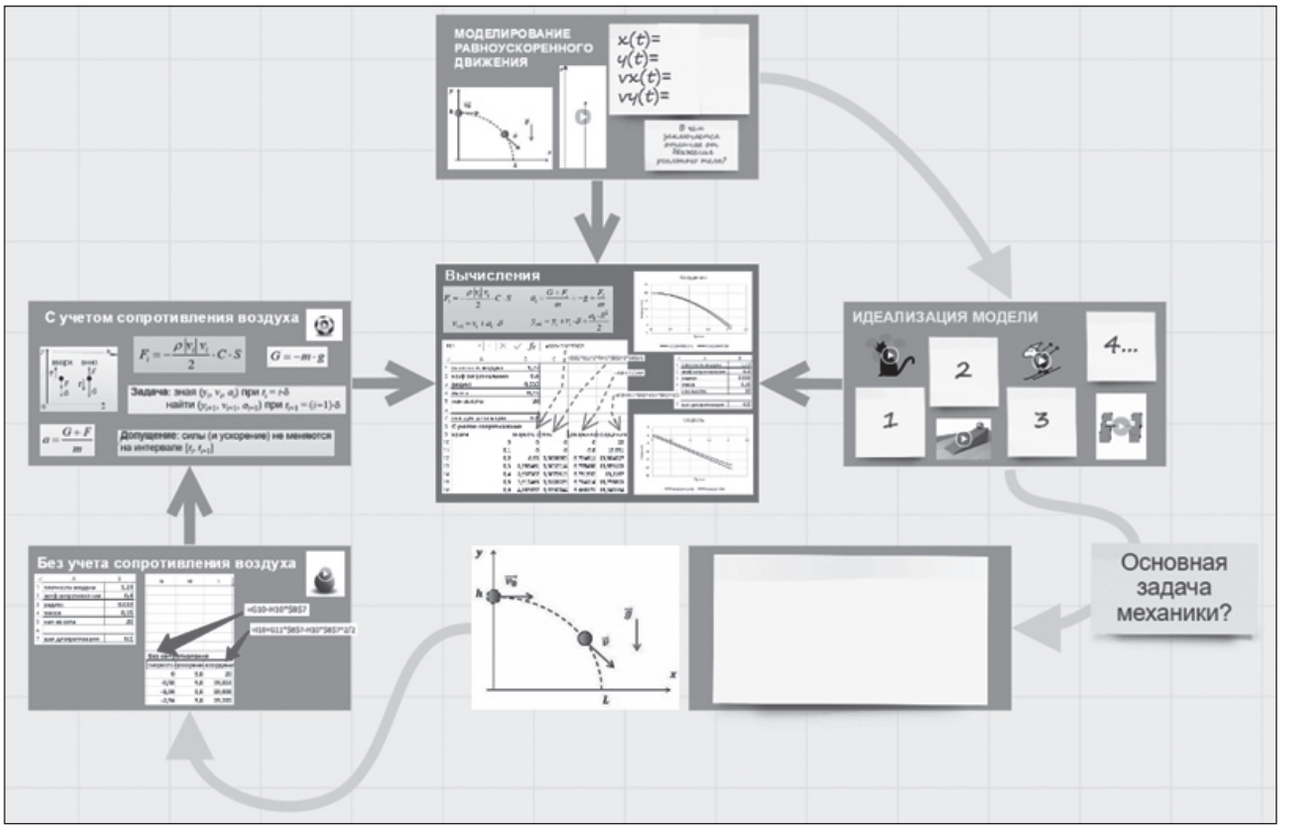


Рис. 1. Общий вид файла «бесконечной доски» для повторения и объяснения принципов моделирования

На уроке учащимся будет предложено провести физический эксперимент по изучению движения тела, брошенного горизонтально, а затем с помощью электронных таблиц провести математическое моделирование для двух случаев: без учета сопротивления воздуха и с учетом этого сопротивления.

К доске вызывается ученик для записи формул, характеризующих движение тела, брошенного горизонтально.

Изменение координат тела по осям x и y:

$$x(t) = v_0 t;$$

$$y(t) = h - \frac{gt^2}{2}.$$

Формулы зависимости скорости от времени:

$$v_x(t) = v_0;$$

$$v_y(t) = -gt.$$

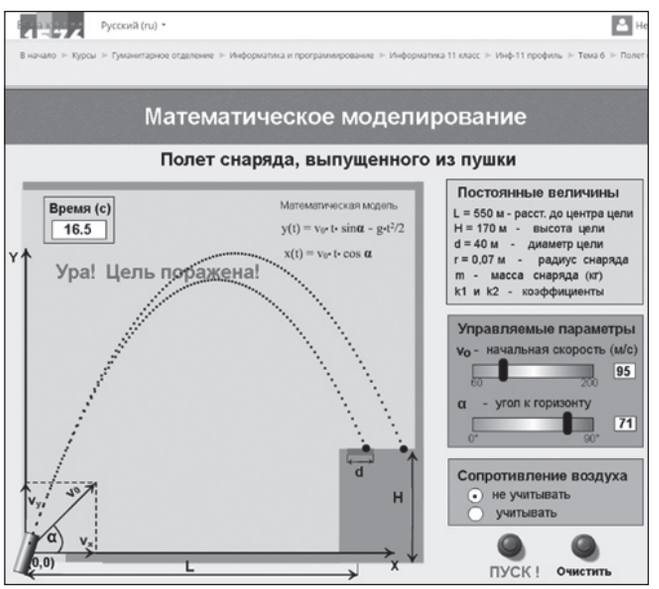


Рис. 2. Интерактивное задание, в котором нужно поразить цель, изменяя начальные условия движения тела: скорость и угол к горизонту

Рис. 3. Иллюстрации для внесения формул и обсуждения принципов моделирования

Рис. 4. Заготовка для вывода уравнения на интерактивной доске

Учитель задает **вопросы**:

- В чем отличие движения реального тела и модели? (*В моделировании не учитываются сила сопротивления воздуха, изменение ускорения свободного падения в зависимости от высоты, плотность воздуха, ветер и т. д.*)
- Всегда ли можно пренебрегать этими отличиями? (*Не всегда. Если сила сопротивления настолько замедляет движение, что меняется его тип, то пренебрегать этими отличиями нельзя.*)

Доказать, что движение тела, брошенного горизонтально в поле силы тяжести, является равноускоренны, можно, если получить уравнение траектории (должна получиться квадратичная зависимость $y(x)$) или построить график этой зависимости (должна получиться парабола). Для экспериментального определения коэффициентов в уравнении $y(x)$ выведем формулу, с помощью которой определяется начальная скорость тела по результатам измерения высоты и дальности полета тела.

К доске вызывается ученик, который показывает, что для определения начальной скорости полета достаточно измерить высоту и дальность полета тела (рис. 4).

$$L = \upsilon t;$$

$$h = \frac{gt^2}{2}.$$

Отсюда:

$$\upsilon_0 = \sqrt{\frac{gL^2}{2h}}.$$

Таким образом, экспериментально определив начальную скорость движения тела, брошенного горизонтально, учащиеся смогут записать уравнение траектории и сделать вывод о характере этого движения.

Второй способ — построение графика — используется при моделировании движения в электронных таблицах.

2. Проведение физического эксперимента

Учащиеся получают оборудование для проведения эксперимента и раздаточный материал для заполнения (рис. 5).

Учащиеся:

- моделируют движение тела, брошенного горизонтально;

Рис. 5. Раздаточный материал для записи результатов эксперимента

- измеряют высоту и дальность полета;
- на основании этих данных по ранее выведенной формуле рассчитывают начальную скорость;
- записывают уравнение траектории;
- убеждаются в равноускоренном характере движения тела.

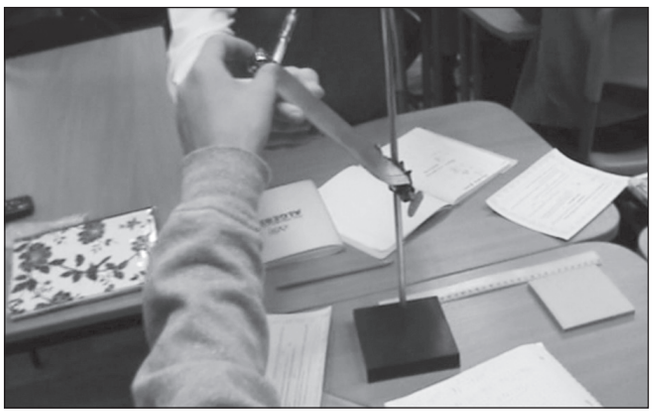


Фото 1. Проведение эксперимента

3. Подведение итогов эксперимента, обсуждение принципов математического моделирования

Учитель обсуждает с учениками, как можно построить траекторию движения тела с помощью электронных таблиц, какие формулы необходимо использовать для вычисления положения тела и его скорости в любой момент времени.

Фото 2. Обсуждение идеализации модели

Учитель объясняет, как можно смоделировать движение с помощью электронных таблиц, построить график зависимости координаты от времени и убедиться в равноускоренном характере движения. На интерактивной доске и на компьютерах учащихся демонстрируются слайды, поясняющие процесс моделирования (рис. 6).

Затем учитель объясняет, как производится расчет с учетом сопротивления воздуха (рис. 7). В этом случае движение уже не будет равноускоренным, ускорение меняется, поскольку меняется сила сопротивления, зависящая от скорости. В свою очередь, от ускорения зависит скорость движения. Получается сложная зависимость физических величин друг от друга, учесть которую в рамках школьных знаний можно только при численном моделировании. В этом случае считается, что на промежутках времени, равных шагу дискретизации,

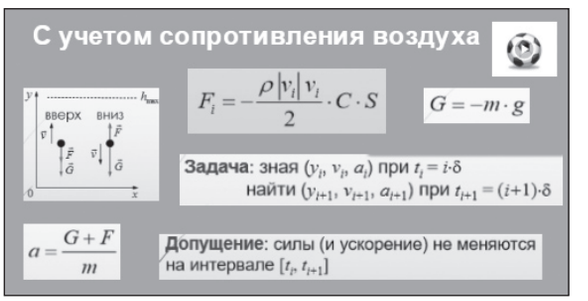


Рис. 6. Иллюстрации к объяснению моделирования в ЭТ

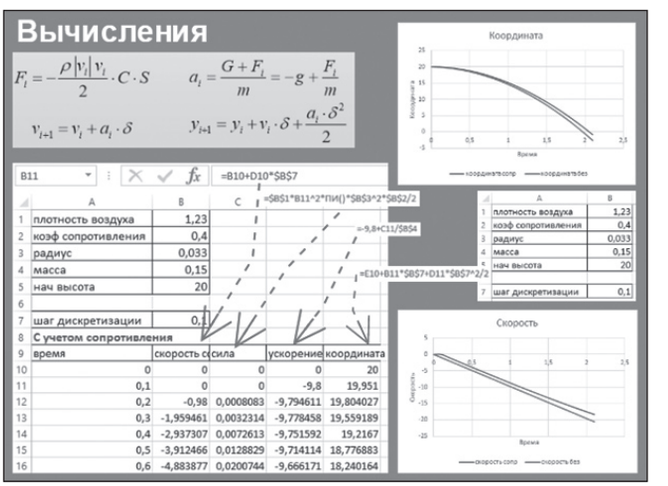


Рис. 7. Моделирование движения с учетом сопротивления воздуха

величины не меняются, что позволяет упростить расчеты на этих малых промежутках времени. Учащиеся отвечают, какие величины зависят друг от друга, какие формулы и в каком порядке позволяют произвести вычисления. По результатам численного моделирования ученики должны с помощью ЭТ построить графики зависимости скорости и координаты от времени для двух случаев моделирования. Сделать вывод о случаях, в которых нельзя пренебречь сопротивлением воздуха.

4. Подведение итогов урока, рефлексия

Учитель спрашивает, остались ли проблемы и трудные места в подготовке к практической работе, отвечает на вопросы учащихся.

Урок «Выполнение практической работы “Моделирование равноускоренного движения”»

Цель урока: закрепить знания учащихся по теме «Моделирование физических процессов в ЭТ MS Excel».

Оснащение урока:
- компьютерный класс или планшеты с подключением к Интернету;
- программа MS Excel.

План урока.
1. Организационный этап — 3 мин.
2. Выполнение моделирования свободного падения мяча без учета сопротивления воздуха — 7–12 мин.
3. Выполнение моделирования свободного падения мяча с учетом сопротивления воздуха — 15–20 мин.
4. Выполнение моделирования свободного падения парашютиста с учетом сопротивления воздуха (дополнительное задание при наличии времени на уроке) — 15 мин.
5. Подведение итогов урока, рефлексия — 5 мин.

Ход урока

1. Организационный этап

Через свои аккаунты учащиеся входят в базу дистанционных курсов и получают индивидуальные задания практической работы (рис. 8).

2. Выполнение моделирования свободного падения мяча без учета сопротивления воздуха

Учащиеся создают файл электронных таблиц, вносят исходные данные, формулы, производят математическое моделирование указанного в задании случая.

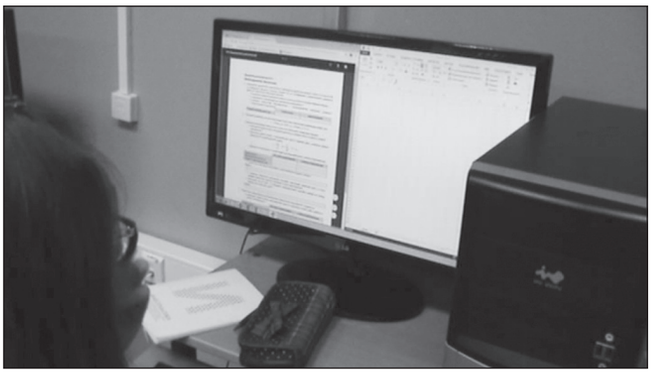


Фото 3. Компьютер ученика

3. Выполнение моделирования свободного падения мяча с учетом сопротивления воздуха

На том же листе MS Excel создается таблица для расчета скорости, силы сопротивления, ускорения и координаты в случае учета силы сопротивления воздуха (рис. 9).

4. Выполнение моделирования свободного падения парашютиста с учетом сопротивления воздуха (дополнительное задание при наличии времени на уроке)

На отдельном листе MS Excel выполняется моделирование движения парашютиста для следующих исходных данных: парашютист массой 90 кг разгоняется в свободном падении до скорости 10 м/с и на высоте 50 м

Практическая работа № 7. Моделирование движения

1. С помощью электронных таблиц смоделируйте полет мяча, брошенного с высоты 20 м без начальной скорости, при

$$r = 33\,\text{мм},\ m = 150\,\text{г},\ \rho = 1{,}23\ \text{кг/м}^3,\ \delta = 0{,}1\,\text{с},\ C = 0{,}4.$$

Выполните следующие задания:
- определите время полета, постройте графики скорости и координаты в зависимости от времени *без учета сопротивления воздуха*;
- определите время полета, постройте графики скорости и координаты в зависимости от времени *с учетом сопротивления воздуха*;
- сравните эти результаты;

	без учёта сопротивления	с учётом сопротивления
Время полета, с		
Скорость приземления, м/с		

- можно ли в этой задаче пренебречь сопротивлением воздуха? почему?

Ответ:

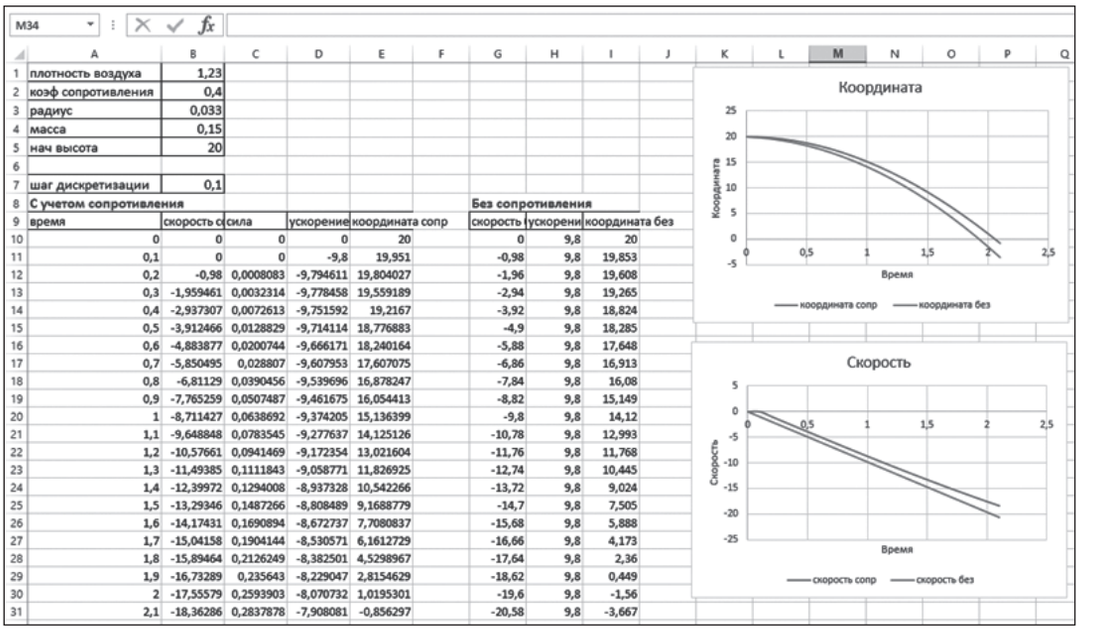
- уменьшите шаг до 0,01 с и повторите моделирование; сделайте выводы по поводу выбора шага в данной задаче.

Ответ:

2. Парашютист массой 90 кг разгоняется в свободном падении до скорости 10 м/с и на высоте 50 м раскрывает парашют, площадь которого 55 м^2. Коэффициент сопротивления парашюта равен 0,9. Выполните следующие задания:
- постройте графики изменения скорости и высоты полета в течение первых 4 секунд;
- определите, с какой скоростью приземлится парашютист?
- сравните результаты моделирования с установившимся значением скорости, вычисленным теоретически.

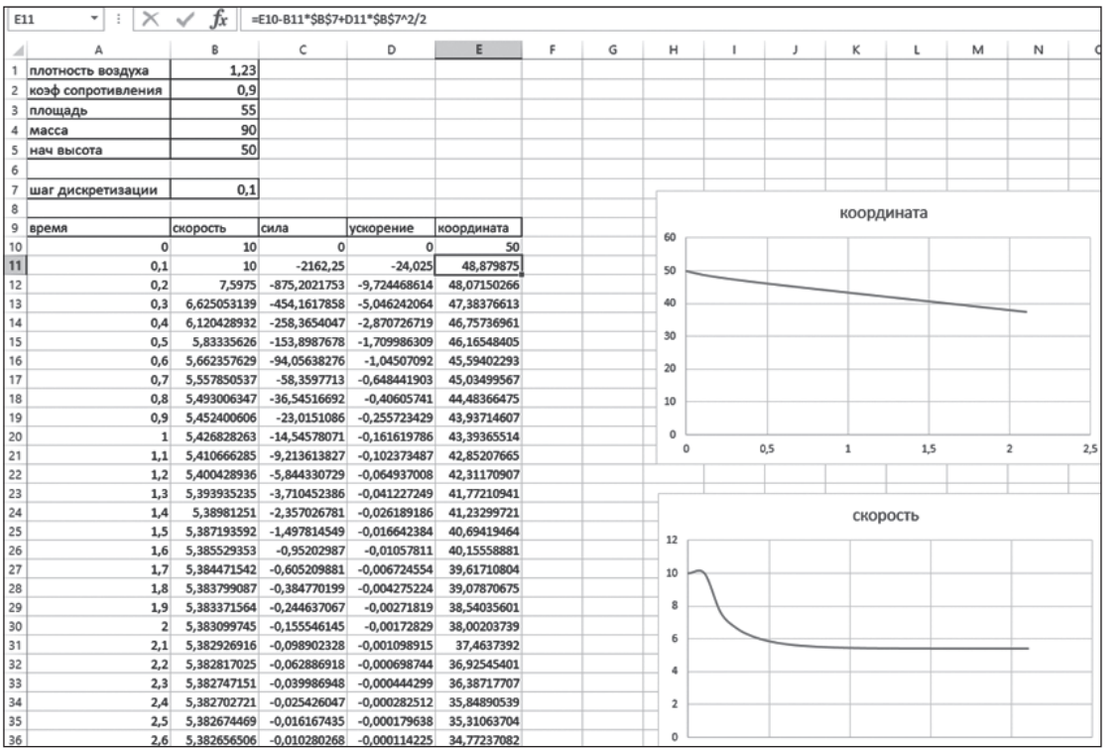
	теоретически	моделирование
Скорость приземления, м/с		

Рис. 8. Задания для практической работы



	A	B	C	D	E	F	G	H	I
1	плотность воздуха	1,23							
2	коэф сопротивления	0,4							
3	радиус	0,033							
4	масса	0,15							
5	нач высота	20							
6									
7	шаг дискретизации	0,1							
8	С учетом сопротивления						Без сопротивления		
9	время	скорость с	сила	ускорение	координата сопр		скорость	ускорени	координата без
10	0	0	0	0	20		0	9,8	20
11	0,1	0	0	-9,8	19,951		-0,98	9,8	19,853
12	0,2	-0,98	0,0008083	-9,794611	19,804027		-1,96	9,8	19,608
13	0,3	-1,959461	0,0032314	-9,778458	19,559189		-2,94	9,8	19,265
14	0,4	-2,937307	0,0072613	-9,751592	19,2167		-3,92	9,8	18,824
15	0,5	-3,912466	0,0128829	-9,714114	18,776883		-4,9	9,8	18,285
16	0,6	-4,883877	0,0200744	-9,666171	18,240164		-5,88	9,8	17,648
17	0,7	-5,850495	0,028807	-9,607953	17,607075		-6,86	9,8	16,913
18	0,8	-6,81129	0,0390456	-9,539696	16,878247		-7,84	9,8	16,08
19	0,9	-7,765259	0,0507487	-9,461675	16,054413		-8,82	9,8	15,149
20	1	-8,711427	0,0638692	-9,374205	15,136399		-9,8	9,8	14,12
21	1,1	-9,648848	0,0783545	-9,277637	14,125126		-10,78	9,8	12,993
22	1,2	-10,57661	0,0941469	-9,172354	13,021604		-11,76	9,8	11,768
23	1,3	-11,49385	0,1111843	-9,058771	11,826925		-12,74	9,8	10,445
24	1,4	-12,39972	0,1294008	-8,937328	10,542266		-13,72	9,8	9,024
25	1,5	-13,29346	0,1487266	-8,808489	9,1688779		-14,7	9,8	7,505
26	1,6	-14,17431	0,1690894	-8,672737	7,7080837		-15,68	9,8	5,888
27	1,7	-15,04158	0,1904144	-8,530571	6,1612729		-16,66	9,8	4,173
28	1,8	-15,89464	0,2126249	-8,382501	4,5298967		-17,64	9,8	2,36
29	1,9	-16,73289	0,235643	-8,229047	2,8154629		-18,62	9,8	0,449
30	2	-17,55579	0,2593903	-8,070732	1,0195301		-19,6	9,8	-1,56
31	2,1	-18,36286	0,2837878	-7,908081	-0,856297		-20,58	9,8	-3,667

Рис. 9. Скриншот выполненного задания



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	плотность воздуха	1,23												
2	коэф сопротивления	0,9												
3	площадь	55												
4	масса	90												
5	нач высота	50												
6														
7	шаг дискретизации	0,1												
8														
9	время	скорость	сила	ускорение	координата									
10	0	10	0	0	50									
11	0,1	10	-2162,25	-24,025	48,879875									
12	0,2	7,5975	-875,2021753	-9,724468614	48,07150266									
13	0,3	6,625053139	-454,1617858	-5,046242064	47,38376613									
14	0,4	6,120428932	-258,3654047	-2,870726719	46,75736961									
15	0,5	5,83335626	-153,8987678	-1,709986309	46,16548405									
16	0,6	5,662357629	-94,05638276	-1,04507092	45,59402293									
17	0,7	5,557850537	-58,3597713	-0,648441903	45,03499567									
18	0,8	5,493006347	-36,54516692	-0,40605741	44,48366475									
19	0,9	5,452400606	-23,0151086	-0,255723429	43,93714607									
20	1	5,426828263	-14,54578071	-0,161619786	43,39365514									
21	1,1	5,410666285	-9,213613827	-0,102373487	42,85207665									
22	1,2	5,400428936	-5,844330729	-0,064937008	42,31170907									
23	1,3	5,393935235	-3,710452386	-0,041227249	41,77210941									
24	1,4	5,38981251	-2,357026781	-0,026189186	41,23299721									
25	1,5	5,387193592	-1,497814549	-0,016642384	40,69419464									
26	1,6	5,385529353	-0,95202987	-0,01057811	40,15558881									
27	1,7	5,384471542	-0,605209881	-0,006724554	39,61710804									
28	1,8	5,383799087	-0,384770199	-0,004275224	39,07870675									
29	1,9	5,383371564	-0,244637067	-0,00271819	38,54035601									
30	2	5,383099745	-0,155546145	-0,00172829	38,00203739									
31	2,1	5,382926916	-0,098902328	-0,001098915	37,4637392									
32	2,2	5,382817025	-0,062886918	-0,000698744	36,92545401									
33	2,3	5,382747151	-0,039986948	-0,000444299	36,38717707									
34	2,4	5,382702721	-0,025426047	-0,000282512	35,84890539									
35	2,5	5,382674469	-0,016167435	-0,000179638	35,31063704									
36	2,6	5,382656506	-0,010280268	-0,000114225	34,77237082									

Рис. 10. Моделирование с учетом сопротивления воздуха

раскрывает парашют, площадь которого 55 м^2. Коэффициент сопротивления парашюта равен 0,9, плотность воздуха 1,23 кг/м^3.

(*Лист решения этой задачи представлен на рисунке 10.*)

5. Подведение итогов урока, рефлексия

Подводятся итоги работы, учащиеся называют условия, при которых нельзя пренебречь при расчетах сопротивлением воздуха.

Ученики загружают свои файлы MS Excel с работой на школьный сервер через Moodle для проверки.

Список использованных источников

1. Авторский курс дистанционной поддержки профильного обучения информатике в 11 классе. http://do.gymn1576.ru/course/index.php?categoryid=25

2. База курсов дистанционной поддержки обучения гимназии № 1576. http://do.gymn1576.ru/

3. *Поляков К. Ю., Еремин Е. А.* Информатика. 11 класс. Базовый и углубленный уровни: учебник в 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2013.

4. Сайт К. Ю. Полякова. http://kpolyakov.spb.ru/

5. Файл «бесконечной доски» к уроку «Моделирование движения». https://realtimeboard.com/app/board/o9J_k0ue1Zc=/

МЕТОДИЧЕСКАЯ КОПИЛКА

А. В. Карандаева,
Мордовский государственный педагогический институт им. М. Е. Евсевьева, г. Саранск, Республика Мордовия

УВЛЕКАТЕЛЬНАЯ РОБОТОТЕХНИКА ДЛЯ ШКОЛЬНИКОВ*

Аннотация

В статье описывается деятельность по обучению робототехнике в Центре молодежного инновационного творчества «МИР-3D», приводится учебно-тематическое планирование.

Ключевые слова: робототехника, программирование, конструирование, дополнительная общеобразовательная программа, робот, LEGO MINDSTORMS Education EV3.

Контактная информация

Карандаева Анастасия Валерьевна, тьютор лаборатории «Увлекательная робототехника» Центра молодежного инновационного творчества «МИР-3D», студентка физико-математического факультета Мордовского государственного педагогического института им. М. Е. Евсевьева, г. Саранск, Республика Мордовия; *адрес:* 430007, Республика Мордовия, г. Саранск, ул. Студенческая, д. 11а; *телефон:* (8342) 33-92-84; *e-mail:* karandaeva.nastena@gmail.com

A. V. Karandaeva,
Mordovian State Pedagogical Institute named after M. E. Evsevjev, Saransk, The Republic of Mordovia

EXCITING ROBOTICS FOR SCHOOLCHILDREN

Abstract

The article describes the work on the training of robotics in the Center for Youth Innovation Creativity "MIR-3D", the curriculum-thematic planning is given.

Keywords: robotics, programming, construction, additional general education program, robot, LEGO MINDSTORMS Education EV3.

Одним из перспективных направлений обучения, в котором заинтересованы и школы, и школьники, является робототехника — «прикладная наука, занимающаяся разработкой автоматизированных технических систем и являющаяся важнейшей технической основой интенсификации производства» [1, с. 3].

В школах обычно закладываются основы робототехники, а сама она изучается на дополнительных занятиях, например в кружках или на факультативах. Как правило, на занятия приходят ученики одного возраста, которые осваивают азы программирования и конструирования роботов. В связи с тем что оснащение школ роботами оставляет желать лучшего, в большинстве случаев на одну группу учащихся приходится по одному-два комплекта роботов. Несколько иначе обстоит дело в центрах дополнительного образования, где и количество комплектов больше, и процесс обучения робототехнике строится иначе. В качестве примера **опишем работу по дополнительной общеобразовательной программе «Увлекательная робототехника»**, реализуемой в Центре молодежного инновационного творчества «МИР-3D» (Мордовский государственный педагогический институт им. М. Е. Евсевьева, г. Саранск).

Обучение по указанной программе предполагает вовлечение в занятия робототехникой **учащихся от 7 до 15 лет**, обучающихся в разных школах города Саранска. При таком подходе дополнительная программа реализуется в разновозрастных группах, когда более старшие ученики выступают помощниками для младших. Совместная деятельность учащихся из разных школ, кроме того, дает возможность ученикам в неявной форме знакомиться и с достижениями в робототехнике в разных школах города.

Программа предполагает продолжительность обучения **72 часа (1 раз в неделю по 2 учебных часа)** и включает в себя **два модуля:**

- первый позволяет сформировать у слушателей навыки конструирования роботов и познакомиться с программой LEGO MINDSTORMS;
- в рамках второго школьники самостоятельно создают и программируют роботов.

На с. 25 представлен **учебно-тематический план программы.**

Содержание и структура программы «Увлекательная робототехника» направлены на формирование инженерного мышления подрастающего поколения, развитие мотивации к познанию и творчеству, осознание общечеловеческих ценностей.

Техническое оснащение программы включает в себя следующее оборудование:

- образовательный конструктор LEGO MINDSTORMS Education EV3 (9 комплектов);
- персональный компьютер (5 шт.);
- планшеты на платформе Android (5 шт.).

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2017/

Учебно-тематический план программы

№ п/п	Наименование разделов и тем	Кол-во часов
1	**Введение в Робототехнику. Основы конструирования роботов**	**36**
1.1	Понятие о робототехнике. Роботы в современном мире	1
1.2	Среда конструирования. Знакомство с деталями конструктора	2
1.3	Моторы и датчики. Простейшая модель робота. Робот «Пятиминутка». Прочность конструкции и способы повышения прочности. Устойчивость модели	9
1.4	Способы передачи движения. Тестирование моторов и датчиков. Проект «Мой первый робот»	7
1.5	Конструирование простейших роботов и управление ими. Проект «Назад в детство. Роботы для развлечений»	8
1.6	Управление моторами и датчиками. Проект «Робот не просто игрушка...»	9
2	**Создание и программирование роботов**	**36**
2.1	Программа LEGO MINDSTORMS. Понятие команды. Программа и программирование. Простейшие программы для управления роботами	5
2.2	Датчик касания. Обнаружение препятствия с помощью датчика касания. Датчик освещенности. Ограничение движения линией. Движение вдоль линии с применением датчика освещенности. Разработка программ для датчиков касания и освещенности	4
2.3	Ультразвуковой датчик. Определение роботом расстояния до препятствия. Изготовление роботов для состязаний «Движение по линии», «Лестница» с использованием одного датчика. Создание программ для роботов	4
2.4	Блок Bluetooth, установка соединения. Загрузка с компьютера	3
2.5	Сборка роботов по инструкции. Испытание роботов. Проект «Уникальный робот». Фотоконкурс «Мой увлекательный мир робототехники»	9
2.6	Создание собственного робота. Написание программы. Творческий проект «Будущее за роботами»	9
2.7	Итоговый проект «Состязание роботов»	2
	ИТОГО:	**72**

В **программное обеспечение** входят приложения:
- LEGO MINDSTORMS EV3;
- LEGO MINDSTORMS Commander;
- LEGO MINDSTORMS Programmer;
- LEGO MINDSTORMS Fix Factory.

Работа по дополнительной общеобразовательной программе реализуется следующим образом.

Первоначально собирается большее количество школьников с целью определения уровня их подготовки в области робототехники и для того, чтобы сформировать небольшие группы (до девяти человек) с примерно одинаковым уровнем знаний и умений.

Учащимся демонстрируется презентация, в которой представлены возможности современной робототехники* (рис. 1).

Далее в форме беседы определяется, что школьники знают о роботах. Кроме того, им предлагается следующая **анкета:**

1. Проходил(а) ли ты ранее обучение по робототехнике?
2. Собирал(а) ли ты роботов? Если да, то из какого набора?
3. Знаком(а) ли ты с конструктором Lego Mindstorms Education EV3?
4. Чему ты хочешь научиться на данной образовательной программе?
5. Каких роботов ты хотел(а) бы собрать?

Благодаря указанному комплексу мероприятий ученики рационально распределяются по группам.

Занятия первого модуля в основном строятся по следующей схеме:
- демонстрируется физический компонент из конструктора, показывается его назначение и демонстрируется способ сборки;
- по предложенной инструкции предлагается собрать простейшего робота (фото 1);
- при успешном создании ученикам выдаются инструкции по созданию более сложных роботов (фото 2).

Практика показала, что у большинства участников появляется желание самостоятельно собрать собственных роботов. Пример одного из них приведен на фото 3.

Параллельно с конструированием роботов ученики осваивают мобильное приложение LEGO MINDSTORMS Commander, с помощью которого учатся управлять созданными роботами (рис. 2). Программа представляет собой управляющее приложение для LEGO MINDSTORMS, подключается к модулю EV3 по Bluetooth.

Работая с программой, ученики убеждаются в том, что с ее помощью можно выполнять лишь элементарные манипуляции с роботом — движение конечностей,

* Презентацию можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2017/

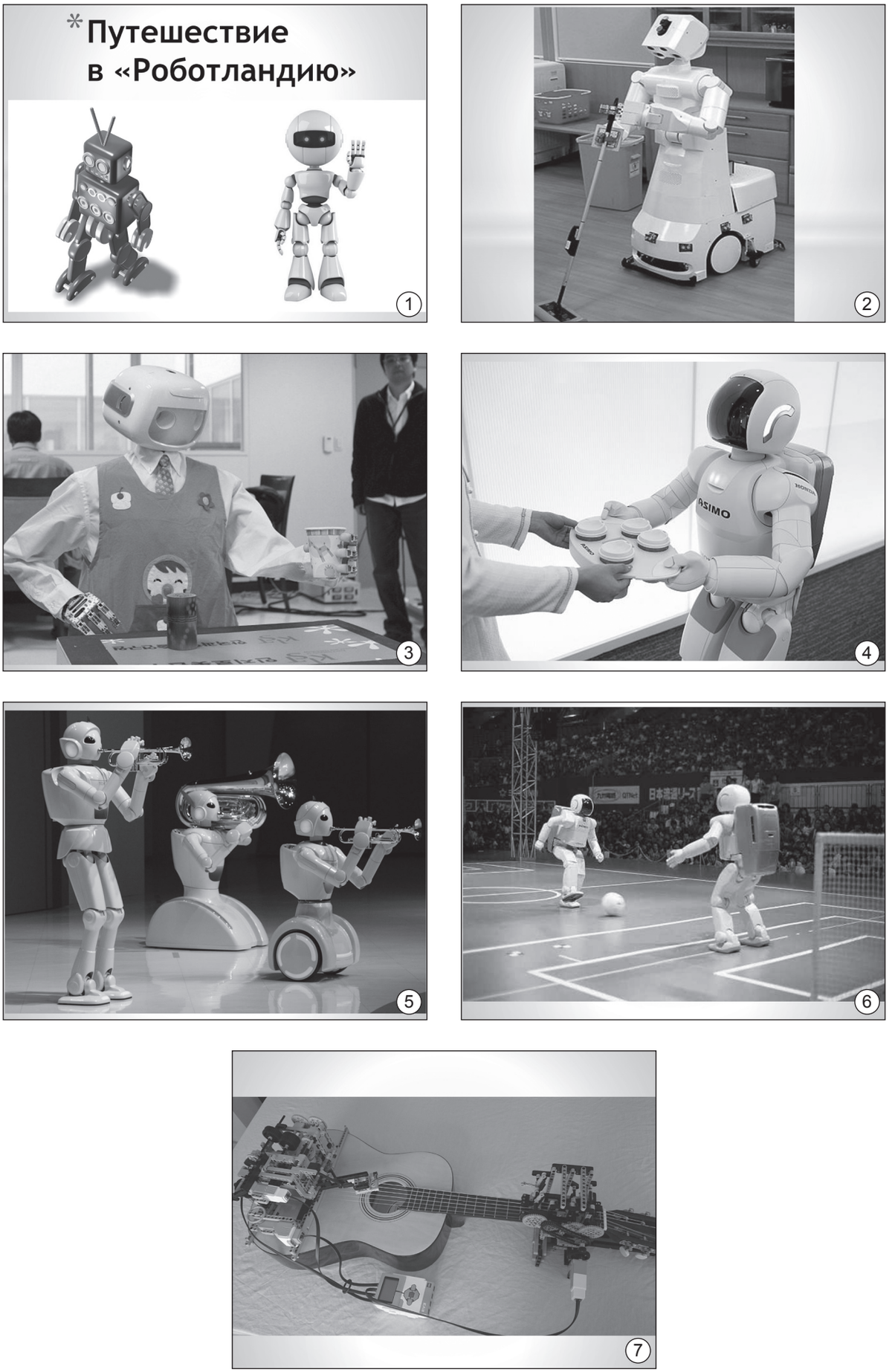


Рис. 1. Слайды презентации

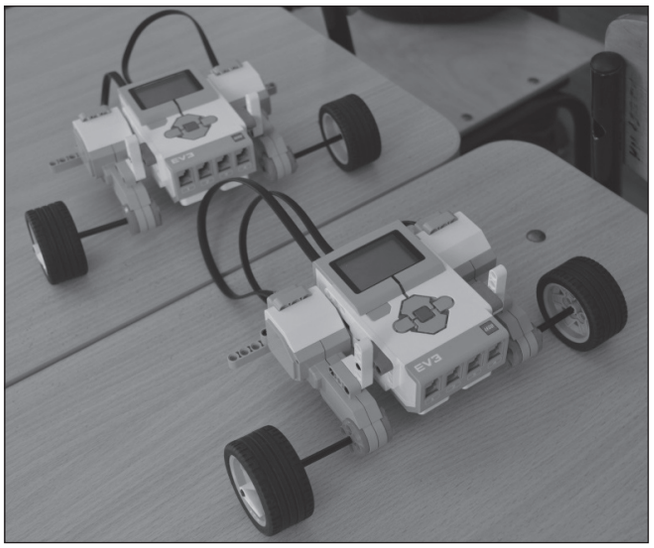


Фото 1. Робот «Пятиминутка»

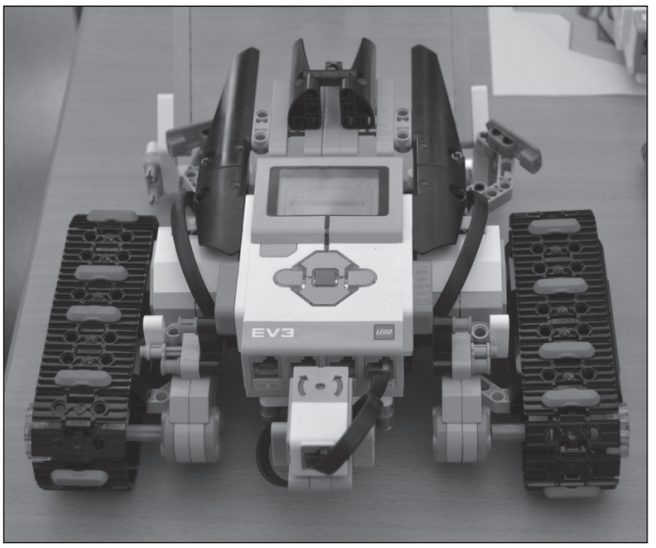


Фото 2. Робот «Танкбот»

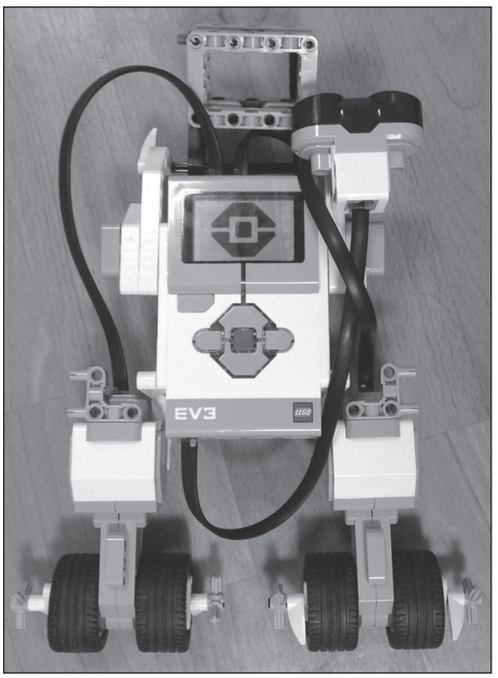


Фото 3. Робот «Перевозчик»

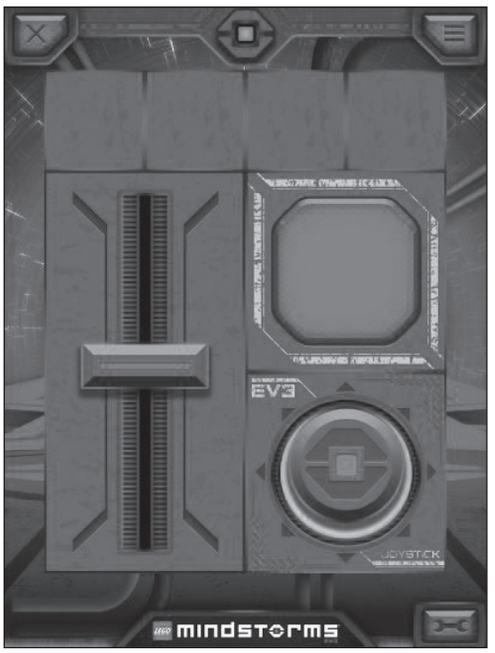


Рис. 2. Интерфейс приложения LEGO MINDSTORMS Commander

повороты или наклоны головы и т. д. Такой недостаток программы служит неявной мотивацией школьников к освоению более функциональных приложений для написания собственных программ для манипуляции роботами.

Тем самым незаметно происходит переход ко **второму модулю,** где школьниками осваивается программа LEGO MINDSTORMS EV3 (рис. 3).

Занятия по второму модулю программы организуются в виде проектной деятельности, когда при минимальной помощи тьютора учащиеся самостоятельно конструируют роботов и программируют управление ими.

Некоторые из результатов деятельности школьников приведены на фото 4 и 5.

Обучение по образовательной программе происходит не только в Центре, но и продолжается дома.

Так как у большинства школьников дома отсутствуют комплекты роботов, ребятам можно выдавать **задания для теоретической работы**:

1. Проанализировать историю появления первых роботов, их первоначальное назначение.
2. Изучить инструменты LEGO MINDSTORMS EV3.
3. Выяснить функционал блоков действий, выполнения программы, датчиков.
4. Определить основное назначение датчиков и моторов.
5. Найти информацию об использовании роботов для развлечения.

В завершение работы по дополнительной общеобразовательной программе организуется **соревнование между роботами, созданными школьниками**. Задание для соревнования состоит в том, что за минимально короткое время сконструированному и запрограммиро-

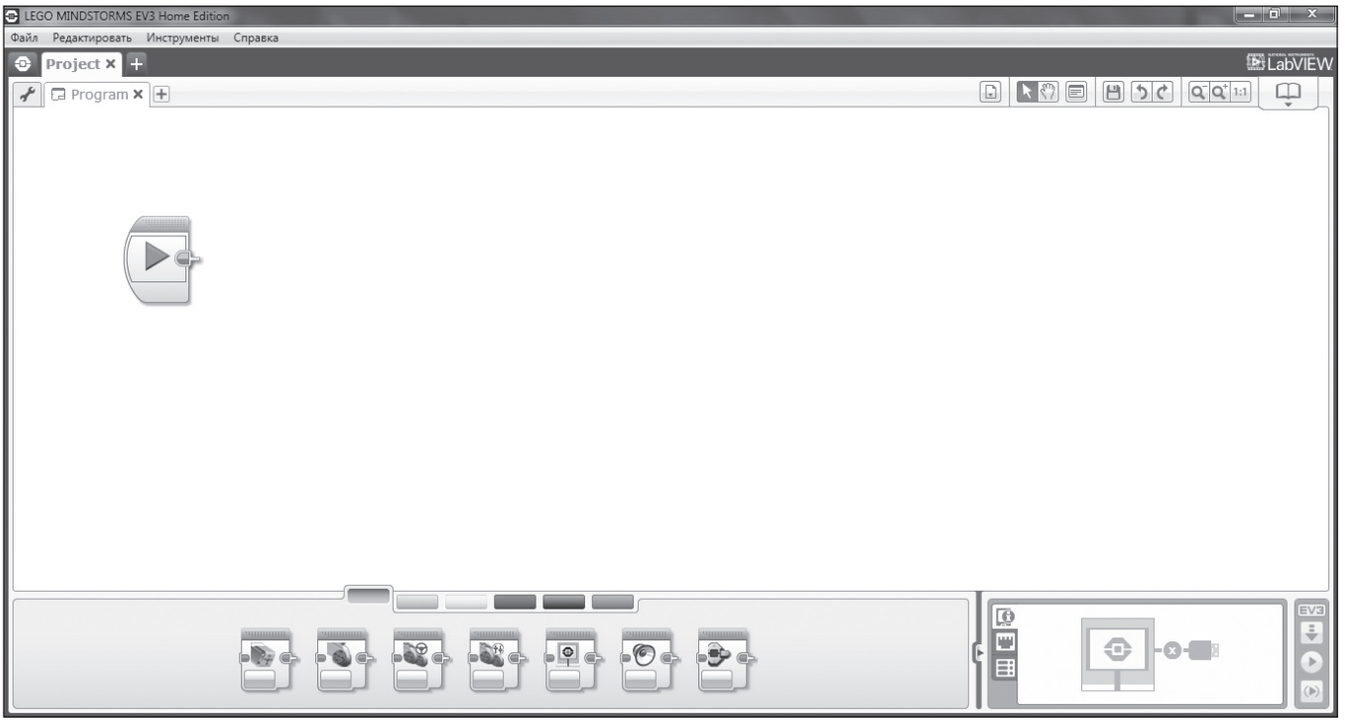


Рис. 3. Рабочая область программы LEGO MINDSTORMS

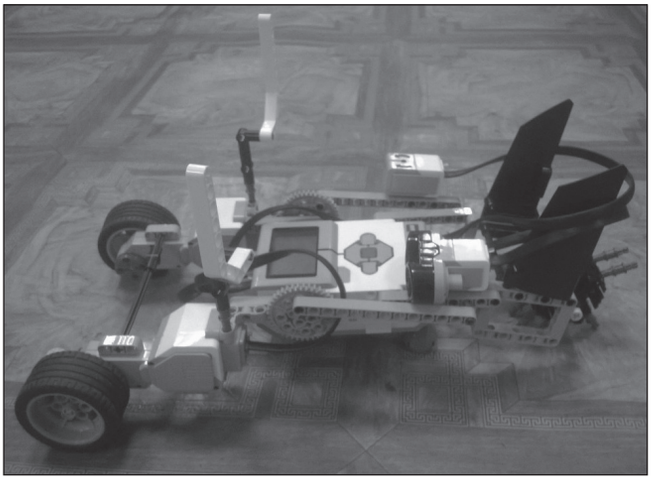


Фото 4. Робот «Драгонобой»

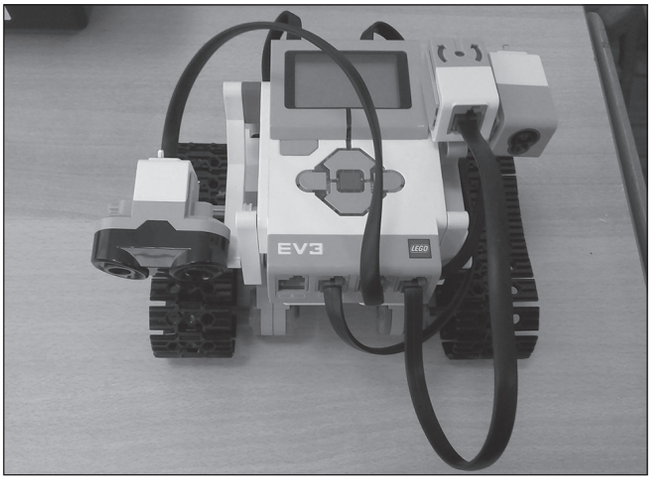


Фото 5. Робот «Роботанк»

ванному роботу необходимо пройти по линии и собрать детали одного конкретного цвета.

Работа с учащимися в Центре молодежного инновационного творчества «МИР-3D» показала, что даже слабые школьники, отстающие по основным учебным предметам (математике, русскому языку, информатике), проявляют интерес к робототехнике, а некоторые из учеников после занятий по рассматриваемой общеобразовательной программе определились со своей будущей профессиональной деятельностью.

Список использованных источников

1. *Иванов А. А.* Основы робототехники. М.: Форум, 2012.
2. *Попов Е. П., Письменный Г. В.* Основы робототехники: Введение в специальность: учебник для вузов по специальности «Робототехнические системы и комплексы». М.: Высшая школа, 1990.

М. А. Плаксин,
Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал);
Пермский государственный национальный исследовательский университет

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ШКОЛЬНОЙ ИНФОРМАТИКЕ. СЕТЕВОЙ ГРАФИК

Аннотация

Описываются понятия «сетевой график», «PERT-диаграмма», «критический путь», «диаграмма Ганта». В качестве примера дается разбор задания из конкурса «ТРИЗформашка-2015».

Ключевые слова: информатика, средняя школа, начальная школа, методика обучения, сетевой график, PERT, PERT-диаграмма, критический путь, диаграмма Ганта, параллельные вычисления, пропедевтика, ТРИЗформатика, ТРИЗформашка.

Контактная информация

Плаксин Михаил Александрович, канд. физ.-мат. наук;

доцент кафедры информационных технологий в бизнесе Пермского филиала Национального исследовательского университета «Высшая школа экономики»; *адрес:* 614070, г. Пермь, ул. Студенческая, д. 38; *телефон:* (342) 205-52-50;

доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета; *адрес:* 614990, г. Пермь, ул. Букирева, д. 15; *телефон:* (342) 239-67-72;

e-mail: mapl@list.ru

M. A. Plaksin,

National Research University Higher School of Economics, Perm Branch;

Perm State National Research University

PROPAEDEUTICS OF PARALLEL COMPUTING IN SCHOOL INFORMATICS. ACTIVITY NETWORK

Abstract

There is the description of the conceptions "activity network" ("project network", "diagram network"), "PERT", "PERT diagram", "critical path", "Gantt diagram" ("Gantt Chart", "Gant diagram", "Gant Chart"). As an example, the investigation of task from the contest "TRIZformashka 2015" is given.

Keywords: informatics, secondary school, primary school, teaching methods, activity network, project network, diagram network, PERT, PERT diagram, critical path, Gantt diagram, Gantt Chart, Gant diagram, Gant Chart, parallel computing, propaedeutics, TRIZformatics, TRIZformashka.

Современный этап развития computer science связан с массовым распространением параллелизма вычислений на всех уровнях (многоядерные процессоры, многомашинные кластеры, многопроцессорные ЭВМ). Это делает актуальным включение пропедевтики параллельного программирования в школьный курс информатики.

В рамках работ над «пермской версией» пропедевтического курса информатики (авторский коллектив: М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова; рабочее название курса «ТРИЗформатика») [7, 8] разработка данной тематики начата в 2013 году. Эффективной площадкой для этого выступил конкурс «ТРИЗформашка» — ежегодный межрегиональный интернет-конкурс по информатике, системному анализу и ТРИЗ (теории решения изобретательских задач) для школьников и студентов [3–6, 9]. В марте 2017 года конкурс состоялся в 17-й раз. Возраст участников — от первого класса школы до четвертого курса вуза. Среднее количество команд — около 100 (рекордное — 202). География конкурса — от Владивостока до Риги. Сайт конкурса: http://www.trizformashka.ru

В процессе подготовки конкурса «ТРИЗформашка» были определены типы задач, направленных на освоение базовых понятий параллельных вычислений, и придуманы по несколько задач этих типов. Одним из таких типов являются задачи на сетевой график.

Вообще говоря, **аппарат сетевых графиков** [10, 11] был создан не для программирования, а для управления проектами. **Причины его включения в курс информатики состоят в следующем:**

- Во-первых, в рамках пропедевтического курса мы занимаемся не узко вопросами программирования, а организацией совместной деятельности любых исполнителей при выполнении любых алгоритмов. В том числе совместной деятельности людей, реализующих проекты.
- Во-вторых, сетевые графики имеют много общего с параллельными формами представления алгоритмов [1, 2, 6]. Более того, в нашей статье [6] для решения задачи на ярусно-параллельную форму алгоритма нам было необходимо одно важное понятие, связанное с сетевым графиком, — критический путь (хотя в той же работе [6] мы на этом внимания не акцентировали).
- В-третьих, сетевые графики — это инструмент планирования. А при организации совместной деятельности как людей, так и устройств планирование является одним из важнейших моментов.

Далее приводятся текст задачи на сетевой график из конкурса «ТРИЗформашка-2015», теоретический материал (описание понятий «сетевой график», «диаграмма Ганта», «критический путь») и разбор задачи конкурса 2015 года.

Что-нибудь получится! (задача из конкурса «ТРИЗформашка-2015»)

Дана схема процесса строительства (рис. 1).

Каждый прямоугольник обозначает какую-то работу, которая должна быть выполнена в ходе строительства. Стрелки определяют порядок выполнения работ (какая работа должна быть выполнена раньше, какая — позже).

Каждую работу выполняет одна бригада (нельзя выполнение одной работы поручить сразу нескольким бригадам).

Для каждой работы указано название и номер (номер служит только для удобства обозначения).

Внизу в квадратных скобках указано количество недель, которое необходимо бригаде для того, чтобы выполнить эту работу.

Требуется:

1. Изобразить (схематично) сооружение, которое будет построено.
2. Определить, сколько недель потребуется для строительства, если работать будет только одна бригада. Ответ обосновать.
3. Определить, за какое минимальное время может быть завершено строительство, если не будет ограничения на рабочую силу. Ответ обосновать.
4. Как повлияет на минимальное время строительства сокращение срока строительства северного бастиона с пяти недель до четырех?
5. Как повлияет на минимальное время строительства увеличение срока строительства западного бастиона с трех недель до четырех?
6. Определить, сколько недель потребуется для строительства, если работать будут восемь бригад. Ответ обосновать.
7. Изобразить (схематично), как будет выглядеть стройка через девять недель после начала строительства, если работать будут пять бригад.
8. Определить, с какой стороны происходит подвоз строительных материалов. Ответ обосновать.
9. Каково минимальное количество бригад, которые способны завершить строительство в минимальное время? Ответ обосновать.

Замечания:

1. Место строительства пересекает река, текущая с северо-запада на юго-восток.
2. Стороны света обозначаются первыми буквами: С — север, Ю — юг, З — запад, В — восток.
3. Бастион имеет форму пятиугольника, у которого четыре наружные стороны представляют собой стены, а пятая — внутренняя — открыта.

Перед продолжением чтения читателю рекомендуется самостоятельно решить предложенную задачу.

Сетевой график, критический путь, PERT, диаграмма Ганта

Сетевой график отображает взаимозависимость работ, составляющих некоторый проект. По способу представления этой зависимости сетевые графики делятся на два вида:

- в одном из них (который и используется в задании конкурса «ТРИЗформашка») работы представляются вершинами графа, а дуги показывают очередность работ;

Рис. 1. Схема строительства

- в другом работы изображаются дугами, а вершины — это события, отмечающие начало/конец той или иной работы (другая трактовка: вершины отображают состояние проекта до и после выполнения соответствующей работы).

Если добавить на сетевой график информацию о продолжительности отдельных работ, можно определить время начала/окончания каждой работы и продолжительность всего проекта.

При оценке продолжительности проекта возникает следующее важное понятие — **критический путь**.

К финишной точке проекта ведут несколько путей. И пройти их надо все (выполнить все лежащие на этих путях работы). Выполнение разных работ требует разного времени. Соответственно, разного времени требует прохождение каждого пути. Имеет смысл выделить самый долгий путь, требующий максимального времени. Дело в том, что указанная нами продолжительность каждой работы — нормативная, планируемая, а на практике всегда возможны задержки по тем или иным причинам. Задержки на «быстрых» путях могут и не повлиять на окончание всего проекта, здесь есть резерв времени. Но задержка на самом долгом пути автоматически означает задержку окончания всего проекта. Такой путь и называется критическим. Он требует особого внимания со стороны руководства проектом. Заметим, что по мере продвижения работ по проекту критический путь может меняться. Например, пусть в начале пути стоит продолжительная работа. Мы сосредоточим на ней много работников и ее выполним. Но сделаем это за счет задержки работ на других путях. Тогда критическим может стать другой путь.

Немного истории. Появление сетевых графиков — побочный результат космической гонки, начатой 4 октября 1957 года с запуска Советским Союзом первого искусственного спутника Земли. В США грянул так называемый спутниковый кризис. Несмотря на индустриальную мощь, преимущества, полученные в результате мировых войн, вывоза в США германских ракетных технологий и ученых, включая ведущих ракетостроителей, оказалось, что Штаты опять (как и в случае с водородной бомбой) отстают от СССР. Кроме того, теперь никто уже не сомневался, что в случае ядерного нападения на Советский Союз, планы которого США разрабатывали начиная с 1945 года, американцам не удастся отсидеться за океаном и они получат ответный удар по своей территории. Среди мер, предпринятых администрацией президента США Дуайта Эйзенхауэра (бывшего главнокомандующего американскими войсками в Европе), были создание в 1958 году Агентства перспективных исследований (ARPA), Национального аэрокосмического агентства (NASA), форсирование работ по созданию межконтинентальной баллистической ракеты «Поларис», предназначенной для базирования на подводных лодках и запуска из-под воды. Кстати, именно ARPA финансировало работы по созданию компьютерных сетей (сети ARPANET, которая стала прародительницей Интернета), протоколов TCP/IP, одной из базовых версий ОС UNIX (так называемой BSD — Berkeley Software Distribution). Реализуемый корпорацией Локхид проект «Поларис» оказался настолько сложен, что для координации работ пришлось создать новую технику управления проектом, получившую название **PERT** — Program/Project Evaluation and Review Technique (методика анализа и оценки программ/проектов). Техника PERT использовала граф вида «работы — дуги, вершины — события». Однако в литературе и во многих компьютерных системах часто любой сетевой график именуют **PERT-диаграммой**.

Сетевой график позволяет отследить взаимозависимость работ, для того он и создан. Но есть еще один важный для управления проектом вопрос, ответить на который по сетевому графику очень сложно. Это вопрос о совпадении различных работ по времени исполнения. Представьте себе, что в вашем проекте задействован один автомобиль и один водитель, а вам надо совершить несколько поездок. Понятно, что, если две поездки будут запланированы на одно и то же время, ваши планы будут нарушены. Для наглядного отображения времени каждой работы используется еще одна технология, предложенная в 1910 году американским инженером Генри Гантом — **диаграмма Ганта**.

Диаграмма Ганта строится следующим образом. Горизонтальная ось представляет собой ось времени. По вертикальной оси располагается список работ (вообще говоря, в произвольном порядке). Напротив каждой работы помещается полоса, начало которой совпадает с моментом начала работы, а конец — с моментом ее окончания. Стрелки соединяют предшествующую работу с последующей. Дополнительно цветом или штриховкой может быть выделен критический путь.

На рисунке 2 изображена диаграмма Ганта для ТРИЗформашкинского задания, построенная в системе GanttProject. Работы упорядочены таким образом, чтобы диаграмма «читалась» (в основном) сверху вниз. К сожалению, в качестве единицы времени GanttProject берет день, хотя по условиям задачи единицей времени является неделя. Это — ограничение системы, преодолеть которое не удалось. Штриховкой выделен критический путь, включающий в себя работы № 19, 18, 3, 4, 2, 7.

На диаграмме Ганта легко определить расположение времени разных работ друг относительно друга, в частности, совпадение времени нескольких работ. Зато зависимость между работами видна плохо: вертикальные стрелки отслеживать трудно, приходится постоянно обращаться к таблице в левой части диаграммы, в которой для каждой работы указаны ее предшественники и продолжительность.

Сетевой график и диаграмма Ганта удачно дополняют друг друга. По этой причине использовать их удобно вместе.

Существует достаточно много компьютерных систем, поддерживающих работу с сетевыми графиками и диаграммами Ганта. Продукт фирмы Microsoft называется MS Project и имеет несколько версий — от персональной («на одного») до корпоративной, для групповой разработки больших проектов. В качестве альтернативы MS Project предлагается несколько свободных систем, распространяемых на условиях лицензии GNU General Public License или схожих с ней, — OpenProj, ProjectLibre и др. Автор данной статьи предпочитает GanttProject.

Наиболее мощная отечественная разработка — система Spider Project. Она ориентирована на крупные проекты и стоит относительно дорого. Однако у системы существует учебная версия, которая обладает полными функциональными возможностями для управления проектами любых типов, но имеет количественные ограничения по размеру проекта: не более 40 работ. Это ограничение можно считать жестким с точки зрения крупных проектов, однако такого количества работ вполне достаточно не только для обучения, но и для практического решения небольших задач.

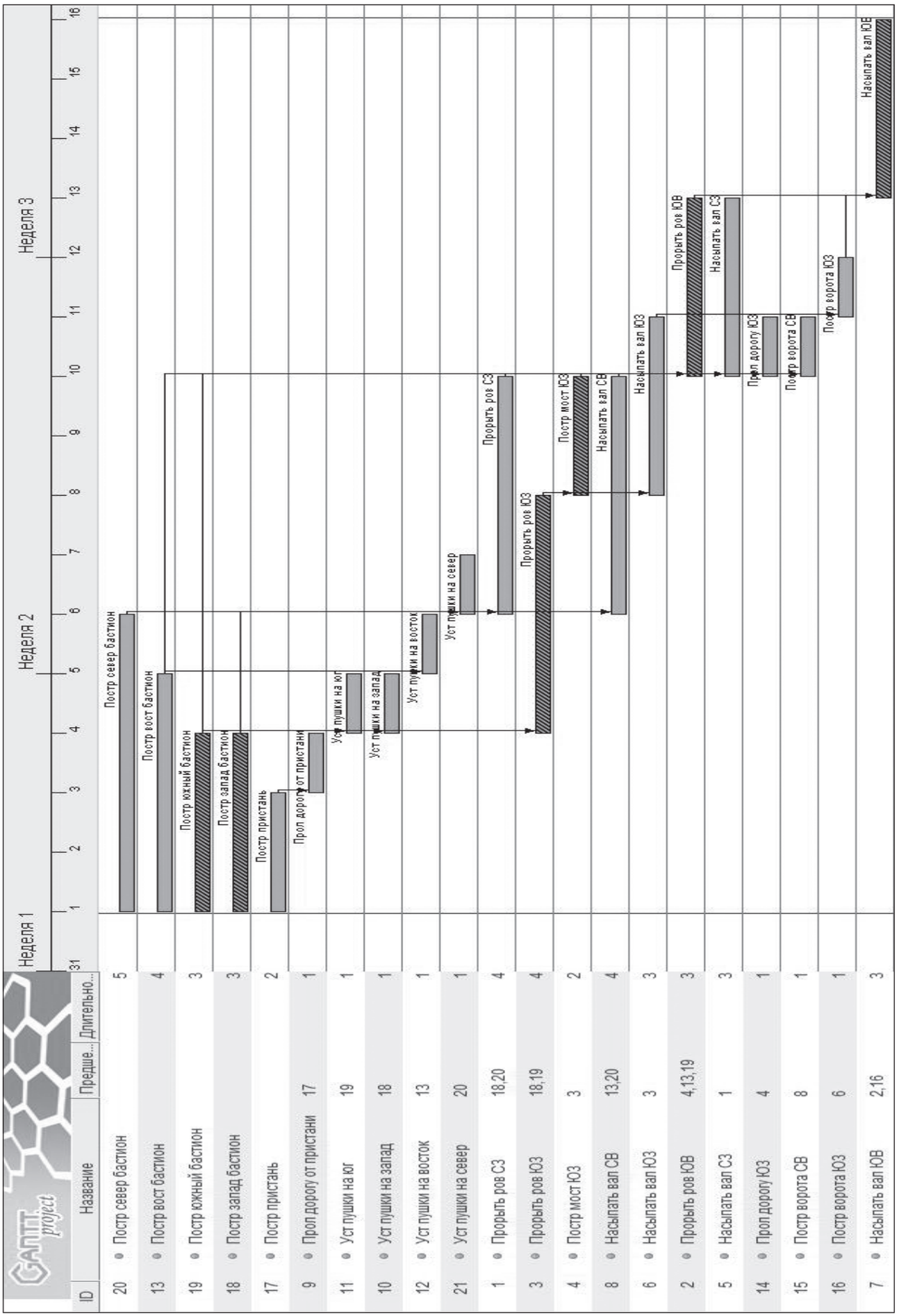


Рис. 2. Диаграмма Ганта

Литература, в которой описаны сетевые графики, PERT-диаграммы, критические пути и пр. весьма обширна. В качестве примера приведем книги [10, 11]. Содержание книги [10] соответствует названию серии, в которой она выпущена, — «для чайников»: материал изложен на предельно простых примерах. В книге [11] изложение гораздо сложнее, строже и обстоятельнее, это книга для профессионалов.

Разбор задачи «Что-нибудь получится!» конкурса «ТРИЗформашка-2015»

1. Изобразить (схематично) сооружение, которое будет построено.

Ответ представлен на рисунке 3.

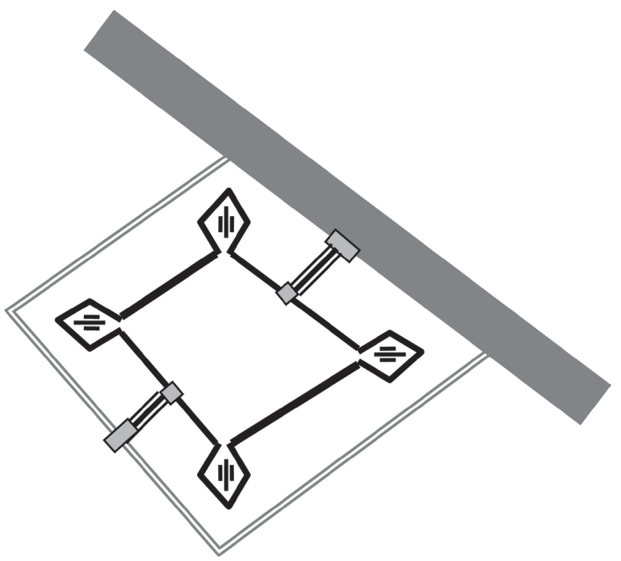


Рис. 3. Сооружение, которое будет построено

2. Определить, сколько недель потребуется для строительства, если работать будет только одна бригада. Ответ обосновать.

Ответ. Потребуется 51 неделя. Нужно сложить продолжительность всех работ.

3. Определить, за какое минимальное время может быть завершено строительство, если не будет ограничения на рабочую силу. Ответ обосновать.

Ответ. Минимальное время — 15 недель. Это длина критического пути:

а) Построить южный бастион (3), Построить западный бастион (3).
б) Прорыть ров ЮЗ (4).
в) Построить мост ЮЗ (2).
г) Прорыть ров ЮВ (3).
д) Насыпать вал ЮВ (3).

4. Как повлияет на минимальное время строительства сокращение срока строительства северного бастиона с пяти недель до четырех?

Ответ. Никак. Строительство северного бастиона не лежит на критическом пути.

5. Как повлияет на минимальное время строительства увеличение срока строительства западного бастиона с трех недель до четырех?

Ответ. Увеличит на неделю. Строительство западного бастиона лежит на критическом пути.

6. Определить, сколько недель потребуется для строительства, если работать будут восемь бригад. Ответ обосновать.

Ответ. Потребуется 15 недель. Это длина критического пути. Максимальная степень распараллеливания — 5. Поэтому работа восьми бригад займет столько же времени, сколько и пяти.

7. Изобразить (схематично), как будет выглядеть стройка через девять недель после начала строительства, если работать будут пять бригад.

Ответ. Построены все четыре бастиона. На всех четырех бастионах установлены орудия.

Построена пристань, проложена дорога к пристани.

Прорыт ров с ЮЗ и ров с СЗ. Построен мост с ЮЗ.

Насыпан вал с СВ.

Насыпаны 2/3 вала с ЮЗ.

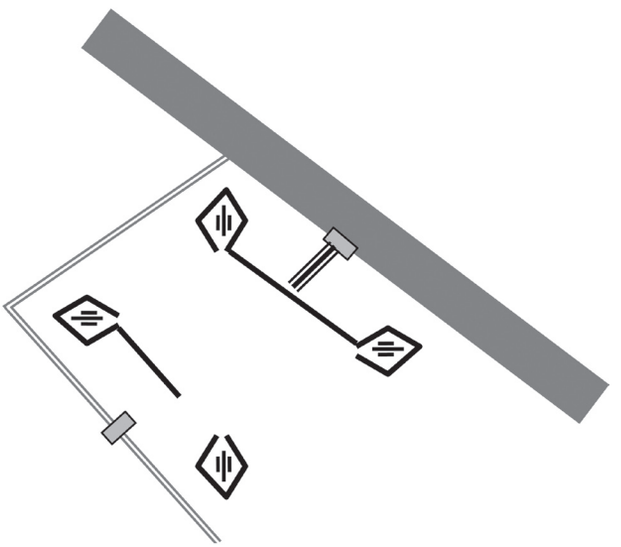


Рис. 4. Состояние стройки через девять недель после начала строительства

8. Определить, с какой стороны происходит подвоз строительных материалов. Ответ обосновать.

Ответ. С юго-востока. Именно эта сторона была в последнюю очередь перегорожена рвом.

9. Каково минимальное количество бригад, которые способны завершить строительство в минимальное время? Ответ обосновать.

Ответ. 4.

Для ответа на этот вопрос посмотрим на сетевой график как на алгоритм строительства крепости и представим этом алгоритм в ярусно-параллельной форме (ЯПФ) [1, 2, 6].

Вообще говоря, можно попробовать обойтись без этого и в качестве ярусно-параллельной формы использовать диаграмму Ганта (рис. 2). Но это довольно неудобно. Во-первых, ЯПФ будет «уложена на бок»: ее высота будет измеряться слева направо, а ширина — сверху вниз. Во-вторых, на ЯПФ все действия одного яруса выполняются одновременно. Поэтому ярусом надо будет считать единицу времени. Это значит, что соседние вершины ЯПФ на диаграмме Ганта будут сливаться в одну линию. Скажем, работа № 20 «Построить северный бастион» занимает 5 единиц времени (5 ярусов). То есть с точки зрения графа ЯПФ эта работа представляет собой 5 вершин,

а на диаграмме Ганта она представлена одной полосой. И, наконец, самое главное: ярусно-параллельная форма должна отображать взаимозависимости между работами, а они на диаграмме Ганта видны плохо.

Для построения ЯПФ воспользуемся алгоритмом из [6]:

- на 1-м ярусе расположим вершины, не имеющие предшествующих;
- на 2-м ярусе — вершины, все предшественники которых расположены на 1-м ярусе;
- на 3-м ярусе — вершины, все предшественники которых расположены на 1-м и 2-м ярусах;
- и т. д.

Каждая вершина попадает на ярус, непосредственно следующий за ярусом, на котором закончено размещение ее предшественников. Расположение вершины на ярусе N означает, что последние предшествующие ей вершины были размещены на ярусе $(N - 1)$.

Построенная по этому алгоритму ярусно-параллельная форма алгоритма строительства крепости будет выглядеть следующим образом (рис. 5).

Напомним, что высота ЯПФ, построенной по описанному выше алгоритму, определяет время выполнения алгоритма, а ширина — степень его распараллеливания (т. е. максимальное количество одновременно работающих исполнителей). В нашем случае высота ЯПФ равна 15, а ширина — 5. То есть 5 бригад смогут закончить строительство за 15 единиц времени.

Вопрос: можем ли мы перестроить параллельную форму таком образом, чтобы уменьшить ее ширину и не увеличить ее высоту? Оценим, возможно ли это.

Распределение вершин по ярусам показано в следующей таблице:

Ширина яруса	Количество ярусов	Номера ярусов	Всего вершин
5	5	1, 2, 3, 4, 10	25
4	3	6, 8, 9	12
3	3	5, 7, 11	9
2	1	12	2
1	3	13, 14, 15	3

Всего у нас работы на 51 неделю. То есть в графе ЯПФ — 51 вершина.

Пятнадцать ярусов шириной 4 вмещают 60 вершин, шириной 3 — 45 вершин, шириной 2 — 30 вершин. Так что графы шириной 3 и 2 нас не устраивают. А граф шириной 4 может устроить.

Для обеспечения ширины 4 нам надо передвинуть с «пятиширинных» ярусов по одной вершине. Место для этого есть. Надо проверить, допускают ли это связи между вершинами ЯПФ. Оказывается, да. Мы можем передвинуть с ярусов 1–3 на более узкие ярусы сцепленную пару работ № 17 «Построить пристань» и № 9 «Проложить дорогу к пристани», с яруса 4 — одну из двух работ № 11 «Установить пушки на юг» или № 10 «Установить пушки на запад», с яруса 10 — одну из двух работ: № 14 «Проложить дорогу ЮЗ» или № 15 «Простроить ворота СВ». Например, сдвинем работы № 10

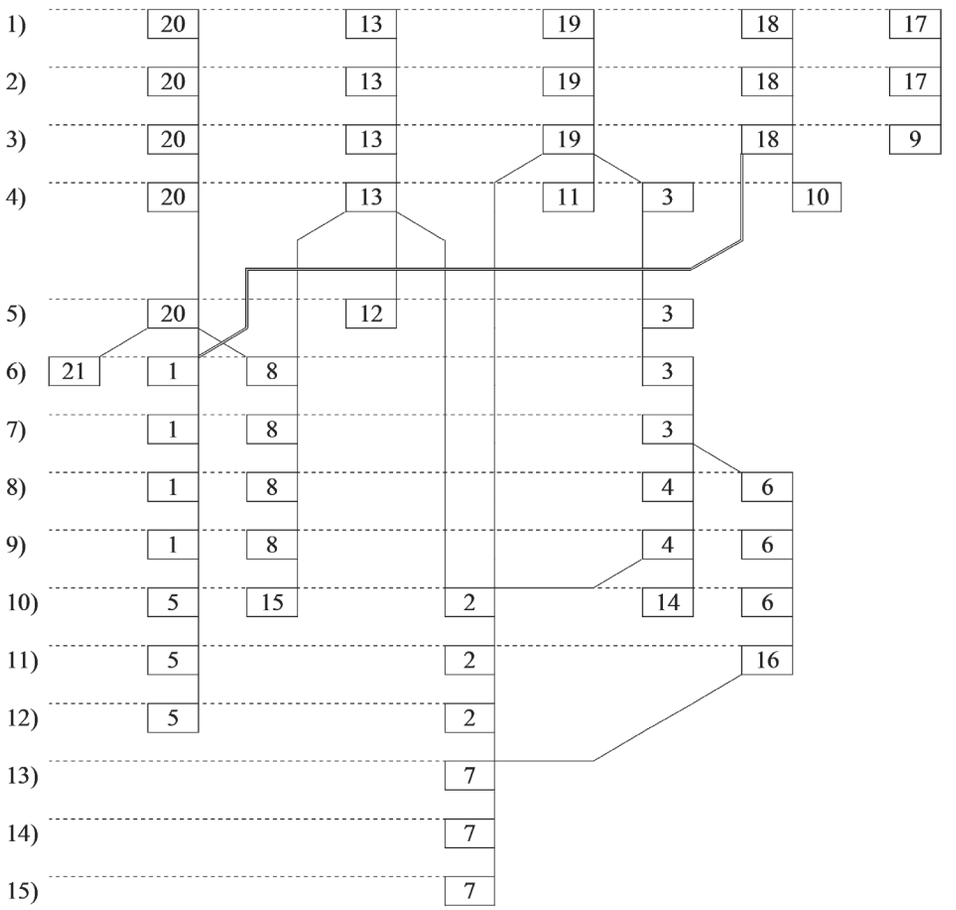


Рис. 5. Ярусно-параллельная форма алгоритма строительства крепости

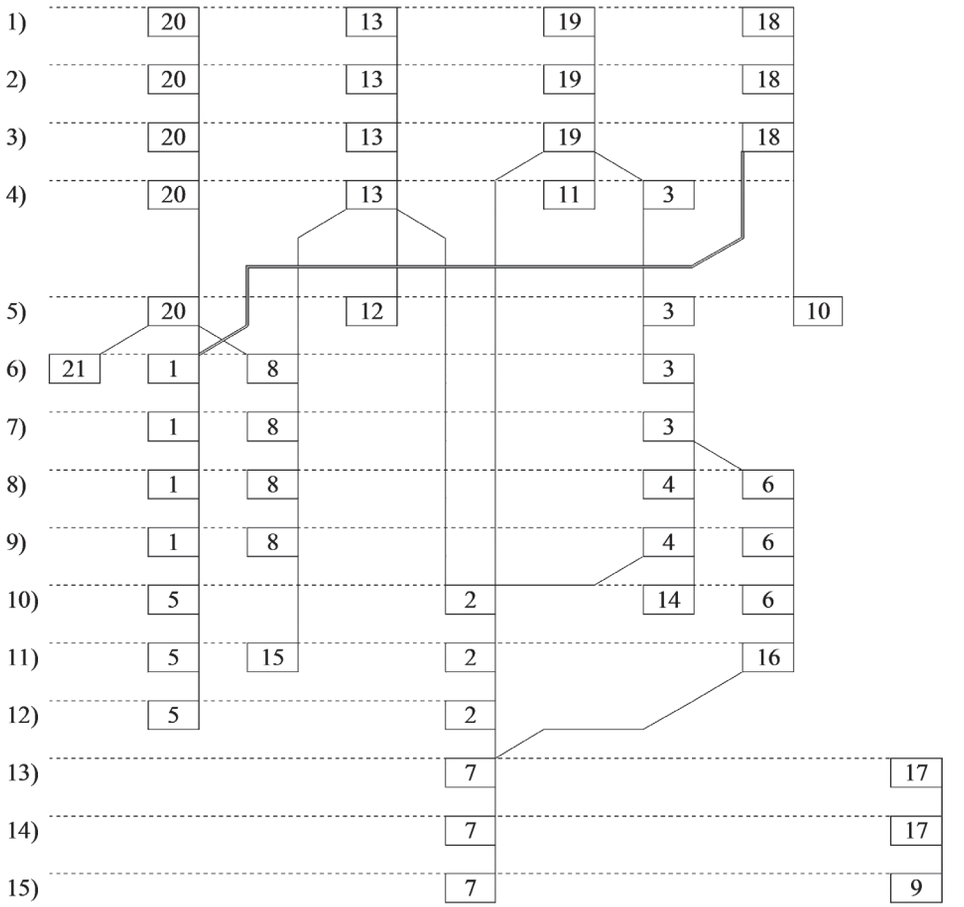


Рис. 6. Ярусно-параллельная форма алгоритма строительства крепости, сокращенная до ширины 4

и № 15 на один ярус (на недели 5 и 11 соответственно), а работы № 17 и № 9 — на последние 3 яруса (на недели 13, 14, 15).

Тогда мы получим ЯПФ шириной 4 при сохранении высоты 15 (рис. 6). То есть строительство за 15 недель могут выполнить 4 бригады.

* * *

Мы разобрали задачу на сетевой график из конкурса «ТРИЗформашка-2015». Аналогичные задачи использовались в конкурсах 2014, 2016 и 2017 годов.

Список использованных источников

1. *Воеводин В. В.* Вычислительная математика и структура алгоритмов: 10 лекций о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности. 2-е изд., стереотип. М.: Изд-во Московского университета, 2010.

2. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002.

3. *Иванова Н. Г., Плаксин М. А., Русакова О. Л.* Задачи на параллельное программирование в конкурсе «ТРИЗформашка-2013» // Информационные технологии в образовании. XXIII Международная конференция-выставка: Сборник трудов. Ч. II. М.: Издательский отдел факультета ВМК МГУ имени М. В. Ломоносова, 2013.

4. *Иванова Н. Г., Плаксин М. А., Русакова О. Л.* Конкурс «ТРИЗформашка» как площадка для апробации заданий на параллельное программирование // Информатика в школе: прошлое, настоящее и будущее: Материалы Всероссийской научно-методической конференции по вопросам применения ИКТ в образовании, 6–7 февраля 2014 г. Пермь: Перм. гос. нац. иссл. ун-т, 2014.

5. *Иванова Н. Г., Плаксин М. А., Русакова О. Л.* ТРИЗформашка // Информатика. 2010. № 5.

6. *Плаксин М. А.* Пропедевтика параллельных вычислений в школьной информатике. Параллельная форма алгоритма // Информатика в школе. 2016. № 10.

7. *Плаксин М. А., Иванова Н. Г., Русакова О. Л.* Информатика: учебник для 3 класса: в 2 ч. М.: БИНОМ. Лаборатория знаний, 2013.

8. *Плаксин М. А., Иванова Н. Г., Русакова О. Л.* Информатика: учебник для 4 класса: в 2 ч. М.: БИНОМ. Лаборатория знаний, 2013.

9. *Плаксин М. А., Иванова Н. Г., Русакова О. Л.* Набор заданий для знакомства с параллельными вычислениями в конкурсе «ТРИЗформашка» // Преподавание информационных технологий в Российской Федерации: Материалы Тринадцатой открытой Всероссийской конференции «ИТ-Образование — 2015» (г. Пермь, 14–15 мая 2015 г.). Пермь: Перм. гос. нац. иссл. ун-т, 2015.

10. *Портни С. И.* Управление проектами «для чайников». М.: Вильямс, 2004.

11. *Троцкий М., Груча М., Огонек К.* Управление проектами. М.: Финансы и статистика, 2006.

ИТОГОВАЯ АТТЕСТАЦИЯ ПО ИНФОРМАТИКЕ

С. М. Кашаев,
Нижегородский государственный лингвистический университет им. Н. А. Добролюбова,

Л. В. Шерстнева, Д. С. Гладских,
Нижегородский государственный технический университет им. Р. А. Алексеева

ВЫПОЛНЕНИЕ СЛОЖНЫХ ЗАДАНИЙ ЕГЭ НА ЯЗЫКЕ PYTHON

Аннотация

Рассматриваются решения сложных задач из наиболее трудной части ЕГЭ (задание под номером 27). Алгоритмы приводятся на очень популярном в настоящее время в школах языке Python. Описанная методика показала свою эффективность при практических занятиях со школьниками.

Ключевые слова: программирование, Python, ЕГЭ по информатике.

Контактная информация

Кашаев Сергей Михайлович, канд. тех. наук, доцент, доцент кафедры математики и информатики Нижегородского государственного лингвистического университета им. Н. А. Добролюбова; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 31а; *телефон:* (831) 416-60-44; *e-mail:* cashaev@rambler.ru

Шерстнева Людмила Владимировна, канд. физ.-мат. наук, зам. директора Института радиоэлектроники и информационных технологий Нижегородского государственного технического университета им. Р. Е. Алексеева; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 24; *телефон:* (831) 436-93-47; *e-mail:* sherlv@rambler.ru

Гладских Дарья Сергеевна, студентка Института радиоэлектроники и информационных технологий Нижегородского государственного технического университета им. Р. Е. Алексеева; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 24; *телефон:* (831) 436-93-47; *e-mail:* inland.holland@gmail.ru

S. M. Kashaev,
Linguistic University of Nizhny Novgorod,

L. V. Sherstneva, D. S. Gladskih,
Nizhny Novgorod State Technical University named after R. E. Alekseev

REALIZING THE MOST DIFFICULT TASKS OF THE USE IN PYTHON

Abstract

Solutions of the tasks from the most difficult part of USE are considered in the article (the question 27). Algorithms are given in the Python language, that is very popular now in schools. The described technique showed the effectiveness at a practical training with school students.

Keywords: programming, Python, Unified State Exam on informatics.

Одно из самых сложных заданий ЕГЭ по информатике — задание 27, в котором от учащихся требуется разработать, как правило, далеко не очевидный алгоритм и представить его на одном из языков программирования. При этом большинство задач не просто заключаются в написании правильной программы, а требуют выполнения ряда ограничений на алгоритм. Обычно эти ограничения касаются производительности алгоритма и объема памяти, отводимой для данных. Сами задачи в задании 27 весьма разнообразны. Мы рассмотрим одну из наиболее трудных категорий таких задач и приведем необходимые пояснения их решения.

Максимальная сумма квадратов

Приведем формулировку задания:

Имеется N отсчетов, каждый из которых является целым положительным числом. Для определенности значение каждого отсчета не более 1000, а N может быть достаточно большим. Необходимо вычислить максимальную сумму квадратов двух различных отсчетов, при этом отсчеты должны отстоять друг от друга не менее чем на три позиции.

В качестве примера рассмотрим ситуацию, когда мы имеем всего пять отсчетов: 5, 9, 12, 2, 3. В этом случае ответ на поставленный вопрос о максимальной сумме — 90 ($81 + 9 = 90$). Это связано с тем, что далеко не все пары отсчетов отвечают условию необходимого интервала в три позиции. Важное условие, касающееся подобных задач ЕГЭ, заключается в том, что алгоритм должен быть максимально эффективным в плане использования памяти и объема вычислений.

Для начала формализуем постановку задачи.

Пусть $a[i]$ — последовательность отсчетов, где индекс i принимает значения от 0 до $N - 1$ включительно. Необходимо найти $\max(a[i]^2 + a[j]^2)$, где разность $i - j$ по модулю не меньше чем 3.

Наиболее простой подход к решению данной задачи заключается в том, чтобы сформировать массив (список на языке Python) a из N элементов. Далее в двойном цикле по переменным i и j выбрать из всех сумм $a[i]^2 + a[j]^2$ максимальную при условии, что разность $i - j$ по модулю должна быть не меньше чем 3.

Такое решение формально правильное, но при оценке на ЕГЭ баллов за него дают немного. Это связано с двумя факторами, упомянутыми в начале статьи.

Рассмотрим наиболее эффективный, на взгляд авторов, подход к решению подобной задачи.

Очевидно, что когда мы имеем только первые три отсчета данных, то не существует пары отсчетов, отстоящих друг от друга не менее чем на три позиции. При появлении четвертого отсчета под это условие попадает только нулевой (самый на-

чальный — с индексом 0). Когда приходит пятый отсчет данных, то для него подходят в плане нахождения максимальной суммы квадратов два начальных элемента. Далее шестой отсчет — и для него подходят уже три начальных, и т. д.

Будем этот набор подходящих отсчетов (в плане интервала в три отсчета) для очередного поступившего называть совокупностью. Фактически для поиска максимальной суммы квадратов нам нужно помнить в этой совокупности только максимум (значение максимального отсчета). Важно отметить, что при появлении каждого следующего отсчета совокупность увеличивается на один отсчет. Так, при появлении 1000-го отсчета нас интересует максимум по первым 997 отсчетам. При появлении 1001-го отсчета в совокупность добавляется 998-й отсчет, и т. д.

С учетом сказанного мы не будем отводить память под все отсчеты последовательности из N элементов. Вполне достаточно отвести память только для трех элементов. И в качестве начального шага создадим список из трех первых элементов последовательности отсчетов: — $a[0]$, $a[1]$, $a[2]$. Обозначим через *maxvr* значение максимального отсчета из совокупности, а через *maxsymkv* — максимальную сумму квадратов отсчетов, отстоящих друг от друга не менее чем на три позиции.

Приведем формальное описание алгоритма обработки отсчетов данных:

- *maxv r* = 0; *maxsymkv* = 0
- цикл по *i* от 0 до 2:
  ввод значений *a*[*i*]
  - цикл по *i* от 3 до *N* – 1:
  - *at* — ввод значения очередного отсчета
  - если *a*[0] > *maxvr*, то
    maxvr = *a*[0]
    (в совокупность добавляется еще один отсчет; если *a*[0] больше максимума, то обновляется максимум)
  - *symkv* = *maxvr*^{2} + *at*^{2} (вычисление суммы квадратов)
  - если *symkv* > *maxsymkv*, то
    maxsymkv = *symkv* (обновление максимальной суммы квадратов)
  - сдвиг элементов в массиве: *a*[0] = *a*[1]; *a*[1] = *a*[2]; *a*[2] = *at*
- печать *maxsymkv*

В качестве примера проиллюстрируем работу данного алгоритма на конкретных данных.

Так, пусть $N = 9$, а входные отсчеты имеют следующие значения: 2, 5, 1, 6, 7, 4, 9, 10, 4. В таблице показаны все вычислительные действия по данному алгоритму.

Таблица

Номера отсчетов	Вычисления
0..2	*a*[0] = 2; *a*[1] = 5; *a*[2] = 1

3	*at* = 6;
	maxvr = 2;
	symkv = $6^2 + 2^2 = 40$;
	maxsymkv = 40;
	a[0] = 5; *a*[1] = 1; *a*[2] = 6

4	*at* = 7;
	maxvr = 5;
	symkv = $7^2 + 5^2 = 74$;
	maxsymkv = 74;
	a[0] = 1; *a*[1] = 6; *a*[2] = 7

5	*at* = 4;
	maxvr = 5;
	symkv = $4^2 + 5^2 = 49$;
	maxsymkv = 74;
	a[0] = 6; *a*[1] = 7; *a*[2] = 4

6	*at* = 9;
	maxvr = 6;
	symkv = $6^2 + 9^2 = 117$;
	maxsymkv = 117;
	a[0] = 7; *a*[1] = 4; *a*[2] = 9

7	*at* = 10;
	maxvr = 7;
	symkv = $7^2 + 10^2 = 149$;
	maxsymkv = 149;
	a[0] = 4; *a*[1] = 9; *a*[2] = 10

8	*a*t = 4;
	maxvr = 7;
	symkv = $4^2 + 7^2 = 65$;
	maxsymkv = 149;
	a[0] = 9; *a*[1] = 10; *a*[2] = 4

В итоге после всех вычислений мы получили 149 — максимальную сумму квадратов двух различных отсчетов, отстоящих друг от друга не менее чем на три позиции. Это действие (100 + 49 = 149) над отсчетами со значениями 7 и 10.

Теперь уже можно запрограммировать данный алгоритм, и на рисунке 1 эта программа представлена на языке Python.

```
n=9                                  # общее число отсчетов
s=3                                  # минимальный интервал между отсчетами
a=[]                                 # определяем a как список (массив)
maxvr=0                              # максимальное число в совокупности
maxsymkv=0                           # максимальная сумма квадратов
for i in range(0,3):
   a.append(int(input()))            # формирование первых  трех отсчетов
for i in range(3,n):
   at=int(input())                   # ввод очередного отсчета
   maxvr=max(a[0],maxvr)             # добавление отсчета в совокупность
                                     # и обновление максимума, если нужно
   symkv=maxvr**2+at**2              # вычисление суммы квадратов
   maxsymkv=max(symkv,maxsymkv)      # обновление максимальной суммы квадратов
   for j in range(0,s-1):            # сдвиг
      a[j]=a[j+1]                    # элементов
   a[s-1]=at
print(maxsymkv)
```

Рис. 1. Программа вычисления максимальной суммы квадратов

Данная программа выполняет правильно вычислительную задачу, а также отвечает необходимым требованиям по вычислительной сложности и объему отводимой памяти. На рисунке 2 показано окно вывода данной программы при подаче на вход той же последовательности отсчетов, которую мы анализировали в таблице.

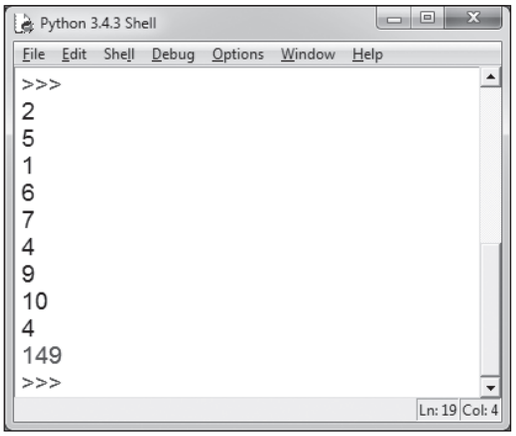


Рис. 2. Выполнение программы вычисления максимальной суммы квадратов

В итоге программно мы получили тот же ответ, что и при непосредственном анализе.

Минимальное четное произведение

Далее мы рассмотрим еще одну (чуть более сложную) задачу с похожим решением:

Считается, что датчик передает каждую секунду по каналу связи неотрицательное целое число, не превосходящее 1000, что представляет собой текущий результат измерений. Необходимо найти в заданной серии показаний датчика минимальное четное произведение двух показаний, между моментами передачи которых прошло не менее пяти секунд. Если получить такое произведение не удается, то программа должна выдать сообщение об этом. Общее количество показаний датчика в серии не превышает 10 000.

Учитывая, что минимальный интервал между отсчетами должен быть не меньше пяти, то до получения пятого отсчета (как и ранее, нумерация отсчетов начинается с нулевого) «претендентов» на минимальное четное произведение нет. Когда же появляется пятый отсчет, то для него подходит только нулевой (только произведение нулевого и пятого может дать минимальное четное произведение с интервалом 5). Если или нулевой, или пятый отсчет, или оба являются четными, то минимальное четное произведение для данной последовательности из шести элементов уже существует. Оно является произведением нулевого и пятого отсчетов. Если же и нулевой, и пятый отсчеты оба нечетные, то минимального четного произведения для данной последовательности из шести элементов нет. Далее поступает шестой отсчет. Он может перемножаться только на нулевой или на первый, чтобы интервал был не менее пяти отсчетов. Следующий, седьмой, отсчет уже может перемножаться на нулевой, первый или второй. Можно сделать общий вывод: если поступает m-й отсчет, то он может перемножаться только на нулевой, первый, ..., $(m-5)$-й. Как и ранее, такой набор из нулевого, первого, ..., $(m-5)$-го отсчетов назовем совокупностью. Всю эту совокупность хранить в памяти нет необходимости. Достаточно из всей совокупности помнить только:

- *ma* — минимальный элемент в совокупности;
- *me* — минимальный четный элемент в совокупности.

Для поиска минимального четного произведения нам потребуется соответствующая переменная, кото-

```
a=[]      # определяем a как список (массив)
s=5       # минимальный
g=1001    # значение большее максимально возможного отсчета
ma=me=g   # за минимумы принимаем число, большее максимально возможного
mp=g**2   # за минимальное произведение принимаем число, большее максимально возможного произведения
N=int(input('Введите N: '))
for i in range(0,s):
   a.append(int(input('Введите очередной отсчет: ')))
for i in range(s,N):
   at=int(input('Введите очередной отсчет: '))
   if a[0]<ma:                  # обновление минимума в совокупности
      ma=a[0]
   if a[0]%2==0 and a[0]<me:    # обновление четного минимума в совокупности
      me=a[0]
   if at%2==0:                  # если очередной отсчет четный вычисляем произведение
      p=at*ma
   elif me<g:                   # если очередной отсчет нечетный и есть четный минимум
      p=at*me
   else:
      p=g**2
   if p<mp:                     # если найдено четное произведение и оно меньше текущего
      mp=p
   for j in range(0,s-1):       # сдвиг элементов
      a[j]=a[j+1]
   a[s-1]=at
if mp==g**2:
  print('Минимального четного произведения нет')
else:
  print(mp)
```

Рис. 3. Программа нахождения минимального четного произведения

рую обозначим *mp*. В начале программы в переменные *ma*, *me* и *mp* внесем значения, превышающие максимально возможные (больше 1000). Как и при анализе предыдущей задачи, приведем формальное описание алгоритма:

- *ma* = 1001; *me* = 1001; $mp = 1001^2$
- цикл по *i* от 0 до 4 включительно:
  ввод значения *a*[*i*]
- цикл по *i* от 5 до *N* – 1 включительно:
- *at* — ввод значения очередного отсчета
- если *a*[0] четное и *a*[0] < *me*,
  то *me* = *a*[0] (обновление четного минимума)
- если *a*[0] < *ma*,
  то *ma* = *a*[0] (обновление минимума)
  (в совокупность добавляется еще один отсчет — *a*[0])
- если *at* четное и *ma* · *at* < *mp*,
  то *mp* = *ma* · *at*
- если *at* нечетное, *me* < 1001 и *me* · *at* < *mp*,
  то *mp* = *me* · *at*
- цикл по *j* от 0 до 3
  a[*j*] = *a*[*j* + 1]
  a[4] = *at*
- печать *mp*

На рисунке 3 представлен текст программы на Python, написанной в соответствии с данным алгоритмом, а на рисунке 4 показано окно вывода программы. По значениям входных отсчетов и полученному минимальному четному произведению видно, что ответ правильный.

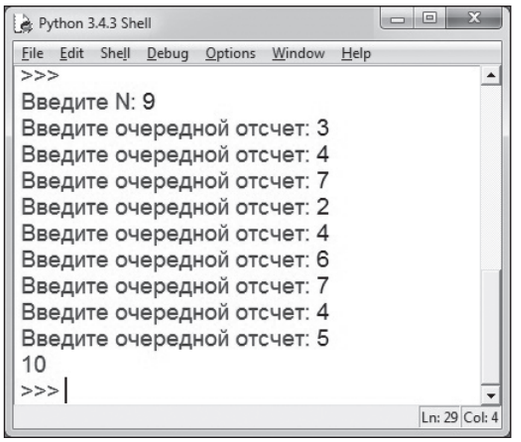


Рис. 4. Выполнение программы нахождения минимального четного произведения

Таким образом, мы разобрали наиболее трудную категорию заданий из ЕГЭ. Язык Python имеет очень ясную структуру, и даже сложные алгоритмы на нем выглядят вполне понятно и не требуют использования блок-схем. Описанные методики показали свою эффективность при практических занятиях со школьниками.

Список использованных источников

1. *Кашаев С. М., Шерстнева Л. В.* Паскаль для школьников. Подготовка к ЕГЭ по информатике. 3-е изд. СПб.: БХВ-Петербург, 2014.

2. *Кашаев С. М., Шерстнева Л. В.* Самостоятельная подготовка к ЕГЭ по информатике. СПб.: БХВ-Петербург, 2009.

НОВОСТИ

В «Росэлектронике» создали оптический диск из веществ растительного происхождения

В холдинге «Росэлектроника» разработали оптический диск, состоящий в том числе из веществ растительного происхождения, сообщает пресс-служба «Ростеха». Многослойные 3D-оптические носители включают в себя фоточувствительные слои, куда записываются данные, и полимерные, служащие для распределения света считывающего лазера. Количество слоев колеблется от 40 до 60. Как пояснили в «Росэлектронике», для формирования функциональных фоточувствительных слоев использованы вещества класса хромонов — фенольные соединения, образующиеся в растениях. В устройстве, разработанном специалистами ЦНИТИ «Техномаш», входящего в «Росэлектронику», запись производится за счет способности хромонов менять свои свойства под воздействием света определенной силы — они начинают светиться. Такой диск может содержать намного большее количество записывающих слоев. Новый носитель может записывать данные объемом до 1 Тбайт, а скорость чтения достигает 12 Гбит/с.

Умные скамейки заряжают гаджеты от солнца и собирают данные для города

Стартап Changing Environments выпускает городские скамейки с зарядными устройствами для гаджетов, работающими на солнечной энергии, и цифровые информационные указатели, которые отображают время прибытия общественного транспорта, сведения о погоде, информацию о событиях и не только. На каждой скамейке — по два порта USB, а указатели энергию тоже получают от солнечных батарей. Каждая скамейка оснащается датчиками, которые собирают сведения о пешеходном движении и передают по беспроводным сетям. Эту информацию муниципальные власти могут использовать для принятия решений о градостроительстве, проведении мероприятий и т. д. Собирается также информация о частоте использования зарядных устройств на скамейках, регистрируются электромагнитные сигналы мобильных устройств проходящих мимо пешеходов.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

ЗАДАЧИ

Т. В. Федотова,
Вяземский железнодорожный техникум, Смоленская область

ИСПОЛЬЗОВАНИЕ ФОРМУЛ МАССИВА В MICROSOFT EXCEL

Аннотация

В статье рассматриваются примеры использования формул массива в Microsoft Excel вместо обычных формул, что позволяет упростить вычисления и в то же время сделать их более эффективными, сократить количество применяемых формул.

Ключевые слова: Excel, массив, константа массива, Мастер функций.

Контактная информация

Федотова Татьяна Вячеславовна, преподаватель Вяземского железнодорожного техникума, Смоленская область; *адрес:* 215111, Смоленская область, г. Вязьма, ул. Плотникова, д. 11; *телефон:* (48131) 3-53-82; *e-mail:* mail@vzt67.ru

T. V. Fedotova,
Vyazma Railway Technical School, Smolensk Region

USING ARRAY FORMULAS IN THE MICROSOFT EXCEL

Abstract

The article discusses examples of using array formulas in Microsoft Excel instead of the or-dinary formulas. This makes possible to simplify calculations and at the same time make them more efficient, reduce the number of using formulas.

Keywords: Excel, array, array constant, Function Wizard.

Представленный материал можно использовать на уроках информатики в девятом классе, а также в средних профессиональных образовательных учреждениях при изучении тем раздела «Обработка числовой информации».

Задачи можно распечатать и предложить учащимся для самостоятельного изучения, так как они содержат подробное описание алгоритма действий для достижения результата.

Формула массива обрабатывает несколько наборов значений, называемых аргументами массива. Она может возвращать как отдельное значение, так и множество значений. Формулы массива часто называют «формулами CSE», поскольку для их ввода используется комбинация клавиш Ctrl+Shift+Enter, а не простое нажатие клавиши Enter.

Формулы массива позволяют выполнять сложные задачи, например следующие:

- подсчет числа знаков в диапазоне ячеек;
- суммирование только тех чисел, которые отвечают определенным условиям, например, наименьших значений в диапазоне чисел, определенном верхней и нижней границами;
- суммирование всех *n*-х значений в диапазоне значений.

Задача 1. Использование формулы массива для вычисления нескольких значений.

Предположим, что для каждого из трех работников нам известна почасовая оплата и количество отработанных часов:

№ п/п	Почасовая оплата	Отработано часов
1	148,12	75
2	153,50	72
3	150,71	85

Найдем, сколько получит за свою работу каждый из работников.

1. Активируем ячейку C4, затем выделим диапазон C4:C6 и введем знак =.
2. Выделим диапазон A4:A6 и введем знак *.
3. Выделим диапазон B4:B6 (рис. 1).

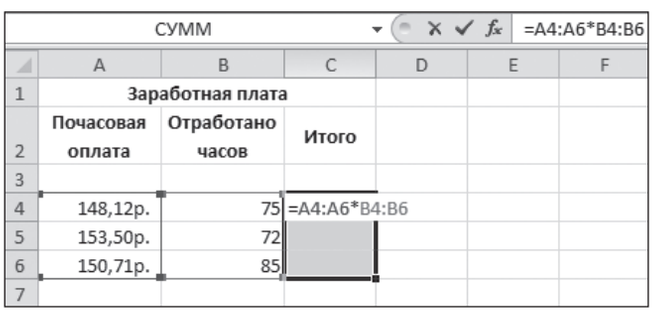


Рис. 1

4. Нажмем Ctrl+Shift+Enter и применим финансовый формат: *Стили ячеек*, *Числовой формат*, *Финансовый* (рис. 2).

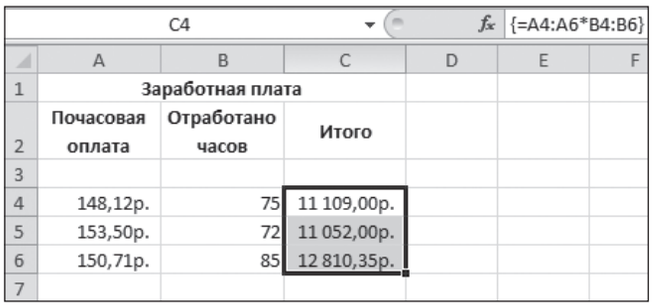


Рис. 2

Задача 2. Использование констант массива.

Константы массива являются компонентами формул массива. Они создаются путем ввода списка элементов с последующей вставкой до и после него фигурных скобок **{ }**. Так как константы массива — это составляющие формул массива, необходимо вручную вставить скобки вокруг них, а затем нажать клавиши Ctrl+Shift+Enter, чтобы ввести формулу целиком.

Для решения предыдущей задачи используем константу массива {75:72:85} (рис. 3).

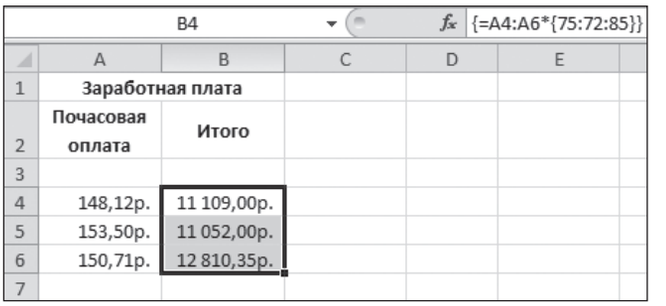


Рис. 3

Задача 3. Использование формулы массива для вычисления одного значения.

Пусть теперь нам надо найти общую сумму денег, которую бухгалтерия должна заплатить трем работникам (рис. 4).

1. Выделим ячейку B7, в ней будет размещаться сумма.
2. Введем с клавиатуры: =СУММ(
3. Выделим диапазон A4:A6 и введем знак *.
4. Выделим диапазон B4:B6 и закроем скобку функции СУММ. Для ввода функции СУММ можно использовать кнопку Σ Автосумма либо Мастер функций.
5. Нажмем Ctrl+Shift+Enter и применим финансовый формат (рис. 5–7).

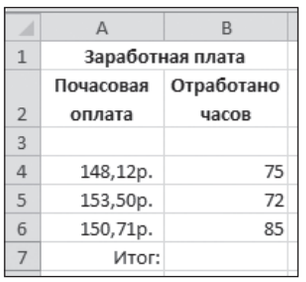


Рис. 4

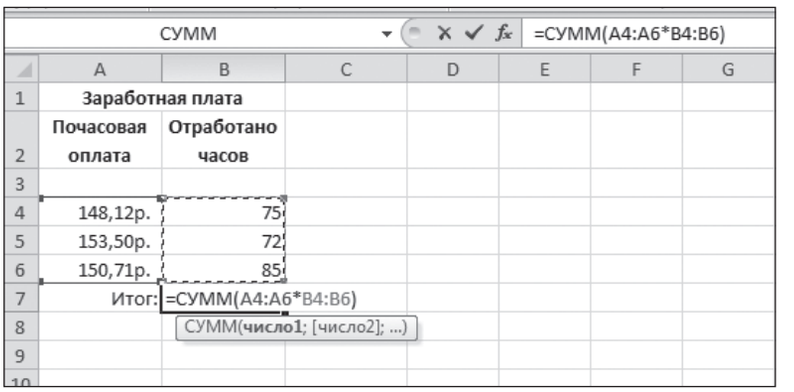


Рис. 5

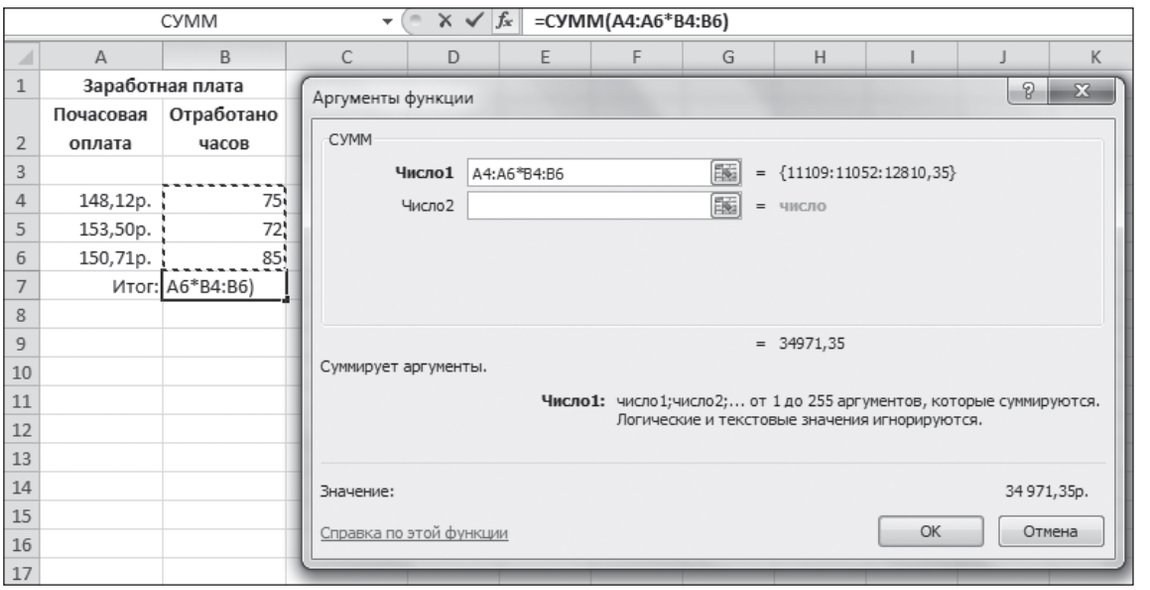


Рис. 6

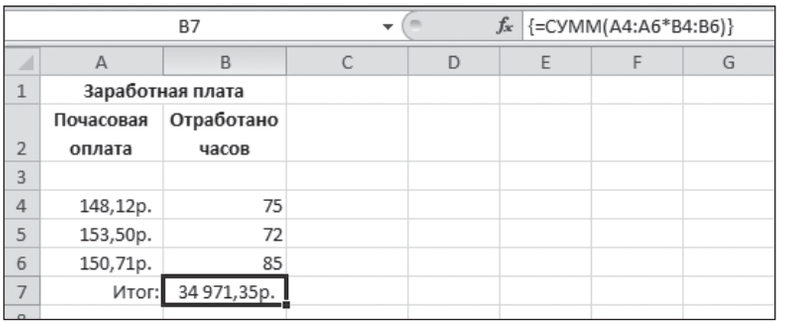


Рис. 7

Задача 4. Создание таблицы значений функции $y = x^2$ (рис. 8).

1. Выделим диапазон C6:M6.
2. С помощью Мастера функций перейдем в описание аргументов функции СТЕПЕНЬ.
3. В качестве аргумента *Число* укажем диапазон C5:M5.
4. В качестве аргумента *Степень* введем 2 (рис. 9).
5. Нажмем Ctrl+Shift+Enter (рис. 10).

Для заполнения таблицы можно использовать и формулу: {=C5:M5^2}

Задача 5. Составление таблицы «Результаты вступительных испытаний».

Дана таблица результатов вступительных испытаний (рис. 11).

Если общее количество баллов больше 40, то в соответствующей ячейке поля *Результат* вывести слово «поступил», иначе — «не поступил».

1. Выделим диапазон F4:F6.
2. Используя Мастер функций, перейдем в описание аргументов функции ЕСЛИ.

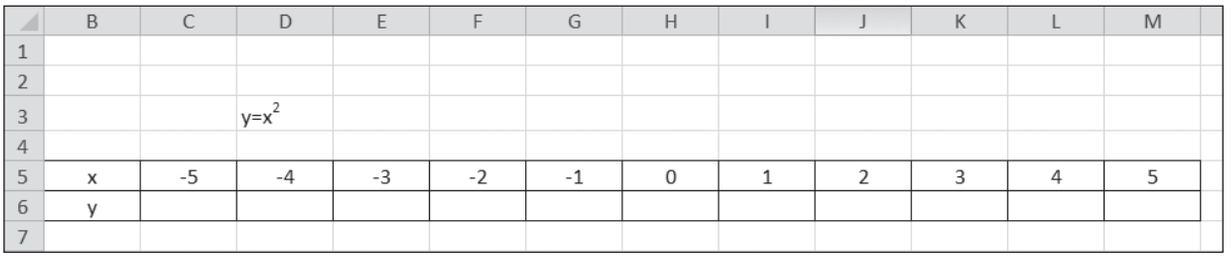


Рис. 8

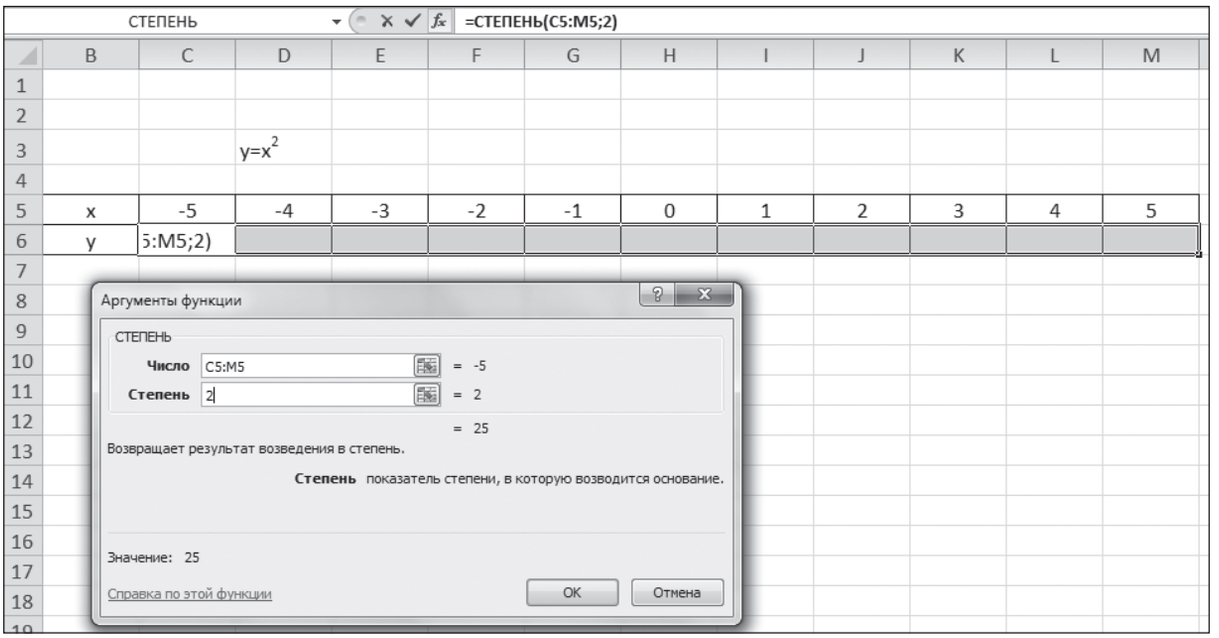
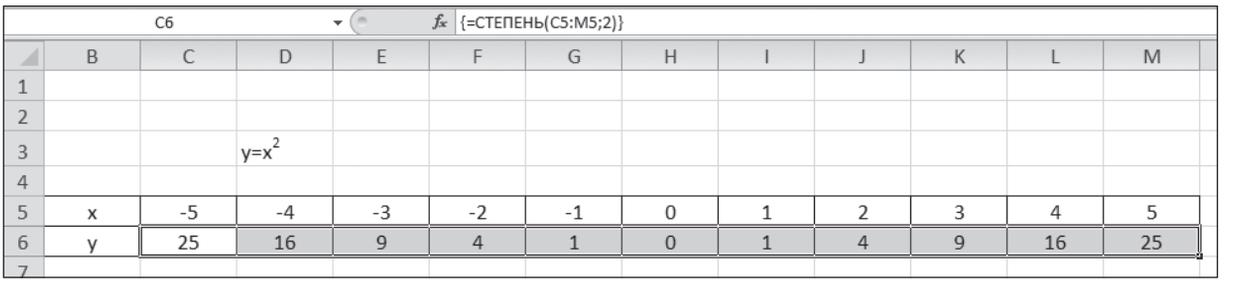


Рис. 9

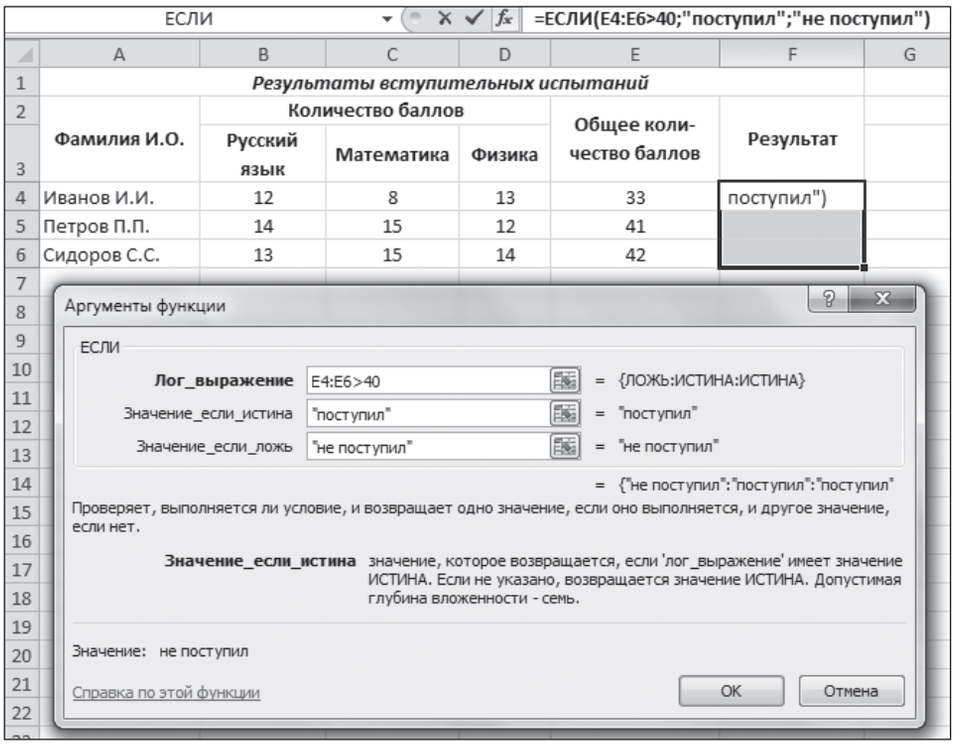


	B	C	D	E	F	G	H	I	J	K	L	M
1												
2												
3			y=x²									
4												
5	x	-5	-4	-3	-2	-1	0	1	2	3	4	5
6	y	25	16	9	4	1	0	1	4	9	16	25
7												

Рис. 10

	A	B	C	D	E	F
1	***Результаты вступительных испытаний***					
2	**Фамилия И.О.**	**Количество баллов**			**Общее количество баллов**	**Результат**
3		**Русский язык**	**Математика**	**Физика**		
4	Иванов И.И.	12	8	13	33	
5	Петров П.П.	14	15	12	41	
6	Сидоров С.С.	13	15	14	42	

Рис. 11



	A	B	C	D	E	F	G
1	***Результаты вступительных испытаний***						
2	**Фамилия И.О.**	**Количество баллов**			**Общее количество баллов**	**Результат**	
3		**Русский язык**	**Математика**	**Физика**			
4	Иванов И.И.	12	8	13	33	поступил")	
5	Петров П.П.	14	15	12	41		
6	Сидоров С.С.	13	15	14	42		

Рис. 12

3. Введем аргументы функции ЕСЛИ, как указано на рисунке 12.
4. Нажмем Ctrl+Shift+Enter (рис. 13).

Задача 6. Вычисление заработной платы работника.

Дана таблица с заработной платой работников (рис. 14).

Найти заработную плату за три месяца И. И. Иванова.

1. Активируем пустую ячейку, например E2.
2. С помощью Мастера функций перейдем в описание аргументов функции СУММ (рис. 15).
3. Щелкнем на стрелке вниз в поле *Фамилия И.О.* и выберем функцию ЕСЛИ.
4. Заполним окно *Аргументы функции ЕСЛИ* значениями (рис. 16).
5. Нажмем Ctrl+Shift+Enter (рис. 17).

F4 | {=ЕСЛИ(E4:E6>40;"поступил";"не поступил")}

	A	B	C	D	E	F	G
1	*Результаты вступительных испытаний*						
2	Фамилия И.О.	Количество баллов			Общее количество баллов	Результат	
3		Русский язык	Математика	Физика			
4	Иванов И.И.	12	8	13	33	не поступил	
5	Петров П.П.	14	15	12	41	поступил	
6	Сидоров С.С.	13	15	14	42	поступил	
7							

Рис. 13

	A	B	C
1	Дата	Фамилия И.О.	Заработная плата
2	05.янв	Иванов И.И.	13 587,57р.
3	05.янв	Петров П.П.	15 130,78р.
4	05.янв	Сидоров С.С.	14 125,12р.
5	05.фев	Иванов И.И.	12 578,14р.
6	05.фев	Петров П.П.	13 000,45р.
7	05.фев	Сидоров С.С.	10 897,10р.
8	05.мар	Иванов И.И.	15 787,00р.
9	05.мар	Петров П.П.	12 777,51р.
10	05.мар	Сидоров С.С.	15 655,00р.

Рис. 14

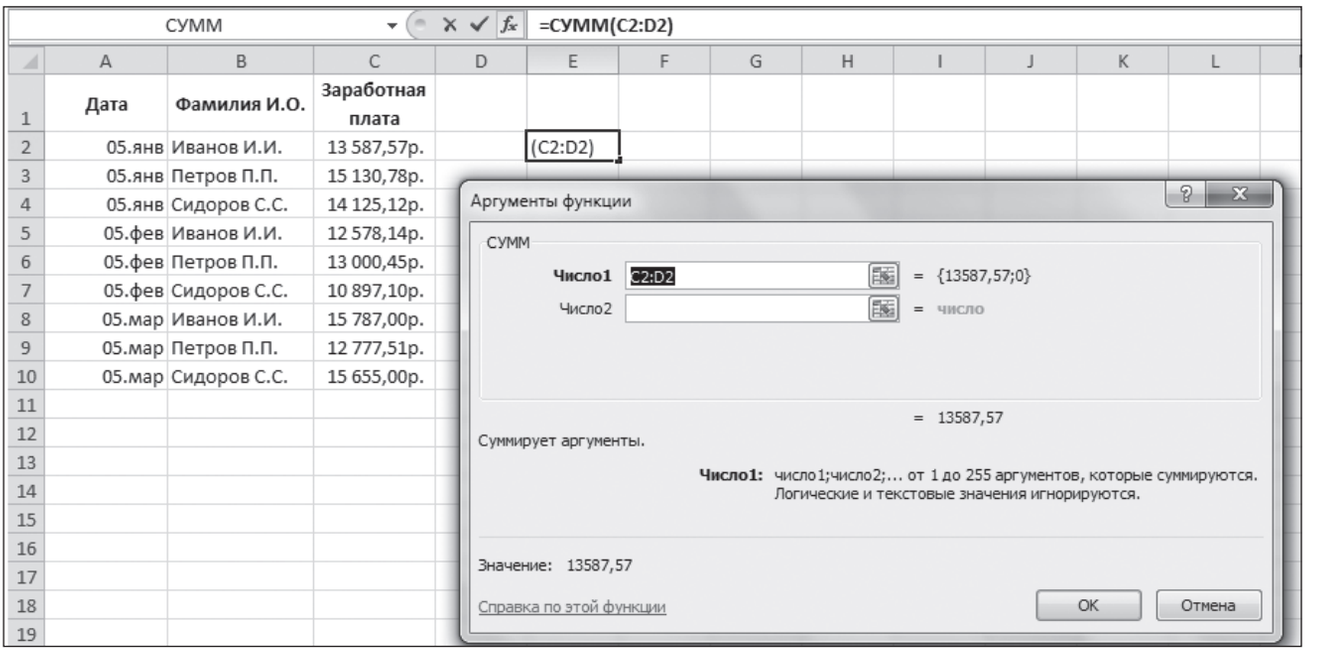
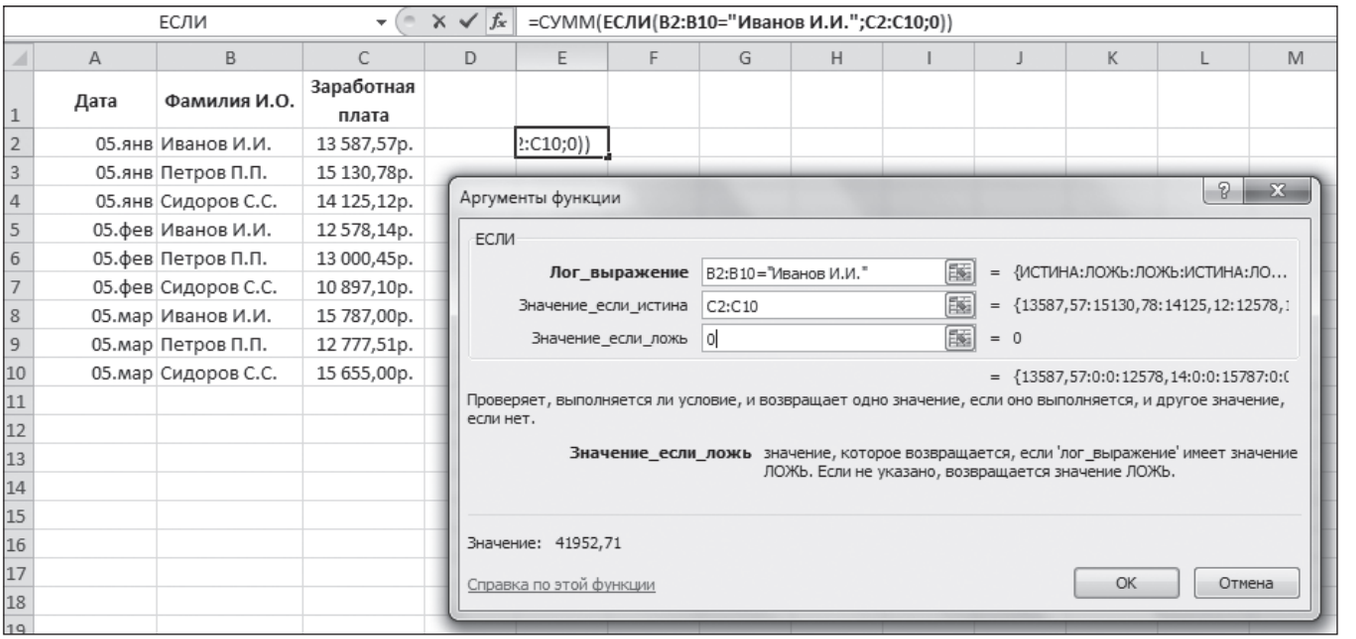
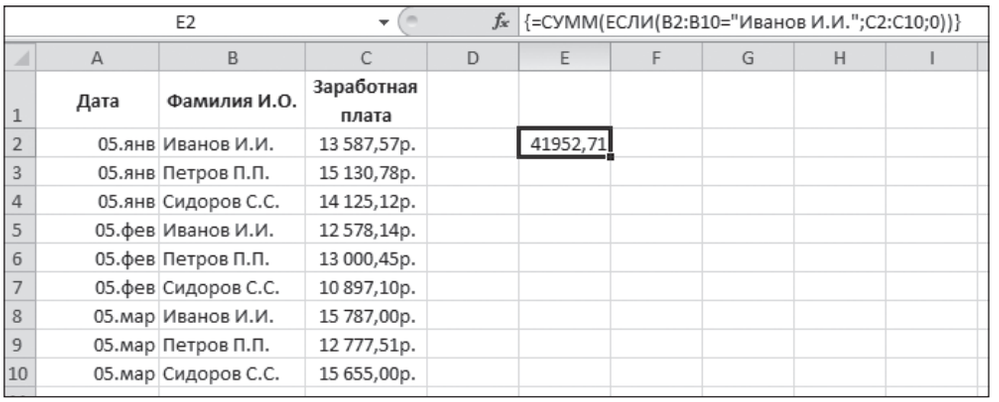


Рис. 15



	A	B	C
1	Дата	Фамилия И.О.	Заработная плата
2	05.янв	Иванов И.И.	13 587,57р.
3	05.янв	Петров П.П.	15 130,78р.
4	05.янв	Сидоров С.С.	14 125,12р.
5	05.фев	Иванов И.И.	12 578,14р.
6	05.фев	Петров П.П.	13 000,45р.
7	05.фев	Сидоров С.С.	10 897,10р.
8	05.мар	Иванов И.И.	15 787,00р.
9	05.мар	Петров П.П.	12 777,51р.
10	05.мар	Сидоров С.С.	15 655,00р.

Рис. 16



	A	B	C	D	E	F	G	H	I
1	Дата	Фамилия И.О.	Заработная плата						
2	05.янв	Иванов И.И.	13 587,57р.		41952,71				
3	05.янв	Петров П.П.	15 130,78р.						
4	05.янв	Сидоров С.С.	14 125,12р.						
5	05.фев	Иванов И.И.	12 578,14р.						
6	05.фев	Петров П.П.	13 000,45р.						
7	05.фев	Сидоров С.С.	10 897,10р.						
8	05.мар	Иванов И.И.	15 787,00р.						
9	05.мар	Петров П.П.	12 777,51р.						
10	05.мар	Сидоров С.С.	15 655,00р.						

Рис. 17

Список использованных источников

1. Методические рекомендации по выполнению контрольных заданий // Мегаобучалка. http://megaobuchalka.ru/1/9649.html

2. Справка по Excel // Служба поддержки Office. https://support.office.com/ru-ru/excel

3. *Уокенбах Дж.* Microsoft Excel 2010. Библия пользователя / пер. с англ. М.: Вильямс, 2011.

НОВОСТИ

В Китае заработал первый в мире спутник квантовой связи

Китай ввел в эксплуатацию первый в мире спутник квантовой связи «Мо-цзы» (Micius). Его разработка началась в 2011 году, а запущен «Мо-цзы» был в августе прошлого года. В середине января спутник завершил тесты на орбите и был введен в эксплуатацию. Программа тестирования на орбите включала проверку работоспособности спутниковой платформы, тест под нагрузкой и проверку разных каналов связи спутника с наземными станциями. По данным Китайской академии наук, к 2030 году Китай может создать глобальную сеть квантовой связи.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

Д. А. Слинкин,
Шадринский государственный педагогический университет, Курганская область

ТЕХНОЛОГИЧЕСКИЕ АСПЕКТЫ ПОДГОТОВКИ ШКОЛЬНИКОВ-ДЕБЮТАНТОВ К ОЛИМПИАДАМ ПО ИНФОРМАТИКЕ

Аннотация

В статье проанализированы особенности организации и проведения школьных олимпиад по информатике, затрудняющие процесс участия дебютантов, предложены подходы к решению возникающих проблем, рассмотрена технология подготовки школьников к участию в олимпиадах по информатике.

Ключевые слова: олимпиада по информатике, технология подготовки школьников-дебютантов к олимпиаде по информатике, программирование.

Контактная информация

Слинкин Дмитрий Анатольевич, канд. пед. наук, доцент кафедры программирования и автоматизации бизнес-процессов Шадринского государственного педагогического университета, Курганская область; *адрес:* 641800, Курганская область, г. Шадринск, ул. К. Либкнехта, д. 3; *телефон:* (35253) 3-63-69; *e-mail:* xdsl@list.ru

D. A. Slinkin,
Shadrinsk State Pedagogical University, Kurgan Region

TECHNOLOGICAL ASPECTS OF TRAINING PUPILS DEBUTANTS TO THE OLYMPIADS IN INFORMATICS

Abstract

The article discusses the features of the organization and conduct of school Olympiads in informatics, complicating the process of participation of debutants, and suggests ways of overcoming problems. It is also considered technology training schoolchildren to participate in the Olympiad in informatics.

Keywords: Olympiad in informatics, technology of training pupils debutants to Olympiad in informatics, programming.

1. Проблемные аспекты организации и проведения олимпиад по информатике

Результаты школьных олимпиад по информатике являются на сегодняшний день одним из эффективных критериев оценки как интеллектуального уровня обучаемых и степени их подготовки в области алгоритмизации и программирования, так и качества работы учителей информатики и математики.

В письме № 08-1755 Министерства образования и науки РФ [10], направленном в адрес органов государственной власти субъектов РФ, осуществляющих государственное управление в сфере образования, указано местоположение в сети Интернет методических рекомендаций для проведения школьного [6], муниципального [5] и заключительного туров олимпиады по информатике [15]. Анализ данных документов, аналогичных материалов прошлых лет, а также многолетний опыт проведения олимпиад различных уровней [8, 9, 12– 14], в том числе муниципальных этапов всероссийской олимпиады школьников по информатике, позволяет нам выделить основные проблемные аспекты их организации и проведения, которые мешают школьникам-дебютантам раскрыть свой потенциал в полном объеме:

1. Программирование в режиме дефицита времени.
2. Жесткая формализация ввода-вывода.
3. Автоматизация проверки решений.
4. Ограничения на общение с членами жюри.
5. Запрет или минимизация использования справочных материалов.

Проанализируем каждый аспект более подробно.

1.1. Программирование в режиме дефицита времени

Ограничение времени, отводимого на решение задач повышенной сложности, является общей проблемой для любых олимпиад. Дефицит времени на олимпиадах по информатике часто порождает стиль программирования, неприменимый в работе инженера-программиста [7]. Реальные инженерные проекты требуют внимательности, аккуратности, скрупулезности, понятного и хорошо откомментированного кода, умения работать в группе. «Штурмовщина» в решении таких задач противопоказана. Если говорить простыми словами, на олимпиаде нужны спринтеры, а в ИТ-компаниях — стайеры. Поэтому систематический подход к программированию, практикуемый при изучении информатики в школе, слабо помогает в решении олимпиадных задач. Школьник, прекрасно и глубоко знающий математику и алгоритмизацию, но теряющийся в стрессовой ситуации, находится на олимпиаде в заведомо проигрышной позиции по сравнению с самоуверенным подростком, обладающим широким кругозором и опытом участия в подобных мероприятиях.

1.2. Жесткая формализация ввода-вывода

Правила четко фиксируют механизмы и формат ввода-вывода. Такой подход является актуальным и востребованным для инженера-программиста [3], но в то же время противоречит всему опыту школьника по разработке алгоритмов на уроках информатики. Любое отклонение от указанного в условии формата считается ошибкой, например, обычный комментарий или подсказка при вводе данных гарантирует дисквалификацию решения, даже при условии представления верного алгоритма. Данная проблема актуальна, прежде всего, при использовании автоматизированной проверки решений.

1.3. Автоматизация проверки решений

Автоматизация проверки предполагает использование эталонных тестов, которые пропускаются сквозь программу участника олимпиады. Таким образом, оптимальность используемого алгоритма и его возможная новизна никак не проверяются. В результате интересные и эффективные решения олимпиадных задач не оцениваются. Кроме того, усугубляется вышеупомянутая проблема формализации вывода данных, что в ситуации первого знакомства школьника с соответствующей системой автоматизированной проверки фактически приводит к значительным потерям времени, так как правила работы с такими системами обычно нетривиальны, а используемая терминология не всегда интуитивно понятна.

1.4. Ограничения на общение с членами жюри

Правила предусматривают ограничения, которые касаются вопросов взаимодействия участников олимпиады с членами жюри. Например, в положении об организации олимпиад зафиксировано, что во время всего тура каждый участник должен иметь возможность задать вопросы членам жюри по условиям задач и получить на них ответы. При этом вопросы следует задавать в письменной форме на специальных бланках или в электронном виде, если это предусмотрено системой проведения соревнований. Обучаемый имеет право представить вопросы, сформулированные с возможностью выбора членами жюри одного из двух видов ответов «да/нет» или «без комментариев». Подросток, ранее не встречавшийся с подобным подходом, либо потратит много времени на поиск формулировок для своего вопроса, либо решит вообще его не задавать, отдавшись на волю случая.

1.5. Запрет или минимизация использования справочных материалов

Правила олимпиады категорически запрещают школьнику использовать любые справочные материалы (это касается личных записей, справочников по математике, таких как справочник И. Н. Бронштейна [2], по языкам и системам программирования). Таким образом, талантливые школьники, умеющие на основе разрозненных исходных данных создавать новые алгоритмы, будут в проигрышном положении по сравнению с подростками, обладающими хорошей памятью, знакомыми с решениями аналогичных задач, но не способными к самостоятельной исследовательской деятельности. Кроме того, указанные правила вступают в конфликт с правилами проведения удаленных (заочных) олимпиад, которые обычно не включают запрет на использование справочной литературы по причине его очевидной бессмысленности в отсутствии контроля.

Анализ вышеизложенных проблемных аспектов, касающихся организации и проведения олимпиад по информатике, а также опыт подготовки школьников-дебютантов к участию в подобных мероприятиях позволили выделить **два возможных подхода к решению указанных проблем:**

1. Совершенствование правил олимпиад с целью обеспечения равных условий для всех участников (как школьников-дебютантов, так и опытных олимпиадников).
2. Разработка технологии подготовки школьников-дебютантов к участию в олимпиаде по информатике с опорой на действующие положения и правила.

2. Совершенствование правил олимпиад

Изменение правил олимпиад является нетривиальной задачей в административном плане, так как сопровождается множеством организационных проблем и для получения максимального положительного эффекта должно применяться одновременно на всех уровнях. Например, снижение строгости в требованиях взаимодействия участников с членами жюри или обязательный анализ эффективности кода на одном из этапов олимпиады позволит дебютантам на равных бороться с опытными участниками и даже занимать призовые места. Однако на следующих этапах, при неизменных правилах, команда талантливых новичков, скорее всего, потерпит поражение, столкнувшись с более строгими и формализованными требованиями. Дополнительно, изменение большинства правил будет сопровождаться все возрастающей нагрузкой на жюри, увеличением роли человеческого фактора при судействе, что может в итоге понизить объективность в оценке решений. На наш взгляд, единственным требованием, которое может быть смягчено без негативных последствий, является запрет на использование справочных материалов. Для сближения правил очных и заочных олимпиад, уменьшения стрессовости процесса, сохранения на прежнем уровне нагрузки на судейство следует предоставить участникам олимпиады возможность использования справочных печатных изданий по математике и языкам программирования из заранее утвержденного списка. Для формирования привычной рабочей среды следует также загрузить электронные копии указанных изданий на рабочие компьютеры участников.

Как было сказано выше, указанное решение приведет к положительным результатам только при его одновременном внедрении на всех уровнях олимпиад, от школьных до всероссийских и международных. Таким образом, предполагается большой объем работы — от формирования универсального списка справочных изданий до решения административно-организационных задач по одновременному изменению правил олимпиад. Решение о начале работ в данном направлении с учетом необходимости влияния на правила международных соревнований может быть принято только на уровне Министерства образования и науки РФ.

3. Разработка технологии подготовки к олимпиаде по информатике школьников-дебютантов

Данный подход характеризуется возможностью его реализации на уровне школы. Подготовку новичков следует проводить исключительно на базе действующих правил организации и проведения всероссийской олим-

пиады школьников по информатике. До начала разработки и анализа алгоритмов решения различных типов задач учитель должен готовить школьников к рефлекторному соблюдению правил, формировать наиболее эффективную стратегию участия в олимпиаде. Основываясь на собственном опыте судейства школьных олимпиад, мы предлагаем планировать первые занятия по олимпиадному программированию следующим образом.

3.1. Выбор языка и среды программирования

Правила Всероссийских олимпиад для школьников декларируют обязательную поддержку языков C++ и Pascal с возможностью использования для обоих языков как свободных (GCC и FreePascal), так и коммерческих (Microsoft Visual C++ и Delphi) сред и компиляторов. Дополнительными языками являются C#, Visual Basic, Python и Java.

Задачей учителя является выбор одной среды программирования (компилятора), полностью (вплоть до номеров версий) соответствующей олимпиадной и наиболее близкой к применяемой на уроках информатики. Например, распространенный в школах PascalABC.NET (http://www.pascalabc.net/) не поддерживается на олимпиадах (хотя и может быть введен предметно-методической комиссией по информатике самостоятельно), поэтому в качестве компилятора/среды должны быть выбраны FreePascal (http://freepascal.org/) и Lazarus (http://www.lazarus-ide.org/). Причины выбора свободного решения в противовес коммерческому очевидны, одна из самых важных — возможность самостоятельных занятий школьника на домашнем компьютере без необходимости приобретать дорогостоящее программное обеспечение.

3.2. Стандартизация источников и приемников данных

Правила определяют источники и приемники данных. Это могут быть:

1) стандартные потоки ввода-вывода;
2) один входной текстовый файл и один выходной текстовый файл, имена которых жестко определены в условии каждой задачи (*input.txt* для ввода и *output.txt* для вывода; реже — другие имена, например, *input.in*, *output.out*).

Учитель при подготовке школьников должен рассмотреть оба варианта, особое внимание уделив файловой обработке. Следует напомнить школьникам о возможности сокрытия расширений в файловых менеджерах различных операционных систем и о вытекающих отсюда проблемах (например, файл *task1.txt* в проводнике ОС Windows по умолчанию будет отображаться как *task1*, а файл, визуально представленный как *task2.txt*, на самом деле имеет имя *task2.txt.txt*). Для решения указанной проблемы, а также для ускорения и упрощения работы с файлами рекомендуется вместо стандартных средств, предлагаемых операционной системой, использовать двухпанельные файловые менеджеры, такие как Far (http://www.farmanager.com/), Double Commander (http://doublecmd.sourceforge.net/), Total Commander (https://www.ghisler.com/), Midnight Commander (https://www.midnight-commander.org/) и др.

На классических олимпиадах входные-выходные данные — исключительно текстовые, поэтому стоит заострить внимание на обработке текстовых файлов. Желательно для выбранного языка разработать шаблон кода, который следует в дальнейшем повсеместно использовать. В шаблоне следует выделить имена обрабатываемых файлов с тем, чтобы участник олимпиады впоследствии менял их на имена, специфичные для каждой конкретной задачи, при этом имена файлов должны указываться без путей, относительных или абсолютных, т. е. варианты вида: *c:\ivanov\problem1\input.txt* или *..\output.txt* совершенно недопустимы. Пример шаблона для языка программирования FreePascal показан в листинге 1.

3.3. Определение форматов данных

Особое внимание ученика следует обратить на требования, предъявляемые к форматам данных, и указать на недопустимость отклонений от них. Вывод любых комментариев неприемлем в решениях олимпиадных задач, количество и вид разделителей должны полностью соответствовать заявленным в условии задачи требованиям. То же самое можно сказать о порядке и форме выводимых данных.

Приведем примеры, конкретизирующие приведенные выше доводы.

Пример 1. Школьник перед вводом данных по привычке вывел фразу: «Введите число: ». Решение такой задачи не будет засчитано при использовании автоматизированной системы проверки, так как указанная фраза будет направлена в стандартный поток вывода, что гарантированно нарушит формат вывода данных.

Пример 2. В условии задачи сказано, что результат должен быть представлен с точностью до двух знаков после десятичной точки. Однако участник олимпиады рассчитал и вывел результат с точностью в шесть знаков.

olymp_template.pas

```
{$mode objfpc}
const   finName='input.txt'; // имя исходного файла
        foutName='output.txt'; // имя результирующего файла
var fin,fout:text;
begin
 assign(fin,finName);
 reset(fin);
 assign(fout,foutName);
 rewrite(fout);
 // Здесь код программы.
 // Чтение данных производится из файла fin,
 // запись данных - в файл fout
 close(fin);
 close(fout);
end.
```

Листинг 1. Шаблон кода для файлового ввода-вывода на FreePascal

Таким образом, вместо эталонного ответа 2.96 было получено значение 2.958543. При проверке автоматизированной системой фактически верный ответ может быть не засчитан.

3.4. Учет ограничений на программу

При организации автоматизированной проверки в большинстве олимпиадных задач декларируют ограничения на:

1) время работы программы;
2) объем памяти, задействованный программой во время исполнения;
3) объем исходного кода (решения задач на *школьной* олимпиаде крайне редко попадают под два последних ограничения).

При ручной или полуавтоматической проверке обычно оставляют только ограничение на время работы программы. Штатным ограничением считается одна секунда.

Учитель должен подготовить школьников к тому, что любое решение задачи, выполняемое более одной секунды, скорее всего, не будет зачтено.

Также рекомендуется продемонстрировать школьникам простейший инструментарий для профилирования программы, чтобы участник олимпиады мог самостоятельно определять время ее работы. Такой инструментарий обычно встроен в операционную систему, язык или среду программирования. Например, в ОС Windows при наличии оболочки powershell можно воспользоваться командой Measure-Command; в ОС Linux — командой time; в программе на FreePascal — функцией MillisecondsBetween из модуля dateutils и т. д. Даже в условиях запрета на использование дополнительных материалов простой инструментарий замера времени может быть быстро изготовлен, как показано в листинге 2. Достаточно подключить такой модуль к программе, чтобы по ее завершении получить общее время работы. При этом вывод генерируется в стандартный поток ошибок, поэтому, даже если участник олимпиады забудет отключить модуль при сдаче решения, программа не нарушит формата вывода данных.

3.5. Получение опыта работы с автоматизированной системой проверки решений

На сегодняшний день существует множество программных продуктов, ориентированных на осуществление автоматизации проверки решений олимпиадных задач.

Распространенным вариантом является использование одного из специализированных веб-сервисов в сети Интернет. Например, в методических рекомендациях по проведению муниципального этапа олимпиады в 2016/2017 учебном году [5, с. 39] предлагается использовать систему Яндекс.Контест (https://contest.yandex.ru/). Также широко известна система Timus Online Judge (http://acm.timus.ru/), обладающая крупнейшим в России архивом задач и поддерживаемая студентами и выпускниками Уральского федерального университета [11].

Учитель в процессе подготовки школьников к олимпиаде по информатике может зарегистрировать команду и принять участие в одном из соревнований. Самостоятельное формирование заданий в подобных веб-сервисах обычно предусматривается только по согласованию с компанией-производителем, поэтому более гибким вариантом является использование автономных систем, которые могут быть загружены и запущены на сервере в локальной сети школы.

Среди автономных систем следует обратить внимание на ejudge (https://ejudge.ru/). Указанная система может быть бесплатно получена в исходных кодах, скомпилирована для конкретного оборудования, загружена в виде заранее подготовленного образа виртуальной машины. Кроме того, данная система присутствует в составе многих дистрибутивов ОС Linux, в том числе во входящих в Единый реестр российских программ для электронных вычислительных машин и баз данных (https://reestr.minsvyaz.ru/). Наиболее универсальным является вариант использования виртуальной машины, которая может быть развернута фактически на любом компьютере.

В целом внедрение автономной системы является более трудоемкой задачей, чем использование существующих веб-сервисов. Таким образом, если задачей является только ознакомление школьников с особенностями работы автоматизированной системы проверки решений, учителю следует выбирать веб-сервис, но если предполагается использовать систему проверки для проведения школьных олимпиад на постоянной основе, следует выбирать автономный вариант.

3.6. Участие в пробном туре

Правила декларируют возможность (в некоторых случаях — необходимость) проведения пробного тура, в течение которого участник может ознакомиться с программным обеспечением, установленным на рабочем месте, и с автоматизированной системой проверки решений, если она используется.

mtime.pp

```
1  {$mode objfpc}
2  unit mtime;
3  interface
4  implementation
5  uses sysutils,dateutils;
6  var dt:TDateTime;
7  initialization
8   dt:=now();
9  finalization
10  writeln(stderr);
11  writeln(stderr,'Elapsed time: ',MillisecondsBetween(dt,now()));
12 end.
```

Листинг 2. Модуль mtime для измерения времени работы программы на FreePascal

Задача педагога — научить школьников активно использовать возможности пробного тура, а именно:

1) проверить наличие и работоспособность компилятора/среды: подготовить простейшую программу с вводом-выводом, скомпилировать, запустить;
2) проверить достаточность прав для создания каталогов и файлов средствами операционной системы и файлов — программными средствами: определить с помощью члена жюри или памятки участника олимпиады местоположение в файловой системе, отведенное организаторами для хранения решений задач, создать там каталог, в нем — подкаталоги, в одном из последних — файл программы, где следует создать и заполнить текстовый файл, затем прочитать и вывести его содержимое;
3) разработать решение задачи пробного тура, проверить доступ к автоматизированной системе проверки решений (если она используется), протестировать свое решение с ее помощью;
4) обратиться к членам жюри в случае возникновения ошибок на любом из этапов с просьбой исправить проблему.

Таким образом участник олимпиады гарантирует себе работоспособность рабочего места к началу олимпиады.

3.7. Подготовка апелляций

Правила гарантируют школьникам возможность подачи апелляции по окончании олимпиады. Однако опыт показывает, что школьники-дебютанты практически не пользуются данной возможностью, если их к этому не обязывает учитель, и часто даже не знают о ней. Вместо этого участники олимпиады замыкаются в себе либо делятся «обидами» со сверстниками в социальных сетях, блогах и форумах. Однако причинами снижения количества баллов могут являться не только объективные ошибки в решениях задач, но и недоработки в программном обеспечении для автоматизированной проверки, в проверочных тестах, в технологии «ручной» проверки решений и т. д. Таким образом, своевременная и обоснованная подача апелляции позволит школьнику получить заслуженную оценку своего труда.

Учитель должен готовить школьников как к участию в разборе решений (вне зависимости от количества полученных баллов), так и к подаче апелляции на неверную оценку решений.

Важным фактом является то, что подаче апелляции должен предшествовать разбор решений задач, проводимый жюри олимпиады. В результате школьник сможет понять, насколько его алгоритм отличается от эталонного, и предметно обосновать свои претензии.

3.8. Формирование общей стратегии

На олимпиадах по программированию очень многое зависит от выбранной школьником-дебютантом стратегии. Учитель, осуществляя подготовку учеников к олимпиаде, должен сформировать у них следующие стратегические императивы:

1) ознакомление с условиями *всех* задач до начала решения;
2) равномерное распределение времени на решение задачи с резервированием времени на сдачу решений: если по истечении запланированного времени основа алгоритма (математическая модель) не разработана, следует осуществить переход к решению другой задачи; если задача решена до запланированного срока, необходимо произвести повторное перераспределение времени;
3) решение в первую очередь задач, схожих с решенными ранее (во время предварительной подготовки, самостоятельно, на предыдущих олимпиадах и т. д.), вне зависимости от количества начисляемых баллов;
4) разработка математической модели алгоритма *до* начала написания программы (как показывает опыт проведения олимпиад, последующие затраты времени на создание и отладку программы при корректной математической модели не превышают 30 минут);
5) для смягчения проблемы дефицита времени:
   - отказ от сложных и многословных языковых концепций (объектно-ориентированного программирования, динамических структур данных и т. п.);
   - отказ от разработки собственных функций, если этого не требует выбранный алгоритм;
   - отказ от оптимизаций алгоритма, если он укладывается в заявленные ограничения по времени, используемой памяти и объему кода;
   - использование только хорошо известных конструкций языка и программных библиотек во избежание трудноуловимых ошибок и увеличения времени на отладку программы.

4. Заключение

Анализируя правила проведения школьных олимпиад по информатике и основываясь на многолетнем опыте судейства, мы приходим к выводу, что подготовка талантливых школьников-дебютантов должна, прежде всего, включать в себя формализацию подходов к участию в олимпиадах. Это позволит без изменения правил олимпиад нивелировать их недостатки, сократив разрыв между дебютантами и опытными участниками соревнований. Изложить и закрепить указанный материал можно в течение трех–пяти занятий, посвятив все последующие изучению и разработке алгоритмов решения олимпиадных задач.

Список использованных источников

1. *Алексеев Е. Р., Чеснокова О. В., Кучер Т. В.* Free Pascal и Lazarus: учебник по программированию. М.: ALT Linux; ДМК-пресс, 2010. http://www.altlinux.org/Books:FreePascal

2. *Бронштейн И. Н., Семендяев К. А.* Справочник по математике для инженеров и учащихся втузов: 13-е изд., испр. М.: Наука, Гл. ред. физ.-мат. лит., 1986.

3. *Крайванова В. А., Крючкова Е. Н.* Олимпиадное программирование как эффективный инструмент подготовки профессиональных программистов // Вестник Новосибирского государственного университета. Серия «Информационные технологии». 2012. Т. 10. № 4.

4. *Мансуров К. Т.* Основы программирования в среде Lazarus. http://www.freepascal.ru/download/book/lazarus_osnovy/osnovy_programmirovanija_v_srede_lazarus.pdf

5. Методические рекомендации по разработке заданий и требований к проведению муниципального этапа всероссийской олимпиады школьников по информатике в 2016/2017 учебном году // ROSOLYMP.RU: Информационный портал всероссийской олимпиады школьников. http://www.rosolymp.ru/

attachments/10646_5 %20Informatics%20Recommendations%20ME_2016-2017.pdf

6. Методические рекомендации по разработке заданий и требований к проведению школьного этапа всероссийской олимпиады школьников по информатике в 2016/2017 учебном году // ROSOLYMP.RU: Информационный портал всероссийской олимпиады школьников. http://www.rosolymp.ru/attachments/10645_5 %20Informatics%20Recommendations%20SHE_2016-2017.pdf

7. Олимпиады по программированию — помогают ли они в реальной работе? // TPROGER.RU: Типичный программист. https://tproger.ru/experts/15/

8. *Пирогов В. Ю.* Олимпиады по программированию в ШГПИ. Избранные задачи и решения // ПРОф (издание ШГПИ). 2010. № 1.

9. *Пирогов В. Ю.* «Шахматные» олимпиадные задачи по программированию // Информатика в школе. 2011. № 5.

10. Письмо Министерства образования и науки № 08-1755 от 26.08.2016 года «О методических рекомендациях для школьного и муниципального этапов всероссийской олимпиады школьников» // ROSOLYMP.RU: Информационный портал Всероссийской олимпиады школьников. http://www.rosolymp.ru/upload/Pismo_%E2 %84 %9608-1755

11. *Самсонов А. В., Ипатов А. В.* Система подготовки олимпиадных задач по программированию в УрГУ // Вестник Новосибирского государственного университета. Серия «Информационные технологии». 2012. Т. 10. № 3.

12. *Слинкин Д. А.* Анализ решения сюжетной олимпиадной задачи по программированию // Информатика и образование. 2010. № 1.

13. *Слинкин Д. А.* «Блондинка против технарей», или Разбор и решение олимпиадной задачи по программированию // Информатика и образование. 2011. № 8.

14. *Слинкин Д. А.* В поисках пиратского клада // ПРОф (издание ШГПИ). 2010. № 1.

15. Требования к проведению заключительного этапа всероссийской олимпиады школьников по информатике в 2016 году // ROSOLYMP.RU: Информационный портал всероссийской олимпиады школьников, http://www.rosolymp.ru/attachments/10616_Informatika_Trebovaniya_ZE_2016.pdf

НОВОСТИ

В планах Seagate — жесткие диски на 14 и 16 Тбайт

Seagate приближается к достижению объявленной ранее цели о создании жесткого диска на 20 Тбайт к 2020 году: в компании сообщили, что в ближайшие полтора года планируется выпустить винчестеры емкостью 14 и 16 Тбайт. Сейчас максимальная вместительность жестких дисков Seagate — 10 Тбайт, идет тестирование накопителя, заполненного гелием, на 12 Тбайт. Seagate также расширяет модельный ряд 10-терабайтных винчестеров. По словам представителей компании, в ней хотят увеличить минимальную емкость винчестеров, используемых в ПК, до терабайта. Жесткие диски в последнее время начинают проигрывать позиции твердотельным накопителям, особенно в сегменте устройств емкостью до 256 Гбайт, но, как утверждают в Seagate, спрос на винчестеры остается высоким на потребительском рынке и среди производителей систем видеонаблюдения. Кроме того, жесткие диски популярны как накопители для систем хранения данных архитектуры NAS. Так или иначе, Seagate, сделавшая себе имя на винчестерах, выходит в лидеры и как поставщик SSD. В прошлом году компания представила SSD на 60 Тбайт и может начать продавать его в нынешнем.

Принтер металлических частей обещает изменить принципы производства

В компании Vader Systems разработали 3D-принтер нового типа, который, как надеются его создатели, изменит основы производства металлических изделий. Он работает, расплавляя алюминиевую нить и затем с помощью магнитного поля выбрасывая капли с высокой частотой через сопло на горячую плиту, которая перемещается, формируя заданную деталь. Этот процесс отличается от более традиционных, когда металлический порошок, нанесенный равномерно, расплавляется лазером или электронным лучом, формирующим очередной слой детали. Как объясняют в Vader, при этом есть вероятность, что отдельные частицы не расплавятся, из-за чего изделие может иметь дефекты. В дальнейшем принтер нового типа можно будет применять для распечатки хирургических инструментов и других сложных изделий на заказ.

Ford направит миллиард долларов на разработку интеллекта для робомобилей

Стартап Argo AI, основанный бывшим руководителем разработки аппаратной платформы самоуправляющихся автомобилей в подразделении Waymo компании Google Брайаном Салески и инженером из исследовательской группы того же направления в компании Uber Питером Рэндером, получит от компании Ford миллиард долларов на ближайшие пять лет. За это время создатели стартапа намереваются разработать интеллектуальную систему автоматического управления автомобилем, соответствующую четвертому уровню из пяти по классификации организации SAE, то есть способную вести машину почти в любых условиях, не полагаясь на вмешательство водителя. Современные системы еще не достигли этого уровня и во многих ситуациях требуют помощи человека. Пятый уровень SAE предусматривает полную автоматизацию вождения в любых ситуациях, соответствующую возможностям человека.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)

А. В. Каплан,
школа № 2009, г. Москва

РЕЗУЛЬТАТЫ АПРОБАЦИИ УЧЕБНО-МЕТОДИЧЕСКОГО КОМПЛЕКТА «ИНФОРМАТИКА ДЛЯ ВСЕХ» В ПЕРВОМ КЛАССЕ

Аннотация

В статье рассматривается учебно-методический комплект «Информатика для всех» с точки зрения основных содержательных линий и их реализации в конкретных заданиях пособий. Описывается опыт внедрения данного УМК в первом классе с указанием объективных и субъективных результатов апробации.

Ключевые слова: информатика, начальное образование, ФГОС НОО, информатика в начальной школе, «Информатика для всех».

Контактная информация

Каплан Адель Викторовна, учитель начальных классов школы № 2009, г. Москва; *адрес:* 117042, г. Москва, ул. Адмирала Руднева, д. 16, корп. 1; *телефон:* (495) 717-19-45; *e-mail:* Kaplan.av@sch2009.net

A. V. Kaplan,
School 2009, Moscow

RESULTS OF APPROBATION OF THE TRAINING KIT "INFORMATICS FOR EVERYONE" IN THE FIRST CLASS

Abstract

The article describes the training kit "Informatics for everyone" in terms of the basic content lines and their implementation in specific tasks. The experience of the introduction of the training kit in the training of the pupils of 1st class is described with an indication of the objective and subjective results of testing.

Keywords: informatics, elementary education, FSES of elementary general education, informatics in elementary school, Informatics for everyone.

Выбор молодого специалиста

В сентябре 2015 года я, начинающий учитель начальной школы, приняла своих первых первоклашек. Небезосновательно я думала, что информатика станет у них одним из самых любимых предметов, и хотела уделить ему особое внимание. Однако при выборе программы обучения возникли некоторые трудности, которые заставили меня обратиться к знаниям, накопленным за период обучения в педагогическом колледже.

В колледже студентам, которым после выпуска предстояло обучать учеников начальных классов по Федеральному государственному образовательному стандарту начального общего образования (ФГОС НОО), при этом не имея багажа работы по ГОС, неоднократно сообщалось, что ФГОС серьезно меняет подход к самому определению образовательных результатов. В ФГОС под образовательными результатами понимается «приращение» в личностных ресурсах обучаемых, которые могут быть использованы ими при решении значимых проблем, в то время как раньше акцент делался на достижении предметных результатов в парадигме «понимать/знать/уметь». Требования стандарта определяют, что развитию мотивационных, инструментальных и когнитивных ресурсов личности соответствуют три группы результатов обучения: предметные, метапредметные и личностные. А следовательно, и выбор УМК предстояло делать, опираясь на эти новые требования. Но, конечно, не только на них.

В ходе освоения курса «Информатика с методикой преподавания в начальных классах» студентов знакомили с основами информатики как учебной дисциплины, с методикой преподавания информатики и, конечно, с современными тенденциями в начальном курсе информатики.

Так, во внимание, безусловно, принимались работы академика РАО А. А. Кузнецова, в которых он рассуждает о серьезном расхождении курса школьной информатики с предметом информатики как науки и определяет, что раздел «социальная информатика» в школьном курсе практически не представлен [4]. А ведь именно навыки управления информационными процессами в социуме можно считать ключевыми в информационном обществе.

Подробно рассматривались и мнения ведущих специалистов в области обучения информатике в начальной школе. В частности, положение о том, что основной путь формирования системно-информационной картины мира на уроках информатики состоит в налаживании уже утраченных связей между информационными единицами, обретении системы знаний, восстановлении связи с реальным миром, развитии интеллектуального и этического потенциала школьников [1].

Особое внимание обращалось на то, что современная пропедевтическая подготовка младших школьников в области информатики представляет собой многогранный процесс, предполагающий взаимосвязь всех его компонентов. Знания и навыки, приобретаемые учениками в ходе изучения началь-

ного курса информатики и являющиеся значимыми для формирования функциональной грамотности, социализации учащихся, значимы для младших школьников и как основа для решения различных личностно важных задач не только в рамках учебной дисциплины «Информатика», но и в ходе всего курса начального образования [2].

Кроме того, важно было учесть, что ФГОС НОО по сути требует более широкого понимания курса информатики, заявляя метапредметными результатами обучения навыки и умения, являющиеся для информатики предметными. Но до сих пор в информатике для начальной школы не задействован огромный потенциал предмета в получении метапредметных результатов на основе изучения и освоения умений эффективных действий в информационных процессах, прежде всего, в процессах получения и передачи информации как основ функциональной грамотности [5].

Обобщив всю имеющуюся информацию, я остановилась на двух УМК:

- проверенном временем УМК «Информатика в играх и задачах» А. В. Горячева, как отвечающем в целом значительной части требований ФГОС НОО (издательство «БАЛАСС»);
- и экспериментальном, разработанном уже непосредственно под требования ФГОС НОО, УМК «Информатика для всех», созданном авторским коллективом под руководством Д. И. Павлова, редактором которого выступил также А. В. Горячев (издательство «БИНОМ. Лаборатория знаний»).

Оставалось сделать выбор между этими двумя УМК. Ход моих рассуждений по результатам сравнительного анализа был следующим.

Во-первых, УМК «Информатика в играх и задачах» ориентирован на развитие логического и алгоритмического мышления (что относится к группе логических познавательных УУД). УМК «Информатика для всех» только половину времени уделяет развитию логического и алгоритмического мышления. Собственно, развитие логического мышления чаще выступает как инструмент, нежели как цель. Вторая же половина сосредоточена на совершенствовании умений получения и передачи информации (это весьма широко отражает практически всю группу коммуникативных и познавательных УУД). По сути, это не плюс или минус, а определенная особенность УМК «Информатика для всех». И, исходя из своих взглядов, я начала склоняться к экспериментальному УМК.

Во-вторых, развитие логического и алгоритмического мышления в УМК «Информатика в играх и задачах» производится через систему игр и заданий нарастающей сложности, а в УМК «Информатика для всех» — через некоторое число усложняющихся теоретических и игровых заданий с переходом к созидательным действиям в компьютерных средах KoduGameLab (третий класс) и Scratch (четвертый класс).

В-третьих, УМК «Информатика в играх и задачах» приобрел итоговый, столь знакомый учителям вид в ходе многолетней апробации, а УМК «Информатика для всех» изначально создавался с учетом накопленного опыта и современных представлений о потребностях как учащихся, так и педагогов. Конечно, это тоже нельзя назвать плюсами или минусами того или другого комплекта, но все указанные особенности повлияли на мой выбор.

В результате выбор был сделан в пользу нового УМК «Информатика для всех» авторов Д. И. Павлова, Ю. А. Аверкина под редакцией А. В. Горячева.

Организационные особенности

Для обучения в первом классе в УМК «Информатика для всех» предлагаются две рабочие тетради и дополнительная рабочая тетрадь с заданиями на освоение компьютера. Предложенная программа рассчитана на два часа в неделю (66 уроков). Учитывая отсутствие вариативности в формировании расписания дисциплин для первого класса, было принято решение проводить занятия в форме *внеурочной* деятельности по одному занятию дважды в неделю (во вторник и четверг) в течение 33 недель.

Было решено включить в обучение компьютерный практикум, так как техническое оснащение класса позволяло проводить занятия на компьютере. Даже «скользкий» момент, связанный с тем, что класс оборудован компьютерами MacBook, не создал проблемы. Практикум, как оказалось, ориентирован на «понятийное» освоение компьютера, а не на привязку к четким действиям в строго определенном программном пакете. А потому использование, например, OpenOffice Writer не стало проблемой (как иногда бывает, если используешь практикум, составленный для Microsoft Word).

Домашние задания практически не задавались. Только трижды из-за отмененных ввиду праздников и иных причин занятий ученикам рекомендовалось выполнить отдельные задания дома. И стоит отметить, что из 24 учеников, посещавших занятия, не менее 20 ребят эти задания выполняли.

Содержательные аспекты

Безусловно, рассмотреть весь УМК в рамках одной статьи не получится. Поэтому **остановимся на основных содержательных элементах, работа с которыми была особенно интересной**.

Прежде всего, отмечу подход к реализации классических заданий — на ***логические цепочки***. Первое время казалось, что дети не в состоянии справиться с таким темпом усложнения материала (рис. 1).

Так, уже на первых уроках, где используются только контуры одноцветных фигур, периодически добавляется компонент вращения или перемещения одной из них.

К четвертому занятию количество фигур возрастает, усложняется их перемещение и появляются различия «закрашенная»/«не закрашенная».

К седьмому занятию наступает пик трудности — закономерность учитывает смену формы и цвета, причем количество форм и цветов в одной цепочке регулярно меняется. На данном этапе самостоятельно справлялись с заданием не более половины учеников. И тут на помощь пришли методические рекомендации, написанные, стоит отметить, языком простым и доступным. Рекомендации предлагали детально объяснять детям каждое неполучившееся задание, в том числе привлекая к объяснению учеников, которые с этим заданием справились.

Уже к пятнадцатому уроку такой подход привел к тому, что с заданием справились самостоятельно свыше 2/3 учеников. А задание было непростое: в закономерности учитывалась смена фигуры, а также цветов заливки и контура. Но это была лишь половина задания. Вторая половина предполагала самостоятельное воспроизведение аналогичной цепочки с изменениями в условии, а именно с заменой фигуры и цвета на другие.

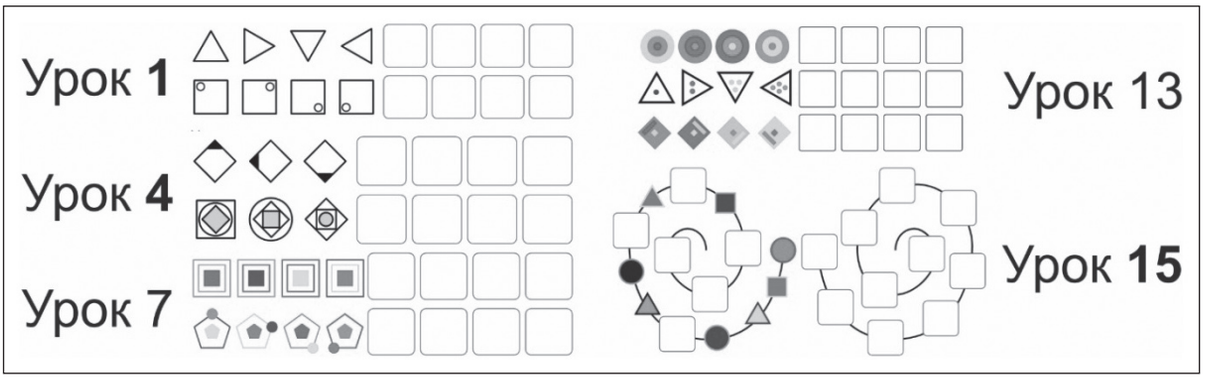


Рис. 1. Задания на логические цепочки

Впрочем, цепочки — это не единственная часть логического блока заданий. В тетрадях присутствуют и так называемые ***задания со спичками***. Не вычислительные, а направленные на работу с формой и поданные не через спички, а через карандаши, которые дети с интересом раскладывают на столе в поисках нужного результата.

Стоит упомянуть и ***малые судоку***, которые появляются уже во втором полугодии (рис. 2). Это, конечно, не новаторство, но сложно недооценить удобство и пользу от заданий этого типа.

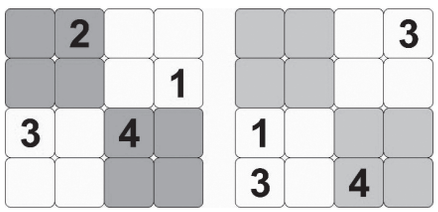


Рис. 2. Задания «Малые судоку»

Следует отметить **несколько организационных моментов**, являющихся яркой особенностью курса.

Первое — ***вариативность правильных решений***. Так, на рисунке 3 приведена иллюстрация к упражнению, в котором ученикам было предложено разделить предметы на группы (закрасив контуры) и озаглавить эти группы.

Рис. 3. Задание на деление на группы

Одни дети предложили четыре группы: «Еда», «Игрушки», «Учебные принадлежности», «Техника»; другие ребята разделили предметы на три группы; а двое детей — и вовсе на две: «Еда» и «Не еда». В соответствии с методическими рекомендациями было предложено дать оценку полученным результатам самим детям — путем обсуждения. При таком подходе к выполнению задания главным становится не столько «правильный» ответ, сколько мотивация, приведенная учеником в пользу своего решения. А при должной аргументации и четыре, и две группы — вполне верное решение.

Хочется выделить также большую ***вариативность в технике выполнения заданий***. Например, в задачах на соответствие ученики устанавливают соответствия между группами объектов, ставя цифры или значки, закрашивая разными цветами или наклеивая разноцветные маркеры. Авторы УМК не предлагают лишь «соединить линиями», мотивируя неопрятностью такого способа решения.

Очень необычно представлена в тетради ***работа с инструкциями и высказываниями***. Так, в примере на рисунке 4 видно, что ученикам предложены три объекта, три возможных цвета, а также три множества, которые необходимо заполнить по условию: скажем, кругами НЕ красного цвета или НЕ ромбами желтого и зеленого цветов.

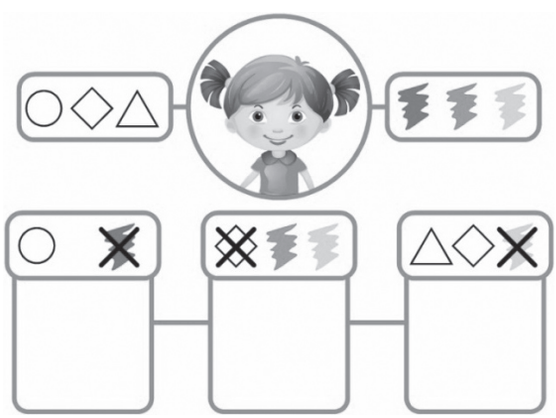


Рис. 4. Пример работы с инструкциями и высказываниями

Ко второму полугодию первого класса набор заданий расширяется. Появляются **задания на получение и анализ информации из изображений, а также на построение логических заключений**.

Рис. 5. Задания на получение и анализ информации из изображений

Так, на рисунке 5 упражнение А — простое: ученики просто обсуждают, как называется такой тип рисунка, а также считают, сколько на рисунке яблок, слив, груш и т. д.

А вот упражнение Б на рисунке 5 — куда сложнее. Обсудив, что на рисунке осень и описав все, что на нем видят, ученики получают задание-усложнение: что изменилось бы на рисунке, если бы на нем был тот же пейзаж зимой?

Опустим задания на простые виды шифрования, а также самые простые варианты чтения таблиц. Они очень заинтересовали детей и в целом оказались поданы в доступной форме.

Привлекла внимание **игра «анаграмма»**, которая во второй части тетради встречается неоднократно. Несмотря на кажущуюся сложность, задания, проверяющие умение составить несколько слов из набора букв исходного слова, крайне увлекательны и очень полезны.

Шестьдесят шесть уроков, предусмотренных программой, разделены на семь тем (разделены, конечно, довольно условно). Каждая тема завершается проверочной работой, выполнение которой позволяет детям оценить свои результаты.

Результаты «экспериментального» года

Говоря о результатах, стоит разделить их на две группы — объективные (измеримые) и субъективные.

К объективным, хотя и косвенным, результатам можно отнести следующие:

- среди 12 первых классов в параллели ученики, занимавшиеся по УМК «Информатика для всех», показали высокие результаты по математике — класс в целом показал третий результат, а без учета детей, не занимавшихся информатикой, — второй;
- было проведено дополнительное тестирование по учебному пособию С. А. Козловой, А. Г. Рубина для первого класса «Самостоятельные и контрольные работы по курсам “Математика” и “Математика и информатика”» [3], а также по методическому пособию Т. П. Хиленко «Типовые задачи по формированию универсальных учебных действий. Работа с информацией. 1 класс» [6]. Был показан стопроцентный уровень обученности, а уровень успешности — 89 %.

К субъективным результатам отнесем следующие:

- высокая явка и заинтересованность в занятиях как у детей, так и у родителей;
- *все* ученики, посещавшие занятия по информатике в первом классе, решили посещать их и во втором; кроме того, к ним добавились учащиеся из числа тех, которые пропустили курс для первого класса;
- в 2016/2017 учебном году двое учителей школы, квалифицированные и опытные специалисты, принявшие очередные первые классы, реализуют УМК для первого класса, опираясь на опыт прошлогоднего эксперимента, и дают положительные отзывы о работе с УМК.

И главный результат для меня как для молодого специалиста: не произошло разочарования в выбранном курсе — ни у меня, ни у администрации, проявлявшей пристальное внимание как к моей работе в целом, так и к реализации экспериментальной программы в частности.

В 2016/2017 учебном году мною проводится обучение школьников с опорой на УМК «Информатика для всех» для второго класса (учебники в электронной форме). Учебники необычно построены и раскрывают начальный курс информатики с интересной и непривычной стороны.

Что же касается программы для первого класса, то, на мой взгляд, она представляет собой компромисс между старым, многократно апробированным подходом и новым. И вполне подойдет для реализации в первом классе, независимо от принятой системы обучения и опыта педагога.

Список использованных источников

1. *Бешенков С. А., Матвеева Н. В.* Обучение информатике в среднем звене общеобразовательной школы // Информатика и образование. 1997. № 8.

2. *Босова Л. Л.* Подготовка младших школьников в области информатики и ИКТ: опыт, современное состояние и перспективы. М.: БИНОМ. Лаборатория знаний, 2012.

3. *Козлова С. А., Рубин А. Г.* Самостоятельные и контрольные работы по курсам «Математика» и «Математика и информатика». М.: БАЛАСС, 2013.

4. Основы общей теории и методики обучения информатике: учебное пособие / под ред. А. А. Кузнецова. 2-е изд. (эл.). М.: БИНОМ. Лаборатория знаний, 2013.

5. *Павлов Д. И.* Начальный курс информатики — новый взгляд в свете изменения характера начального общего образования // Наука, образование и инновации. Сборник статей международной научно-практической конференции (г. Уфа, 28 октября 2016 г.). Т. 3. Уфа: Омега сайнс, 2016.

6. *Хиленко Т. П.* Типовые задачи по формированию универсальных учебных действий. Работа с информацией. 1 класс. М.: Просвещение, 2012.

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

В. Ф. Очков, А. В. Калинина,
Национальный исследовательский университет «МЭИ», г. Москва

ПО ПОРЯДКУ СТАНОВИСЬ!

Аннотация

В статье рассмотрены вопросы последовательного и параллельного выполнения операторов при реализации алгоритма сортировки, а также создания программ сортировки вектора и анимации этой процедуры в среде Mathcad.

Ключевые слова: вектор, сортировка, анимация, Mathcad.

Контактная информация

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

Калинина Алена Владимировна, студентка Национального исследовательского университета «МЭИ», г. Москва; *адрес:* 111250, г. Москва, Красноказарменная ул., д. 14; *e-mail:* kalinina.alena22@gmail.com

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

THE ORDER GETS!

Abstract

The article discusses the issues of sequential and parallel execution of operators when implementing the sorting algorithm, as well as the creation of programs for sorting the vector and the animation of this procedure in Mathcad.

Keywords: vector, sorting, animation, Mathcad.

Как-то раз первый автор статьи разговорился со своей бывшей одноклассницей. Вспоминали хорошее и плохое школьных лет. Эта одноклассница призналась, что для нее одними из самых неприятных моментов в школе были... построения на уроках физкультуры. Девочки и мальчики строились в спортзале по росту и вставали друг против друга в две шеренги в стиле «белый верх — черный низ»: в белых маечках и черных трусиках. Эта одноклассница была самая высокая в классе и довольно крупная. В обычной жизни и в обычной одежде это было не так заметно, но в спортивном зале она оказывалась во главе шеренги полуголая (по стандартам тех времен) и как бы... у позорного столба. Мальчики ее разглядывали, а потом дразнили дылдой. Автор также вспомнил неприятный момент таких построений: у него был друг довольно низкого роста, который всегда оказывался в конце «мужской» шеренги, и на него из «женской половины» презрительно поглядывала девочка, которая ему нравилась[*].

Да, нужно признать, что процедура построения по росту в школе или в армии несколько дискриминационная. В армии к тому же низкорослые солдаты оказываются в конце марширующей колонны и «глотают пыль». Можно предложить альтернативный порядок построения[**], например, по возрасту, цвету глаз, волос или по какому-то другому атрибуту школьников или солдат, не имеющему признаков дискриминации. Но, так или иначе, тут нужно будет использовать процедуру *сортировки*, которая часто становится темой школьных уроков информатики. Операция вроде бы рутинная, разобранная «по косточкам», но давайте внесем в нее что-то новое.

Разработано множество алгоритмов сортировки. В «Википедии» можно найти такие их названия: сортировка пузырьком, сортировка перемешиванием, гномья сортировка, сортировка вставками, сортировка слиянием, сортировка с помощью двоичного дерева, сортировка Timsort, сортировка подсчетом, блочная сортировка, поразрядная сортировка, сортировка выбором, сортировка Шелла, сортировка расческой, пирамидальная сортировка, плавная сортировка, быстрая сортировка, интроспективная сортировка, терпеливая сорти-

---

[*] Говорят, что наш великий полководец Александр Суворов однажды выстроил своих крепостных по росту в две шеренги: неженатые парни с одной стороны, незамужние девицы — с другой. Так сложились пары, которым он повелел идти в церковь венчаться. Мнение новобрачных никого не интересовало. (https://lenta.ru/articles/2016/04/26/russian_slavery)

[**] Вообще, без построения по какому-либо признаку тут не обойтись: школьники или солдаты будут спорить и толкаться, стараться встать ближе к правому (приоритетному) краю шеренги или, наоборот, «затеряться в середине толпы». Можно предложить строить школьников в том порядке, в каком они по алфавиту записаны в классном журнале. Но тут также можно узреть элементы дискриминации: кого чаще вызывают к доске — Абрамова, Яковлева или, наоборот, Иванова-Петрова-Сидорова из середины списка. Кстати, к сортировке мы очень часто прибегаем, выстраивая файлы на экране компьютера по имени, типу, размеру или времени создания. Эти значения «в шеренге файлов» могут либо возрастать, либо убывать после повторного щелчка по имени атрибута.

ровка, рекурсивная сортировка, глупая сортировка, блинная сортировка, сортировка перестановкой*...

Эти алгоритмы реализованы на разных языках программирования. Попробуем и мы это сделать, но в среде пакета Mathcad [9, 11], который в настоящее время широко используется в школах и вузах**. Так, в списке использованных источников [1–3, 5, 8–12] приведены данные о статьях в журналах издательства «Образование и Информатика», где в названии присутствует слово Mathcad. Самих же статей, где речь идет об использовании этого пакета в учебном процессе, но где нет слова Mathcad в названии, гораздо больше.

Основа многих алгоритмов сортировки — это обмен численных значений у двух переменных. В языке программирования Basic, например, эту процедуру выполняет оператор swap(a, b). Но в математическом пакете Mathcad этого оператора нет, хотя его нетрудно реализовать другими средствами.

На рисунке 1 можно видеть два способа реализации оператора swap(a, b) в среде Mathcad: традиционный («треугольный» — с использованием вспомогательной переменной) и «экзотический», о котором многие пользователи Mathcad и не догадываются, — не через последовательное присваивание, а в некоем параллельном режиме, когда операторы присваивания записаны в матричном виде и выполняются независимо друг от друга. Эта технология обсуждалась на сайте пользователей Mathcad: https://www.ptcusercommunity.com/message/448295

Рис. 1. Два способа перестановки значений переменных в среде Mathcad

* В студенческие годы первый автор подрабатывал на овощной базе. Самой непыльной и интеллигентной работой там считалась... сортировка лимонов. В те времена лимоны были единственными фруктами, какие продавались не по весу, а поштучно (к чаю) за 25, 30, 35, 40 и 45 копеек или что-то в этом роде: чем лимон крупнее, тем он, естественно, дороже. На овощных базах их сортировали так: просовывали в круглые дырки с разным диаметром на специальном сортировочном столе. Лимоны падали в отдельные ящики, которые потом развозили по разным магазинам, чтобы не было пересортицы. Автор же в те годы, глядя на эти ящики с отсортированными лимонами, вспоминал... вузовский курс математической статистики с его нормальным распределением: см. [6], а также рисунок 8, где можно увидеть нормальное распределение... студентов по росту. Анекдот тех времен: «Где подрабатываешь? — На овощной базе. — Устаешь? — Очень! — Ящики таскаешь? — Нет, лимоны сортирую — голова устает!»

** В пакете Mathcad есть, кстати, функция sort для этой работы, но мы пока забудем о ее существовании. Кроме того, встроенные в Mathcad функции csort и rsort позволяют сортировать матрицы по столбцам (c) или рядам (r), задавая ключ сортировки — номер столбца или строки матрицы. На сайте https://www.ptcusercommunity.com/videos/6867 размещена анимация сортировки матрицы, когда значения ее элементов отображаются цветом.

Если под рукой есть оператор перестановки swap(a, b) в явном («треугольном») или в матричном виде, то несложно написать в среде Mathcad функцию-программу (рис. 2), реализующую, наверно, самый простой алгоритм сортировки, суть которого такова.

Школьники произвольно выстраиваются в ряд (у нас на рисунке 2 это делает функция runif, возвращающая вектор с десятью элементами и со случайными числами в интервал от 150 до 200), а учитель физкультуры (физрук) проходит вдоль шеренги слева направо (вложенный цикл for) и меняет местами пары школьников, если левый школьник (V_{i-1}) окажется выше правого (V_i). Эта процедура выполняется до тех пор, пока (while) очередной проход физрука вдоль шеренги не выявит беспорядка и переменная Flag (признак окончания сортировки) останется равной нулю.

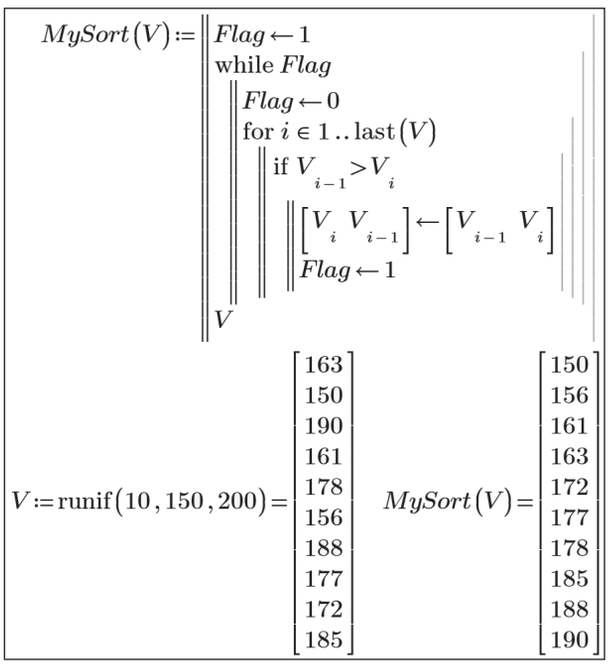


Рис. 2. Наипростейшая сортировка вектора (Mathcad Prime)

Пакет Mathcad 15 имеет удобные средства анимации [4, 7], которые можно применить и для визуализации сортировки векторов. На рисунке 3 показано, как в программу на рисунке 2 были вставлены три дополнительных оператора (они в рамках), позволившие показать сортировку вектора в анимации: столбики на диаграмме последовательно меняются местами, выстраиваясь в правильную шеренгу. При этом очередной «высокий столбик» перемещается вправо, как бы толкаясь и передавая эстафету столбику с более высоким ростом. Эта анимация выложена на сайте: https://www.ptcusercommunity.com/videos/6795

Управляет анимацией системная переменная FRAME, которая при создании анимации автоматически будет менять свое значение от 0 (рисуется исходный вектор в виде столбиков) до 999 (отсортированный вектор, часть которого показана на рисунке 3). Перед созданием анимации можно вручную менять значение системной переменной FRAME и видеть изменения на графике — на столбчатой диаграмме. Так программа отлаживается перед анимацией. В программу на рисунке 3 вставлен счетчик перестановок соседних элементов векторов n. Если он становится равным номеру кадра анимации

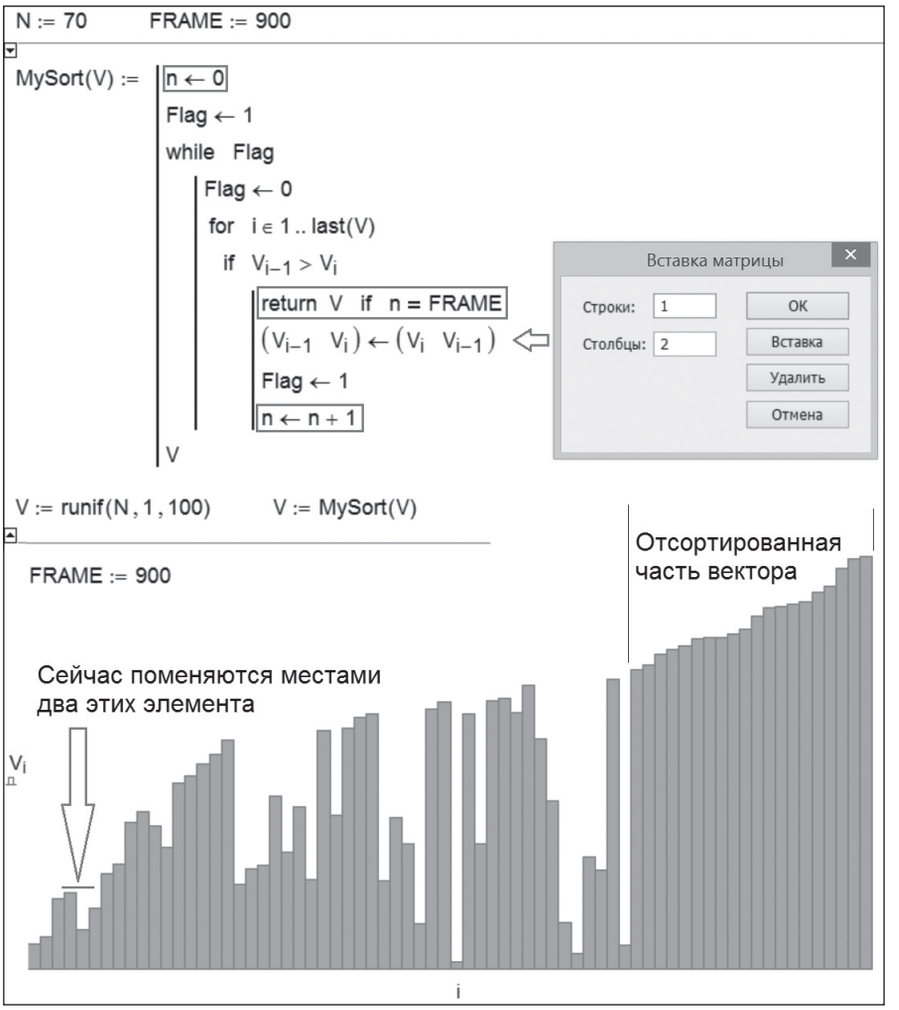


Рис. 3. Создание анимации сортировки по простейшему алгоритму (Mathcad 15)

FRAME, то программа оператором return возвращает частично отсортированный вектор, который рисуется на графике в виде столбчатой диаграммы.

Характер анимации, один кадр которой показан на рисунке 3, можно кардинально поменять, если в конце плеча оператора if (после оператора n ← n + 1) записать оператор break. Он прерывает выполнение цикла for и передает управление программой в его конец: физрук, переставив местами очередную пару школьников, не пойдет дальше с проверками вдоль шеренги, а пробежит в ее конец* и перейдет к циклу while. С оператором break отсортированные элементы вектора будут группироваться не на правом (рис. 3), а на левом фланге шеренги, но конечный результат будет тот же: слева окажутся самые низкие, а справа — самые высокие. Эта анимация размещена на сайте: https://www.ptcusercommunity.com/videos/6824

Программа сортировки на рисунке 3 имеет переменное число выполнения вложенных циклов while... for, и это число (оно хранится в переменной *n*) зависит от вида исходного вектора. Если этой программе «подсунуть» уже отсортированный вектор, то вложенный цикл while...for будет выполнен минимальное число раз, равное размеру вектора (у нас на рисунке 3 это 70 раз, как указано первым оператором). Если же этой программе дать уже отсортированный, но «перевернутый» вектор (этот «переворот» делает встроенная в Mathcad функция reverse), то вложенный цикл for будет «глупо»** выполняться максимальное число раз, переставляя пары элементов вектора. Эта анимация размещена здесь на сайте: https://www.ptcusercommunity.com/thread/135862 как некий добавок к вопросу на форуме Mathcad по «перевертыванию» векторов.

Но в среде Mathcad несложно создать очень простую программу, сортирующую любой исходный вектор одним циклом. При этом число выполнения тела цикла будет равно числу элементов в исходном векторе (рис. 4).

Алгоритм в программе на рисунке 4 такой: физрук оценивает взглядом неотсортированную шеренгу школьников, находит самого высокого и отправляет его

* В среде Mathcad, как и в языке Pascal, есть три «прерывающих» оператора: return, break и continue. Первый из них прерывает выполнение всей программы и возвращает из нее свой операнд (см. рис. 3). Второй оператор (он без операнда) прерывает выполнение цикла for или while и передает управление программой в конец цикла. Третий оператор тоже прерывает выполнение цикла, но передает управление не в его конец, а в его начало. Кстати, мало кто даже из англичан знает, что означает break в слове завтрак — a breakfast. A fast — это по-английски «пост» (в смысле, не послание в Интернете, а ограничение в питании), a breakfast — это *прерывание* поста — разговение, которое имеет место на Пасху или в Рождество.

** Если «глупой» обезьяне дать очищенный банан, то она попытается его еще раз очистить. Так и наша программа на рисунке 3 будет еще раз пытаться отсортировать уже отсортированный вектор.

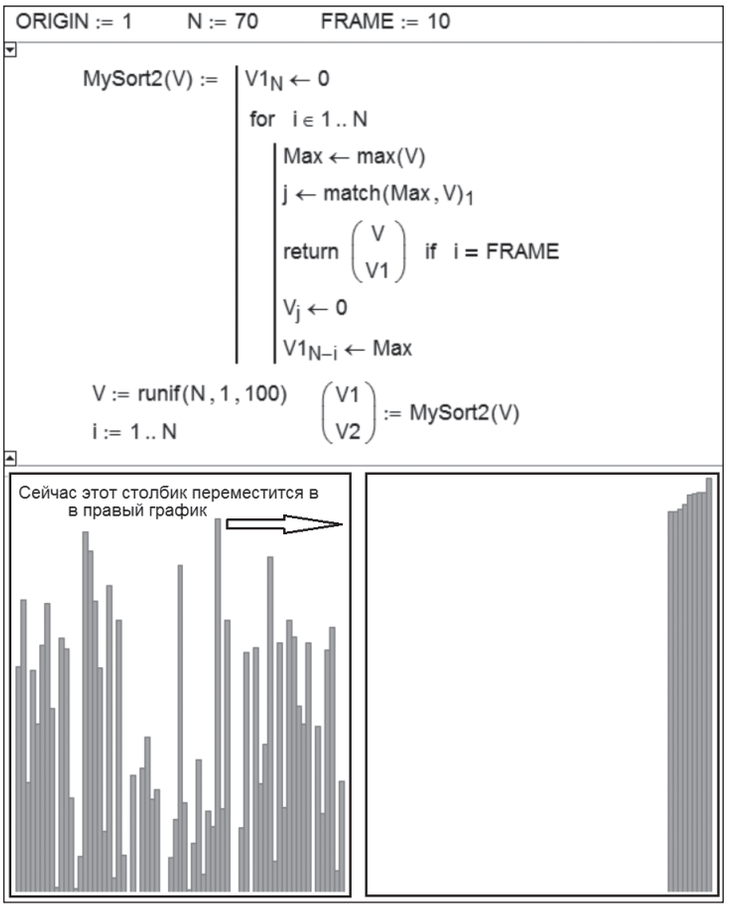


Рис. 4. Создание анимации сортировки по второму алгоритму (Mathcad 15)

на левый фланг отдельной формируемой шеренги. Эта операция повторяется N раз, где N — число элементов в исходном векторе. На столбчатых диаграммах рисунка 4 видно, что из исходного вектора в формируемый вектор перенесено девять «высокорослых» элементов (см. пустые места в левой диаграмме) и очередного переноса ожидает очередной «высокорослый» элемент. Программа на рисунке 4 исполняет этот простой алгоритм, работая с двумя специфическими функциями пакета Mathcad:

- max — определение максимального элемента массива (в нашем случае вектора);
- match — определение номера или номеров элементов массива с заданным значением (эта функция выдает вектор с одним или более элементами, поэтому-то в программе на рисунке 4 эта функция записана с единичным индексом).

Примечание. Самый «левый» элемент вектора в среде Mathcad по умолчанию имеет нулевой номер. Это умолчание присутствовало в программах на рисунках 2 и 3. Но в программе на рисунке 4 это умолчание нарушено: второй важной системной переменной Mathcad — переменной ORIGIN — присваивается единица. После этого нумерация векторов будет начинаться с единицы*. Но в программировании тут более привычен нуль. И еще. Алгоритм, реализованный в программе на рисунке 4, применим к исходному вектору с положительными элементами. Но его несложно доработать и для вектора с любыми числовыми элементами.

Можно развить алгоритм, реализованный в программе на рисунке 4, — выбирать из неотсортированной части вектора не только максимальный, но и минимальный элемент и отправлять его на правый фланг формируемого нового вектора.

Анимация, создание которой отображено на рисунке 4, расположена по адресу: https://www.ptcusercommunity.com/videos/6837.

Школьники могут попробовать реализовать в среде Mathcad и, главное, анимировать алгоритмы сортировки, описанные в «Википедии» (см. выше) или свои собственные. При этом можно использовать встроенную в Mathcad функцию time для оценки скорости работы тех или иных пользовательских функций сортировки по сравнению со встроенной функцией sort. При этом нужно иметь в виду, что скоростные изощренные алгоритмы сортировки создавались во времена тихоходных вычислительных машин, когда время счета было одним из главных лимитирующих факторов. Теперь же право на жизнь имеют и медленные алгоритмы, у которых их суть лежит на поверхности — как в наших программах.

Есть такая наука бионика. Она позволяет решать технические проблемы, наблюдая за поведением живых организмов. Например, один из самых оптимальных алгоритмов решения задачи коммивояжера [5] подсказали... муравьи, притаскивающие грузы в муравейник наикратчайшим маршрутом. Мы, создавая алгоритм сортировки, тоже можем обратиться к «живым организмам» — построить в спортзале шеренгу школьников,

* Программиста призвали в армию, поставили в начале шеренги и отдали команду: «По порядку рассчитайсь!» Программист сначала уточнил, по какой системе (двоичная, восьмеричная, десятичная, шестнадцатеричная или какая-то другая особая) нужно вести счет, а потом выкрикнул: «Нулевой!»

например, по алфавиту, а затем приказать им перестроиться по росту. Все это можно записать на видеокамеру, а затем с помощью замедленного просмотра на мониторе компьютера проанализировать и попытаться воссоздать данный «живой» алгоритм самосортировки. Если сортировку, проводимую физруком (см. выше), можно реализовать на однопроцессорной машине, то самосортировка школьников потребует уже многопроцессорной системы. При этом физрук может вмешиваться в процесс такой сортировки, контролировать его. Тут на память приходят реальные технические объекты или объекты живой природы. В современном самолете есть и центральный компьютер, и локальные вычислительные устройства (контроллеры), один из которых, например, регулирует подачу топлива в двигатель, прекращает подачу топлива при нештатной ситуации. У человека есть «центральный» мозг, но человек отдергивает руку от горячего предмета рефлекторно, без задействования «умственных способностей». Все это повышает надежность работы сложных технических и живых систем, повышает их управляемость.

И вторая, более простая тема для исследования. Как изменится характер сортировки, если в конце плеча альтернативы программы на рисунке 2 поставить не оператор break, а оператор continue? И вообще, стоит ли вставлять их в программу? Как это будет влиять на скорость сортировки исходных векторов с разным распределением векторов «по росту»?

Статистическая добавка к статье.

На рисунке 2 показано, как встроенная в Mathcad функция runif генерирует вектор, содержащий случайные числа в интервале от 150 до 200. Числа 150 и 200 — это условно минимальный и максимальный рост человека в сантиметрах. При этом было сделано допущение, что процент людей с определенным диапазоном роста одинаков. На рисунке 5 показана гистограмма (частота встречи) чисел, генерируемых функцией runif. При этом мы отказались от сантиметров и перешли к старым русским аршинам и вершкам и допустили, что рост человека может меняться от двух аршин до сажени (три аршина). Почему мы отказались от сантиметров?

Когда-то перед лекцией по регрессионному анализу в курсе «Информационные технологии» первый автор статьи подбирал пример статистической выборки для такого исследования. Но когда он «взошел на кафедру» и взглянул на аудиторию, то понял, что эта выборка находится прямо перед его глазами. Была проведена перекличка студентов. Юноши вставали и сообщали свой вес и рост. Студентки, одной из которых была второй автор статьи, по понятным причинам, были освобождены от этой процедуры. Числа заносились в два вектора с именами «Вес» и «Рост» в среде математической программы Mathcad. В векторах получилось по 50 элементов. Эти массивы чисел послужили хорошей затравкой для лекции и основой для построения графика на рисунке 6 и вывода соответствующей регрессионной линейной зависимости веса, вернее, массы и роста юношей.

Захотелось также построить гистограмму роста студентов. Но один сантиметр в качестве элементарного диапазона роста тут не подошел: никакой статистической зависимости из такой гистограммы получить не удавалось. Не подошли тут и английские футы с дюй-

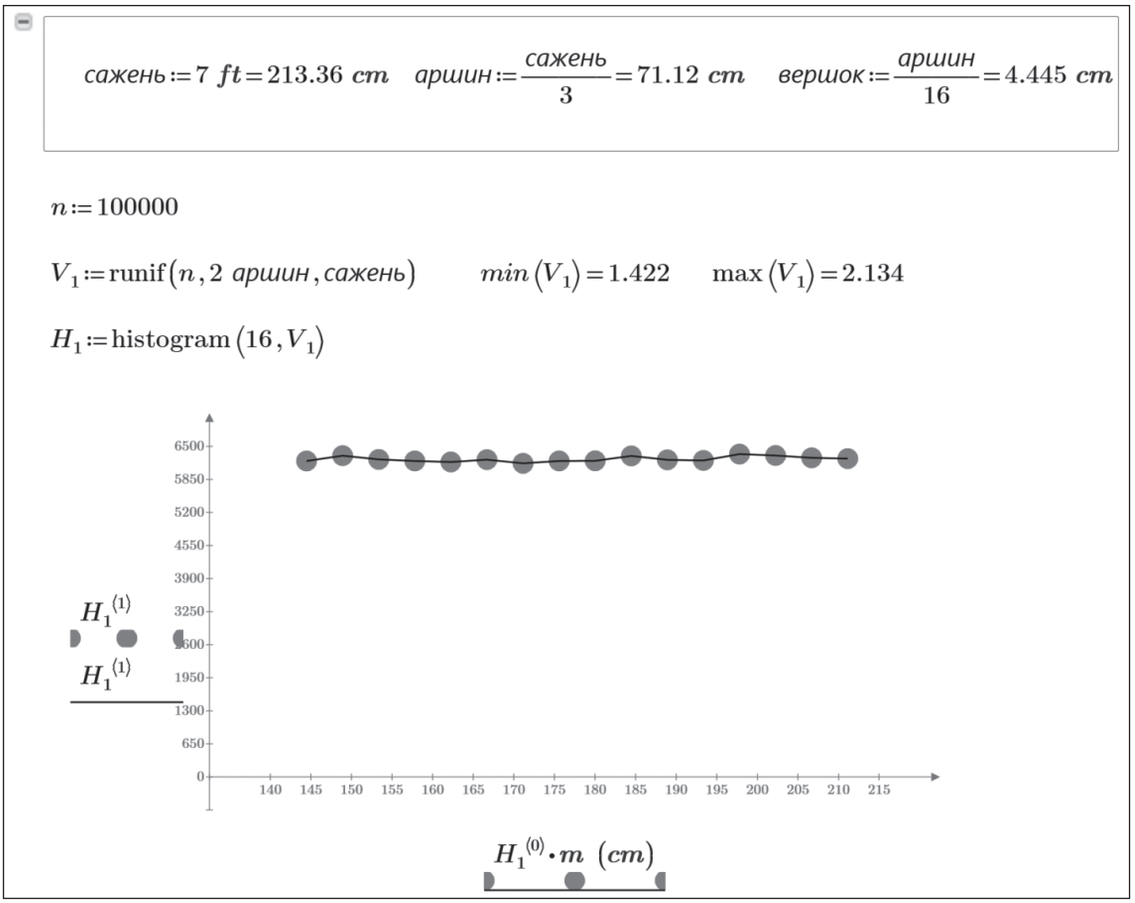


Рис. 5. Гистограмма равномерного распределения

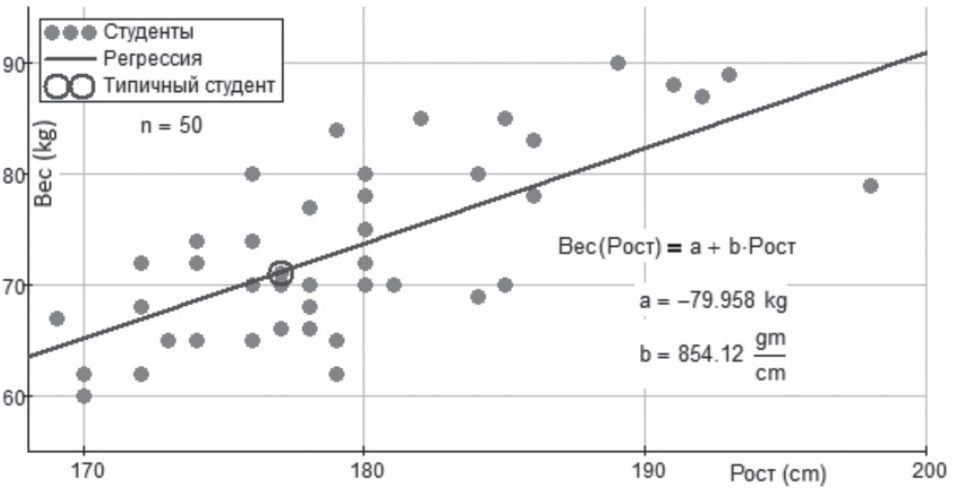


Рис. 6. Статистическая обработка параметров студентов

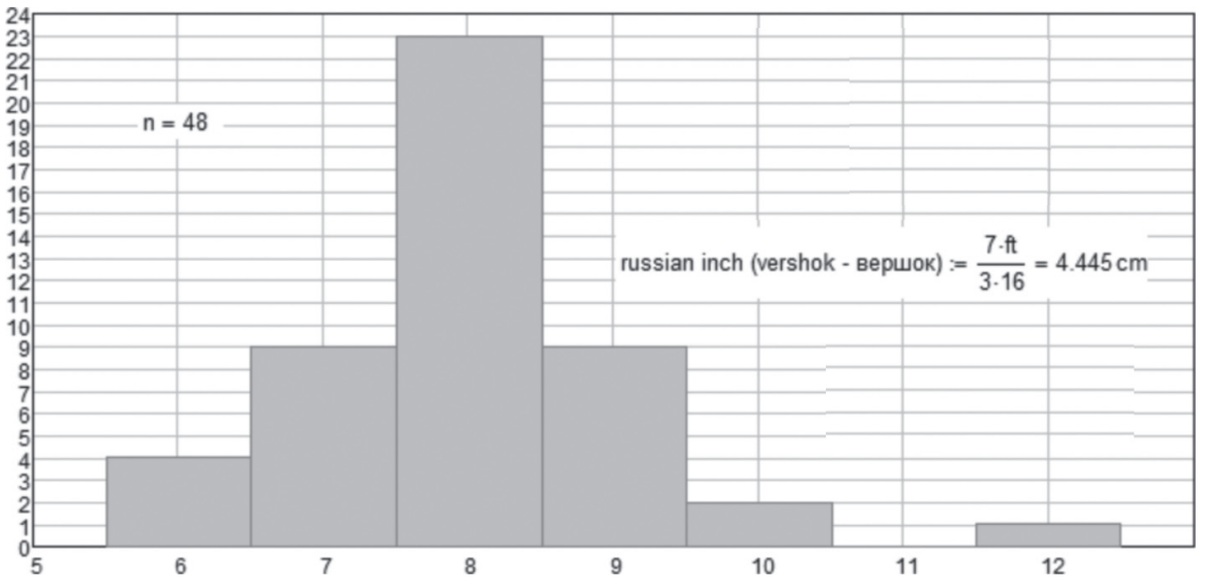


Рис. 7. Гистограмма роста студентов

мами*. Но старые добрые русские вершки оказались что надо!

На «вершковой» гистограмме (рис. 7) четко просматривается некий «статистический колокол» — кривая нормального распределения. Молодые люди по росту разбиваются на пять условных групп:

- низкого роста (четыре человека с шестивершковым ростом);
- скорее низкого, чем среднего роста (девять человек — семь вершков);
- среднего роста (23 человека — восемь вершков);
- скорее высокого, чем среднего роста (девять человек — девять вершков);
- высокого роста (два человека, 10 вершков).

Один студент выпал из нашей «статистики» и оказался ростом с тургеневского Герасима из повести «Муму» в 12 вершков**: исключение только подтверждает статистическое правило.

В Интернете можно найти не очень качественную, но очень интересную фотографию «живой» гистограммы роста людей (рис. 8): «девочки в белом, мальчики в черном».

Группа игроков выстроилась на поле по росту, но не в одну шеренгу, как на уроках физкультуры (см. выше), а в колонну с разным числом людей в каждом ряду. Каждый ряд соответствовал определенному росту: пять футов (152,4 см), пять футов и один дюйм (154,94 см) и так далее до 6 футов и 5 дюймов (195,58 см). В среде Mathcad есть функция rnorm, позволяющая генерировать случайные числа с нормальным (рис. 9), а не с равномерным (см. рис. 5) распределением. Подбирая значения первого (математическое ожидание) и второго (среднеквадратическое отклонение) аргументов этой функции, можно

* В одном из романов Агаты Кристи можно прочитать, что рост Эркюля Пуаро равнялся пяти футам и пяти дюймам. Такой же рост имел предок Форсайтов, описанный в знаменитой «Саге» Джона Голсуорси. Знание старинных и иностранных единиц измерения помогает лучше чувствовать «дух» классической литературы.

** Читаем у Тургенева: «Из числа всей ее челяди самым замечательным лицом был дворник Герасим, мужчина двенадцати вершков роста, сложенный богатырем и глухонемой от рожденья». Рост Герасима был равен двум аршинам и 12 вершкам, но аршины не указывали по умолчанию.

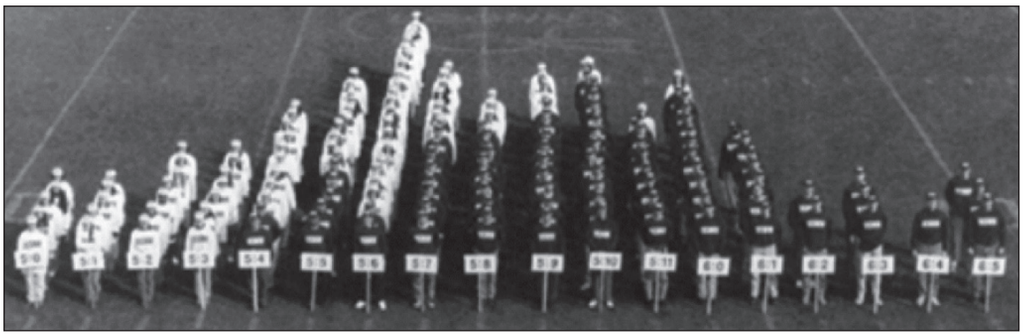


Рис. 8. «Живая» гистограмма роста человека

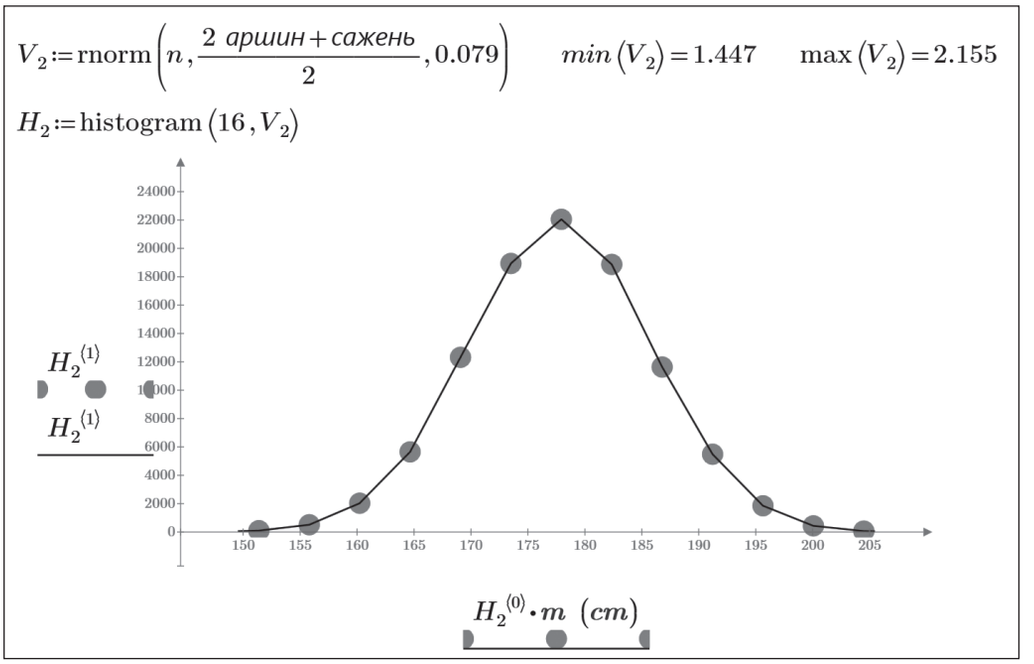


Рис. 9. Генерация вектора с нормальным распределением в среде Mathcad

добиться требуемого распределения студентов по росту. Но на сортировку характер распределения школьников, студентов, солдат по росту не влияет.

Список использованных источников

1. *Андрафанова Н. В., Попова Г. И.* Использование возможностей пакета Mathcad для генерации и проверки индивидуальных заданий // Информатика и образование. 2014. № 10.

2. *Бгатова О. В.* Применение Mathcad при обучении математике в колледже // Информатика и образование. 2007. № 11.

3. *Очков В. Ф.* Mathcad и криптография // Информатика в школе. 2013. № 10. http://twt.mpei.ac.ru/ochkov/Mathcad-15/MATHCAD-CRYPTOGRAPHY.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.* Пойти туда, зная куда // Информатика в школе. 2014. № 10. http://twt.mpei.ac.ru/ochkov/Commi-maket.pdf

6. *Очков В. Ф., Богомолова Е. П.* Интерполяция, экстраполяция, аппроксимация, или Ложь, наглая ложь и статистика // Cloud of Science. 2015. Т. 2. № 1. http://twt.mpei.ac.ru/ochkov/stat.html

7. *Очков В. Ф., Богомолова Е. П., Иванов Д. А.* Физико-математические этюды с Mathcad и Интернет: учебное пособие. СПб.: Лань, 2016. https://www.ptcusercommunity.com/groups/etudes

8. *Очков В. Ф., Чудов В. Л., Соколов А. В.* Использование форума PTC Community/Mathcad на школьных занятиях по информатике // Информатика в школе. 2015. № 10. http://twt.mpei.ac.ru/ochkov/PTC-Community-Lyc.pdf

9. *Суханов М. Б.* Применение пакета Mathcad в обучении программированию на языках высокого уровня // Информатика и образование. 2011. № 10.

10. *Черняк А. А., Черняк Ж. А., Якимович А. А.* Mathcad и Excel для школьников: решение уравнений и неравенств // Информатика и образование. 2009. № 3–7.

11. *Шамсутдинова Т. М.* Программируем в системе Mathcad // Информатика и образование. 2006. № 5.

12. *Шушкевич С. В.* Обучение построению графических объектов в Mathcad // Информатика и образование. 2009. № 5, 6.

Уважаемые авторы и читатели!

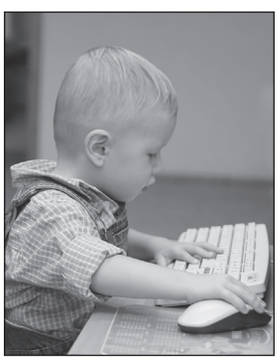
Издательство «Образование и Информатика» объявляет о проведении конкурса цифровых изображений и фотографий, которые будут размещены на обложке журнала «Информатика в школе» во втором полугодии 2017 года.

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:

Артур Норкин,
ученик гимназии № 5, г. Сергиев Посад, Московская область

Виктория Иванова,
воспитанница Дворца детского (юношеского) творчества Всеволожского района Ленинградской области, ученица средней общеобразовательной школы «Всеволожский центр образования», г. Всеволожск, Ленинградская область



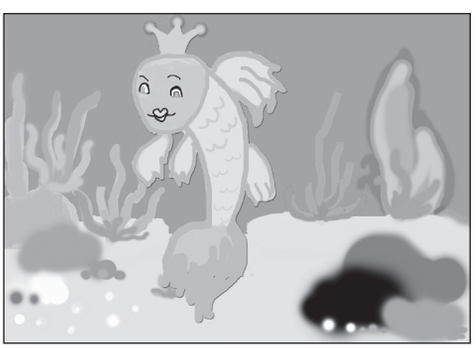
Ю. А. Фесенко,
учитель информатики гимназии № 4, г. Лыткарино, Московская область

Елизавета Громова,
ученица Наро-Фоминской средней общеобразовательной школы № 1, Московская область

Марина Цыганова,
ученица средней общеобразовательной школы № 4 имени Героя Советского Союза Д. П. Левина, г. о. Сызрань, Самарская область

Анастасия Осетрова,
ученица Наро-Фоминской средней общеобразовательной школы № 6 с углубленным изучением отдельных предметов, Московская область

Е. А. Конопко,
доцент кафедры информатики Института информационных технологий и телекоммуникаций Северо-Кавказского федерального университета, г. Ставрополь



Ксения Лысова,
воспитанница творческого объединения «Информационные технологии» Станции юных техников, ученица средней общеобразовательной школы № 49, г. Чебоксары, Чувашская Республика

Журнал «Информатика в школе»

Индексы подписки (агентство «Роспечать») на 2-е полугодие 2017 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**17** 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 **Информатика в школе** (наименование издания)

Стоимость			Количество комплектов	
	подписки	руб.		
	каталожная	руб.		
	переадресовки	руб.		

На 20**17** год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

			Город
почтовый индекс			село
			область
код улицы			Район
			улица
дом	корпус	квартира	Фамилия И.О.