

Национальный институт образования

# ИНФОРМАТИКА

ФАКУЛЬТАТИВНЫЕ ЗАНЯТИЯ



А.И. Лапо, О.Л. Сапун, Е.В. Пазюра

## ИНФОРМАТИКА В ЗАДАЧАХ

8–9 КЛАССЫ

ISBN 978-985-471-689-3



9 789854 716893

Пособие для учителей

УДК 373.3.016:004  
ББК 74.263.2  
Л24

**Лапо, А. И.**

Л24 Информатика в задачах : 8—9-е кл. : пособие для учителей учреждений общ. сред. образования с белорус. и рус. яз. обучения / А. И. Лапо, О. Л. Сапун, Е. В. Пазюра. — 2-е изд. — Минск : Адукацыя і выхаванне, 2014. — 60 с. : ил. — (Информатика. Факультативные занятия)

ISBN 978-985-471-689-3.

Предлагаемые факультативные занятия направлены на обучение учащихся методам решения задач по информатике на языке программирования. Программа ориентирована на поддержку базового курса информатики, расширяет и углубляет знания учащихся в области программирования и основ алгоритмизации.

УДК 373.3.016:004  
ББК 74.263.2

ISBN 978-985-471-689-3

© НМУ «Национальный институт образования», 2010  
© Оформление. РУП «Издательство “Адукацыя і выхаванне”», 2010



# УЧЕБНАЯ ПРОГРАММА

---

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данная программа расширяет и углубляет знания и умения учащихся в области теории построения и записи алгоритмов решения практических задач, знакомит с методологией и технологией программирования, имеет алгоритмическую направленность.

Программа имеет системно-информационный характер. Целью программы является формирование у учащихся системно-информационного взгляда на мир, включающего абстрагирование, моделирование и алгоритмическое мышление, а также навыков владения компьютером на уровне начинающего программиста. В рамках этих задач значительное внимание уделено изучению программирования и методам алгоритмизации на примерах конкретных задач из различных разделов программирования и методов алгоритмизации.

Программа факультатива дополняет и расширяет программу базового курса информатики в рамках изучения основ программирования и основ алгоритмизации.

**Основными целями** учебной программы являются:

- развитие логического и алгоритмического мышления;
- развитие системных подходов к решению задач;
- подготовка учащихся к активной полноценной жизни в условиях современного информационного общества;
- формирование у учащихся интереса к выбору профессий, связанных с программированием.

**Основными задачами** учебной программы являются:

- формирование умений использования компьютера для решения различных практических задач;
- формирование у учащихся навыков формализации задач;
- формирование знаний учащихся в области компьютерного моделирования;
- формирование у учащихся навыков структурного программирования;

- формирование у учащихся умений тестирования и отладки программ;
- развитие их интересов к компьютерным технологиям решения задач и методам обработки информации.

### **Рекомендуемые формы и методы проведения занятий**

Успех решения задач предлагаемого курса во многом зависит от организации учебного процесса. Обучение предполагает прежде всего наполнение учебного материала упражнениями и задачами различной сложности. Одни из них служат для закрепления пройденного материала, в других модифицируются рассматриваемые алгоритмы и реализации структур данных.

Особое внимание в учебном процессе должно быть уделено самостоятельной работе учащихся: самостоятельному решению заданий, проработке дополнительного учебного материала, разбору готовых решений.

Выбор базового языка программирования остаётся за учителем. Для представления рассматриваемых структур данных и алгоритмов удобен язык Pascal, подмножество которого изучается в базовом курсе информатики.

## **СОДЕРЖАНИЕ ПРОГРАММЫ**

### **VIII класс (35 часов)**

#### **1. Этапы решения задач с помощью компьютера (3 ч).**

Постановка задачи. Построение модели. Разработка алгоритма. Реализация алгоритма в виде программы. Тестирование и отладка программы. Анализ результатов.

#### **2. Интегрированная среда разработки программ (2 ч).**

Среда разработки программ на примере профессиональной IDE (FreePascal или Delphi): создание проекта, ввод и редактирование текста программы, выполнение по шагам, возможности отладки.

#### **3. Логические задачи (7 ч).**

Логические величины. Логические операции. Таблицы истинности. Управляющие структуры языка Pascal, использующие логические величины. Оператор выбора.

#### **4. Использование процедур и функций для решения задач (9 ч).**

Правила организации, описания и вызова процедур и функций, фактические и формальные параметры, глобальные и локальные переменные.

#### **5. Решение геометрических задач (9 ч).**

Представление геометрических объектов: точка, отрезок, прямая. Уравнения прямой, размещение точек плоскости относительно заданной прямой. Расстояние между точками. Расстояние от точки до прямой. Взаимное расположение отрезков. Точка пересечения прямых и отрезков.

#### **6. Творческая работа (3 ч).**

Резерв — 2 часа.

### **IX класс (35 часов)**

#### **7. Решение задач с использованием массивов (10 ч).**

Организация ввода массива данных из файла. Вывод результатов в файл. Упорядочение элементов массива. Двоичный поиск. Оценка сложности алгоритмов сортировки.

Двумерные массивы. Обработка строк и столбцов таблицы: перестановка строк и столбцов. Заполнение массива по заданному правилу.

### **8. Задачи целочисленной арифметики (10 ч).**

Выделение цифр числа. Поиск делителей числа, простые числа, разложение на простые множители, поиск наибольшего общего делителя и наименьшего общего кратного натуральных чисел, операции с дробями, сокращение дробей.

### **9. Системы счисления (10 ч).**

Позиционные и непозиционные системы счисления. Базис, алфавит, основание системы счисления. Представление целых чисел и арифметические операции в позиционных системах счисления. Перевод целых чисел из одной позиционной системы счисления в другую. Алгоритмы перевода целых чисел.

### **10. Творческая работа (3 ч).**

Резерв — 2 часа.

## **Ожидаемые результаты**

После успешного прохождения учебного материала *учащиеся получают представление:*

- о разнообразии задач, решаемых с использованием языка программирования;
- методах решения задач;
- разработке алгоритмов решения задач.

*Изучение данного курса предполагает:*

- развитие познавательных способностей школьников;
- формирование у них алгоритмического мышления;
- получение реального опыта творческой и исследовательской деятельности;
- повышение интереса учащихся к программированию.

## **Список литературы**

1. *Бентли, Д.* Жемчужины программирования / Д. Бентли. — СПб. : Питер, 2002. — 272 с.

2. *Златопольский, Д. М.* Сборник задач по программированию / Д. М. Златопольский. — СПб. : БХВ-Петербург, 2007. — 240 с.

3. *Кормэн, Т. Х.* Алгоритмы: построение и анализ / Т. Х. Кормэн, Ч. И. Лейзерсон, Р. Л. Ривест. — М. : МЦНМО, 2000. — 960 с.

4. *Котов, В. М.* Информатика: методы алгоритмизации : 10—11 классы / В. М. Котов, О. И. Мельников. — Минск : Народная асвета, 2000. — 221 с.

5. *Котов, В. М.* Информатика: методы алгоритмизации / В. М. Котов, И. А. Волков, А. И. Лапо. — Минск : Народная асвета, 2000. — 300 с.

6. *Котов, В. М.* Структуры данных и алгоритмы: теория и практика / В. М. Котов, Е. П. Соболевская. — Минск : БГУ, 2004. — 255 с.

7. *Меньшиков, Ф. В.* Олимпиадные задачи по программированию. — СПб. : Питер, 2006. — 315 с.

8. *Окулов, С. М.* Программирование в алгоритмах / С. М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2002. — 341 с.

9. *Окулов, С. М.* Основы программирования / С. М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2005. — 440 с.

10. *Радион, В. С.* Олимпиады по информатике. Задачи. Решения. Тесты / В. С. Радион. — Минск : Аверсэв, 2007. — 366 с.

11. *Лапо, А. И.* Дидактические материалы по основам и методам алгоритмизации 8—9 кл. / А. И. Лапо. — Минск : Беларуская энцыклапедыя, 2000. — 96 с.

# ПОСОБИЕ ДЛЯ УЧИТЕЛЕЙ

---

## ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ С ПОМОЩЬЮ КОМПЬЮТЕРА

### Цели обучения:

- формирование умений записывать решение задачи в виде алгоритма;
- формирование умений реализации алгоритма в виде программы;
- формирование умений тестировать и отлаживать программу.

### Цели развития:

- развитие пооперационного мышления учащихся;
- обучение учащихся самостоятельной работе с учебным материалом факультатива.

*Учащиеся должны знать* основные этапы решения задачи.

*Учащиеся должны уметь:*

- построить математическую модель;
- разработать алгоритм;
- составить программу на языке Pascal;
- реализовать программу на ПЭВМ;
- проанализировать полученные результаты.

**Программное и методическое обеспечение урока:** среда программирования с языком Pascal, дидактические материалы к факультативу.

### Методические рекомендации по теме (1—3-й уроки)

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (33 вопроса). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает завершить предложения:

Алгоритм — это \_\_\_\_\_ .

Программа — это \_\_\_\_\_ .



Линейный алгоритм — это алгоритм, в котором ... \_\_\_\_ .  
 Перечислите основные способы записи алгоритмов: \_\_\_\_ .  
 Тестирование — это \_\_\_\_ .  
 Отладка — это \_\_\_\_ .

3. Предлагается правильно указать цифровую последовательность решения задачи на ПЭВМ:

Построение математической модели	1
Постановка задачи	2
Составление программы	3
Разработка алгоритма	4
Реализация программы на компьютере	5
Анализ результатов	6

4. Выбрать правильное соответствие определениям:

<b>Величина</b>	Величина, имеющая значение определённого типа и меняющаяся в процессе выполнения программы
<b>Константа</b>	Элемент данных, который имеет имя и значение
<b>Переменная</b>	Величина, значение которой не изменяется в ходе выполнения программы

5. Выбрать правильные идентификаторы:

**Label2, Block\_Calcul, A1, Stop2, Do\_Goto\_Output, 1\_th-Block, lab+1, 1234567, Лето, Блок 56, R.89, W..Z. SDF, аaaa, аSDF, q12345678.**

6. Какие из перечисленных слов являются названиями типов данных в языке Pascal:

**Byte, Number, Integer, Real, Const?**

7. Какие операторы присваивания будут правильными, если переменные описаны следующим образом: a, b, C1, S2, X, Z, R: integen, Cena, S1, C2, Y: real?

- а) X:= a + b;
- б) Cena:= 12000 div 5;
- в) S1:= S2 + C1;
- г) S2:= C2;
- д) Y:= 1/(x-a) + 2/(x+b);

е)  $Z := \sin(2\pi x) + \cos(\pi/4 - 3x)$ ;

ж)  $Cena := 12000/5$ ;

з)  $C2 := S2$ ;

и)  $Z+1 := Z$ ;

к)  $R := \sqrt{x-a} + \sqrt{y-b}$ .

8. Записать на языке программирования следующие выражения:

а)  $2 + 3|x| - 7|1+x|$ ;

б)  $(1-x)(2 - \frac{1}{x})(3 - \frac{2}{x})$ ;

в)  $x^4 + 2x^3 - 3x^2 + 7,5x - 8,999$ ;

г)  $\frac{a+b}{c+d} - 45,4 + \frac{3x+a}{5,8(2b-a)}$ ;

д)  $\frac{a + 2,3x + 8a}{23,9 - ax^2} - \frac{23}{39 - x\sqrt{2}}$ .

9. В выражении на языке программирования  $1/2 + 3/4 + 5/6 + 7/8$  расставить скобки так, чтобы оно соответствовало следующему выражению в традиционной математической форме:

$$\frac{1}{2 + \frac{3}{4 + \frac{5}{6 + \frac{7}{8}}}}$$

10. Написать по правилам языка программирования выражение, значением которого является:

а) периметр квадрата, площадь которого равна  $a$ ;

б) площадь равностороннего треугольника с периметром  $p$ ;

в) сумма цифр натурального трёхзначного числа  $N$ .

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

1. Предложить учащимся выполнить задание 1 из факультатива: набрать программу, выполнить её несколько раз, подставляя различные числа, отладить и проанализировать полученные результаты вариантов а) и б).

2. Предложить учащимся выполнить задание 2 из факультатива, используя операции *div* и *mod*.

3. Написать программу для решения геометрической задачи, используя геометрические формулы (задание 3).

4. Написать программу для решения задачи (задания 4—5). При решении воспользоваться операциями *div* и *mod*.

5. Не пользуясь никакими другими арифметическими действиями, кроме умножения, написать программу решения задания 6.

### **Уровень 3. Творческий уровень.**

Предложить учащимся выполнить индивидуальные задания 1—3 по следующему сценарию:

- разработать алгоритм решения задания;
- написать программу на языке программирования;
- провести тестирование программы на наличие ошибок;
- проверить правильность решения с различными числовыми данными.

## **ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ ПРОГРАММ**

### **Цели обучения:**

- формирование умений работать в среде программирования;
- формирование умений ввода и редактирования программы;
- формирование умений отладки и тестирования программы.

### **Цели развития:**

- развитие пооперационного мышления учащихся;
- обучение учащихся самостоятельной работе с учебным материалом факультатива.

*Учащиеся должны знать* основные этапы решения задачи.

*Учащиеся должны уметь:*

- разработать алгоритм решения задачи;
- составить программу на языке Pascal;
- реализовать программу на ПЭВМ;
- проанализировать полученные результаты.

**Программное и методическое обеспечение урока:** среда программирования с языком Pascal, дидактические материалы к факультативу.

## Методические рекомендации по теме (уроки 4—5)

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

Изучение языка программирования логично начинать с рассмотрения простейших программ, чтобы на этих примерах изучить основные конструкции языка.

1. Учитель может задать следующие вопросы на повторение материала:

- Какими словами начинается и заканчивается тело любой программы?

- Какова структура программы?

- Как загрузить среду языка программирования Pascal?

- Как запустить программу на выполнение?

- Что называется переменной?

- Что значит описать переменные?

- Какие графические команды вы знаете?

- Как задаются координаты точки на экране?

- Из каких геометрических фигур можно создать изображение, используя графические возможности PascalABC?

2. Учитель предлагает завершить предложения:

Оператор — это \_\_\_\_\_ .

Простой оператор — это \_\_\_\_\_ .

Составной оператор — это \_\_\_\_\_ .

Приведите пример заголовка программы: \_\_\_\_\_ .

Приведите пример тела программы: \_\_\_\_\_ .

3. Являются ли следующие строки идентичными для компилятора языка Pascal?

**program** Hello;

**Program** hello;

**PROGRAM** HELLO.

4. Дана программа, написанная на языке программирования:

```
program Sum2;
```

```
var a, b, S: integer;
```

```
begin
```

```
  writeln('Введите два целых числа: ');
```

```
  readln(a, b);
```

```
  S:= a + b;
```

```
  writeln('Сумма этих чисел равна ', S);
```

```
End.
```

а) Какие символы будут выведены на печать в результате выполнения программы, если  $a=73$ ,  $b=17$ ?

б) Назовите имена переменных.

в) Какого типа названные переменные?

5. Написать определение понятиям:

а) поиск синтаксических ошибок, допущенных программистом при написании программы, осуществляет \_\_\_\_\_ ;

б) преобразование исходного текста (текст на языке программирования) в последовательность команд, закодированных в машинных кодах, производит \_\_\_\_\_ ;

в) программа, которая преобразует исходный текст в последовательность машинных команд, называется \_\_\_\_\_ .

6. Написать, что позволяют выполнить команды меню среды PascalABC:

Файл	
Правка	
Вид	
Программа	
Сервис	
Помощь	

7. Показать учителю на экране:

- строку меню;
- рабочую область;
- строку состояния.

8. Написать, какие операции выполняют с помощью сочетания клавиш:

<i>CTRL+Z</i>	
<i>CTRL+X</i>	
<i>CTRL+C</i>	
<i>CTRL+V</i>	
<i>CTRL+F</i>	



9. Выбрать правильное расширение файла программы, написанной на языке программирования Pascal в среде PascalABC:

- а) doc;
- б) pas;
- в) xls;
- г) ppt.

10. Выбрать правильное расширение файла программы, написанной на языке программирования Pascal в среде Delphi:

- а) doc;
- б) pas;
- в) dpr;
- г) ppt.

11. Что следует отнести к синтаксическим ошибкам:

- а) неправильная запись команды;
- б) пропуск апострофов или точек с запятой;
- в) неправильный алгоритм решения задачи;
- г) использование неподходящей команды?

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

1. Предложить учащимся набрать программу из факультатива (задание 2).

Что конкретно будет выведено на печать после выполнения данной программы при  $a=10$ ; 78; 99?

Выполнить эту программу в режиме отладки.

2. Предложить учащимся набрать следующую программу:

```
Program ABC;
```

```
Var a: integer;
```

```
Begin
```

```
  a:=5;
```

```
  Writeln('введите значение a = ');
```

```
  a:=2*a;
```

```
  a:=a+1;
```

```
  Write('искомое число = ', a);
```

```
End.
```

Что будет выведено на печать после выполнения данной программы?

Выполнить программу в режиме отладки. Понаблюдать за изменением значения переменной *a*.

3. Предложить учащимся набрать следующую программу:

```
Program Cel;  
Var a, b, c, d: integer;  
Begin  
  a:=13;  
  b:=5;  
  c:=a div b;  
  Writeln('div= ', c);  
  d:= a mod b;  
  Writeln('mod= ', d);  
End.
```

Что будет выведено на печать после выполнения данной программы?

Выполнить программу в режиме отладки. Понаблюдать за изменением значений переменных.

4. Что будет выведено на экран в результате выполнения следующей программы:

```
Program Graf;  
Uses GraphAbc;  
Begin  
  SetPenWidth (10);  
  SetPenColor (clRed);  
  Line (100, 100, 270, 90);  
End.
```

Объяснить команды, которые выводят графику на экран.

Выполнить программу в режиме отладки. Понаблюдать за последовательным выводом графических примитивов.

5. Используя команду *line*, нарисовать на экране фигуры задания 7 факультатива.

### **Уровень 3. Творческий уровень.**

1. Предложить учащимся выполнить индивидуальные задания 4—7 по следующему сценарию:

- разработать алгоритм решения задания;

- написать программу на языке программирования;
- провести тестирование программы на наличие ошибок;
- проверить правильность решения с различными числовыми данными.

2. Используя графические возможности языка программирования Pascal, нарисовать следующие рисунки:



## ЛОГИЧЕСКИЕ ЗАДАЧИ

### Цели обучения:

- формирование знаний учащихся об логических величинах, операциях и выражениях;
- формирование умений использовать составные условия в операторах ветвления и цикла.

### Цели развития:

- развитие логического и пооперационного мышления учащихся;
- развитие познавательных интересов учащихся.

### Учащиеся должны знать:

- правила описания и использования логических переменных;
- логические операции, используемые в языке Pascal;
- правила записи и использования оператора выбора.

### Учащиеся должны уметь:

- составлять логические выражения;
- использовать логические операции для создания составных условий;
- использовать составные условия в операторах ветвления и цикла;
- использовать оператор выбора.

**Программное и методическое обеспечение урока:** среда программирования с языком Pascal, дидактические материалы к факультативу.

## Методические рекомендации по теме (уроки 6—12)

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (8 вопросов). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает завершить предложения:

Логическая переменная в языке Pascal описывается ключевым словом \_\_\_\_\_.

Результатом операции AND является истинное выражение «тогда и только тогда, когда» \_\_\_\_\_.

Результатом операции OR, является ложное выражение «тогда и только тогда, когда» \_\_\_\_\_.

Результатом операции XOR, является истинное выражение «тогда и только тогда, когда» \_\_\_\_\_.

Операция NOT используется для \_\_\_\_\_.

Составное условие образуется из \_\_\_\_\_.

3. Предлагается правильно указать цифровую последовательность, соответствующую приоритету выполнения логических операций в языке Pascal:

AND	1
OR, XOR	2
NOT	3

4. Выбрать правильное соответствие определениям:

Логическая операция AND для двух выражений	меняет значение логического выражения на противоположное: истинное на ложное, а ложное на истинное
Логическая операция NOT	возвращает значение TRUE в том и только том случае, когда оба выражения истинны
Логическая операция XOR для двух выражений	возвращает значение FALSE в том и только в том случае, когда оба выражения ложны
Логическая операция OR для двух выражений	возвращает значение TRUE в том и только в том случае, когда оба выражения имеют разные значения (одно — TRUE, другое — FALSE)

5. Определить истинность логического выражения:

а)  $3 < 7$ ;

б)  $2+2*2=8$ ;

в)  $\text{abs}(-5) > \text{abs}(3)$ ;

г)  $y \geq \text{sqr}(x)$ , если  $x=2$  и  $y=10$ ;

д)  $y \geq \text{sqr}(x)$ , если  $x=10$  и  $y=2$ ;

е)  $(45 \bmod 3=0)$  and  $(45 \bmod 5=0)$ ;

ж)  $(45 \bmod 2=0)$  or  $(45 \bmod 5=0)$ ;

з)  $(45 \bmod 3=0)$  xor  $(45 \bmod 5=0)$ ;

и)  $\text{Not}(x > y)$ , если  $x=2$  и  $y=10$ ;

к)  $\text{Not}(x > y)$ , если  $x=10$  и  $y=2$ .

6. Какое значение будет иметь переменная  $t$  после выполнения программы:

```
program zadach;  
var x, y: real;  
    t: boolean;  
begin  
    read(x);  
    t:=x<trunc(x);  
    read(y);  
    t:=t or (y<trunc(y));  
    writeln(t);  
End.
```

а) если  $x=1.5$ ,  $y=-0.8$ ;

б) если  $x=-1.5$ ,  $y=-0.8$ ;

в) если  $x=-1.5$ ,  $y=0.8$ ;

г) если  $x=1.5$ ,  $y=0.8$ ?

7. Указать ошибки в записи операторов:

a.	if $x > y$ then $x:=0$ and $y:=0$ ; else write(z);	b.	if $a:=b$ or $b:=c$ then $b:=b*2$ ; $a:=b+1$ else 100;
c.	if $x > y > z$ then $a:=a+1$ ; else $b:=b+1$ ; write(b);	d.	$x:=1$ ; $y:=1$ ; while $x < 100$ and $y < 100$ do ( $x:=x+1$ ) and ( $y:=y+2$ );



**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Урок 6. Предложить учащимся выполнить задание 1.

Уроки 7—8. Предложить учащимся выполнить задания 3, 4, 6—8.

Урок 9. Предложить учащимся выполнить задания 9, 10.

Уроки 10—11. Предложить учащимся выполнить задания 2, 5, 11.

Урок 12. Предложить учащимся выполнить задание 12.

**Уровень 3. Творческий уровень.**

1. Предложить учащимся изобразить в графическом режиме решение задания 6. Точки, попадающие внутрь области, рисовать зелёным цветом, а точки снаружи — красным цветом.

2. Предложить учащимся изобразить в графическом режиме решение задания 8. Рисовать фигуру из условия задачи в том случае, если она существует.

## **ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ**

**Цели обучения:**

- формирование умений по разбиению алгоритма решения задачи на отдельные составляющие: основной и вспомогательные алгоритмы;

- формирование умений по использованию механизма построения процедур и функций для решения поставленных задач.

**Цели развития:**

- развитие процедурного стиля мышления учащихся;
- обучение учащихся самостоятельной работе с учебным материалом факультатива

**Учащиеся должны знать:**

- формат и порядок описания и использования процедур и функций;

- понятия глобальных и локальных переменных;
- понятия фактических и формальных параметров;
- различия между функциями и процедурами.

**Учащиеся должны уметь:**

- оформлять вспомогательные алгоритмы (подпрограммы) с помощью процедур и функций;
- реализовывать алгоритмы с использованием подпрограмм.

**Программное и методическое обеспечение урока:** среда программирования с языком Pascal, дидактические материалы к факультативу.

### Методические рекомендации по теме (уроки 13—21)

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (14 вопросов). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает завершить предложения:

Описание процедуры — это \_\_\_\_\_ .

Вызов процедуры — это \_\_\_\_\_ .

В заголовке функции на языке Pascal последним указывается... \_\_\_\_\_ .

Область действия локальной переменной: \_\_\_\_\_ .

Область действия глобальной переменной: \_\_\_\_\_ .

Фактический параметр подменяет... \_\_\_\_\_ .

3. Предлагается правильно указать цифровую последовательность элементов заголовка функции:

Ключевое слово <i>Function</i>	1
Список формальных параметров с их типами	2
Имя функции	3
Тип результирующего значения функции	4

4. Выбрать правильное соответствие определениям:

<b>Локальные переменная</b>	Область действия распространяется на все вложенные процедуры и функции, если в них не описана переменная с тем же именем
<b>Глобальная переменная</b>	Область действия заканчивается с окончанием процедуры (функции)
<b>Ключевое слово <i>procedure</i></b>	Используется при описании результирующих формальных параметров
<b>Ключевое слово <i>Var</i></b>	Используется первым при описании процедуры

5. Записать определение функции для вычисления:

- а) периметра квадрата, площадь которого равна  $a$ ;
- б) площади равностороннего треугольника с периметром  $p$ ;
- в) суммы цифр натурального трёхзначного числа  $N$ .

6. Указать ошибки в записи заголовков функций:

- а) Function (a,b:integer):integer;
- б) Function sum(a,b:integer);
- в) Function f(x:double;):Boolean;

7. Оформить процедурой вспомогательный алгоритм сравнения действительных чисел, описанный с помощью следующей функции:

```
Function comp1(a,b:Real): integer;
Begin
    If abs(a-b)<= 0.00001 then comp1:=0
    Else
        If a>b then comp1:=1
        Else comp1:= -1;
```

End;

Записать пример вызова функции и процедуры.

8. Оформить функцией вспомогательный алгоритм нахождения суммы квадратов двух действительных чисел, описанный с помощью следующей процедуры:

```
Procedure sum1(a,b:Real; var s:Real);
Begin
    S:= a*a+b*b;
```

End;

Записать пример вызова функции и процедуры.

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Урок 13. Предложить учащимся оформить разработанные ранее алгоритмы задания 3 темы «Логические задачи» в виде функции, применить функцию для параметра, изменяющегося от  $-10$  до  $10$  с интервалом  $0.5$ , вывести результаты в виде двух столбцов:  $x$  и  $f$ .

Уроки 14—15. Предложить учащимся выполнить задания 1 и 2, оформив вспомогательные алгоритмы в виде функций.

Урок 16. Предложить учащимся выполнить задания 1 и 2, оформив вспомогательные алгоритмы в виде процедур.

Уроки 17—18. Предложить учащимся выполнить задание 3, выбрав самостоятельно набор глобальных и локальных переменных.

Уроки 19—21. Предложить учащимся самостоятельно выполнить задания 4 и 5, используя математические формулы, приведённые в приложении.

### **Уровень 3. Творческий уровень.**

1. Вывести формулу для вычисления площади треугольника по заданным координатам вершин (все координаты — положительные числа), проведя перпендикуляры из вершин на ось абсцисс и рассмотрев площади образовавшихся трёх трапеций. Оформить соответствующую функцию.

2. Обобщить решение предыдущей задачи для треугольника, координаты вершин которого являются не обязательно положительными числами.

## **РЕШЕНИЕ ГЕОМЕТРИЧЕСКИХ ЗАДАЧ**

### **Цели обучения:**

- формирование умений учащихся представлять простейшие геометрические фигуры (точка, отрезок, прямая);
- формирование умений решать геометрические задачи с помощью компьютера.

### Цели развития:

- развитие логического и пооперационного мышления учащихся;

- развитие познавательных интересов учащихся.

#### *Учащиеся должны знать:*

- способы представления точки, отрезка и прямой.

#### *Учащиеся должны уметь:*

- записывать уравнение прямой различными способами;
- определять местоположение точек относительно прямой;
- определять взаимное расположение отрезков;
- вычислять расстояние между точками и от точки до прямой.

**Программное и методическое обеспечение урока:** система программирования с языком Pascal, дидактические материалы к факультативу.

### Методические рекомендации по теме (уроки 22—30)

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (16 вопросов). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает завершить предложения:

Для того, чтобы задать точку в языке Pascal, нужно \_\_\_\_\_ .

Для того что бы задать отрезок в языке Pascal, нужно \_\_\_\_ .

Прямая в языке Pascal может быть задана путём указания \_\_\_\_\_ .

Для определения принадлежности точки прямой нужно \_\_\_\_\_ .

Две прямые могут совпадать, быть параллельными и \_\_\_\_\_ .

Расстоянием от точки до прямой является \_\_\_\_\_ .

3. Выбрать правильное соответствие:

$y = kx + b$	Общее уравнение прямой
$Ax + By + c = 0$	Уравнение прямой с угловым коэффициентом
$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$	Уравнение прямой с угловым коэффициентом, которая проходит через заданную точку
$y - y_0 = k(x - x_0)$	Уравнение прямой, проходящей через две точки



4. Для двух прямых, расположенных на плоскости, записать:

а) условие параллельности, если прямые заданы общими уравнениями;

б) условие параллельности, если прямые заданы уравнениями с угловым коэффициентом;

в) условие пересечения, если прямые заданы уравнениями с угловым коэффициентом;

г) условие пересечения, если прямые заданы общими уравнениями.

5. Определить расстояния:

а) между точками (4; 5) и (2; 9);

б) между точками (-5; 3) и (3; -2);

в) между точками (6; 4) и (-7; -1);

г) от точки (2; 5) до прямой;

д) от точки (2; 5) до прямой;

е) от точки (2; 5) до прямой, проходящей через точки (-3; 4) и (7; -5);

6. Определить, что делает данная программа, и дописать вместо многоточия необходимые слова.

а) begin

```
writeln('введи коэффиц. общ. уравнения прямой');
```

```
readln(A, B, C);
```

```
if (A<>0) and (B=0) then writeln('...x');
```

```
if (A=0) and (B<>0) then writeln('...y');
```

```
End.
```

б) begin

```
writeln('введи коэфф. общ. уравнения прямой');
```

```
readln(A, B, C);
```

```
if (A=0) and (C=0) then writeln('...x');
```

```
if (B=0) and (C=0) then writeln('...y');
```

```
End.
```

в) begin

```
writeln('введи коэфф. общ. уравнения прямой');
```

```
readln(A, B, C);
```

```
if (A=0) and (B<>0) and (C<>0) then writeln('...x');
```

```
if (A<>0) and (B=0) and (C<>0) then writeln('...y');
```

```
End.
```

```

d)begin
  writeln('введи коэффиц. ур-ния линейной функции');
  readln(k, b);
  if k=0 then
    if b=0 then writeln('...')
    else writeln('...')
  else writeln('...');
End.

```

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Урок 22. Предложить учащимся графически изобразить точки и отрезки по заданным координатам.

Уроки 23—24. Предложить учащимся выполнить задания 1—3.

Урок 25. Предложить учащимся выполнить задания 4—5.

Урок 26. Предложить учащимся выполнить задания 6—7.

Уроки 27—28. Предложить учащимся выполнить задания 8—9.

Уроки 29—30. Предложить учащимся выполнить задание 10.

**Уровень 3. Творческий уровень.**

Предложить учащимся решение олимпиадных задач.

#### *Задача 1. Треугольник и точка*

В декартовой системе координат на плоскости заданы координаты вершин треугольника и ещё одной точки. Определить, принадлежит ли эта точка треугольнику.

**Ограничения:** координаты вершин — целые числа, для любой точки выполняются следующие условия:  $-10\,000 \leq x, y \leq 10\,000$ , время 1 с.

**Ввод.** В четырёх строках находятся пары чисел — координаты точек. Числа в первых трёх строках — это координаты вершин треугольника, в четвёртой строке — координаты тестируемой точки.

**Вывод.** Вывести слово *In*, если точка находится внутри треугольника, или *Out* — если снаружи.

*Примеры.*

Ввод 1	Ввод 2	Ввод 3	Ввод 4
0 0	0 0	0 0	0 0
100 0	100 0	100 0	100 0
0 100	0 100	0 100	0 100
100 100	10 10	50 50	0 0
Вывод 1	Вывод 2	Вывод 3	Вывод 4
Out	In	In	In

*Задача 2. Стена*

Путник, перемещаясь из пункта А в пункт В, всегда видит перед собой часть пути и может заблаговременно решить, можно ли двигаться по прямой или придётся огибать какое-то препятствие. Компьютер, управляющий движением какого-либо объекта, тоже должен уметь «видеть», т. е. анализировать данные о расположении возможных препятствий и положении начальной и конечной точек маршрута.

Итак, на плоскости своими целочисленными координатами заданы четыре точки  $A, B, C, D$ , не лежащие на одной прямой. Отрезок  $CD$  — «стена», через которую невозможно ни перелезть, ни перепрыгнуть. Требуется написать программу, проверяющую, можно ли, перемещаясь из точки  $A$  в точку  $B$ , двигаться по прямой или придётся обходить стену. Стена не может проходить ни через начальную, ни через конечную точки.

Ввод состоит из 8 целых чисел:  $xa ya xb yb xc yc xd yd$  — целочисленных координат четырёх точек  $A, B, C, D$ .

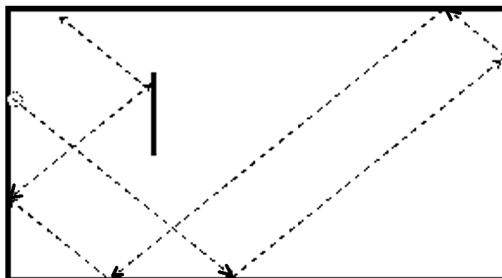
Программа должна вывести сообщение: «Надо обходить стену» или «Можно идти по прямой».

*Примеры.*

Ввод 1	Ввод 2
2 2 0 0 -1 1 1 1	2 2 0 0 -1 1 0 1
Вывод 1	Вывод 2
Надо обходить стену	Можно идти по прямой

## ТВОРЧЕСКАЯ РАБОТА

В качестве творческой работы можно предложить учащимся реализовать движение мячика в прямоугольной области. Можно добавить дополнительные стенки.



При реализации этого проекта учащимся понадобятся как знания графического режима, так и знания, полученные в процессе изучения тем факультатива.

## РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ МАССИВОВ

### Цели обучения:

- формирование представления о массиве, основных свойствах массива и действиях над элементами массива;
- формирование знаний об упорядочивании элементов массива, о различных способах сортировки и оценке сложности сортировки;
- формирование умений обработки строк и столбцов, перестановки элементов в двумерном массиве.

### Цели развития:

- развитие умений работать за ограниченный промежуток времени;
- развитие навыков самостоятельной работы;
- развитие навыков работы с учебным материалом факультатива;
- повышение мотивации к учебной деятельности.

*Учащиеся должны знать:* методы сортировки массивов, правила организации двоичного поиска.

**Учащиеся должны уметь:**

- осуществлять ввод и вывод элементов массива;
- сортировать элементы массива;
- осуществлять двоичный поиск;
- переставлять строки и столбцы двумерного массива;
- заполнять массив по заданному образцу.

**Программное и методическое обеспечение урока:** система программирования с языком Pascal, дидактические материалы к факультативу.

### **Методические рекомендации по теме (уроки 1—10)**

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (11 вопросов). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает ответить на следующие вопросы теста:

а) Какое ключевое слово используется для объявления массива в программе на языке Pascal:

- 1) RUN;
- 2) DIM;
- 3) LET;
- 4) ARRAY;
- 5) REM?

б) Числой одномерный массив A заполнен последовательно числами 2, 5, 7, 34. Каково будет значение элемента A(3):

- 1) 34;
- 2) 5;
- 3) 7;
- 4) 2?

в) Доступ к элементу массива осуществляется по его...

- 1) адресу;
- 2) имени;
- 3) номеру;
- 4) значению.



3. Учитель предлагает фрагменты программ и работу над ними:

а) Указать правильное описание массива:

- 1) Var a:array[1..1000] of integer;
- 2) Var A, B,C: ARRAY [L. 50] OF REAL;
- 3) Var A: ARRAY [1..50 OF REAL];

б) Что делает следующий фрагмент программы:

```
for i:=1 to n do
```

```
begin
```

```
  write('a[',i,'] ');
```

```
  readln(a[i]);
```

```
end;
```

- 1) ввод элементов массива с клавиатуры;
- 2) вывод элементов массива на экран монитора;
- 3) ввод элементов массива из файла?

в) Что делает следующий фрагмент программы:

```
randomize;
```

```
write( 'число элементов ');
```

```
readln(n);
```

```
for i:=1 to n do
```

```
begin
```

```
  a[i]:=random(45)-22;
```

```
  write(a[i], ' ');
```

```
end;
```

1) заполняет массив случайными числами и не выводит их на печать;

2) заполняет массив случайными числами и выводит их на печать;

3) заполняет массив одинаковыми числами и выводит их на печать?

г) Что делает следующий фрагмент программы:

```
imax:=1; max:=a[1];
```

```
for i:=1 to n do
```

```
  if a[i]>= max then
```

```
  begin
```

```
    max:=a[i];
```

```
    imax:=i;
```

```
  end;
```

```
writeln(max, ' номер ', imax);
```

- 1) находит максимальный элемент и печатает на экране;
- 2) находит максимальный элемент, его индекс и печатает их на экране;
- 3) находит индекс максимального элемента и печатает на экране?

е) Что делает следующий фрагмент программы:

```
k:=0;
for i:=1 to n do
  if a[i] mod 2=0 then k:=k+1;
writeln('k=', k);
```

- 1) находит максимальный элемент и печатает на экране;
- 2) находит чётный элемент и печатает его на экране;
- 3) находит число чётных элементов и печатает их количество на экране?

4. Предлагается заполнить одномерный массив элементами, отвечающими следующему фрагменту программы, если  $n=10$ :

```
Read(n); {Ввод количества элементов}
a[1]:= 1;
a[2]:= 1;
for i :=3 to n do
  a[i] := a[i - 1] + a[i - 2];
```

5. Рассмотреть задание 4 из факультатива и предложить учащимся на основании предложенного материала выполнить следующее задание. Что выполняет данный фрагмент программы?

```
Begin
  L:=1; R:=N; P:=false;
  While (L<=R) and (P=false) do
    begin
      m:=(L+R) div 2;
      if a[m]=y then
        p:=true
      else
        if a[m]<y then
          L:=m+1
        else
          R:=m-1;
    end;
  poisk:=m;
End;
```

6\*. Как описать массив, предложенный ниже?

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

7\*. Квадратная матрица, у которой все элементы, исключая элементы главной диагонали, равны нулю, называется \_\_\_\_\_

---

$$A = \begin{pmatrix} 100 \\ 050 \\ 009 \end{pmatrix}$$

8\*. Диагональная матрица, у которой все элементы, стоящие на главной диагонали, равны 1, называется \_\_\_\_\_

---

$$A = \begin{pmatrix} 100 \\ 010 \\ 001 \end{pmatrix}$$

9\*. Если в матрице  $A$  ( $m, n$ ) поменять местами строки и столбцы, то получится матрица  $A^t$  ( $m, n$ ), которая называется \_\_\_\_\_

---

$$A = \begin{pmatrix} a_{11} a_{12} \dots a_{1n} \\ a_{21} a_{22} \dots a_{2n} \\ \dots \\ a_{m1} a_{m2} \dots a_{mn} \end{pmatrix} \quad A^t = \begin{pmatrix} a_{11} a_{21} \dots a_{m1} \\ a_{12} a_{22} \dots a_{m2} \\ \dots \\ a_{1n} a_{2n} \dots a_{mn} \end{pmatrix}$$

10. Исправить неточности в описании двумерного массива:

Var A= array[1..n,1..m] of integer;

**11. Найти ошибки в программе:**

```
Program SR;  
  var I,S,K:integer;  
  B : aray[1...5; 1..5] of integer;  
Begin  
  S:=0; K:=0;  
  For I =1 to N do  
    A[I] = random(10);  
  For I = 1 to 5 do  
  Begin  
    For J = 1 to 5 do  
      Writeln (A [I,J]:3);  
    Writeln;  
  end;  
for I = 1 to 5 do;  
  for J = 1 to 5 do;  
    S:=S+A[I]; K:=K+1;  
    SR:=S/K;  
    Writeln('ср.арифм. = ', SR);  
end.
```

**12. Указать, что будет напечатано в результате выполнения следующей программы:**

```
Var a:array[1..3, 1..3] of integer;  
  i,j: integer;  
Begin  
  For i:=1 to 3 do  
    For j:=3 downto 1 do  
      If i > j then  
        a[i,j]:=10 * i + j;  
      else  
        a[i,j]:=10 * i - j;  
  for i:=1 to 3 do  
  begin  
    For j:=1 to 3 do  
      write (a[i,j], ' ')  
    writeln (' ');  
  end;  
End.
```

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Уроки 1—2. Предложить учащимся выполнить задания 1, 2.

Уроки 3—4. Предложить учащимся выполнить задание 3.

Урок 5. Предложить учащимся выполнить задание 4.

Урок 6. Предложить учащимся выполнить задание 3. Протестировать программы, написанные на уроках 3—4, для различных исходных данных. Оценить зависимость времени работы от алгоритма и исходных данных.

Урок 7. Предложить учащимся выполнить задания 5, 6.

Урок 8. Предложить учащимся выполнить задания 7, 8.

Уроки 9—10. Предложить учащимся выполнить задания 9, 10.

#### **Дополнительные задания.**

1. Ввести и отладить программу для подсчёта различных чисел в одномерном массиве.  $N = 10$ . Элементы массива — 1, 2, 2, 2, -1, 1, 0, 34, 3, 3. Ответ: \_\_\_\_\_ .

```
Program Razlichnye_Elementy;
```

```
Var I, N, K, Kol : Integer;
```

```
    A : Array [1..50] Of Integer;
```

```
    Lo : Array [1..50] Of Boolean;
```

```
Begin
```

```
    Write('Введите количество элементов массива: ');
```

```
    ReadLn(N);
```

```
    FOR I := 1 TO N DO
```

```
        Begin
```

```
            Write('A[', I, '='); ReadLn (A[I]);
```

```
            {Заполняем вспомогательный массив значениями True}
```

```
            Lo[I] := True;
```

```
        End;
```

```
        {переменная, в которой будет храниться  
        количество различных чисел}
```

```
        Kol := 0;
```

```
    FOR I := 1 TO N DO
```

```
        IF Lo[I] THEN
```

```

Begin
    Kol:= Kol + 1;
    FOR K := I TO N DO
        {Во вспомогательный массив заносим значение False,
        если число уже встречалось ранее или совпадает с
        текущим элементом A[I]}
        Lo[K] := (A[K] <> A[I]) And Lo[K];
    end;
    WriteLn('Количество различных чисел: ', Kol)
End.

```

**2. Ввести и отладить программу, которая определяет среднее арифметическое элементов и количество положительных элементов в каждой строке массива:**

```

program sred_n;
const m=3;
n=4;
var
a: array[1..m,1..n] of integer;
i,j,n_pos_el:integer;
sred:real;
begin
    for i:=1 to m do
        for j:=1 to n do
            read(a[i,j]);
        sred:=0;
        for i:=1 to m do
            begin
                n_pos_el:=0;
                for j:=1 to n do
                    begin
                        sred:=sred+a[i,j];
                        if a[i,j]>0 then
                            inc(n_pos_el);
                    end;
                writeln('V ', i, '-oi stroke ', n_pos_el, ', polozitelnix
                elementov');
            end;
        sred:=sred/m/n;
        writeln('Srednee arifmeticheskoe:',sred:6:2);
    End.

```

3. Ввести и отладить программу, которая определяет номер самого первого слева столбца, содержащего только положительные элементы. Если такого столбца нет, выводится сообщение.

```

program num_posit;
const m=3;
      n=4;
var
      a: array[1..m,1..n] of integer;
      i,j,num:integer;
      all_posit:boolean;
begin
      randomize;
      for i:=1 to m do
      begin
          for j:=1 to n do
          begin
              a[i,j]:=random(100)-10;
              write(a[i,j]:4);
          end;
          writeln;
      end;
      num:=0;
      for j:=1 to n do
      begin
          all_posit:=true;
          for i:=1 to m do
              if a[i,j] < 0 then
              begin
                  all_posit:=false;
                  break;
              end;
              if all_posit then
              begin
                  num:=j;
                  break;
              end;
          end;
      end;
      if num = 0 then
          writeln('Takix stolbcov net')
      else
          writeln('Nomer stolbca:',num);
  
```

End.

4. Ввести и отладить программу. Дан двумерный массив. Найти в каждой строке наибольший элемент и поменять его местами с элементом главной диагонали.

```
Program Obmen;
Var
  N, I, J, Max, Ind, Vsp : Integer;
  A : Array [1..15, 1..15] Of Integer;
Begin
  WRITE('Введите количество элементов в массиве: ');
  Readln(N);
  FOR I := 1 TO N DO
  Begin
    FOR J := 1 TO N DO
      Read (A[I, J]);
      Readln;
    end;
    FOR I := 1 TO N DO
    Begin
      Max := A[I, 1];
      Ind := 1;
      FOR J := 2 TO N DO
        IF A[I, J] > Max THEN
          Begin
            Max := A[I, J];
            Ind := J
          end;
        Vsp := A[I,1]; A[I,1] := A[I,Ind]; A[I,Ind] := Vsp;
      end;
    FOR I := 1 TO N DO
    Begin
      Writeln;
      FOR J := 1 TO N Do
        Write(A[I, J] : 3);
      end;
      Writeln
    End.
```



### Уровень 3. Творческий уровень.

1. Переписать элементы массива  $X$  в массив  $Y$  в обратном порядке.

2. Сформировать массив  $Y(1)=X(1)+X(N)$ ,  $Y(2)=X(2)+X(N-1)$ ,  $Y(3)=X(3)+X(N-2)$  и т. д. ( $N$  — чётное).

3. Записать в массив  $Y$  номера элементов массива  $X$ , лежащих на отрезке  $[0,1]$ .

4. Записать в массив  $Y$  элементы массива  $X$ , имеющие чётные индексы, а в массив  $Z$  элементы, имеющие нечётные индексы.

5. Записать в начало массива  $Y$  положительные, а в конец отрицательные элементы массива  $X$ , сохранив порядок элементов.

6. Заполнить двумерный массив порядка  $n$  по следующему образцу:

1	2	3	...	$n-2$	$n-1$	$n$
2	1	2	...	$n-3$	$n-2$	$n-1$
3	2	1	...	$n-4$	$n-3$	$n-2$
...	...	...	...	...	...	...
$n-1$	$n-2$	$n-3$	...	2	1	2
$n$	$n-1$	$n-2$	...	3	2	1

## ЗАДАЧИ ЦЕЛОЧИСЛЕННОЙ АРИФМЕТИКИ

### Цели обучения:

- формирование знаний учащихся о способах решения задач целочисленной арифметики с помощью компьютера;
- формирование умений использовать алгоритмы целочисленной арифметики при работе со стандартными типами данных.

### Цели развития:

- развитие логического и пооперационного мышления учащихся;
- развитие познавательных интересов учащихся.

### Учащиеся должны знать:

- смысл понятий «наибольший общий делитель» (НОД), «наименьшее общее кратное» (НОК), «простой множитель», «делитель числа».

**Учащиеся должны уметь:**

- находить простые множители, делители числа;
- находить НОД и НОК;
- сокращать дроби, выполнять действия с дробями.

**Программное и методическое обеспечение урока:** система программирования с языком Pascal, дидактические материалы к факультативу.

### **Методические рекомендации по теме (уроки 11—20)**

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (13 вопросов). Учитель задаёт выборочно вопросы и предлагает учащимся ответить на них.

2. Учитель предлагает завершить предложения:

Целое число, делящееся на 2 без остатка, называют \_\_\_\_ .

Число  $a$  называют кратным числа  $b$ , если \_\_\_\_\_ .

Число 2 — единственное чётное число, являющееся \_\_\_\_\_ .

Числа  $a$  и  $b$  называют взаимно простыми, если \_\_\_\_\_ .

Для того чтобы найти НОК трёх чисел, нужно \_\_\_\_\_ .

3. Выбрать правильный ответ:

а) число 10 может являться для числа 100:

- 1) делителем,
- 2) кратным,
- 3) множителем,
- 4) простым множителем;

б) число 2 может являться для числа 100:

- 1) делителем,
- 2) кратным,
- 3) множителем,
- 4) простым множителем;

в) число 1000 является для числа 100:

- 1) делителем,
- 2) кратным,
- 3) множителем,
- 4) простым множителем.

4. Определить, что проверяется в приведённой части программы:

- a) if (a mod 3=0) and (a mod 2=0) then  
    writeln('да')  
    else  
        writeln('нет');
- b) if (a mod 3=0) and ((a mod 10=0) or (a mod 10=5)) then  
    writeln('да')  
    else  
        writeln('нет');
- c) if (a mod 3=0) and ((a mod 100) mod 4=0) then  
    writeln('да')  
    else  
        writeln('нет');

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Уроки 11—13. Предложить учащимся выполнить задания 1—3.

Уроки 14—15. Предложить учащимся выполнить задание 4.

Уроки 16—17. Предложить учащимся выполнить задания 5—6.

Уроки 18—20. Предложить учащимся выполнить задание 7.

При изучении тем «Разложение числа на простые множители» и «Нахождение НОД» можно предложить учащимся знакомство с рекурсивной реализацией данных алгоритмов. Примеры можно посмотреть в приложении 2.

**Уровень 3. Творческий уровень.**

Предложить учащимся решение олимпиадных задач.

### *Задача 1. Строка цифр*

Из строки цифр вычеркнуть минимальное количество цифр так, чтобы оставшееся число было максимальным из

всех возможных, делящихся на 3 без остатка. Цифры вводятся по одной. Признак конца ввода — число -1. Вывести количество удаляемых цифр и их номера. Если исходное число уже делится на 3, то вывести 0.

*Примеры.*

<b>Ввод 1</b>	<b>Ввод 2</b>
1 1 1 1 -1	1 1 1 -1
<b>Вывод 1</b>	<b>Вывод 2</b>
1 1	0

### *Задача 2. Умножение столбиком*

Вывести на экран картинку, изображающую умножение «столбиком» двух заданных целых чисел. Результат умножения заданных чисел не превышает девять десятичных разрядов.

## **СИСТЕМЫ СЧИСЛЕНИЯ**

### **Цели обучения:**

- углубление знаний о позиционных системах счисления на примере десятичной, двоичной, шестнадцатеричной систем;
- формирование умения переводить числа из одной системы счисления в другую;
- формирование знаний о программной реализации алгоритмов перевода чисел из одной системы счисления в другую.

### **Цели развития:**

- развитие логического и пооперационного мышления учащихся;
- развитие познавательных интересов учащихся.

### **Учащиеся должны знать:**

- представление чисел и правила выполнения арифметических операций в позиционных системах счисления;
- правила и алгоритмы перевода целых чисел из одной позиционной системы счисления в другую;
- правила и алгоритмы выполнения арифметических действий с числами, записанными в поразрядном представлении позиционной системы счисления.

### ***Учащиеся должны уметь:***

- представлять целые числа в виде последовательности значений разрядов;
- программно реализовывать алгоритмы перевода целых чисел из одной позиционной системы счисления в другую;
- программно реализовывать алгоритмы выполнения арифметических действий (сложения, вычитания, умножения, вычисления остатка от деления на некоторое число) с числами, записанными в поразрядном представлении позиционной системы счисления.

**Программное и методическое обеспечение урока:** система программирования с языком Pascal, дидактические материалы к факультативу.

### **Методические рекомендации по теме (уроки 21—30)**

Изучение этой темы строится на базе углубления и расширения материала базового курса IX класса. Основное внимание уделяется алгоритмам перевода чисел из одной позиционной системы в другую, их программной реализации, а также выполнению арифметических операций над числами, представленными последовательностью разрядов.

**Уровень 1. Репродуктивный уровень** (проверка теоретических знаний).

1. Учащимся предлагается устно ответить на вопросы для повторения в предлагаемом факультативе (7 вопросов). Учитель задаёт вопросы и предлагает учащимся ответить на них. Необходимо добиться, чтобы учащиеся безошибочно могли ответить на все поставленные вопросы, прежде чем переходить к решению задач.

2. Предложить учащимся рассмотреть запись времени в формате «часы:минуты:секунды» как число в позиционной 60-ричной системе счисления и выполнить операцию сложения (с переносом в следующие разряды):

- а) 12:00:59 и 00:00:02;
- б) 03:58:55 и 02:01:06.

3. Напомнить учащимся о непозиционных системах счисления на примере римской, в которой в качестве цифр используются латинские буквы: I обозначает 1, V — 5, X — 10, L — 50, C — 100, D — 500, M — 1000.

Выполнить преобразование римских чисел в обычные десятичные:

а) MCIX;

б) XLII.

4. Учитель предлагает дополнить таблицу перевода чисел между десятичной, двоичной и шестнадцатиричной системами счисления для чисел от 0 до 16.

Десятичная система	Двоичная система	Шестнадцатиричная система
0	0	0
1	1	1
2		2
3	11	3
	100	4
5	101	5
		6
7	111	7
8		8
9	1001	9
		A
11	1011	
12		
	1101	
14		
15	1111	
16	10000	

5. Предлагается перевести числа из десятичной системы счисления в двоичную, выполняя деление на 2 и выписывая остатки в обратном порядке:

а) 255;

б) 1000;

в) 520.

6. Предлагается перевести числа из десятичной системы счисления в шестнадцатеричную, выполняя деление на 16 и выписывая остатки (с учётом наличия специальных обозначений шестнадцатеричных цифр) в обратном порядке:

- а) 2550;
- б) 10200;
- в) 520.

7. Предлагается перевести шестнадцатеричные числа в двоичное представление, выписывая тетрады (четвёрки) двоичных цифр для каждой шестнадцатеричной цифры:

- а)  $9F6_{16}$ ;
- б)  $27D8_{16}$ ;
- в)  $520_{16}$ .

**Уровень 2. Продуктивный уровень** (проверка умений и навыков).

Задания сформулированы в соответствующих темах дидактических материалов к факультативу.

Уроки 2—6. Предложить учащимся выполнить задания 1—6, используя в качестве основы программы, приведённые в дидактических материалах факультатива.

Уроки 7—10. Предложить учащимся самостоятельно разработать программы для взаимного перевода чисел в 2-10-16-ричных системах счисления, используя ранее полученные знания. Числа хранить в виде массива значений соответствующих разрядов.

При изучении темы «Перевод чисел из десятичной системы счисления в системы счисления с основанием, меньшим 10», можно предложить учащимся знакомство с рекурсивной реализацией данного алгоритма. Пример можно посмотреть в приложении 3.

**Уровень 3. Творческий уровень.**

Предложить учащимся изучить в качестве непозиционной систему счисления остаточных классов (СОК), основанную на понятии вычета и китайской теоремы об остатках (см. реализацию представления в такой системе счисления, например, на [http://www.e-maxx.ru/algo/chinese\\_theorem](http://www.e-maxx.ru/algo/chinese_theorem)).

## ТВОРЧЕСКАЯ РАБОТА

В качестве творческой работы можно предложить учащимся следующие виды работ:

- решение олимпиадных задач, подбор тестов для олимпиадных задач;
- разработку тестирующей программы для проверки знаний, полученных во время работы на факультативе;
- разработку простейших логических игр.

### *Приложение 1 к теме 2*

#### *Отладка программ в среде Borland Delphi*

В помощь пользователю Delphi предоставляет средства, необходимые для отладки программы, которые позволяют тщательно протестировать программу и устранить все ошибки в ней. Среда Delphi содержит мощный и гибкий отладчик, который позволяет выполнить программу построчно, просматривать и модифицировать переменные и выражения. Отладчик встроен в интегрированную среду и поэтому пользователь может редактировать, компилировать и отлаживать программу, не выходя из Delphi.

Существует три основных типа ошибок: ошибки компиляции, ошибки выполнения и логические ошибки.

#### **Ошибки компиляции.**

Ошибки **компиляции**, или **синтаксические** ошибки, возникают в том случае, если не описана некоторая переменная, передается неправильное количество параметров процедуре или функции, присваивается вещественное значение целочисленной переменной и т. д. С исправлением ошибок, обнаруженных во время компиляции, сложностей практически не возникает. Компилятор, обнаружив в программе ошибки, завершает работу и выдает информацию об ошибках в окне редактора кода, на вкладке Build (рис. 1). Для каждой ошибки указывается номер строки, в которой обнаружены ошибка и описание этой ошибки. Индикатор ► указывает, какая из ошибок выделена в программе строкой коричневого цвета (рис. 2). Двойной щелчок по ошибке в окне Build покажет её местоположение в



программе. Исправив ошибки, обнаруженные таким образом, можно повторить компиляцию программы.

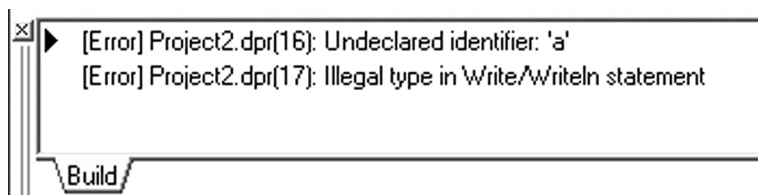


Рис. 1

```
Write('введите значение z=');  
Readln(z);  
a:=(2*x+3*y-z) / (3+2*x);  
Writeln('a=', a:7:3);  
Readln;
```

Рис. 2

### Ошибки выполнения.

Ошибки выполнения, или семантические ошибки, возникают на этапе выполнения при попытке открыть несуществующий файл, делить на нуль и т. д. При этом выполнение программы прерывается, и на экран выдётся сообщение об ошибке (рис. 3).

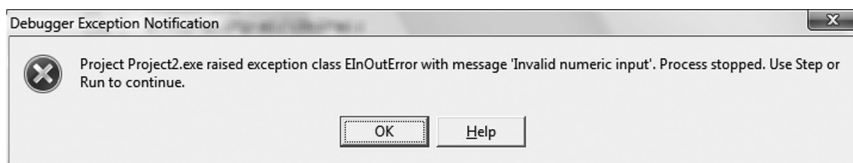


Рис. 3

Возникновение фатальных ошибок приводит к немедленному прекращению выполнения программы.

### **Логические ошибки.**

Программа может содержать и **логические** ошибки. Логические ошибки позволяют компилятору выполнить программу, однако результат выполнения будет отличаться от ожидаемого. Наличие таких ошибок определяется с помощью тестирования. Обнаружить ошибку этой группы, не имея никаких отладочных средств, бывает достаточно сложно. Наличие в программе таких ошибок является одной из основных причин возникновения необходимости использования отладчика, входящего в состав интегрированной среды Delphi. Отладчик Delphi позволяет выполнить программу в диалоговом режиме, наблюдая за изменением значений отдельных переменных или выражений. Имеет возможность остановиться в заданной точке программы и изменить значения некоторых переменных и выражений.

Команды отладчика собраны в меню Run.

**Трассировка** — это построчное выполнение исходной программы, при котором после выполнения каждой строки можно остановиться и посмотреть результаты. Если в программе вызывается процедура или функция, предоставляется выполнить этот вызов как один шаг трассировки, либо войти в вызываемую подпрограмму и протрассировать её построчно.

Наличие возможности выполнения исходной программы до строки, на которую указывает курсор, позволяет пропустить трассировку малоинтересных частей программы и сразу перейти в точку начала отладки.

Некоторые строки программы могут быть помечены как точки прерывания. Для этого достаточно щёлкнуть мышью слева от строки программы (рис. 4).

Когда в процессе выполнения программы достигается точка прерывания, выполнение программы приостанавливается, и на экране отображается фрагмент исходного текста программы, содержащий точку прерывания. Остановившись в точке прерывания, можно просмотреть значения переменных, начать трассировку или выполнить программу до следующей точки прерывания.

```
Repeat
  d:=d*2;
  g:=g*3;
  n:=n+1;
  a:=(n+d)/g;
  S:=S+a;
```

Рис. 4

Отладчик интегрированной среды Delphi предоставляет возможность **наблюдения** за изменением некоторых (выбранных программистом) переменных или выражений. Наблюдаемые объекты помещаются в окно наблюдения (Watch List), отражая изменения в программе при пошаговом выполнении. Окно Watch List (рис. 5) появляется «висячим», при желании его можно прицепить к окну редактора кода (рис. 6). Для этого нужно переместить окно Watch List к окну редактора кода и выбрать место его размещения. В процессе перемещения окна место, куда его можно «прицепить», будет отображаться рамкой с границей серого цвета. Управлять окном Watch List позволяет его контекстное меню.

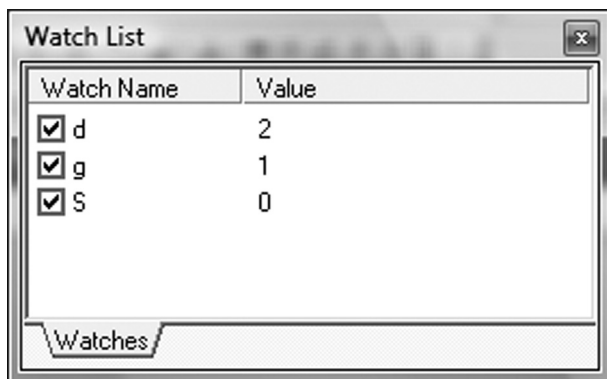


Рис. 5

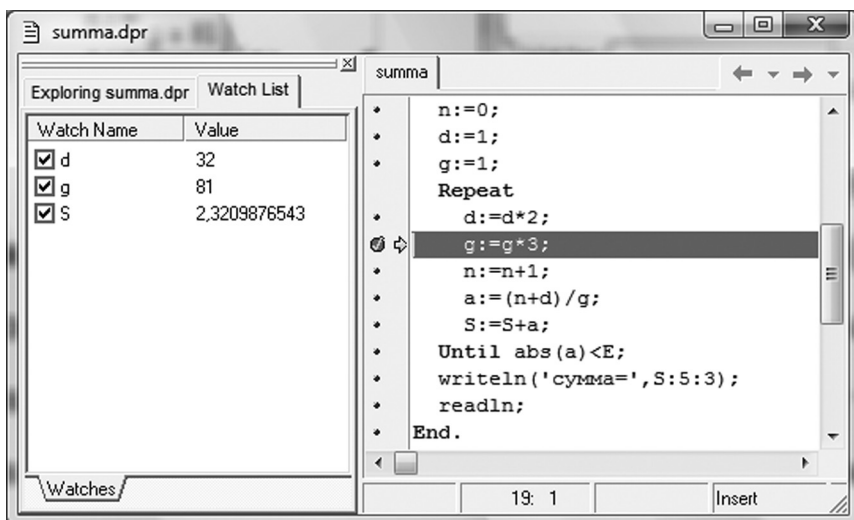


Рис. 6

В режиме отладки, наведя указатель мыши на имя переменной, в окне редактора кода, можно сразу увидеть её значение (рис. 7).

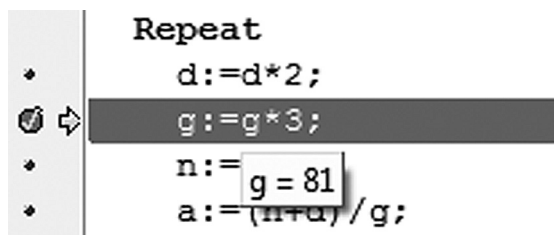





Рис. 7

Некоторые команды отладки, собранные в меню Run, приведены в таблице. Вместе с командами меню приведены также некоторые кнопки, расположенные на панели быстрого доступа, и клавиши, связанные с ними.

## Команды отладки в среде Delphi

Команда	Функция	Клавиша
Run 	Выполняет программу. Если необходимо, сначала её перекомпилирует	F9
Program Reset	Завершает сеанс отладки, освобождает память, закрывает файлы, готовит программу к новому сеансу отладки	Ctrl+F2
Run to cursor	Выполняет программу до строки, на которой установлен курсор	F4
Trace into 	Выполняет строку программы. При наличии вызова подпрограммы в этой строке переходит к трассировке подпрограммы	F7
Step over 	Выполняет строку программы. При наличии вызова подпрограммы в этой строке перехода к трассировке подпрограммы не происходит	F8
Run until return	Выполняет подпрограмму до конца и передаёт управление на команду, следующую за вызовом данной подпрограммы	Shift+F8
Evaluate/Modify	Отображает на экране окно для вычислений, позволяющее просмотреть, и, если необходимо, скорректировать значения переменных и выражений	Ctrl+F7
Add watch	Добавить объект в окно наблюдения. Это же можно сделать, если, находясь в окне наблюдения, воспользоваться клавишей Insert	Ctrl+F5

## Некоторые команды отладчика в среде PascalABC

Команда меню	Клавиша	Кнопка	Действие
Программа → Выполнить	F9		Выполняет программу. Если необходимо, сначала её перекомпилирует
Программа → Шаг с входом в подпрограмму	F7		Выполняет строку программы с входом в подпрограмму
Программа → Шаг без входа в программу	F8		Выполняет строку программы без входа в подпрограмму
Программа → Идти к курсору	F4		Выполняет программу до строки, на которой установлен курсор
Программа → Завершить	Ctrl+F2		Завершает сеанс отладки

### Приложение 2 к теме 7

#### *Рекурсивная реализация алгоритмов*

Разложение числа на простые множители:

```

var n:longint;
procedure razl(x,y:longint);
begin
  if x>1 then
    if x mod y=0 then
      begin
        write(y:5);
        razl(x div y,y);
      end
    else razl(x, y+1);
end;
begin
  writeln('chislo');
  readln(n);
  razl(n,2);
  readln;
end.

```

### Нахождение НОД:

```
var a,b:integer;
function nod(x,y:integer):integer;
begin
  if (x=0) or (y=0) then
    nod:=x+y
  else
    if x>y then
      nod:=nod(x mod y, y)
    else
      nod:=nod(y mod x, x);
end;
begin
  writeln('vvedi 2 chisla');
  readln(a,b);
  writeln('nod('a','b')=',nod(a,b));
  readln;
end.
```

### Приложение 3 к теме 9

#### *Рекурсивная реализация алгоритма перевода*

```
var s,i,n:longint;
m:byte;
a:array[1..20] of integer;

procedure per(x:longint;p:byte);
begin
  if x>=p then per(x div p,p);
  n:=n+1;
  a[n]:=x mod p;
end;

begin
  writeln('chislo');
  readln(s);
  writeln('v kakou systemu schislenia');
  readln(m);
  n:=0;
```

```

per(s,m);
for i:=1 to n do
  write(a[i]);
readln;
End.

```

**Краткое описание представления числа в виде набора вычетов (из Википедии).**

*Система остаточных классов (СОК).*

Представление числа в системе остаточных классов основано на понятии вычета и китайской теореме об остатках. СОК определяется набором взаимно простых *модулей* ( $m_1, m_2, \dots, m_n$ ) с произведением  $M = m_1 \cdot m_2 \cdot \dots \cdot m_n$  так, что каждому целому числу  $x$  из отрезка  $[0, M - 1]$  ставится в соответствие набор вычетов  $(x_1, x_2, \dots, x_n)$ , где указаны остатки от деления  $x$  на  $m_1, m_2, \dots, m_n$  соответственно.

$$x \equiv x_1 \pmod{m_1};$$

$$x \equiv x_2 \pmod{m_2};$$

...

$$x \equiv x_n \pmod{m_n}.$$

При этом китайская теорема об остатках гарантирует однозначность представления для чисел из отрезка  $[0, M - 1]$ .

В СОК арифметические операции (сложение, вычитание, умножение, деление) выполняются покомпонентно, если о результате известно, что он является целочисленным и также лежит в  $[0, M - 1]$ .

Недостатками СОК являются возможность представления только ограниченного количества чисел, а также отсутствие эффективных алгоритмов для сравнения чисел, представленных в СОК. Сравнение обычно осуществляется через перевод аргументов из СОК в смешанную систему счисления по основаниям  $(m_1, m_1 \cdot m_2, \dots, m_1 \cdot m_2 \cdot \dots, m_{n-1})$ .

*Перевод чисел из СОК в десятичную систему счисления.*

Формула перевода имеет вид:

$A = a_1 * B_1 + a_2 * B_2 + \dots + a_n * B_{n-r} * P$ , где  $a_1, \dots, a_n$  — представление числа  $A$  в СОК с основаниями  $p_1, p_2, \dots, p_n$ ;

$$P = p_1 * p_2 * \dots * p_n;$$

$r = 0, 1, 2, \dots$  (целые числа), причём  $r$  выбирают так, чтобы разность между левой и правой частью выражения была меньше  $P$ ;



$B_i = (P/p_i) * k_i$ , где  $k_i = 1, 2, \dots, p_i$ , причём  $k_i$  выбирается таким, чтобы остаток от деления  $B_i/p_i$  был равен 1.

*Пример.*

$A = (2, 4, 6)$  в системе с основаниями:  $p_1 = 3, p_2 = 5, p_3 = 7$ .

$P = p_1 * p_2 * p_3 = 3 * 5 * 7 = 105$ .

$B_1 = 105/3 * k_1 = 35 * 2 = 70$ ;

$B_2 = 105/5 * k_2 = 21 * 1 = 21$ ;

$B_3 = 105/7 * k_3 = 15 * 1 = 15$ ;

$A = 2 * 70 + 4 * 21 + 6 * 15 - r * 105$ ;

$A = 314 - r * 105 = 104$ , где  $r = 2$ .

## Приложение 4

### Тематическое планирование курса «Информатика в задачах»

№ урока	№ в теме	Тема	Решение задач
1	2	3	4
<b>VIII КЛАСС</b>			
<b>Этапы решения задач с помощью компьютера (3 ч)</b>			
1	1	Постановка задачи. Построение модели. Разработка алгоритма	Задание 1
2	2	Реализация алгоритма в виде программы	Задания 2—3
3	3	Тестирование и отладка программы. Анализ результатов	Задания 4—5
<b>Интегрированная среда разработки программ (2 ч)</b>			
4	1	Создание проекта, ввод и редактирование текста программы	Задания 5—6
5	2	Выполнение по шагам, возможности отладки	Задание 7
<b>Логические задачи (7 ч)</b>			
6	1	Логические величины. Логические операции. Таблицы истинности	Задание 1

1	2	3	4
7—8	2—3	Использование логических величин в операторе ветвления	Задания 3, 4, 6—8
9	4	Использование логических величин в операторе цикла	Задания 9, 10
10—11	5—6	Управляющие структуры языка Pascal, использующие логические величины для работы с графикой	Задания 2, 5, 11
12	7	Оператор выбора	Задание 12
<b>Использование процедур и функций для решения задач (9 ч)</b>			
13	1	Правила организации, описания и вызова функций с одной переменной	Задание 3 (из пред. темы)
14—15	2—3	Фактические и формальные параметры	Задания 1—2
16	4	Правила организации, описания и вызова процедур	Задания 1—2
17—18	5—6	Глобальные и локальные переменные	Задание 3
19—21	7—9	Решение задач с использованием процедур и функций	Задания 4—5
<b>Решение геометрических задач (9 ч)</b>			
22	1	Представление геометрических объектов: точка, отрезок	
23—24	2—3	Прямая. Уравнения прямой	Задания 1—3
25	4	Размещение точек плоскости относительно заданной прямой	Задания 4—5
26	5	Расстояние между точками, расстояние от точки до прямой	Задания 6—7
27—28	6—7	Взаимное расположение отрезков	Задания 8—9
29—30	8—9	Точка пересечения прямых и отрезков	Задание 10

1	2	3	4
<b>Творческая работа (3 ч)</b>			
31—33(35)	1—3(5)	Творческая работа	
<b>IX КЛАСС</b>			
<b>Решение задач с использованием массивов (10 ч)</b>			
1—2	1—2	Организация ввода массива данных из файла. Вывод результатов в файл	Задания 1, 2
3—4	3—4	Упорядочение элементов массива	Задание 3
5	5	Двоичный поиск	Задание 4
6	6	Оценка сложности алгоритмов сортировки	Задание 3
7	7	Двумерные массивы	Задания 5, 6
8	8	Обработка строк и столбцов таблицы: перестановка строк и столбцов	Задания 7, 8
9—10	9—10	Заполнение массива по заданному правилу	Задания 9,10
<b>Задачи целочисленной арифметики (10 ч)</b>			
11—13	1—3	Выделение цифр числа	Задания 1—3
14—15	4—5	Поиск делителей числа	Задание 4
16—17	6—7	Простые числа. Разложение на простые множители	Задания 5, 6
18—20	8—10	Поиск наибольшего общего делителя и наименьшего общего кратного натуральных чисел. Операции с дробями, сокращение дробей	Задание 7
<b>Системы счисления (10 ч)</b>			
21	1	Позиционные и непозиционные системы счисления	Задание 1

*Окончание таблицы*

1	2	3	4
22	2	Базис, алфавит, основание системы счисления	Задание 2
23—26	3—6	Представление целых чисел и арифметические операции в позиционных системах счисления	Задания 3—6
27—30	7—10	Перевод целых чисел из одной позиционной системы счисления в другую. Алгоритмы перевода целых чисел	
<b>Творческая работа (3 ч)</b>			
31—33(35)	1—3(5)	Творческая работа	

# Содержание

Учебная программа .....	3
Пояснительная записка .....	3
Содержание программы .....	5
Пособие для учителей .....	8
Этапы решения задачи с помощью компьютера .....	8
Интегрированная среда разработки программ.....	11
Логические задачи .....	16
Использование процедур и функций для решения задач .....	19
Решение геометрических задач.....	22
Творческая работа.....	27
Решение задач с использованием массивов .....	27
Задачи целочисленной арифметики .....	37
Системы счисления.....	40
Творческая работа.....	44
Приложение 1 к теме 2 .....	44
Приложение 2 к теме 7 .....	50
Приложение 3 к теме 9 .....	51
Приложение 4 .....	53

## ДЛЯ ЗАМЕТОК

---

## **ДЛЯ ЗАМЕТОК**

---

Учебное издание

**Лапо** Анжелика Ивановна  
**Сапун** Оксана Леонидовна  
**Пазюра** Екатерина Васильевна

# Информатика в задачах

8—9 классы

Пособие для учителей учреждений  
общего среднего образования  
с белорусским и русским языками обучения

2-е издание

Редактор	<i>Я. Н. Важник</i>
Компьютерный набор	<i>В. Ю. Зарецкая</i>
Компьютерная вёрстка	<i>В. Ю. Зарецкая</i>
Корректор	<i>Я. Н. Важник</i>

Подписано в печать 10.03.2014 г. Формат 60×84 <sup>1</sup>/<sub>16</sub>.

Бумага газетная. Печать офсетная.

Усл. печ. л. 3,48. Уч.-изд. л. 1,9.

Тираж 550 экз. Заказ № 24.

Издатель и полиграфическое исполнение:

РУП «Издательство “Адукацыя і выхаванне”».

Свидетельство о государственной регистрации  
издателя, изготовителя, распространителя печатных изданий

№ 1/19 от 02.08.2013.

№ 2/17 от 26.11.2013.

Ул. Будённого, 21, 220070, г. Минск.