

# ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

---

## Урок 25. Алгоритм и его формальное исполнение

**Цели:** ввести понятие «кибернетика», разобрать свойства алгоритма.

### *Требования к знаниям и умениям*

*Учащиеся должны знать:*

- что такое кибернетика; предмет и задачи этой науки;
- что такое алгоритм управления; какова роль алгоритма в системах управления;
- основные свойства алгоритма;
- исполнители алгоритмов: формальные и неформальные;
- способы записи алгоритмов: блок-схемы, учебный алгоритмический язык.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- выполнять трассировку алгоритма для известного исполнителя.

**Программно-дидактическое оснащение:** Сем., § 25, 27, с. 140 и 147; Угр., § 4.1.1, с. 105; демонстрация «Алгоритм»; проектор.

## Ход урока

### I. Организационный момент

### II. Постановка целей урока

Вы уже знаете множество областей применения компьютера. Это обработка текстов, графики, передача и получение информации, создание справочников, произведение расчетов. Еще одно из важнейших направлений применения компьютеров — управле-

ние. Управление объектами происходит по алгоритму. Изучению этого раздела мы и посвятим следующие уроки.

### III. Работа по теме урока

(С помощью демонстрации «Алгоритм» показать примеры алгоритмов, исполнителей алгоритмов: формальные и неформальные, свойства алгоритма, примеры для определения набора исходных данных алгоритма.)

В 1948 г. в США и Европе вышла книга Норберта Винера «Кибернетика, или Управление и связь в животном и машине». С этого момента начали говорить о новой науке — кибернетике. **Кибернетика** — это наука об общих свойствах процессов управления в живых и неживых системах. Управление — это целенаправленное воздействие одних объектов (управляющих) на другие объекты — управляемые. Все управляющие воздействия производятся в форме команд. Команды отдаются с определенной целью. Последовательность команд по управлению объектом, выполнение которых приводит к достижению заранее поставленной цели, называется **алгоритмом управления**. Объект управления — исполнитель алгоритма.

#### Задание 1.

1. Кто играет роль управляющего и исполнителя в следующих системах: школа, класс, самолет, стая волков, стадо коров? (*Администрация — коллектив, учащиеся; учитель — ученики; пилот — самолет, стюардессы — пассажиры; вожак — остальные волки; пастух — коровы.*)

2. Для этих систем назовите некоторые команды управления и скажите, в какой форме их отдают.

Слово «алгоритм» происходит от имени выдающегося средневекового математика Мухаммеда ибн Мусы аль-Хорезми (787–850). Он предложил приемы вычисления с многозначными числами, в Европе их называли алгоритмами. Позже понятие «алгоритм» значительно расширилось.

**Исполнитель алгоритма.** Это может быть человек или живое существо (неформальный исполнитель, т. е. он может отойти от заранее продуманного плана), или техническое устройство (формальный исполнитель не вносит элементов творчества в выполнение). Каждый исполнитель понимает свой набор команд — система команд исполнителя СКИ.

**Задание 2.** Назовите исполнителей следующих видов работы: уборка мусора во дворе; перевозка пассажиров; выдача заработной платы; прием экзамена; сдача экзамена; обучение детей в школе. Продумайте их СКИ.

**Свойства алгоритма.**

1. Дискретность — процесс решения задачи разбит на последовательно выполняемые шаги.
2. Понятность — алгоритм должен состоять из команд, понятных исполнителю, которые входят в его СКИ.
3. Точность (выполнимость) — каждая команда определяет однозначно действия исполнителя.
4. Конечность — исполнение алгоритма должно завершаться за конечное число шагов.
5. Результативность — исполнение алгоритма должно приводить к конкретному результату.
6. Массовость — один и тот же алгоритм можно применять к большому количеству исходных данных.
7. Детерминированность — последовательность команд алгоритма должна выполняться строго в определенной последовательности.
8. Понятность — команды в алгоритме должны быть написаны на понятном исполнителю языке.

Еще один важный момент в теории алгоритмов: для успешного решения задачи исполнитель должен иметь полный набор исходных данных. Если исходных данных не хватает, то задачу совсем нельзя решить, либо она решается неверно.

**Задание 3.** Определите полный набор данных для решения следующих задач:

- 1) вычисления стоимости покупок в магазине;
- 2) определения времени показа по телевизору интересующего вас фильма;
- 3) вычисления площади треугольника;
- 4) определения месячной платы за расход электроэнергии;
- 5) алгоритм письменно можно оформить с помощью блок-схем (используя графические примитивы), алгоритмического языка или на одном из языков программирования.

**СКИ алгоритмического языка:**

- 1) **алг** Название алгоритма (решение задачи должно начинаться с этой команды);
- 2) **нач** Начало тела алгоритма;
- 3) тело алгоритма, последовательность команд из СКИ;
- 4) **кон** Конец тела алгоритма.

Язык программирования — это фиксированная система обозначений для описания алгоритмов и структур данных. Программа — алгоритм, записанный на языке программирования. Иначе: алгоритм и программа не отличаются по содержанию, но отличаются по форме.

#### IV. Подведение итогов урока

- Что такое алгоритм?
- Что такое исполнитель алгоритма?
- Что такое СКИ?
- Назовите основные свойства алгоритма.

#### Домашнее задание

Написать алгоритм нахождения гипотенузы прямоугольного треугольника, если известны его катеты.

### Урок 26. Алгоритмы работы с величинами

**Цель:** ввести понятие «величина» и показать ее назначение в программировании.

#### *Требования к знаниям и умениям*

*Учащиеся должны знать:*

- основные виды и типы величин;
- назначение языков программирования;
- назначение систем программирования;
- основные команды работы с величинами.

*Учащиеся должны уметь:*

- отличать типы величин;
- составлять алгоритмы на алгоритмическом языке, используя команды присваивания, ввода, вывода.

**Программно-дидактическое оснащение:** Сем., § 33, с. 180; Угр., § 4.1.3, с. 109, § 4.4, с. 123; карточки: тест по теме «Алгоритм», задания на урок, домашнее задание.

### Ход урока

#### I. Организационный момент

#### II. Актуализация знаний

**Тест по теме «Алгоритм»**

1. Как называется алгоритм, записанный на «понятном» компьютеру языке?

- |               |                        |
|---------------|------------------------|
| а) программа; | в) протокол алгоритма; |
| б) текст;     | г) набор команд.       |

2. Как называется свойство алгоритма, заключающееся в том, что каждое действие и алгоритм в целом должны иметь возможность завершения?

- |                         |
|-------------------------|
| а) дискретность;        |
| б) детерминированность; |
| в) результативность;    |
| г) конечность.          |

а) дискретность;                      в) понятность;  
б) детерминированность;          г) точность.

а) дискретность;                      в) конечность;  
б) детерминированность;          г) массовость.

- а) схематичное изображение в произвольной форме;
- б) система обозначения правил для единообразной и точной записи алгоритмов их исполнения;
- в) способ представления алгоритма с помощью геометрических фигур;
- г) представление алгоритма в форме таблиц и расчетных формул.

### III. Постановка целей урока

- Как вы думаете, на какие категории можно разделить людей, работающих на компьютере? (*Пользователи и программисты.*)
- Какие задачи решают программисты? (*Создают операционные системы, редакторы, графические пакеты, компьютерные игры и т. д.*)

Теперь вам предстоит поближе познакомиться еще с одним разделом информатики – «Программирование».

#### IV. Работа по теме урока

**Программирование** необходимо для разработки программ управления компьютером с целью решения различных информационных задач. **Система программирования** – это программное обеспечение компьютера, предназначенное для разработки, отладки, исполнения программ, записанных на определенном языке программирования.

Компьютер-исполнитель работает с определенными данными по определенной программе. Данные (числа, символы, строки, таблицы и т. д.) — **величины**. Величины бывают: 1) числовые; 2) символьные; 3) логические. С последними двумя мы встречались, изучая БД и ЭТ. С этого момента мы будем работать с **числовыми величинами**. Они бывают двух типов — **переменные** и **константы**.

**Константы** записываются своими десятичными значениями (59, 25.4). **Переменные** обозначаются символическими именами (идентификаторами, например g, has, f5g и т. д.).

**Система команд.** Любой алгоритм работы с величинами может быть составлен из следующих команд:

- присваивание;
- ввод;
- вывод;
- обращение к вспомогательному алгоритму;
- цикл;
- ветвление.

Присваивание:  $\langle \text{переменная} \rangle := \langle \text{выражение} \rangle$ .

Например,  $X:=5$ ;  $Y:=X+1$  после этого в ячейке с именем Y будет лежать 6.

Ввод: ввод  $\langle \text{список переменных} \rangle$ .

Например, ввод A, B, C. Значения переменных, являющихся исходными данными решаемой задачи, задаются вводом.

Вывод: вывод  $\langle \text{список вывода} \rangle$ .

Например, вывод  $x_1, x_2, x_3$ . Результаты решения задачи сообщаются компьютером пользователю путем выполнения команды вывода.

## V. Практическая работа

**Задание 1.** В схематическом виде отразите изменения в ячейках, соответствующих переменным A и B, в ходе последовательного выполнения команд присваивания:

- 1)  $A:=1$ ;  $B:=2$ ;  $A:=A+B$ ;  $B:=2*A$ .
- 2)  $A:=1$ ;  $B:=2$ ;  $C:=A$ ;  $A:=B$ ;  $B:=C$ .
- 3)  $A:=1$ ;  $B:=2$ ;  $A:=A+B$ ;  $B:=A-B$ ;  $A:=A-B$ .

**Задание 2.** Вместо многоточия впишите в алгоритм несколько команд присваивания, в результате чего должен получиться алгоритм возведения в четвертую степень введенного числа: ввод A ... вывод A.

## VI. Подведение итогов урока

- Чем отличаются переменные и постоянные величины?
- Назовите основные типы величин в программировании.
- Как записываются команды: присваивания, ввода, вывода?

## Домашнее задание

1. Написать последовательность команд на алгоритмическом языке для наиболее быстрого вычисления выражения:  $(x + x^2 + 5x^4)^4$ .
2. Написать алгоритм вычитания двух дробей.

## Урок 27. Знакомство с языком Паскаль

**Цель:** выяснить структуру программы, операторы, правила оформления и пунктуацию языка Паскаль.

### **Требования к знаниям и умениям**

*Учащиеся должны знать:*

- назначение языка Паскаль;
- структуру программы;
- операторы ввода, вывода, присваивания;
- правила записи арифметических выражений.

*Учащиеся должны уметь:*

- анализировать готовую программу;
- составлять арифметические выражения;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 35, с. 191; карточки: тест по теме «Алгоритмы работы с величинами», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

**Тест по теме «Алгоритмы работы с величинами»**

1. Укажите последовательность команд, в результате выполнения которых значения переменных  $X$  и  $Y$  поменяются местами:

- а)  $X:=X+Y$ ;  $Y:=X-Y$ ;  $X:=X-Y$ ;
- б)  $V:=X$ ;  $X:=Y$ ;  $Y:=X$ ;
- в)  $Y:=X$ ;  $V:=X$ ;  $X:=Y$ ;
- г)  $X:=Y$ ;  $Y:=X$ .

2. Чему будет равно значение переменной  $X$  после выполнения команды  $X:=X+2$ , если до ее выполнения оно было равно 3?

- а) 1;
- б) 2;
- в) 3;
- г) 5.

3. При каких исходных значениях переменных  $X$  и  $Y$  в результате выполнения команды  $X:=X^Y-Y$  значение переменной  $X$  станет равным 2?

- а) 4, 1;
- б) 3, 2;
- в) 2, 2;
- г) 5, 1.

4. Значение выражения  $2*4*2-5^3*2$  равно:

- а) -186;
- б) -15661;
- в) -234;
- г) -61.

5. Команда присваивания означает, что:

- а) вводятся значения переменных, являющихся исходными данными решаемой задачи;
- б) выводятся результаты решения задачи компьютером пользователю;
- в) переменная величина получает значение;
- г) происходит обращение к новой программе.

*Ответы:* 1а; 2г; 3в; 4в; 5в.

### III. Постановка целей урока

После того как продуман алгоритм решения задачи, составляется программа на одном из языков программирования. Одним из самых популярных языков является Паскаль. С этого урока мы начнем рассматривать основы программирования на этом языке.

### IV. Работа по теме урока

Этот язык был разработан в 1971 г. швейцарским профессором Никлаусом Виртом и назван в честь французского ученого Блеза Паскаля. Команды языка называются *операторами*. Программа на языке Паскаль близка по своему описанию алгоритмическому языку.

**Структура программы** (рассмотрим упрощенный вариант):

**Program** <Имя программы>;

**Var** <Раздел описаний>

**Begin**

<Тело программы>

**End.**

Имя программы — любое имя.

Раздел описаний — список переменных (латинских букв, наборов букв, букв и цифр) через запятую, после двоеточия — тип переменных. Числовые типы: вещественный (real) и целый (integer). Например, var a :integer; b1 :real.

Тело программы — основная часть программы. Операторы внутри тела программы отделяются точкой с запятой, а в конце служебного слова **end** ставится точка.

Оператор ввода — при выполнении этого оператора компьютер ожидает ввод данных с клавиатуры.

**Read(<список переменных>)** или **readln(<список переменных>)**. Их отличие в том, что по оператору **readln** курсор перемещается в новую строку. Если в операторе несколько переменных, то данные вводятся через пробел, а в конце нажимается клавиша **Enter**.

Оператор вывода — компьютер выводит результат работы программы на экран.

**Write(<список вывода>)** или **writeln(<список вывода>)** — аналогично оператору ввода экранный курсор переводится на новую строку во втором случае. Например, `write(5,c); write(a,b,c); write('С Новым годом!');` `write(x*y+25-4/3).`

При выводе числовых данных можно задать формат вывода: 1) для целых чисел — количество позиций на экране; 2) для вещественных — общее количество чисел и количество чисел в дробной части. Например, `a=6` оператор `write(a:5)`, тогда на экране будет `_____6`, где `_` пробел; `a=511,64` оператор `write(a:10:4)`, при этом на экране `_____511,6400`.

Оператор присваивания: `<числовая переменная>:=<арифметическое выражение>`, арифметическое выражение может содержать: числовые константы, переменные, арифметические операции, круглые скобки. Приоритет операций: `*`, `/`, `+`, `-`. Возведение в степень — знак `^`, `x2 — sqrt(x)`, `√x — sqrt(x)`.

Например, выражение  $\left(x^2 - y \times \frac{3}{4}\right)^2 - \sqrt{x^3}$  на языке Паскаль выглядит так: `sqr(sqr(x)-y*3/4)-sqrt(x^3)`.

## V. Практическая работа

**Задание 1.** Запишите по правилам языка Паскаль следующие выражения:

1.  $\sqrt{5x + x^2}$ ;
2.  $\operatorname{tg} 3x + 1$ ;
3.  $3x^3 + 2x^2 + 4$ .

**Задание 2.** Определите, какие из приведенных ниже имен нельзя использовать в качестве заголовка программы и почему.

`Iwanov_Petr_10r`; `Dog-Cat`; `Nomera_doma`; `F 22`.

**Задание 3.** Найдите ошибки в программе.

```
Program control;  
Var x, y, z: integer;  
Begin  
x:=5  
y:=7;  
z=x+y;  
writeln (z=x+y, z);  
end.
```

**Задание 4.** Какая задача решается по следующей программе?

```
Program Test;  
var A, B, C: integer;
```

```
begin  
readln(A, B);  
C:=(A+B)*(B-A);  
writeln(C)  
end.
```

Какой результат будет получен, если в качестве исходных значений  $A$  и  $B$  ввести соответственно 7 и 8?

**Задание 5.** Напишите программу, которая вычисляет площадь и периметр прямоугольника со сторонами  $a$  и  $b$ .

## VI. Подведение итогов урока

- Когда появился язык Паскаль и кто его автор?
- Какова структура программы на языке Паскаль?
- Какие операторы на данный момент вы знаете?
- Какие правила пунктуации в Паскале вам известны?

## Домашнее задание

1. Записать по правилам языка Паскаль следующее выражение:

$$\frac{x^2 - 3\sqrt{x^4 - 25\sqrt{x}}}{3,6 - x^3}.$$

2. Написать программу на алгоритмическом языке и на языке Паскаль, по которой можно найти расстояние между двумя точками с заданными координатами.

## Урок 28. Линейные вычислительные алгоритмы

**Цель:** отработать навык составления линейных программ.

### Требования к знаниям и умениям

*Учащиеся должны знать:*

- алгоритмические конструкции следования;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять линейные программы;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Угр., § 4.2.1, с. 113; карточки: тест по теме «Основы языка Паскаль», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

Тест по теме «Основы языка Паскаль»

1. Какого раздела не существует в программе, написанной на языке Паскаль?

- а) заголовков;
- б) операторов;
- в) описания;
- г) примечаний.

2. Какой из операторов не относится к группе ввода – вывода на языке Паскаль?

- а) Read(a1, a2, a3);
- б) Write(a, 'a');
- в) Println;
- г) Writeln.

3. В результате выполнения следующих операторов:  $a:=3$ ;  $b:=4$ ; `write('x=',sqrt(sqr(a)+sqr(b)),a+b=7); writeln(a,b,'a','b');` будет напечатано:

- а)  $x=5$ TRUE34ab;
- б)  $x=5a+b=7$ 34ab;
- в)  $x=\sqrt{sqr(a)+sqr(b)}$ ,  $a+b=7$ 34ab;
- г)  $x=5.000000000000000E+0000$ TRUE34ab.

4. Что будет выведено на печать в результате выполнения следующих операторов:  $a:=5$ ; `write('квадрат числа', a=sqr(a))`?

- а) квадрат числа  $a = 25$ ;
- б) квадрат числа  $5 = 25$
- в) квадрат числа FALSE;
- г) квадрат числа  $a = \sqrt{5}$ .

Ответы: 1г; 2в; 3б; 4в.

### III. Постановка целей урока

– Какой алгоритм называется линейным? (*Алгоритм, в котором команды выполняются друг за другом последовательно.*)

Сегодня на уроке мы будем программировать линейные алгоритмы.

### IV. Работа по теме урока

Для начала давайте изучим еще несколько операторов:  $a \text{ div } b$  – выдает целую часть от деления  $a$  на  $b$ ;  $a \text{ mod } b$  – выдает остаток от деления  $a$  на  $b$ ;  $\text{trunk}(x)$  – выдает целую часть дробного числа  $x$ ;  $\text{round}(x)$  – округляет число  $x$  по правилам математики. Например,  $20 \text{ div } 3 = 6$ ,  $5 \text{ mod } 2 = 1$ ,  $\text{trunk}(3.545) = 3$ ,  $\text{round}(3.545) = 4$ .

### V. Практическая работа

**Задание 1.** Найдите значение выражения  $A \text{ mod } (B \text{ div } C + 5) - 10$ , если  $A = 35$ ,  $B = 6$ ,  $C = 4$ .

**Задание 2.** Отметьте выражения, которые относятся к целому типу.

1.  $1+0,0$
2.  $20/5$
3.  $\text{sqr}(5,0)$
4.  $\text{trunk}(3.1415)$
5.  $\text{sqrt}(4)$
6.  $5+\text{round}(12.7)$

**Задание 3.** Запишите выражение по правилам языка Паскаль.

$$c + \frac{\frac{a+b-1,7}{d}}{e+f+0,5}$$

**Задание 4.** Перепишите выражение  $1 + \text{sqr}(\cos((x+y)/2))$  в традиционной математической форме.

**Задание 5.** Дана неполная программа, предназначенная для вычисления площади треугольника, заполните пропуски, а в операторе вывода укажите параметры так, чтобы значение площади вывелось с двумя знаками после запятой.

```
Program square_treug;
Var a, h: _____;
S: _____;
Begin
Writeln('введите длины стороны и высоты треугольника');
Read(_____);
S:= _____;
Write(S: _____);
End.
```

**Задание 6.** Введите свой возраст, рост и вес. Выведите на экран сообщение.

Для ...-летнего возраста рост ... сантиметров и вес ... кг очень подходят!

**Задание 7.** Вычислите среднее арифметическое и среднее геометрическое двух чисел.

*Ответы:*

1.  $-5$ .
2. 1, 3, 4, 5, 6.
3.  $(a+b-1.7)/(c+d/(e+f+0.5))$ .
4.  $1 + \sqrt{\cos\left(\frac{x+y}{2}\right)}$ .
5. Real; real; a, h;  $(a*h)/2$ ; 4:2.
6. Program vozr;

```

Var a, b, c: integer;
Begin
  Readln(a,b,c);
  Write ('Для',a, '-летнего возраста рост', b, 'сантиметров и вес',
с, 'кг очень подходят!');
End.
7. Program sr;
Var a, b, c, d: integer;
Begin
  Writeln('введите два числа');
  Readln(a,b);
  c:=(a+b)/2;
  d:=sqrt(a*b);
  Write('c=',c:5:2,'d=',d:5:2);
End.

```

## VI. Подведение итогов урока

- В какой последовательности происходит выполнение команд в линейном алгоритме?
- Что является результатом вычисления функций div, mod, trunk, round?

## Домашнее задание

1. Расставить правильно скобки в выражении:  $a+b/x-2*y$  – так, чтобы оно соответствовало выражению:  $a + \frac{b}{x-2} \times y$ .
2. Удалить лишние скобки.  
 $(a*(b/(c*(d/(e*f))))))$ .
3. Написать программу для нахождения суммы цифр заданного трехзначного числа.

*Ответы:*

1.  $a+b/(x-2)*y$ .
2.  $a*b/(c*d/(e*f))$ .
3. Program sr;  
Var a, b, c, d, sum: integer;  
Begin  
Writeln('введите трехзначное число');  
Readln(a);  
b:=a div 100;  
c:=(a-b\*100) div 10;  
d:=a mod 10;  
sum:=b+c+d;  
Write('сумма цифр числа равна', sum);  
End.

## Урок 29. Алгоритмы с ветвящейся структурой

**Цель:** отработать навык составления программ с ветвящейся структурой.

### *Требования к знаниям и умениям*

*Учащиеся должны знать:*

- конструкции ветвления, полную и неполную;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания, работы с целыми числами.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с разветвляющимися алгоритмами;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 36, с. 197; Угр., § 4.2.2, с. 114; карточки: тест по теме «Линейный алгоритм», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

Тест по теме «Линейный алгоритм»

1. Определите значение целочисленной переменной  $a$  после выполнения фрагмента алгоритма.

$a := 247;$

$b := (a \text{ div } 100) * 10 + 9;$

$a := (10 * b - a) \bmod 10;$

**Примечание.** Операции  $\text{div}$  и  $\text{mod}$  вычисляют результат деления нацело первого аргумента на второй и остаток от деления соответственно.

а) 3;

в) 2;

б) 454;

г) 4.

2. Служебное слово `var` в программе на языке Паскаль фиксирует начало раздела программы, содержащего:

а) описание переменных;

б) описание меток;

в) описание констант;

г) описание сложных типов данных.

3. Переменная  $y$  — вещественного типа, а  $n$  — целого типа. Выберите корректное использование оператора присваивания.

а)  $y := n + 1;$

в)  $n := 4.0;$

б)  $n := y - 1;$

г)  $y := \text{trunk}(y).$

4. Укажите значения для переменных  $x$  и  $y$  после выполнения операторов  $x:=2$ ;  $y:=5$ ;  $x:=y$ ;  $y:=x$ .

- |          |          |
|----------|----------|
| а) 2, 2; | в) 5, 5; |
| б) 5, 2; | г) 2, 5; |

5. Какую задачу решает данная программа?

```
Program primer;  
Var a, b, c: integer;  
Begin  
  Writeln('введите двузначное число');  
  Readln(a);  
  b:=adiv10;  
  c:=amod10;  
  Write(c*10+b);  
End.
```

- а) находит сумму цифр числа;
- б) вычисляет целую часть и остаток от деления числа на 10;
- в) меняет цифры числа местами;
- г) находит цифры числа, а на экран выводит само число.

*Ответы:* 1а; 2а; 3а; 4в; 5в.

### III. Постановка целей урока

При решении задач часто возникают ситуации, когда на вопрос «что дальше делать» мы даем ответ в зависимости от выполнения некоторого условия, например находим решение линейного уравнения  $ax = b$ , если коэффициент  $a$  не нулевой. В таких случаях используется конструкция ветвления, реализуемая в Паскале условным оператором.

### IV. Работа по теме урока

В отличие от линейных алгоритмов, в которых команды выполняются одна за другой, конструкция ветвления включает в себя проверку условия. В качестве условия такого оператора используется значение логического выражения. Существуют два вида условного оператора:

- IF B THEN S1 — если выражение B — TRUE (истина), то выполняется выражение, стоящее после THEN.
- IF B THEN S1 ELSE S2 — если выражение B — TRUE (истина), то выполняется выражение, стоящее после THEN, иначе, после ELSE.

B — выражение булевского типа; S1, S2 — отдельные операторы или операторы, сгруппированные при помощи операторных скобок BEGIN END. Перед Else после End «;» не ставится.

S1 и S2, в свою очередь, тоже могут быть условными операторами. При этом каждому ELSE соответствует предыдущий THEN.

## V. Практическая работа

**Задание 1.** Среди предложенных выражений выберите логические.

1.  $(x=3)$  or  $(x<>4)$
2.  $2*x+5$
3.  $x \bmod 2 = 0$
4.  $x \div 3$
5.  $x+y=10$
6.  $x+y$
7.  $(x+y>5)$  or  $(x-y>5)$
8.  $(x>3)$  and  $(x\leq 10)$

**Задание 2.** Вычислите:

- а)  $t$  and  $(p \bmod 3 = 0)$  при  $t = \text{true}$ ,  $p = 101010$ ;
- б)  $(x*y<>0)$  and  $(y>x)$  при  $x = 2$ ,  $y = 1$ ;
- в)  $(x*y<>0)$  or  $(y>x)$  при  $x = 2$ ,  $y = 1$ ;
- г)  $f$  or  $(\text{not}(b))$  при  $f = \text{false}$ ,  $b = \text{true}$ .

**Задание 3.** Составьте условие для условного оператора в программе, которая выводит на экран приветствие в зависимости от времени суток. Пусть утро с 8 до 12 ч, день с 12 до 17 ч, вечер с 17 до 23 ч, ночь – все остальное время.

```
Program primer;
```

```
Var x:real;
```

```
Begin
```

```
Writeln('Который час?');
```

```
Read(x);
```

```
If _____ then writeln('Доброе утро!');
```

```
If _____ then writeln('Добрый день!');
```

```
If _____ then writeln('Добрый вечер!');
```

```
If _____ then writeln('Доброй ночи!') else writeln('введенное число выходит за рамки интервала [0,24]');
```

```
End.
```

**Задание 4.** Какие значения примут переменные  $x$  и  $y$  в результате выполнения следующих фрагментов?

а) Read(a,b);

$x:=0$ ;  $y:=0$ ;

if  $a=b$  then  $x:=a+b$ ;  $y:=a*b$ ;

Ответ: при  $a=2$ ,  $b=3$ ,  $x=$ \_\_\_\_,  $y=$ \_\_\_\_;

при  $a=5$ ,  $b=5$ ,  $x=$ \_\_\_\_,  $y=$ \_\_\_\_\_.

б) Read(a,b);

$x:=0$ ;  $y:=0$ ;

if  $a=b$  then begin  $x:=a+b$ ;  $y:=a*b$ ; end;

Ответ: при  $a=2$ ,  $b=3$ ,  $x=$ \_\_\_\_,  $y=$ \_\_\_\_;

при  $a=5$ ,  $b=5$ ,  $x=$ \_\_\_\_,  $y=$ \_\_\_\_\_.

в) Read(a,b);

x:=0; y:=0;

if a=b then x:=a+b; else x:=a-b; y:=a\*b;

Ответ: при a=2, b=3, x=\_\_\_\_, y=\_\_\_\_;

при a=5, b=5, x=\_\_\_\_, y=\_\_\_\_.

г) Read(a,b);

x:=0; y:=0;

if a=b then x:=a+b else begin x:=a-b; y:=a\*b; end;

Ответ: при a=2, b=3, x=\_\_\_\_, y=\_\_\_\_;

при a=5, b=5, x=\_\_\_\_, y=\_\_\_\_.

д) Read(a,b);

x:=0; y:=0;

if a=b then begin x:=a+b; y:=2\*a\*b; end else begin x:=a-b; y:=a\*b;

end;

Ответ: при a=2, b=3, x=\_\_\_\_, y=\_\_\_\_;

при a=5, b=5, x=\_\_\_\_, y=\_\_\_\_.

**Задание 5.** Составьте программу, заменяющую меньшее из двух данных чисел на среднее арифметическое этих чисел, а большее из двух данных чисел — на среднее геометрическое этих чисел.

Ответы:

1. 1, 3, 5, 7, 8.

2. а) True; б) false; в) true; г) false.

3.  $(x \geq 8) \text{ and } (x < 12)$ ;  $(x \geq 12) \text{ and } (x < 17)$ ;  $(x \geq 17) \text{ and } (x < 24)$ ;  $(x > 0) \text{ and } (x < 8)$ .

4. а) (0, 6), (10, 25); б) (0, 0), (10, 25); в) (-1, 6), (10, 25); г) (-1, 6), (10, 0); д) (-1, 6), (10, 50).

## VI. Подведение итогов урока

- Почему отношение неравенства можно назвать логическим выражением?
- В каком случае для числовой переменной следует указывать целый тип, а в каком — вещественный?

## Домашнее задание

1. Составить алгоритм нахождения меньшего из двух значений.
2. Определить, является ли данное число четным или нечетным.

Ответы:

1. Program dz1;

Var a, b, min: integer;

Begin

Writeln('введите два числа');

Readln(a,b);

if a>b then min:=b else min:=a;

Write('наименьшее число', min);

End.

```
2. Program dz2;  
Var a: integer;  
Begin  
  Writeln('введите число');  
  Readln(a);  
  if amod2=0 then write('число четное') else write('число нечет-  
ное');  
End.
```

## Урок 30. Алгоритмы с ветвящейся структурой

**Цель:** отработать навык составления программ с ветвящейся структурой.

### *Требования к знаниям и умениям*

*Учащиеся должны знать:*

- конструкции ветвления, полную и неполную;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания, работы с целыми числами.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с разветвляющимися алгоритмами;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 36, с. 197; Угр., § 4.2.2, с. 114; карточки: задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

Дана программа:

```
Program urok;  
Var m, n, s, p: integer;  
Begin  
  Read(m,n); s:=m+n; p:=m*n;  
  if m>n then begin m:=s; n:=p; end  
  else begin m:=p; n:=s; end;  
  Write ('m=',m, 'n=',n);  
End.
```

- Для решения какой задачи она предназначена? (*Большее число заменить суммой этих чисел, меньшее – произведением.*)
- Какие структуры называются ветвлением? (*Конструкция ветвления включает в себя проверку истинности или ложности условия.*)

— Какие конструкции ветвления бывают? (*Полная и неполная.*)

### III. Постановка целей урока

Сегодня на уроке мы закрепим пройденный материал, решая задачи.

### IV. Практическая работа

**Задание 1.** Составьте программу, определяющую, является ли число  $A$  кратным числу  $B$ .

*Решение:*

```
Program primer;
```

```
Var a, b :integer;
```

```
Begin
```

```
Writeln('введите два числа');
```

```
Read(a,b);
```

```
If a mod b=0 then writeln('a кратно b') else ('a не кратно b');
```

```
End.
```

**Задание 2.** Вводятся две четвертные оценки по математике и русскому языку. Выведите на экран надпись «Молодец!», если их сумма больше или равна 9, иначе надпись — «Подтянись!».

*Решение:*

```
Program primer;
```

```
Var a, b:integer;
```

```
Begin
```

```
Writeln('введите четвертные оценки по математике и русскому языку');
```

```
Read(a,b);
```

```
If a+b>=9 then writeln('Молодец!') else writeln('Подтянись!');
```

```
End.
```

Использование операторных скобок внутри условного оператора называется **составным оператором**.

**Задание 3.** Составьте программу сортировки по возрастанию значений трех переменных:  $A$ ,  $B$ ,  $C$ .

*Решение:*

```
Program Pr;
```

```
Var S1, S2, S3, C: Integer;
```

```
Begin
```

```
Writeln('Введите три числа');
```

```
Readln(S1,S2,S3);
```

```
If S1>S2 then begin C:=S1; S1:=S2; S2:=C; end; {Меняем местами содержимое S1 и S2}
```

```
If S2>S3 then begin C:=S2; S2:=S3; S3:=C; end; {Меняем местами содержимое S2 и S3}
```

If  $S1 > S2$  then begin  $C := S1$ ;  $S1 := S2$ ;  $S2 := C$ ; end; {Меняем местами содержимое  $S1$  и  $S2$ }

Writeln('Числа в порядке возрастания:',  $S1, S2, S3$ )

End.

**Задание 4.** Составьте программу решения линейного уравнения вида  $Ax = B$ .

*Решение:*

Program linear\_equation;

Var a, b: real;

Begin

Write('введите коэффициенты a и b:');

Readln(a, b);

If  $a \neq 0$  then write('x0',  $-b/a$ )

Else

If  $b = 0$  then write('корень любое число')

Else write('корней нет');

End.

**Задание 5.** Составьте программу решения квадратного уравнения.

*Решение:*

program kv;

var

a, b, c, D, x1, x2: real;

begin

Read(a, b, c);

$D := b^2 - 4 * a * c$ ;

if  $D < 0$

then Writeln('КОРНЕЙ НЕТ')

else

if  $D = 0$  then begin  $D := \text{sqrt}(D)$ ;  $x1 := (-b)/(2 * a)$ ; Writeln('x=', x1);

else

begin

$D := \text{sqrt}(D)$ ;

$x1 := (-b + D)/(2 * a)$ ;

$x2 := (-b - D)/(2 * a)$ ;

Writeln('x1=', x1);

Writeln('x2=', x2);

end;

end.

## V. Подведение итогов урока

- Как программируется на языке Паскаль полное и неполное ветвление?
- Что такое составной оператор?

**Домашнее задание**

1. Дано:  $a$ ,  $b$ ,  $c$  — стороны предполагаемого треугольника. Требуется сравнить длину каждого отрезка с суммой двух других. Если хотя бы в одном случае отрезок окажется больше суммы двух других, то треугольника с такими сторонами не существует.
2. Написать программу, определяющую по координатам точки, в какой четверти она находится.

*Ответы:*

```
1. var a, b, c: integer;
begin
write ('Длины сторон:');
readln (a,b,c);
if (a<b+c) and (b<a+c) and (c<a+b) then
writeln('Треугольник существует.')
else
writeln('Треугольник не существует.');
```

```
2. var x, y: real;
begin
write('x='); read(x);
write('y='); readln(y);
write('Номер четверти координатной плоскости:');
if (x>0) and (y>0) then writeln('I')
else
if (x<0) and (y>0) then writeln('II')
else
if (x<0) and (y<0) then writeln('III')
else
if (x>0) and (y<0) then writeln('IV')
else
writeln('— . Точка лежит на оси.');
```

```
readln
end.
```

**Урок 31. Алгоритмы со структурой «выбор»**

**Цель:** научить применять еще один оператор ветвления — структуру «выбор».

**Требования к знаниям и умениям**

**Учащиеся должны знать:**

- конструкции ветвления, полную и неполную;

- правила записи арифметических выражений, операторы ввода, вывода, присваивания, работы с целыми числами;
- конструкцию «выбор».

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с разветвляющимися алгоритмами и операторами «выбор»;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Угр., § 4.2.3, с. 115; карточки: задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

- Какие структуры называются ветвлением?
- Какие конструкции ветвления бывают?

### III. Постановка целей урока

Что делать, если в задаче необходимо проверить множество условий? В этом случае, кроме оператора `if` в языке программирования Паскаль, предусмотрен оператор ветвления по ряду условий — `case`. Данный оператор используется реже из-за существующих ограничений для него.

### IV. Работа по теме урока

Оператор `case` служит для выбора одного из возможных вариантов в ситуации, которая может иметь несколько (более двух) возможных исходов. В зависимости от значения переменной (селектора) выполняется оператор, помеченный соответствующей константой. Если же ни одна константа не подходит, выполняется оператор, записанный после `else`.

**Оператор `case` имеет вид:**

```
Case порядковая_переменная of  
  константа_1: оператор_1;  
  константа_2: оператор_2;  
  константа_n: оператор_n;  
Else оператор;  
End;
```

Возможно использование сокращенной формы оператора `case`, в которой строка `else` отсутствует. В этом случае, если ни одна константа не совпадает со значением порядковой переменной, управление передается оператору, следующему за `end`. Например:

Case n of

1: writeln('красный');

2: writeln('зеленый');

3: writeln('белый');

End;

Запись типа 'a'...'z' включает диапазон значений от символа 'a' до символа 'z' включительно. Для решения задач на эту тему нам понадобится еще один тип переменных: `char` – простой тип данных, предназначенный для хранения одного символа в определенной кодировке.

## V. Практическая работа

**Задание 1.** Напишите программу, по которой выводится название школьного звена в зависимости от класса.

*Решение:*

```
var n: integer;
```

```
begin
```

```
write('Введите класс школы:');
```

```
readln(n);
```

```
case n of
```

```
1..4: writeln('Младшие классы.');
```

```
5..8: writeln('Средняя школа.');
```

```
9,11: writeln('Старшие классы. Выпускной.');
```

```
10: writeln('Старшие классы.');
```

```
else writeln('Error')
```

```
end;
```

```
end.
```

**Задание 2.** С помощью конструкции `case` сравните значение переменной `ch` с различными диапазонами значений. В случае совпадения выводится соответствующая надпись (цифра, английская буква, русская буква).

*Решение:*

```
Var ch: char;
```

```
begin
```

```
write('Введите символ: ');
```

```
readln(ch);
```

```
case ch of
```

```
'0'..'9': write('Это число');
```

```
'a'..'z', 'A'..'Z': write('Это английская буква');
```

```
'а'..'я', 'А'..'Я': write('Это русская буква')
```

```
end;
```

```
end.
```

*Примечание.* Символы перечисления помещаются в апострофы.

**Задание 3.** В зависимости от введенного символа  $L$ ,  $S$ ,  $V$  программа должна вычислять длину окружности; площадь круга; объем цилиндра.

*Решение:*

```
Var L, S, V, r, h: real; n: char;  
begin  
  write('Задайте радиус круга'); readln(r); write('Высоту цилиндра'); readln(h);  
  write('Введите символ:');  
  readln(n);  
  case ch of  
    L: write('Длина окружности L=',  $2*3.14*r$ );  
    S: write('Площадь круга S=',  $3.14*sqr(r)$ );  
    V: write('Объем цилиндра V=',  $3.14*sqr(r)*h$ );  
  end;  
end.
```

**Задание 4.** Напишите программу преобразования цифр в слова.

*Решение:*

```
Program Number4;  
Var a: integer;  
Begin  
  writeln('Введите цифру');  
  readln(a);  
  case a of  
    0: writeln('ноль');  
    1: writeln('один');  
    2: writeln('два');  
    3: writeln('три');  
    4: writeln('четыре');  
    5: writeln('пять');  
    6: writeln('шесть');  
    7: writeln('семь');  
    8: writeln('восемь');  
    9: writeln('девять')  
  else writeln('Это число не является цифрой');  
  end;  
End.
```

## VI. Подведение итогов урока

- Как применяется оператор выбора?
- Когда он особенно необходим?

## Домашнее задание

1. По номеру дня недели вывести его название.

2. Написать программу, которая по введенному числу из промежутка  $[0,24]$  выдает время суток.

*Ответы:*

1. Var a: integer;

Begin

writeln('Введите номер дня недели');

readln(a);

case a of

1: writeln('понедельник');

2: writeln('вторник');

3: writeln('среда');

4: writeln('четверг');

5: writeln('пятница');

6: writeln('суббота');

7: writeln('воскресенье');

else writeln('Это число не является номером дня недели');

end;

End.

2. Var a: real;

Begin

writeln('Введите время суток');

readln(a);

case a of

8,01..12,00: writeln('Утро');

12,01..17,00: writeln('День');

17,01..23,00: writeln('Вечер');

23,01..24,00: writeln('Ночь');

1,00..8,00: writeln('Ночь');

else writeln('Это число выходит за пределы интервала  $[0,24]$ ');

end;

End.

## Урок 32. Проверочная работа по теме «Условные алгоритмы»

**Цель:** проверить знания и умения по составлению программ с применением конструкций if и case.

**Требования к знаниям и умениям**

*Учащиеся должны знать:*

- конструкции ветвления, полную и неполную;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания, работы с целыми числами;
- конструкцию «выбор».



```
writeln(n);
```

end.

a) 3;

B) 1;

6)  $-1$ ;

г) 2.

**A6.** Какое значение будет иметь переменная  $z$  после выполнения операторов?

if x>0 then if y>0 then z:=1 else z:=2; при x=1; y=-1;

a) 1:

B) -1;

6) 2:

г) 3.

**В1.** Есть ли в следующих текстах пустые операторы? (В ответе перечислите буквы через запятую.)

a) if  $x > 0$  then  $x := 3$  else:  $y := x - 2$

6) if a mod b=0 then else b:=0

в) begin x:=2; v:=5; end

r) begin a:=true; b:=b or a end

д) begin if  $x=0$  then  $x:=2$ ;  $y:=x$

**B2.** Найдите ошибки в следующих операторах, запишите исправленные операторы в тетрадь.

a) begin if  $x^2 < 1$  then  $y := \sin x$ ;

$$x := \frac{x}{2}; \text{end};$$

6) if  $0 < x < 2$

```

then if  $y < 1$  then  $y := y + 3$  else  $y := y^2$ 

```

```
else y:=sqr(y)
```

B) if  $1 < x < 2$

then

$$x:=x+1; v:=0;$$

else

$$x:=0; v:=v+1;$$

r) if  $1 < x$  and  $x < 2$

then begin  $x:=x+1$ ;  $y:=0$  end;

```
else begin x:=0; y:=y+1 end
```

**Напишите программу для решения задачи.**

**С1.** Даны целые числа  $a, b, c$ . Если  $a \leq b \leq c$ , то все числа замените квадратами, если  $a > b > c$ , то каждое число замените наибольшим из них, в противном случае смените знак каждого числа.

**С2.** По номеру месяца выдайте квартал, которому он принадлежит.

### Вариант 2

**A1.** К какому типу относится переменная  $A:=0.25$ ?

a) char;

**B) boolean;**

6) integer;

г) real.



б) if  $-2 < y < 5$   
 then if  $y \geq 1$  then  $y = y^3$  else  $y = y^2$   
 else  $y = \text{sqr}(y)$   
 в) if  $-1 < y < 2$   
 then  
 $x := 3x + 1$ ;  $y := 0$ ;  
 else  
 $x := 0$ ;  $y := 5y + 1$ ;  
 г) if  $-1 < y$  and  $y < 2$   
 then begin  $x := x^3 + 1$ ;  $y := 0$  end  
 else begin  $x := 0$ ;  $y := y^2 + 1$  end

**Напишите программу для решения задачи.**

**С1.** Даны целые числа  $a, b, c$ . Если  $a \leq b \leq c$ , то все числа замените кубами, если  $a > b > c$ , то смените знак у каждого, в противном случае каждое замените наименьшим из них.

**С2.** По возрасту ребенка выдайте название группы, в которую он ходит.

**Ответы**

| Вариант 1                                 |                                                                                                                                                                                                                                                                                                                                                      |    |    |    |    | Вариант 2                                 |                                                                                                                                                                                                                                                                                                                                                                               |    |    |    |    |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|----|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|----|
| A1                                        | A2                                                                                                                                                                                                                                                                                                                                                   | A3 | A4 | A5 | A6 | A1                                        | A2                                                                                                                                                                                                                                                                                                                                                                            | A3 | A4 | A5 | A6 |
| б                                         | а                                                                                                                                                                                                                                                                                                                                                    | а  | а  | а  | б  | г                                         | а                                                                                                                                                                                                                                                                                                                                                                             | а  | в  | г  | а  |
| B1                                        | B2                                                                                                                                                                                                                                                                                                                                                   |    |    |    |    | B1                                        | B2                                                                                                                                                                                                                                                                                                                                                                            |    |    |    |    |
| а, б,<br>г, д                             | а) begin if $\text{sqr}(x) < 1$ then<br>y:=sin(x);<br>x:=x/2; end;<br>б) if (x>0) and (x<2)<br>then if y<1 then y=y+3 else<br>y:=sqr(y);<br>else y:=sqr(y);<br>в) if (x>1) and (x<2)<br>then<br>begin x:=x+1; y:=0; end<br>else<br>begin x:=0; y:=y+1; end;<br>г) if (x>1) and (x<2)<br>then begin x:=x+1; y:=0 end;<br>else begin x:=0; y:=y+1 end; |    |    |    |    | б,<br>г, д                                | а) begin if $\text{sqr}(y) > 1$ then<br>y:=cos(y);<br>y:=(y-3)/2; end;<br>б) if (y>-2) and (y<5)<br>then if y>=1 then y=y*y*y<br>else y:=sqr(y)<br>else y:=sqr(y)<br>в) if (y>-1) and (y<2)<br>then<br>begin x:=3*x+1; y:=0; end<br>else<br>begin x:=0; y:=5y+1; end<br>г) if (y>-1) and (y<2)<br>then begin x:=x*x*x+1; y:=0<br>end;<br>else begin x:=0; y:=sqr(y)+1<br>end; |    |    |    |    |
| C1                                        |                                                                                                                                                                                                                                                                                                                                                      |    |    |    |    | C1                                        |                                                                                                                                                                                                                                                                                                                                                                               |    |    |    |    |
| Program PR1;<br>Var<br>a, b, c : integer; |                                                                                                                                                                                                                                                                                                                                                      |    |    |    |    | Program PR1;<br>Var<br>a, b, c : integer; |                                                                                                                                                                                                                                                                                                                                                                               |    |    |    |    |

| C1                                                                                                                                                                                                                                                                                       | C1                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Begin writeln('Введите числа a, b, c'); readln(a,b,c); if (a&lt;=b) and (b&lt;=c) then begin a:=sqr(a); b:=sqr(b); c:=sqr(c); end else if (a&gt;b) and (b&gt;c) then begin b:=a; c:=a; end else begin a:=-a; b:=-b; c:=-c; end writeln(a,b,c); readln; End.</pre>                  | <pre> Begin writeln('Введите числа a, b, c'); readln(a,b,c); if (a&lt;=b) and (b&lt;=c) then begin a:=sqr(a)*a; b:=sqr(b)*a; c:=sqr(c)*a; end else if (a&gt;b) and (b&gt;c) then begin a:=-a; b:=-b; c:=-c; end else begin a:=c; b:=c; end writeln(a,b,c); readln; End.</pre> |
| C2                                                                                                                                                                                                                                                                                       | C2                                                                                                                                                                                                                                                                            |
| <pre> Program pr2; Var MONTH: integer; Begin writeln('Введите номер месяца'); readln(MONTH); case MONTH of 1, 2, 3 : writeln ('Первый квартал'); 4, 5, 6 : writeln ('Второй квартал'); 7, 8, 9 : writeln ('Третий квартал'); 10, 11, 12 : writeln ('Четвертый квартал'); end; End.</pre> | <pre> Program pr2; Var data: integer; Begin writeln('целое число лет'); readln(data); case data of 1, 2 : writeln ('ясли'); 3,4 : writeln ('младшая'); 5 : writeln ('старшая'); 6,7 : writeln ('подготовительная'); end; End.</pre>                                           |

#### IV. Подведение итогов урока

(Учитель выставляет оценки за урок.)

За каждое верно выполненное задание части А начисляется 0,5 балла, части В – 1 балл, части С – 2 балла.

**Критерии оценивания в зависимости  
от количества набранных баллов**

| Баллы   | Оценка |
|---------|--------|
| 8–9     | «5»    |
| 6–7     | «4»    |
| 4–5     | «3»    |
| Менее 4 | «2»    |

## Урок 33. Программирование циклов

**Цель:** освоить программирование циклов с предусловием.

**Требования к знаниям и умениям**

*Учащиеся должны знать:*

- конструкции цикла с предусловием;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами с предусловием;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 39, с. 211; Угр., § 4.2.4, с. 117; карточки: задания на урок, домашнее задание.

### Ход урока

#### I. Организационный момент

#### II. Постановка целей урока

В процессе решения множества задач часто требуется повторять те или иные действия. При этом бывают разные ситуации. Например:

- 1) количество повторений известно до выполнения тела цикла;
- 2) до выполнения тела цикла количество повторений неизвестно;
- 3) тело цикла должно выполниться хотя бы один раз.

Для разных ситуаций существуют различные циклические конструкции. В языке программирования Паскаль их три, в соответствии с пунктами, описанными выше. Самой универсальной и поэтому чаще других используемой конструкцией цикла является второй вариант (цикл while).

### III. Работа по теме урока

Цикл **while** является **циклом с предусловием**.

```
while <логическое выражение> do begin
```

```
  группа операторов
```

```
end;
```

В заголовке цикла находится логическое выражение. Если оно истинно, то тело цикла выполняется, если ложно — то нет. Если тело цикла было выполнено, то ход программы снова возвращается в заголовок цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение вернет true. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в заголовке цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация false. Иначе произойдет так называемое заикливание. Например, нужно распечатать  $n$  звездочек.

*Решение:*

```
Var i, n: integer;
```

```
begin
```

```
  write('Количество знаков: '); readln(n);
```

```
  i:=1;
```

```
  while i<=n do begin
```

```
    write('*'); i:=i+1; end;
```

```
end.
```

### IV. Практическая работа

**Задание 1.** Найдите сумму всех натуральных чисел из промежутка  $[1, m]$ .

*Решение:*

```
Program Summa;
```

```
Var i, m: integer; x, S: real;
```

```
Begin
```

```
  write('Сколько чисел для сложения?'); readln(m);
```

```
  S:=1; i:=1;
```

```
  while i<=m do begin S:=S+i; i:=i+1; end;
```

```
  write('Сумма равна ', s:5:2);
```

```
End.
```

**Задание 2.** Найдите сумму  $n$  чисел, вводимых с клавиатуры.

*Решение:*

```
Program Summa;
```

```
Var i, N : integer; x, S : real;
```

```
Begin
```

```
  write('Сколько чисел для сложения?'); readln(N);
```

```
  S:=0; i:=1;
```

```
while i<=N do begin
write('Введите ',i,'-е число '); readln(x); S:=S+x; i:=i+1;
end;
write('Сумма введенных чисел равна ',s:5:2);
End.
```

**Задание 3.** Выведите квадраты натуральных чисел, не превосходящих 50.

*Решение:*

```
var x,n: integer;
begin
write('n='); readln(n);
x:=1;
while sqr(x)<n do begin
write(sqr(x),' '); x:=x+1
end;
end.
```

**Задание 4.** Возведите число в степень.

*Решение:*

```
Var num, deg: integer; res: real; i: byte;
begin
write('Число:'); readln(num);
write('Степень:'); readln(deg);
res:=1; i:=0;
while i<abs(deg) do begin
res:=res*num; i:=i+1
end;
if deg<0 then
res:=1/res;
writeln(res:10:5);
end.
```

## V. Подведение итогов урока

- Как программируется цикл с предусловием на языке Паскаль?

### Домашнее задание

1. Вывести все четные числа, начиная с числа  $N$  и кончая числом  $M$ . Числа  $N$  и  $M$  задает пользователь.
2. Вводятся 14 чисел. Определить, сколько среди них положительных (включая 0) и сколько отрицательных. (Числа вводятся в одну переменную в цикле.)

*Ответы:*

1. Program dz1;  
var x, n, m: integer;

```
begin  
write('n='); readln(n); write('m='); readln(m);  
x:=n;  
while x<=m do begin  
if xmod2=0 then write(x, ' ');  
x:=x+1  
end;  
end.
```

2. Program dz2;

Var i, N : integer; x, S : real;

Begin

S1:=0; S2:=0; i:=1;

while i<=14 do begin

write('Введите ', i, '-е число '); readln(x);

if x>=0 then s1:=s1+1 else s2:=s2+1; i:=i+1;

end;

write('положительных чисел', s1:5, 'отрицательных чисел', s2:5);

End.

## Урок 34. Программирование циклов

**Цель:** освоить программирование циклов с постусловием.

**Требования к знаниям и умениям**

*Учащиеся должны знать:*

- конструкции цикла с постусловием;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами с постусловием;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 39, 40, с. 211; Угр., § 4.2.4, с. 117; карточки: проверочная работа «Цикл с предусловием», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

**Проверочная работа «Цикл с предусловием»**

1. Запишите конструкцию цикла с предусловием. В каких случаях применяется цикл с предусловием?

2. Запишите правильную структуру последовательности операторов.

```
a:=2; b:=1; while a+b<8 do begin a:=a+1; b:=b+1 end; s:=a+b.
```

Сколько раз будет повторен цикл и какими будут значения переменных  $a$ ,  $b$ ,  $s$  после его завершения?

3. Запишите правильную структуру последовательности операторов. Какими будут значения переменных  $a$  и  $b$  после выполнения операторов?

```
a:=1; b:=1; while a<=8 do a:=a+1; b:=b+1.
```

4. Дана последовательность операторов, вычисляющих факториал  $f$  числа  $n$ , которая содержит пять ошибок. Найдите эти ошибки, предварительно записав операторы в правильном виде.

```
k:=1; f:=0;  
while k<n do f=f*k  
k:=k+1,
```

*Примечание.* Факториалом натурального числа называется произведение всех натуральных чисел от 1 до этого числа, т. е.  $n! = 1 * 2 * 3 * \dots * (n-2) * (n-1) * n$ .

5. Найдите и исправьте ошибки в следующем фрагменте программы, определяющей для заданного натурального числа  $n$  число, записанное цифрами числа  $n$  в обратном порядке.

```
p:=n;  
while p>=0 do  
begin  
a:=a+p mod 10;  
p:=p div 10  
end;
```

*Ответы:*

1. while <логическое выражение> do begin группа операторов end; Когда до выполнения цикла количество повторов неизвестно.

```
2. a:=2;  
b:=1;  
while a+b<8 do  
begin  
a:=a+1;  
b:=b+1  
end;  
s:=a+b.
```

2 раза;  $a = 5$ ,  $b = 3$ ,  $s = 8$ .

```
3. a:=2;  
b:=3;  
while a<=7 do
```

```
a:=a+1;  
b:=b+1;  
a = 8, b = 4.  
4. k:=1; f:=1;  
while k<=n do f:=f*k;  
k:=k+1;  
end;  
5. read(n);  
p:=n; a:=0;  
while p>=0 do  
begin  
a:=a*10+p mod 10;  
p:=p div 10  
end;
```

### III. Постановка целей урока

Вы уже умеете организовать цикл при помощи оператора `while`. Напомним, что при выполнении этого оператора компьютер вычисляет значение условия. Если условие истинно, то исполнительная часть оператора `while` будет выполняться до тех пор, пока это условие не примет значение `false`. Если значение условия есть `false` в самом начале, то исполнительная часть оператора `while` вообще не будет выполняться.

Иногда при решении задач возникает необходимость выполнить тело цикла хотя бы один раз, а потом исследовать условие, повторять ли его еще раз. Эту задачу выполнит другой вид цикла `Repeat`.

### IV. Работа по теме урока

**repeat** повторяй  
операторы

**until** <условие>; до тех пор, пока условие не будет верным

Есть небольшое отличие в организации цикла `repeat` по сравнению с `while`: для выполнения в цикле `repeat` нескольких операторов не следует помещать эти операторы в операторные скобки `begin ... end`. Зарезервированные слова `repeat` и `until` действуют как операторные скобки.

Конструкция `repeat ... until` работает аналогично циклу `while`. Различие заключается в том, что цикл `while` проверяет условие до выполнения действий, в то время как `repeat` проверяет условие после выполнения действий, это гарантирует хотя бы одно выполнение действий до завершения цикла. Например:

```
а) repeat  
read (Number);
```

```
Sum := Sum+Number;  
until Number=-1  
b) repeat  
i := i+1;  
writeln (Sqr(i))  
until Number=-1
```

## V. Практическая работа

**Задание 1.** Определите, является ли данное число простым.

*Примечание.* Простым называется число, которое не имеет делителей, кроме 1 и самого себя.

*Решение:*

```
Var i, Number : integer; {возможный делитель и исследуемое  
число}
```

```
Begin  
writeln ('Какое число должно быть проверено?'); read  
(Number);  
i:=1;  
repeat  
i:=i+1;  
until Number mod i=0;  
if Number=i then writeln(Number, 'является простым')  
else writeln(Number, 'делится на', i);  
End.
```

**Задание 2.** Выведите на экран строку из звездочек.

*Решение:*

```
Var i, n: integer;  
begin  
write('Количество знаков:'); readln(n);  
i:=1;  
repeat  
write('*');  
i:=i+1  
until i>n;  
end.
```

**Задание 3.** Выведите квадраты натуральных чисел, не превосходящих 50.

*Решение:*

```
var x, n: integer;  
begin  
write('Enter n:'); readln(n);  
x:=1;  
repeat
```

```
y:=x*x;  
if y<n then write(y, ' ');  
x:=x+1;  
until y>n;  
end.
```

**Задание 4.** Найдите НОД двух чисел.

*Решение:*

```
Var x, y: integer;  
Begin  
Writeln('введите два числа'); readln(x, y);  
Repeat  
If x>y then x:=x mod y else y:=y mod x;  
Until (x=0) Or (y=0);  
Writeln ('НОД=', x+y);  
End.
```

## VI. Подведение итогов урока

- Как программируется цикл с постусловием на языке Паскаль?
- Для каких целей лучше его использовать?

## Домашнее задание

1. Составить программу нахождения НОД трех чисел.  
 $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$ .

2. Составить программу нахождения НОК двух чисел, используя формулу  $A \times B = \text{НОД}(A, B) \times \text{НОК}(A, B)$ .

*Ответы:*

```
1. program dz1;  
Var A, B, C, nod: integer;  
Begin  
Readln(A, B, C);  
Repeat  
If A>B then A:=A mod B else B:=B mod A;  
Until (A=0) or (B=0);  
Writeln('НОД= ', A+B);  
nod:=A+B;  
Repeat  
If nod>C then nod:=nod mod C else C:=C mod nod;  
Until (nod=0) or (C=0);  
Writeln ('НОД= ', nod+C);  
End.
```

```
2. program dz2;  
var n, m, i, nok: integer;  
begin
```

```
write('введите два числа'); readln(n, m);
if (m<>0) and (n<>0) then
begin
nok:=n*m; i:=nok;
while i>0 do begin
if (i mod n=0) and (i mod m=0) then nok:=i;
i:=i-1; end;
writeln('NOK',m,'i',n,'raven',nok)
end else writeln('Na nol delit nelzya');
end.
```

## Урок 35. Программирование циклов

**Цель:** освоить программирование циклов со счетчиком.

### **Требования к знаниям и умениям**

*Учащиеся должны знать:*

- конструкции цикла со счетчиком;
- правила записи арифметических выражений, операторы ввода, вывода, присваивания.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами со счетчиком;
- использовать операторы ввода, вывода, присваивания.

**Программно-дидактическое оснащение:** Сем., § 39, 40, с. 211; карточки: проверочная работа «Цикл с постусловием», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

#### **Проверочная работа «Цикл с постусловием»**

1. Что будет напечатано на экране в результате выполнения следующего фрагмента программы?

```
A:=1; B:=5;
repeat
writeln(A, B); A:=A+1;
until A>=B;
```

2. Определите значение переменной *s* после выполнения фрагмента алгоритма:

```
s:=0; i:=1; repeat s:=s+5 div i; i:=i+1 until i>=5.
```

3. Запишите правильную структуру последовательности операторов. Какими будут значения переменных  $a$  и  $b$  после выполнения операторов:  $a:=1$ ;  $b:=1$ ; repeat  $a:=a+1$  until  $a>8$ ;  $b:=b+1$ ?

4. Дана последовательность операторов, вычисляющих факториал  $f$  числа  $n$ , которая содержит ошибки. Найдите эти ошибки, предварительно записав операторы в правильном виде.

```
k:=1; f:=0;
repeat f=f*k
k:=k+1, until k<n
```

5. Найдите и исправьте ошибки в следующем фрагменте программы, определяющей для заданного натурального числа  $n$  число, записанное цифрами числа  $n$  в обратном порядке.

```
p:=n;
repeat do
begin
a:=a+p mod 10;
p:=p div 10
end;
until p>=0
```

*Ответы:*

1. 1 5; 2 5; 3 5; 4 5; 5 5.

2. 10.

3.  $a:=1$ ;  $b:=1$ ;

```
repeat
a:=a+1;
until a>=8;
```

$b:=b+1$ ;

$a = 8, b = 2$ .

4.  $k:=1$ ;  $f:=1$ ;

```
repeat
f=f*k;
k:=k+1;
until k>n;
```

5. read( $n$ );

$p:=n$ ;  $a:=0$ ;

```
repeat
a:=a*10+p Mod 10;
p:=p Div 10
until p<0;
```

### III. Постановка целей урока

Циклы со счетчиком составляют такой класс, в котором выполнение исполнительской части должно повторяться заранее определенное число раз. Циклы со счетчиком используются до-

вольно часто, и поэтому в языке Паскаль для этих целей имеется специальная конструкция. Можно, конечно, циклы со счетчиком моделировать при помощи операторов **while** и **repeat**, но структура цикла со счетчиком проще.

#### IV. Работа по теме урока

**Общая форма записи цикла со счетчиком:**

```
for i:=A to B do
```

```
begin
```

```
...
```

```
end;
```

или

```
for i := A downto B do
```

```
begin
```

```
...
```

```
end;
```

Здесь  $i$  — управляющая переменная или переменная цикла,  $A$  — начальное значение переменной цикла,  $B$  — конечное значение переменной цикла.

В языке программирования Паскаль отсчет идет всегда с шагом, равным 1 (для **for...to**), или уменьшается на 1 (для **for...downto**). Легко понять, что количество повторений цикла определяется разностью между вторым и первым значением плюс единица. Тело цикла не должно содержать выражений, изменяющих счетчик, и тип управляющей переменной не должен быть **real**. Управляющая переменная должна описываться, как и любая другая переменная. Обычно переменная цикла имеет тип **integer**.

Нужно понимать, что количество итераций цикла **for** известно именно до его выполнения, но не до выполнения всей программы.

#### V. Практическая работа

**Задание 1.** Вычислите сумму ряда чисел  $1/1^2 + 1/2^2 + 1/3^2 + \dots + 1/n^2$ , где  $n$  определяется пользователем.

*Решение:*

```
Var n, i: integer; sum: real;
```

```
begin
```

```
write('n='); readln(n);
```

```
sum:=0;
```

```
for i:=1 to n do sum:=sum+1/(i*i);
```

```
writeln(sum:6:5);
```

```
end.
```

**Задание 2.** Выведите на экран строку из звездочек.

*Решение:*

```
Var i, n: integer;
```

```
begin  
write('Количество знаков: '); readln(n);  
for i:=1 to n do write ('(*)');  
end.
```

**Задание 3.** Выведите квадраты натуральных чисел, не превосходящих 50.

*Решение:*

```
Var i, y: integer; begin  
write('Enter n: '); readln(n);  
for i:=1 to n do begin y:=i*i; write(y, ' ') end;  
end.
```

**Задание 4.** Выведите на экран ряд чисел Фибоначчи, состоящий из  $n$  элементов.

*Решение:*

```
Var a, b, c, i, n: integer;  
begin  
write('n='); readln(n);  
a:=0; write(a, ' '); b:=1; write(b, ' ');  
for i:=3 to n do begin  
write(a+b, ' '); c:=b; b:=a+b; a:=c  
end;  
end.
```

## VI. Подведение итогов урока

- Как программируется цикл со счетчиком на языке Паскаль?
- Для каких целей лучше его использовать?

## Домашнее задание

1. Составить программу вычисления степени числа  $a$  с натуральным показателем  $n$ .
2. Найти значение выражения  $1*1+2*2+...+n*n$ .

*Ответы:*

```
1. program dz1;  
var a, n, i, p: integer;  
begin  
write('n='); readln(n); write('a='); readln(a);  
p:=1;  
for i:=1 to n do p:=p*a;  
write('p=', p);  
end.
```

```
2. program dz2;  
var n, s, i: integer;  
begin  
write('n='); readln(n);
```

```
s:=0;  
for i:=1 to n do s:=s+i*i;  
writeln('s=',s);  
end.
```

## Урок 36. Массивы в Паскале

**Цель:** ввести понятие «массив».

**Требования к знаниям и умениям**

*Учащиеся должны знать:*

- что такое массив;
- правила описания и ввода значений в массив;
- циклические конструкции;
- нахождение суммы, произведения, среднего значения элементов массива.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами со счетчиком;
- использовать операторы ввода, вывода, присваивания;
- описывать и заполнять массив данными.

**Программно-дидактическое оснащение:** Сем., § 41, 42, с. 222; карточки: проверочная работа «Цикл со счетчиком», задания на урок, домашнее задание.

### Ход урока

#### I. Организационный момент

#### II. Актуализация знаний

**Проверочная работа «Цикл со счетчиком»**

1. Сколько раз выполнится цикл и что будет выдано на печать?  
var i,k:integer;

...

k:=0;

for i:=-10 to -9 do k:=i; write(k);

а) 1 раз, -10;

в) 2 раза, -10;

б) 1 раз, -9;

г) 2 раза, -9.

2. Как называется цикл for?

а) цикл с предусловием;

в) цикл с постусловием;

б) цикл со счетчиком;

г) ветвление.

3. Что выведется на экран в результате выполнения фрагмента программы?

```
s:=0;
```

```
for j:=10 to 15 do begin
  s:=s+2*j; write('j=', j:2, 's=', s:4) end;
```

4. Что выведется на экран в результате выполнения фрагмента программы?

```
p:=1;
for k:=5 downto 1 do begin
  p:=p*k end;
write('k=', k:2, 'p=', p:3)
```

5. Тело цикла для вычисления значений функций  $f(x) = x^4$  с шагом 0,1 будет иметь вид:

- а)  $f := x * x * x * x$ ;  $x := x + 1$ ; writeln (f,x);
- б)  $f := x^4$ ;  $x := x + 1$ ; writeln (f, x);
- в)  $f := x * x * x * x$ ;  $x := x + 0.1$ ; writeln (f,x);
- г)  $f := x * x * x * x$ ;  $x := x + 0.1$ ; writeln (f,x).

Ответы: 1. г; 2. б; 3.  $j = 10, s = 20$ ;  $j = 11, s = 42$ ;  $j = 12, s = 66$ ;  $j = 13, s = 92$ ;  $j = 14, s = 120$ ;  $j = 15, s = 150$ ; 4.  $k = 1$ ;  $p = 120$ ; 5. в.

### III. Постановка целей урока

Предположим, что программа работает с большим количеством однотипных данных. Нужно обработать около ста разных целых чисел, выполнив над ними те или иные вычисления.

Представьте 100 переменных в программе. И для каждой переменной нужно написать одно и то же выражение вычисления значения. Это очень неэффективно. Есть более простое решение — это использование такой структуры (типа) данных, как массив.

### IV. Работа по теме урока

Массив представляет собой последовательность ячеек памяти, в которых хранятся однотипные данные. При этом существует всего одно имя переменной, связанной с массивом, а обращение к конкретной ячейке происходит по ее индексу (номеру) в массиве. Нужно четко понимать, что индекс ячейки массива не является ее содержимым. Содержимым являются хранимые в ячейках данные, а индексы только указывают на них. Действия в программе над массивом осуществляются путем использования имени переменной, связанной с областью данных, которая отведена под массив.

**Массив** — это именованная группа однотипных данных, хранящихся в последовательных ячейках памяти. Каждая ячейка содержит элемент массива. Элементы нумеруются по порядку, но необязательно начиная с единицы (хотя в языке программирования Паскаль чаще всего именно с нее). Порядковый номер элемента массива называется индексом этого элемента.

```
var <имя_массива> : <имя_типа> ;
```

или

var<имя\_массива>: array [t1..tn] of <тип\_элементов\_массива>;  
t1,...,tn — тип индекса (перечислимый или интервальный).

Помним, что *все элементы определенного массива имеют один и тот же тип*. У разных массивов типы данных могут различаться. Например, один массив может состоять из чисел типа integer, а другой — из чисел типа real.

Индексы элементов массива обычно целые числа, однако могут быть и символами, а также описываться другими порядковыми типами.

Массив можно создать несколькими способами.

Обращение к определенному элементу массива осуществляется путем указания имени переменной массива и в квадратных скобках — индекса элемента.

Простой массив является *одномерным*. Он представляет собой линейную структуру.

```
var ch: array [1..11] of real; i: integer;
begin
  for i := 1 to 11 do read (ch[i]); {заполнение массива вручную}
  for i := 1 to 11 do write (ch[i]:3);
end.
```

В примере выделяется область памяти под массив из 11 чисел. Их индексы от 1 до 11. В процессе выполнения программы пользователь вводит 11 любых чисел, которые записываются в ячейки массива. Текущее значение переменной *i* в цикле for используется в качестве индекса массива. Второй цикл for отвечает за вывод элементов массива на экран.

## V. Практическая работа

**Задание 1.** Найдите сумму, произведение и среднее значение элементов массива.

*Примечание.* Задачи данного типа сводятся к поэлементному перебору массива и выполнению соответствующих алгебраических операций.

Программа на языке Паскаль:

```
var arr: array[1..100] of real; {под массив отводится 100 элементов, но пользователь сам задает нужное количество меньше 100}
n: integer; sum, op, sr: real; i: integer;
begin
  write('Количество элементов: '); readln(n);
  write('Укажите значения: ');
  for i:=1 to n do read(arr[i]); {пользователь сам заполняет массив значениями}
  sum:=0; {пустая переменная для суммы}
```

```

for i:=1 to n do sum:=sum+arr[i]; {сумма находится путем при-
бавления к предыдущему значению нового значения}
ор:=1; {пустая переменная для произведения}
for i:=1 to n do ор:=ор*arr[i]; {произведение находится путем
умножения предыдущего значения на новое значение}
sr:=sum/n;
writeln('Сумма элементов', sum:10:2);
writeln('Произведение элементов', ор:20:2);
writeln('Среднее значение элементов', sr:20:2);
end.

```

**Задание 2.** Заполните массив числами, которые вводит пользователь, и вычислите их сумму. Если пользователь вводит ноль или превышен размер массива, то запросы на ввод должны прекратиться.

*Примечание.* Для решения подобного рода задач иногда бывает уместно использовать цикл с постусловием repeat.

Программа на языке Паскаль:

```

var arr: array[1..10] of integer;
sum, i: integer;
begin
i:=1; sum:=0;
repeat
write('Число: '); readln(arr[i]);
sum:= sum+arr[i]; i:=i+1
until (arr[i-1]=0) or (i>n);
for i:=1 to n do write(arr[i], ' ');
writeln('sum = ', sum);
end.

```

## VI. Подведение итогов урока

- Что такое массив?
- Придумайте примеры данных, которые можно организовать в виде массива.

## Домашнее задание

Подсчитать средний балл за экзамен по математике у учащихся, сдавших его.

*Ответ:*

```

var A: array[1..30] of integer;
i, x, y: integer; s: real;
begin
for i:=1 to N do readln(A[i]);
x:=0; y:=0;
for i:=1 to N do
if A[i]>2 then begin

```

```
x:= x+1;  
y:= y+A[i];  
end;  
s:=y/x;  
writeln('Средний балл ', s:10:3);  
end.
```

## Урок 37. Решение задач с использованием массивов

**Цель:** освоить способ заполнения массивов, используя датчик случайных чисел.

### *Требования к знаниям и умениям*

*Учащиеся должны знать:*

- что такое массив;
- правила описания и ввода значений в массив;
- циклические конструкции;
- нахождение элемента массива с данным свойством;
- датчик случайных чисел `random(x)`.

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами со счетчиком;
- использовать операторы ввода, вывода, присваивания;
- описывать и заполнять массив данными, используя датчик случайных чисел.

**Программно-дидактическое оснащение:** Сем., § 43, с. 232; карточки: проверочная работа «Массивы в Паскале», задания на урок, домашнее задание.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

#### **Проверочная работа «Массивы в Паскале»**

1. Значения двух массивов  $A[1..100]$  и  $B[1..100]$  задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do A[n]:= n - 20;
```

```
for n:=1 to 100 do B[n]:= A[n]*n;
```

Сколько элементов массива  $B$  будет иметь положительные значения?

а) 20;

в) 80;

б) 50;

г) 100.

2. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do  
  A[i]:= i + 1;  
for i:=1 to 10 do  
  A[i]:= A[i-1];
```

Как изменяются элементы этого массива?

- а) все элементы, кроме последнего, сдвигаются на 1 элемент вправо;
- б) все элементы, кроме первого, сдвигаются на 1 элемент влево;
- в) все элементы окажутся равны 1;
- г) все элементы окажутся равны своим индексам.

3. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do  
  A[i]:= i + 1;  
for i:=10 downto 0 do  
  A[i]:= A[10-i];
```

Чему будут равны элементы этого массива?

- а) 10 9 8 7 6 5 4 3 2 1 0;
- б) 11 10 9 8 7 6 5 4 3 2 1;
- в) 11 10 9 8 7 6 7 8 9 10 11;
- г) 1 2 3 4 5 6 5 4 3 2 1.

4. Значения элементов двух массивов  $A[1..100]$  и  $B[1..100]$  задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do  
  A[n]:=n-50;  
for n:=1 to 100 do  
  B[101-n]:=A[n]*A[n];
```

Какой элемент массива  $B$  будет наименьшим?

- а)  $B[1]$ ;                                      в)  $B[51]$ ;
- б)  $B[50]$ ;                                    г)  $B[100]$ .

5. Дан фрагмент программы, обрабатывающий массив  $A$  из 10 элементов:

```
n:=10;  
for i:=1 to n do A[i]:=i;  
j:=1;  
for i:=1 to n-1 do  
  if A[i]<A[i+1] then j:=j+1;
```

Чему будет равно значение переменной  $j$  после выполнения этого алгоритма?

а) 1;

в) 10;

б) 2;

г) 11.

*Ответы:* 1в; 2в; 3г; 4в; 5в.

### III. Постановка целей урока

В языках программирования, как правило, имеется аналог игрального кубика, который позволяет получать случайные числа. Он называется датчиком случайных чисел. Это стандартная функция `random(x)`. С помощью его мы можем организовать заполнение массива.

### IV. Работа по теме урока

Результатом оператора `random(x)` будет целое число в диапазоне  $[0, x]$ . Но, чтобы оператор `random` при многократном выполнении выдавал разную последовательность чисел, необходимо изначально описать процедуру `randomize`, которая случайным образом устанавливает начальное состояние датчика.

### V. Практическая работа

**Задание 1.** Замените отрицательные элементы на противоположные по знаку.

*Решение:*

```
Var i: integer;
```

```
Begin
```

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

```
if m[i]<0 then m[i]:=-1*m[i];
```

```
End;
```

**Задание 2.** Найдите и выведите на экран номера четных элементов.

*Решение:*

```
Var i: integer;
```

```
Begin
```

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

```
if m[i] mod 2 = 0 then write(i:5);
```

```
End;
```

**Задание 3.** Найдите количество положительных и отрицательных элементов в данном массиве.

Опишем процедуру, которой будем отправлять три параметра — массив и два счетчика, один для элементов, больших нуля, а второй — для отрицательных элементов.

*Решение:*

```
Var i: integer;
```

```
Begin
```

```
k1:=0; k2:=0;  
for i:=1 to n do  
  if m[i]>0 then Inc(k1) else  
  if m[i]<0 then Inc(k2);  
End;
```

## VI. Подведение итогов урока

- Какие значения может принимать целочисленная переменная  $Y$ , если в программе записано  $Y:=10+\text{random}(5)$ ? (10; 15.)
- Как, используя функцию  $\text{random}(x)$ , можно получать числа в диапазонах: от 1 до 10, от  $-10$  до  $+10$ , от 50 до 100? ( $\text{random}(9)+1$ ,  $\text{random}(20)-10$ ,  $\text{random}(50)+50$ .)

## Домашнее задание

Найти максимальный элемент массива.

*Ответ:*

```
Var a: array[1..50] of integer;  
i, num, index: integer;  
begin  
  randomize;  
  for i:=1 to 50 do begin  
    a[i]:=random(100);  
    write(a[i]:3);  
  end;  
  index:=1; num:=a[1];  
  for i:=2 to 50 do  
    if a[i]>num then begin  
      index:=i;  
      num:=a[i];  
    end;  
  writeln('Max = ', num);  
  writeln('position: ', index);  
  readln;  
end.
```

## Урок 38. Решение задач с использованием массивов

**Цель:** освоить «метод пузырька» для сортировки массива.

**Требования к знаниям и умениям**

*Учащиеся должны знать:*

- что такое массив;
- правила описания и ввода значений в массив;
- циклические конструкции;

- нахождение элементов массива с данным свойством;
- датчик случайных чисел `random(x)`;
- сортировку массива;
- «метод пузырька».

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами со счетчиком;
- использовать операторы ввода, вывода, присваивания;
- описывать и заполнять массив данными, используя датчик случайных чисел;
- сортировать заданный массив.

**Программно-дидактическое оснащение:** Сем., § 6.2, с. 323; карточки с заданиями на урок и домашним заданием.

## Ход урока

### I. Организационный момент

### II. Актуализация знаний

Составьте программу заполнения массива из 100 чисел случайными значениями из диапазона от  $-20$  до  $20$ .

*Решение:*

```
Var a: array[1..100] of integer;  
i: integer;  
begin  
  randomize;  
  for i:=1 to 100 do begin  
    a[i]:=random(40)-20;  
    write(a[i]:3);  
  end;
```

### III. Постановка целей урока

Известно, что данные в таблице можно отсортировать по возрастанию или убыванию значений в столбцах. В программировании существует несколько способов для решения этой задачи. Мы рассмотрим один из самых распространенных — «метод пузырька».

### IV. Работа по теме урока

Суть «метода пузырька»:

1. Последовательно сравниваются пары соседних чисел и упорядочиваются по убыванию (возрастанию). В результате минимальное (максимальное) число всплывает в конце массива.

2. Упорядочивается массив еще  $n-1$  раз, где  $n$  — количество элементов массива. Каждый раз рассматривается на один элемент меньше, так как упорядоченные элементы всплывают в конце.

3. Следовательно, циклы, реализующие проходы, сами должны циклически повторяться. Структура алгоритма должна представлять собой два вложенных цикла.

Давайте вспомним, как поменять два числа местами. ( $a:=b$ ;  $b:=c$ ;  $c:=a$ .)

Программа на языке Паскаль:

```
Program primer;
Var b: array[1..20] of integer;
I, k, x: integer;
Begin
  randomize;
  for i:=1 to 20 do begin
    a[i]:=random(40)-20;
    write(a[i]:3);
  end;
  for k:=1 to 19 do
    for i:=1 to 20-k do
      if (b[i]>b[i+1]) then
        begin
          x:=b[i]; b[i]:=b[i+1]; b[i+1]:=x;
        end;
    for i:=1 to 20 do writeln(b[i]:2);
  readln;
end.
```

## V. Практическая работа

На соревнованиях по прыжкам в длину получен массив  $b(20)$ . Определите три лучших результата. Массив сформируйте случайным образом.

*Решение:*

```
Program primer;
Var b: array[1..20] of integer;
I, k, x: integer;
Begin
  randomize;
  writeln('Результаты прыжков спортсменов:');
  for i:=1 to 20 do begin
    a[i]:=random(10);
    write(a[i]:3);
  end;
```

```
for k:=1 to 20 do
for i:=1 to 19 do
if (b[i]>b[i+1]) then
begin
x:=b[i]; b[i]:=b[i+1]; b[i+1]:=x;
end;
writeln('Лучшие результаты:');
for i:=1 to 3 do writeln(i, '-е место:', b(i):2, 'см');
readln;
end.
```

## VI. Подведение итогов урока

- Как пояснить название метода сортировки – «метод пузырька»?

## Домашнее задание

Подготовиться к контрольной работе.

# Урок 39. Контрольная работа по теме «Основы алгоритмизации и программирования»

**Цель:** проверить полученные знания по теме «Основы алгоритмизации и программирования».

### **Требования к знаниям и умениям**

*Учащиеся должны знать:*

- что такое массив;
- правила описания и ввода значений в массив;
- циклические конструкции;
- нахождение элементов массива с данным свойством;
- датчик случайных чисел `random(x)`;
- правила сортировки массива;
- «метод пузырька».

*Учащиеся должны уметь:*

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- составлять программы с циклами со счетчиком;
- использовать операторы ввода, вывода, присваивания;
- описывать и заполнять массив данными, используя датчик случайных чисел;
- сортировать заданный массив.

**Программно-дидактическое оснащение:** карточки с текстами контрольной работы в двух вариантах.

## Ход урока

### I. Организационный момент

### II. Постановка целей урока

Сегодня на уроке мы подведем итог по теме «Основы алгоритмизации и программирования».

### III. Работа по теме урока

#### *Вариант 1*

**A1.** В результате выполнения программы `a:='x'; b:=1; c:=2; writeln('b+c=3',a); write(a,'x',b,c);` будет напечатано:

- |                |                 |
|----------------|-----------------|
| а) 1+23xxx12;  | в) truexxx12;   |
| б) b+c=3xxx12; | г) b+c=3x;xx12. |

**A2.** Служебное слово `const` в программе на языке Паскаль фиксирует начало раздела программы, содержащего:

- а) описание переменных;
- б) описание меток;
- в) описание констант;
- г) описание сложных типов данных.

**A3.** Чему будет равняться  $k$  после выполнения программы?

```
var m,k:byte;  
begin  
  m:=1; k:=0;  
  while (m<5) do m:=m+1; k:=k+m;  
  writeln(k)  
end.
```

- |       |       |
|-------|-------|
| а) 2; | в) 4; |
| б) 3; | г) 5. |

**A4.** Сколько раз выполнится оператор `while` в следующем фрагменте программы?

```
s:=0; k:=1;  
while s<=7 do begin k:=k+1; s:=s+k end;
```

- |       |       |
|-------|-------|
| а) 1; | в) 3; |
| б) 2; | г) 4. |

**A5.** Чему будет равняться  $x$  после выполнения фрагмента программы?

```
x:=0;  
repeat x:=x+8 until x>=21;
```

- |        |        |
|--------|--------|
| а) 21; | в) 23; |
| б) 22; | г) 24. |

**A6.** Сколько раз выполнится цикл и что будет выдано на печать?

```
var i, k:integer;
```

...

k:=0;

for i:=10 downto 10 do k:=i; write(k);

а) 1 раз, 10;

в) 11 раз, 0;

б) 10 раз, 10;

г) 2 раза, 0.

**A7.** Дано описание массива mas:array[-22..0] of integer. Сколько элементов содержит данный массив?

а) 20;

в) 22;

б) 21;

г) 23.

**A8.** Значения двух массивов  $A[1..100]$  и  $B[1..100]$  задаются с помощью следующего фрагмента программы:

нц для n от 1 до 100  $A[n]=n-10$  кцнц для n от 1 до 100  $B[n]=A[n]*n$  кц

Сколько элементов массива  $B$  будет иметь положительные значения?

а) 10;

в) 90;

б) 30;

г) 100.

**B1.** Какую задачу решает следующая программа?

Var x,y,N:integer;

begin

write('Enter N: ');

readln(N);

x:=1; while y&lt;N do

begin

y:=x\*x;

if y&lt;n then write(y, ' ');

x:=x+1;

end;

writeln;

end.

**B2.** Каким условиям должны удовлетворять значения переменной  $k$ , чтобы следующие циклы были бесконечны: while  $k < > 0$  do  $k:=1+k$ ?

**B3.** Определите, что делает следующая программа и какой результат она выдаст на заданном массиве (1,4,0,-1,6).

const n=5;

var a:array[1..n] of integer; i, b:integer;

begin

b:=a[1];

for i:=2 to n do if a[i]&lt;b then b:=a[i];

writeln(b)

end.

**C1.** Допустим, имеется одномерный массив, содержащий числа от 0 до 49 включительно. Требуется исключить из него все элементы, значения которых меньше 15.

**Вариант 2**

**A1.** В результате выполнения программы  $a:='po'; b:=4; c:=6;$   
`writeln('b-c=-2',a); write(a,'po',b,c);` будет напечатано:

а)  $b-c=-2$ ророро46;

б) 4-6-2ророро46;

в) trueророро46;

г)  $b-c=-2$ po;poro12.

**A2.** Служебное слово `label` в программе на языке Паскаль фиксирует начало раздела программы, содержащего:

а) описание переменных;

б) описание меток;

в) описание констант;

г) описание сложных типов данных.

**A3.** Чему будет равно значение  $k$  после выполнения программы?

`Var m, k:integer;`

`begin`

`m:=0; k:=1;`

`while (m<5) do m:=m+1; k:=k+m;`

`writeln(k)`

`end.`

а) 2;

в) 4;

б) 3;

г) 5.

**A4.** Сколько раз выполнится оператор `while` в следующем фрагменте программы?

`S:=0; k:=1;`

`while s<=11 do begin k:=k+1; s:=s+k end;`

а) 1;

в) 3;

б) 2;

г) 4.

**A5.** Чему будет равняться  $x$  после выполнения фрагмента программы?

`x:=0;`

`repeat x:=x+7 until x>=19;`

а) 21;

в) 23;

б) 22;

г) 24.

**A6.** Сколько раз выполнится цикл и что будет выдано на печать?

`Var i, k:integer;`

`...`

`k:=0;`

for i:=5 to 5 do k:=i; write(k);

а) 1 раз, 5;

в) 6 раз, 0;

б) 5 раз, 5;

г) 2 раза, 0.

**A7.** Дано описание массива mas: array[-20..0] of integer. Сколько элементов содержит данный массив?

а) 20;

в) 22;

б) 21;

г) 23.

**A8.** Значения двух массивов  $A[1..100]$  и  $B[1..100]$  задаются с помощью следующего фрагмента программы:

нц для n от 1 до 100  $A[n] = n - 20$  кц

нц для n от 1 до 100  $B[n] = A[n] * n$  кц

Сколько элементов массива  $B$  будет иметь положительные значения?

а) 10;

в) 90;

б) 80;

г) 100.

**B1.** Какую задачу решает следующая программа?

Var x,y,N:integer;

begin

write('Enter N: ');

readln(N);

x:=1; while y<N do

begin

y:=x\*x\*x;

if y<n then write(y, ' ');

x:=x+1;

end;

writeln;

end.

**B2.** Каким условиям должны удовлетворять значения переменной  $k$ , чтобы следующие циклы были бесконечны: while  $k < > 0$  do  $k := k - 2$ ?

**B3.** Определите, что делает следующая программа и какой результат она выдаст на заданном массиве (1,4,0,-1,6).

Var a:array[1..5] of integer; i, b:integer;

begin

b:=a[1];

for i:=2 to n do if  $a[i] > b$  then  $b := a[i]$ ;

writeln(b)

end.

**C1.** Допустим, имеется одномерный массив, содержащий числа от -20 до 20 включительно. Требуется исключить из него все элементы, значения которых больше 15.



**IV. Подведение итогов урока**

(Учитель выставляет оценки за урок.)

За каждое верно выполненное задание части А начисляется 0,5 балла, части В – 1 балл, части С – 3 балла.

**Критерии оценивания в зависимости  
от количества набранных баллов**

| Баллы   | Оценка |
|---------|--------|
| 9–10    | «5»    |
| 7–8     | «4»    |
| 5–6     | «3»    |
| Менее 5 | «2»    |