

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И МОЛОДЕЖНОЙ ПОЛИТИКИ
СТАВРОПОЛЬСКОГО КРАЯ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«КУРСАВСКИЙ РЕГИОНАЛЬНЫЙ КОЛЛЕДЖ «ИНТЕГРАЛ»**

**ОСНОВЫ
ПРОГРАММИРОВАНИЯ**

**Методические указания для проведения
лабораторных работ
по ОП.05 «Основы программирования»**

**для студентов специальности: 230115 «Программирование в
компьютерных системах»**

с. Курсавка
2014 г.

Методические рекомендации для проведения лабораторных работ по
ОП.05 «Основы программирования» для студентов специальности:
230115 «Программирование в компьютерных системах»

Организация - разработчик: **ГБОУ СПО КРК «Интеграл»**

Разработчик:

**Сыроватский Константин Борисович, преподаватель ГБОУ СПО КРК
«Интеграл»**

**Рассмотрено, утверждено и рекомендовано к применению на заседании
методического Совета ГБОУ СПО КРК «Интеграл»**

Протокол № _____ от «__» _____ 20__ г.

Председатель

М.А. Уманская

**357070 Ставропольский край,
Андроповский район,
с. Курсавка, ул. Титова, 15
тел.: 8(86556)6-39-82, 6-39-83
факс:6-39-79
kurs_integrall@mail.ru**

ВВЕДЕНИЕ

Данные методические указания для проведения лабораторных работ по дисциплине «Основы алгоритмизации и программирования» предназначены для реализации государственных требований к минимуму содержания и уровню подготовки выпускников специальности **230115 «Программирование в компьютерных системах»** с целью закрепления теоретических знаний и практических умений

В сборнике содержатся методические указания по выполнению 18 лабораторных работ в интегрированной системе программирования ABCPascal.

При выполнении лабораторных работ студент должен

знать:

- алфавит и лексику языка;
- типы данных языка программирования;
- правила записи выражений и операций.
- операторы ввода-вывода, ветвления, цикла;
- синтаксис объявления массива, строк, множеств, выполняемые операции с массивами, строками, множествами, стандартные функции для работы с массивами, со строками;
- синтаксис объявления процедур и функций;
- синтаксис объявления файла, порядок работы с файлами различного типа;
- назначение библиотек подпрограмм;

уметь:

- составлять программы линейной структуры;
- составлять программы с ветвлениями;
- составлять циклические программы;
- осуществлять ввод и вывод массивов, сравнивать массивы;
- выполнять операции со строками и над множествами;

- применять стандартные функции для работы с массивами и строками;
- создавать программы, используя процедуры и функций;
- работать с файлами разных типов;
- использовать библиотеку подпрограмм.

Каждая лабораторная работа имеет следующую структуру: тема, цели, краткие теоретические сведения, порядок проведения работы, требования к составлению отчета.

После выполнения лабораторной работы студент должен представить отчет о проделанной работе. Оценку по практической работе студент получает, если студентом работа выполнена в полном объеме, студент может пояснить выполнение любого этапа работы, отчет выполнен в соответствии с требованиями к выполнению работы, студент отвечает на контрольные вопросы на удовлетворительную оценку и выше.

Зачет по выполнению лабораторных и практических работ студент получает при условии выполнения всех предусмотренных программой лабораторных работ с отчетами по всем работам.

Лабораторная работа №1

Программирование линейных алгоритмов

Цель: Ознакомиться с системой программирования ABCPascal, получить основные навыки работы с ней, освоить приёмы создания, компиляции и исполнения программы, научиться создавать программы для решения линейных задач.

Теоретические сведения:

Работа в системе программирования ABCPascal:

- Активация меню: Клавиша <F10> (щелчок левой кнопкой мыши на нужном пункте меню)
- Ввод текста: текст вводится на английском языке. После ввода каждой строки нажимается клавиша Enter. Смена алфавита: на русский язык - <Ctrl>+<Shift>(справа), на английский язык - <Ctrl>+<Shift>(слева)
- Компиляция: меню Compile, команда Compile (клавиши <Alt>+<F9>)
- Запуск программы: меню Run, команда Run (клавиши <Ctrl>+<F9>)
- Открытие нового окна: Каждая программа создается в новом окне. Чтобы открыть новое окно, выберите меню File, команду New
- Переключение между окнами: Клавиши <Alt>+<номер окна>
- Открытие сохраненной программы: меню File, команда Open (клавиша <F3>). В открывшемся окне диалога в поле «Имя» введите полное имя нужного файла. Выберите команду Open.
- Выход из программы: меню File, команда Exit (клавиши <Alt>+<X>).

Порядок выполнения работы:

- 1) Запустите программу ABCPascal.
- 2) Введите текст программы решения задачи: *Найти сумму, разность, произведение и частное двух целых чисел.*

```
Program Primer;  
Var A, B, S, R, P: integer;  
    Ch: real;  
Begin  
    Writeln('Введите два целых числа');  
    Readln(A, B);  
    S:=A+B;  
    R:=A-B;  
    P:=A*B;  
    Ch:=A/B;
```

```

Writeln('Сумма чисел', A, 'и', B, 'равна', S);
Writeln('Разность чисел', A, 'и', B, 'равна', R);
Writeln('Произведение чисел', A, 'и', B, 'равно', P);
Writeln('Частное чисел', A, 'и', B, 'равно', Ch);
Readln;
End.

```

- 3) Откомпилируйте программу.
- 4) Запустите программу и проверьте правильность решения задачи на числах:
20 и 5; 36,6 и 6.
- 5) Исправьте ошибку ввода данных, введя числа 36 и 6. Замените операторы Writeln и Readln на Write и Read.
- 6) Рассмотрите, что изменилось в ходе выполнения программы. Покажите результат преподавателю.

7) Разработать программу нахождения значения следующего выражения:

№1 $\frac{b+\sqrt{b^2+4ac}}{2a} - a^3c+b-2$; №2 $\frac{d}{c} * \frac{b}{d} - \frac{ab-c}{cd}$; №3 $\frac{\sin x + \cos y}{\cos x - \sin y} x \operatorname{arctg} xy$;

№4 $\frac{x+y}{x+1} - \frac{xy-12}{34+x}$; №5 $3 - 4x + (y - \sqrt{|x|})$; №6 $x - \frac{x^3}{3} + \frac{x^5}{5}$;

№7 $\ln|(y - \sqrt{|x|})(x - \frac{y}{x + \frac{x}{4}})|$; №8 $e^x - x - 2 + (1 + x)^2$; №9 $\frac{\ln|\cos x|}{\ln(1 + x^2)}$;

№10 $\frac{1 + \sin\sqrt{x+1}}{\cos(12y - 4)}$; №11 $e^x - \frac{y^2 + 12xy - 3x^2}{18y - 1}$; №12

$\frac{x^2 - 7x + 10}{x^2 - 8x + 12}$;

№13 $\frac{\cos x}{\pi - 2x} + 16x \cos(xy) - 2$; №14 $\sin\sqrt{x+1} - \sin\sqrt{x-1}$; №15 $x - \ln x +$

$\frac{y}{\cos x - \frac{x}{3}}$.

8) Составьте программу решения задачи:

Вычислить корни квадратного уравнения $ax^2 + bx + c = 0$, заданного коэффициентами a, b и c (предполагается, что $a \neq 0$).

- 9) Составьте программу решения задачи:
Дано трёхзначное число. Поменять местами вторую и третью его цифры.
Откомпилируйте программу. Проверьте правильность решения задачи на любом целом трехзначном числе.
- 10) Составьте программу решения задачи:
Дана длина окружности. Найти площадь круга, ограниченного этой окружностью.
Откомпилируйте программу.
- 11) Составьте программу решения задачи: *Найти площадь кольца, внутренний радиус которого R_1 , а внешний R_2 ($R_1 < R_2$).*
Откомпилируйте программу. Используя составленную программу, вычислите площадь кольца, внутренний радиус которого – 16 см, а внешний – 25 см.
- 12) Запишите полученные результаты в тетрадь.
- 13) Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 14) Выйдите из программы ABCPascal.

Лабораторная работа №2

Программирование разветвляющихся алгоритмов

Цель: научиться составлять программы для решения задач, содержащих условия, продолжить освоение работы в системе программирования ABCPascal.

Теоретические сведения:

Условный оператор используется для выполнения одного из двух возможных вариантов программы.

Формат записи команды ветвления:

| | |
|------------------------------|-------------------------------|
| if логическое_условие | если логическое_условие верно |
| then оператор_1 | то выполняется оператор_1 , |
| else оператор_2; | иначе оператор_2; |

Перед else точка с запятой не ставится!

| | |
|--|--|
| Формат полного оператора ветвления: | Формат неполного оператора ветвления: |
|--|--|

| | |
|---------------------------|---------------------------|
| if <логическое выражение> | if <логическое выражение> |
| then <оператор 1> | then <оператор>; |
| else <оператор 2>; | |

Логическое выражение:

- Простой формой логического выражения является операция отношения:

- < меньше,
- > больше,
- <= меньше или равно,
- >= больше или равно,
- = равно,
- <> не равно.

- Сложные формы логических выражений составляются с использованием логических операций:

- not логическое отрицание (НЕ),
- and логическое умножение (И),
- or логическое сложение (ИЛИ),
- xor исключающее ИЛИ

- Порядок действий: Отрицание, умножение, сложение, операции отношений.

Операторные скобки:

- Если после слов then или else нужно выполнить не один оператор, а несколько, то эти операторы заключают в операторные скобки: begin ... end

- Конструкция такого вида: `Begin <последовательность операторов> end` относится к составным операторам.
- Операторы ветвления могут быть вложены друг в друга, необходимо только следить за тем, чтобы `then` и `else` одного и того же оператора располагались друг под другом.

Порядок выполнения работы:

- 1) Запустите программу ABCPascal.
- 2) Составьте программу решения задачи: *Дано целое число. Если оно является положительным, то прибавить к нему 1, в противном случае вычесть из него два. Вывести полученное число.*
- 3) Составьте программу решения задачи: *Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, то есть имеют ли они равные площади.*
- 4) Составьте программу решения задачи: *Ввести два числа. Меньшее заменить полусуммой, а большее – удвоенным произведением.*
- 5) Составьте программу решения задачи: *Вычислить значение Y при значении X=:*

$$\begin{cases} X^2 - 4X^3, & \text{при } X > 0, \\ |X|, & \text{при } X \leq 0. \end{cases}$$
- 6) Составьте программу решения задачи: *Даны целые числа a и b. Определить, является ли число a делителем числа b.*
 Откомпилируйте программу. Используя составленную программу, определите, является ли число 43 делителем числа 553, а число 56 делителем числа 672.
- 7) Запишите полученные результаты в тетрадь.
- 8) Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 9) Выйдите из программы ABCPascal.

Лабораторная работа №3

Программирование алгоритмов с использованием оператора выбора.

Цель: научиться составлять программы с использованием оператора множественного выбора Case, продолжить освоение работы в системе программирования ABCPascal.

Теоретические сведения:

Оператор варианта Case является обобщением оператора if и позволяет сделать выбор из произвольного числа имеющихся вариантов. Он состоит из выражения, называемого ключом выбора, и списка операторов, каждому из которых предшествует список констант выбора. Как и в операторе if, в операторе Case может присутствовать слово else, имеющее тот же смысл. Ключ выбора может быть *целого, символьного или логического* типа. Список констант должен иметь тот же тип, что и ключ выбора, и может состоять из одной константы, списка констант, разделенных запятыми или списка констант, заданных диапазоном.

Общий вид:

```
case <ключ выбора> of
    <список 1>: <оператор 1>;
    <список 2>: <оператор 2>;
    ...
    <список N>: <оператор N>
else <оператор>
end;
```

Порядок выполнения работы:

- 1) Запустите программу ABCPascal.
- 2) Написать программу, которая по номеру дня недели (натуральному числу от 1 до 7) выдает в качестве результата расписание занятий в вашей группе в этот день.
- 3) Составить программу, которая по заданному номеру месяца *t* определяет количество дней в этом месяце.
- 4) Составьте программу решения задачи: В старояпонском календаре был принят двенадцатилетний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки и свиньи. Написать программу, которая по номеру года определяет его название в старом японском календаре.

- 5) Составьте программу решения задачи: *Для целого числа и от 1 до 99 напечатать фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.*
- 6) Составьте программу решения задачи: *Введите номер дня и месяца. Определите, сколько дней осталось до Нового года.*
- 7) Запишите полученные результаты в тетрадь.
- 8) Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

Лабораторная работа №4

Программирование циклических алгоритмов: цикл с параметром.

Цель: закрепить практические навыки работы с системой ABCPascal, научиться правильно использовать оператор цикла с параметром; научиться составлять программы решения задач с использованием циклических структур.

Теоретические сведения:

Для реализации в языке Pascal используется составной оператор, состоящий из операторов for, to, downto, do и при необходимости из операторных скобок. Переменная параметр обязательно объявляется в декларационной части программы и может принадлежать одному из порядковых типов. Если при изменении переменной параметра необходимо использовать переход к следующему значению, то используется оператор to; если переход необходимо осуществить к предыдущему значению, то используется оператор downto. Тогда в общем виде цикл записывается так:

```
for I:=I0 to In do
begin
  <оператор 1>;
  <оператор 2>;
  ...
  <оператор n>;
end;
```

где I₀ и I_n – начальное и конечное значения.

Порядок выполнения работы:

1) Составьте программу, выводящую на экран квадраты и кубы чисел от 10 до 20.

Откомпилируйте программу. Проверьте правильность решения задачи.

2) Составьте программу решения задачи: *Найти значение выражения $5x^9 + 13x^7 - 2x^{11}$, используя оператор цикла с параметром.*

3) Составьте программу, выводящую на экран таблицу умножения чисел от 1 до 9.

Откомпилируйте программу. Проверьте правильность решения задачи.

4) Составьте программу решения задачи: Дано натуральное число n. Вычислить

$$S=1!+2!+3!+4!+\dots+n! \quad (n>1).$$

5) Составьте программу решения задачи: *Сколько можно купить ручек, тетрадей и карандашей, платя за ручку 5 рублей, за тетрадь – 10 рублей, за карандаш – 0,5 рублей, если на 100 рублей нужно купить 100 предметов.* Откомпилируйте программу. Проверьте правильность решения задачи.

6) Запишите полученные результаты в тетрадь.

7) Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

8) Выйдите из программы ABCPascal.

Лабораторная работа №5

Программирование циклических алгоритмов: цикл с предусловием.

Цель: закрепить практические навыки работы с системой ABCPascal, научиться правильно использовать операторы цикла с условием; научиться составлять программы решения задач с использованием циклических структур.

Теоретические сведения:

Цикл с предусловием.

Для реализации циклов с предусловием используется составной оператор, включающий оператор while, do, операторные скобки.

В общем виде цикл реализуется записью:

```
while <условие> do <действие>;
```

Если тело цикла содержит более одного действия, то необходимо использовать операторные скобки:

```
while <условие> do
begin
  <оператор 1>;
  <оператор 2>;
  ...
  <оператор n>;
end;
```

Порядок выполнения работы:

- 1) Составьте программу решения задачи: *Дано натуральное число n . Вычислить $S=1!+2!+3!+4!+\dots+n!$ ($n>1$).* Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.
- 2) Составьте программу решения задачи: *Программа должна запрашивать пароль (например, четырехзначное число) до тех пор, пока он не будет правильно введён.* Откомпилируйте программу. Проверьте правильность решения задачи.
- 3) Составьте программу решения задачи: *Найти сумму и произведение всех трехзначных чисел, кратных 7.* Откомпилируйте программу. Проверьте правильность решения задачи.
- 4) Составьте программу решения задачи: *Одноклеточная амёба каждые 3 часа делится на 2 клетки. Определить сколько амёб будет через 3, 6, 9, 12,*

15, 18, 21,24 часа. Откомпилируйте программу. Проверьте правильность решения задачи.

5) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

6) Выйдите из программы ABCPascal.

Лабораторная работа №6

Программирование циклических алгоритмов: цикл с постусловием.

Цель: закрепить практические навыки работы с системой ABCPascal, научиться правильно использовать операторы цикла с условием; научиться составлять программы решения задач с использованием циклических структур.

Теоретические сведения:

Цикл с постусловием.

Для реализации цикла используется составной оператор, состоящий из операторов repeat и until. В общем виде цикл записывается так:

```
repeat  
  <действие>;  
until <условие>;
```

Порядок выполнения работы:

- 1) Составьте программу решения задачи: *В банк положили 1000 рублей под проценты. Каждый месяц сумма увеличивается на 5% от суммы предыдущего месяца. Через сколько месяцев сумма на счету достигнет 2000 рублей?* Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.
- 2) Составьте программу решения задачи: *Введите целое число и определите количество цифр в нем.* Откомпилируйте программу. Проверьте правильность решения задачи.
- 3) Составьте программу решения задачи: *Найти среднее арифметическое натуральных чисел, меньших 200 и кратных 7.* Откомпилируйте программу. Проверьте правильность решения задачи.
- 4) Составьте программу решения задачи: *Найти все трехзначные числа, средняя цифра которых равна сумме первой и последней.* Откомпилируйте программу. Проверьте правильность решения задачи.
- 5) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 6) Выйдите из программы ABCPascal.

Лабораторная работа №7

Обработка одномерных и двумерных массивов.

Цель: научиться описывать, заполнять, выводить и обрабатывать одномерные и двумерные массивы.

Теоретические сведения:

Массив – группа элементов одного типа, объединенных под общим именем.

Описание массивов

Массивы описываются в разделе описания переменных Var.

Общий вид описания одномерного массива:

<имя массива>: array [*<начальный индекс>*..*<конечный индекс>*] of *<тип элемента>*;

где имя - имя переменной-массива; array - ключевое слово, обозначающее, что переменная является массивом; нижний_индекс и верхний_индекс - целые числа, определяющие диапазон изменения индексов (номеров) элементов массива и, неявно, количество элементов (размер) массива; тип - тип элементов массива.

Общий вид описания двумерного массива:

<имя массива>:array[*<m₁>*..*<m₂>*,*<n₁>*..*<n₂>*] of *<тип>*;

где *Имя* - имя массива; array - слово языка Pascal, показывающее, что описываемый элемент данных - массив; *m₁*, *m₂*, *n₁*, *n₂*- константы или выражения типа INTEGER, определяющие диапазон изменения индексов и, следовательно, число элементов массива; *Тип* - тип элементов массива.

Заполнение массива

Под вводом массива понимается ввод значений элементов массива. Ввод удобно реализовать при помощи инструкции FOR. Чтобы пользователь программы знал, ввода какого элемента массива ожидает программа, следует организовать вывод подсказок перед вводом очередного элемента массива. В подсказке обычно указывают индекс элемента массива. Заполнение массива можно производить:

- 1) с клавиатуры: For i:=1 to n do readln(a[i]);
- 2) через датчик случайных чисел: Randomize; For i:=1 to n do begin a[i]:=random(i);

Если требуется, чтобы значения элементов массива выбирались из определенного интервала [a,b], то a+Random(b-a+1);

- 3) через оператор присваивания (по формуле): For i:=1 to n do a[i]:=i*3;

Вывод массива

Если в программе необходимо вывести значения всех элементов массива, то для этого удобно использовать инструкцию FOR, переменная-счётчик которой может быть реализована как индекс элемента массива. Например,

For i:=1 to n do writeln(a[i]);

Удаление элементов из одномерного массива.

Для того, чтобы удалить из массива k-ый элемент нужно: найти номер элемента k; сдвинуть все элементы, начиная с k-го, на один элемент влево; последнему элементу присвоить значение, равное 0; уменьшить количество элементов массива на единицу.

Вставка элемента в одномерный массив.

Вставлять элемент можно до или после данного элемента, номер этого элемента можно вводить с клавиатуры или искать при определенных условиях. Пусть k - это номер элемента, после которого мы должны вставить элемент x. Тогда вставка осуществляется следующим образом: первые k элементов массива остаются без изменения, все элементы, начиная с (k+1)-го, необходимо сдвинуть на один назад, на место (k+1)-го элемента записываем значение x; увеличить количество элементов в массиве на единицу.

При решении задач с использованием двумерных массивов организуются вложенные циклы:

For i:=1 to m do begin

изменяется номер строки

For j:=1 to n do begin *изменяется номер столбца*

Запись элемента массива: a[i , j]

| Главная диагональ $i = j$ | Побочная диагональ $i + j = n + 1$ |
|---|---|
| a₁₁ a ₁₂ a ₁₃ a ₁₄ | a ₁₁ a ₁₂ a ₁₃ a₁₄ |
| a ₂₁ a₂₂ a ₂₃ a ₂₄ | a ₂₁ a ₂₂ a₂₃ a ₂₄ |
| a ₃₁ a ₃₂ a₃₃ a ₃₄ | a ₃₁ a₃₂ a ₃₃ a ₃₄ |
| a ₄₁ a ₄₂ a ₄₃ a₄₄ | a₄₁ a ₄₂ a ₄₃ a ₄₄ |

Порядок выполнения работы:

- 1) Запустите программу АВСРascal.
- 2) Ввести одномерный массив, состоящий из 10 элементов. Заменить отрицательные элементы на противоположные по знаку. Вывести полученный массив на экран.
- 3) Ввести одномерный массив, состоящий из n элементов. Найти и вывести на экран номера четных элементов.

- 4) Ввести одномерный массив, состоящий из t элементов. Найти количество положительных и отрицательных элементов в данном массиве.
- 5) Ввести одномерный массив, состоящий из n элементов. Удалите из массива третий элемент.
- 6) Ввести одномерный массив, состоящий из n элементов. Вставьте в массив число 100 после пятого элемента.
- 7) Ввести двумерный массив, состоящую из $n*t$ элементов. Найдите сумму всех элементов.
- 8) Заполните двумерный массив размером $N*N$ следующим образом:
- ```
0 1 1 1 1 0
2 0 1 1 0 4
2 2 0 0 4 4
2 2 0 0 4 4
2 0 3 3 0 4
0 3 3 3 3 0.
```
- 9) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 10) Выйдите из программы ABCPascal.

## Лабораторная работа №8

### **Сортировка массивов. Поиск элемента массива.**

**Цель:** научиться производить сортировку массивов различными способами, а также отыскивать необходимый элемент в массиве.

#### **Теоретические сведения:**

##### **Алгоритмы поиска элемента массива.**

Пусть значения элементов линейного массива  $x$  сформированы. Требуется среди чисел  $x[1], x[2], \dots, x[n]$  найти такое, что  $x[j] = \max(x[1], x[2], \dots, x[n])$ . Если в задаче требуется найти порядковый номер этого элемента, то значит необходимо найти еще и значение индекса  $j$ . Основная идея алгоритма состоит в том, что переменной **max** присваивается значение любого элемента массива (чаще всего первого по порядку). В случае нахождения порядкового номера переменной **ind** присваивается значение индекса этого элемента (т.е. 1).

Далее просматриваются все элементы массива, значения которых сравниваются со значением переменной **max**. Если окажется, что значение какого-либо элемента массива превосходит значение переменной **max**, то переменная **max** меняет свое значение на значение большего элемента. Если поиск наибольшего элемента идет в двумерном массиве, то необходимо просматривать поочередно все элементы каждой из строк.

#### **Методы сортировки массивов.**

##### **1. Метод «пузырька»**

*Идея метода:* весь массив просматривается несколько раз, причем при каждом просмотре сравниваются значения двух соседних элементов. Если эти значения следуют не в порядке возрастания, то производится их перестановка. Так происходит до тех пор, пока не будет сделано ни одной перестановки.

##### **2. Метод простых обменов.**

*Идея метода:* весь массив просматривается несколько раз и при каждом просмотре ищется минимальный по значению элемент, который меняется местами с первым, вторым, третьим, ..., предпоследним элементов массива и исключается из дальнейшего рассмотрения.

##### **3. Метод вставки и сдвига.**

*Идея метода:* делается предположение, что первые  $q$  элементов массива упорядочены, и если  $q+1$ -ый элемент меньше, чем какой-либо из первых  $q$ , то он записывается на свое место среди упорядоченных, при этом «хвост» массива «сдвигается» к концу.

### **Порядок выполнения работы:**

- 1) Запустите программу ABCPascal.
- 2) Введите массив из 10 целых чисел. Найдите элемент с минимальным значением.
- 3) Введите массив из 10 целых чисел и отсортируйте его в порядке возрастания и убывания элементов всеми рассмотренными методами.
- 4) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 5) Выйдите из программы ABCPascal.

## Лабораторная работа №9

### *Работа со строковым типом данных.*

**Цель:** научиться составлять программы решения задач с использованием символьных и строковых типов данных.

#### **Теоретические сведения:**

Строка – последовательность символов. В языке Pascal для строк отведен свой тип данных: string. String – строка размером 255 символов. String[n] – строка размером n символов, где n должно быть больше нуля и меньше или равно 255.

Так же мы можем и вручную присваивать переменной строкового типа значения через оператор присваивания: **Stroka:=’строка’;**

Таким образом, мы присвоили переменной stroka значение «строка», которое находится между апострофами.

Для переменных строкового типа предусмотрены некоторые функции и процедуры:

| <b>Функция</b>                                | <b>Действие</b>                                                                  | <b>Пример</b> |
|-----------------------------------------------|----------------------------------------------------------------------------------|---------------|
| Length(s)<br>Len:=Length(stroka);             | Длина строки s                                                                   |               |
| Copy(s, start, finish)<br>S2:=Copy(s1, 3, 5); | Копирует часть строки s<br>с позиции start до<br>позиции finish                  |               |
| Delete(s, start, finish)<br>6);               | Удаляет из строки символы<br>с позиции start до позиции<br>Finish                | Delete(s1, 4, |
| Insert(s, s1, posit)<br>2);                   | Вставляет в строку s<br>строку s1 в позицию posit                                | Insert(s1,s2, |
| Pos(sub, s)<br>N:=Pos(‘asd’, s1);             | Позиция подстроки sub в<br>строке s. В случае, если<br>строка не содержит данной |               |

подстроки, то функция  
возвращает значение 0.

Str(x, s) Преобразует числовое значение x в строковое s Str(124, s1);

Val(s, x, error) Преобразует строку s в  
Val(s1,123.22,error\_code);  
числовое значение x.

### Порядок выполнения работы:

- 1) Запустите программу АВSPascal.
- 2) Составьте программу решения задачи: *Ввести строку символов, заменить все буквы «а» на буквы «о».* Откомпилируйте программу. Проверьте правильность решения задачи на примере.
- 3) Составьте программу решения задачи: *Даны две фамилии определить, какая из них длиннее.* Откомпилируйте программу. Проверьте правильность решения задачи на примере.
- 4) Составьте программу решения задачи: *Определить и вывести на экран количество слов во введенной строке.* Откомпилируйте программу. Проверьте правильность решения задачи на примере.
- 5) Составьте программу решения задачи: *Заданы фамилия, имя и отчество учащегося, разделенные пробелами. Напишите программу, печатающую фамилию ученика и его инициалы.*  
Откомпилируйте программу. Проверьте правильность решения на конкретном примере.
- 6) Составьте программу решения задачи: *Введённую с клавиатуры строку А записать в обратном порядке в строку В. Строку В вывести на экран.* Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.
- 7) Составьте программу решения задачи: *Составить программу, определяющую, является ли введенное слово перевертышем.*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

8) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

9) Выйдите из программы ABCPascal.



## Лабораторная работа №10

### *Работа с множественным типом данных.*

**Цель:** научиться составлять программы решения задач с использованием множественного типа данных.

#### **Теоретические сведения:**

В языке Паскаль типом-множеством называется множество различных сочетаний элементов исходного (базового) типа. Число элементов исходного множества в Turbo-Pascal не может быть больше 256, а порядковые номера элементов должны находиться в пределах от 0 до 255.

Для объявления типа-множества используются зарезервированные слова `set of`, после которых указываются элементы этого множества, как правило в виде перечисления или диапазона. Объявить тип-множество можно в разделе программы `Type` или при объявлении переменной в разделе `Var`. Например:

`Type`

`Lat=set of 'A'..'Z'; {множество прописных латинских букв}`

`Oper=set of (Plus, Minus, Mult, Divide) {множество знаков арифметических операций}`

`Chisla=set of 12..27; {множество чисел от 12 до 27}`

`Digit=set of '0'..'9'; {множество символов-цифр}`

После определения типа-множества можно объявить переменные или типизированные константы этого типа.

`Var l: Lat; c1,c2: chisla; op: Oper; d: Digit;`

`g: set of ('a', 'я', 'o', 'ё', 'y', 'ю', 'э', 'e', 'ы', 'и');`

В операциях над множествами могут участвовать переменные типа-множество, константы и конструкторы множеств.

Для множеств определены следующие операции.

`+` объединение множеств;

`-` разность множеств;

`*` пересечение множеств;

`=` проверка эквивалентности двух множеств;

`<>` проверка неэквивалентности двух множеств;

`<=` проверка, является ли левое множество подмножеством правого;

`>=` проверка, является ли правое множество подмножеством левого;

`in` проверка принадлежности элемента множеству.

Результатом операций объединения, разности и пересечения является множество. Результатом операций проверки эквивалентности и вхождения будет значение логического типа.

### Порядок выполнения работы:

- 1) Запустите программу ABCPascal.
- 2) Составьте программу решения задачи: *Вычислить значения отношений:*  
А)  $[2] \diamond [2, 2, 2]$ ;    Б)  $[4, 5, 6] = [5, 4, 6]$ ;    В)  $[2, 3, 5, 7] \leq [1..9]$ ;    Г)  $[3, 6..8] \geq [2..7, 9]$
- 3) Составьте программу решения задачи: *Вычислить значения выражений:*  
А)  $[1, 3, 5] + [2, 4]$ ;    Б)  $[1, 3, 5] * [2, 4]$ ;    В)  $[1, 3, 5] - [2, 4]$ ;    Г)  $[1..8] - [3..6]$
- 4) Составьте программу решения задачи: *Из множества целых чисел 1..20 выделить: 1) множество чисел, делящихся на 6 без остатка; 2) множество чисел, делящихся на 2 или на 3 без остатка. Вывести содержимое этих множеств на экран. Откомпилируйте программу. Проверьте правильность решения задачи на примере.*
- 5) Составьте программу решения задачи: *Ввести строку символов, состоящую из латинских букв, цифр и пробелов. Осуществить проверку правильности введенных символов.*  
Откомпилируйте программу. Проверьте правильность решения задачи на примере.
- 6) Составьте программу решения задачи: *Определить количество гласных букв в предложении, введенном с клавиатуры. Откомпилируйте программу. Проверьте правильность решения задачи на примере.*
- 7) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 8) Выйдите из программы ABCPascal.

## Лабораторная работа №11

### **Организация процедур**

**Цель:** научиться составлять программы решения задач с использованием процедур.

#### **Теоретические сведения:**

В языке Паскаль имеется два вида подпрограмм - **процедуры** и **функции**.

Структура описания процедур до некоторой степени похожа на структуру Паскаль-программы: у них также имеются заголовок, раздел описаний и исполняемая часть. Раздел описаний содержит те же подразделы, что и раздел описаний программы: описания констант, типов, меток, процедур, функций, переменных. Исполняемая часть содержит собственно операторы процедур. Одна и та же подпрограмма может вызываться неоднократно, выполняя одни и те же действия с разными наборами входных данных. Параметры, используемые при записи текста подпрограммы в разделе описаний, называют **формальными**, а те, что используются при ее вызове - **фактическими**.

Формат описания процедуры имеет вид:

```
procedure имя процедуры (формальные параметры);
 раздел описаний процедуры
begin
 исполняемая часть процедуры
end;
```

**Вызов процедуры** производится оператором, имеющим следующий формат:

**имя процедуры (список фактических параметров);**

При вызове фактические параметры как бы подставляются вместо формальных, стоящих на тех же местах в заголовке. В стандарте языка Паскаль передача параметров может производиться двумя способами - по значению и по ссылке. Параметры, передаваемые по значению, называют **параметрами-значениями**, передаваемые по ссылке - **параметрами-переменными**. Последние отличаются тем, что в заголовке процедуры (функции) перед ними ставится служебное слово `var`. При первом способе (*передача по значению*) значения фактических параметров копируются в соответствующие формальные параметры. При изменении этих значений в ходе выполнения процедуры (функции) исходные данные (фактические параметры) измениться не могут. При втором способе (*передача по ссылке*) все изменения, происходящие в теле процедуры (функции) с

формальными параметрами, приводят к немедленным аналогичным изменениям соответствующих им фактических параметров.

Имена, описанные в заголовке или разделе описаний процедуры называют **локальными** для этого блока. Имена, описанные в блоке, соответствующем всей программе, называют **глобальными**.

### Порядок выполнения работы:

1) Запустите программу ABCPascal.

2) *Напишите программу, состоящую из трех процедур и основной программы. Первая процедура организует ввод двух целых чисел  $X$  и  $Y$ , вторая проверяет их сумму, третья выводит результат. Используйте эти процедуры в основной программе. Используйте  $X$  и  $Y$  как глобальные переменные. Откомпилируйте программу. Проверьте правильность решения задачи.*

3) *Найти площадь десятиугольника, вершины которого имеют координаты  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  ...  $(x_{10}, y_{10})$ .*

Откомпилируйте программу. Проверьте правильность решения задачи на любом примере.

4) *Вычислить разность двух простых дробей:  $a/b - c/d$  ( $a, b, c, d$  — натуральные числа). Результат получить в виде простой несократимой дроби  $e/f$ .*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

5) *Вычислить* 
$$y = \frac{\sum_{i=1}^7 i + \sum_{k=18}^{91} k}{\sum_{j=5}^{47} j}$$

*Оформить вычисление суммы в виде подпрограмм. Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.*

6) *Дан одномерный массив, состоящий из 10 элементов. Используя процедуру, найдите максимальный и минимальный элементы этого массива.*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

- 7) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.
- 8) Выйдите из программы ABCPascal.

## Лабораторная работа № 12

### Организация функций

**Цель:** научиться составлять программы решения задач с использованием функций.

#### Теоретические сведения:

Функции, это такие подпрограммы, результатом которых обязательно является некоторое значение. Описание функции во многом совпадает с описанием процедуры. Но если имя процедуры используется только для её вызова, то с именем функции, кроме того, связывается её результат. В описании функции заголовок выглядит следующим образом:

```
Function<имя>(<Параметры>): <Тип_возвращаемого_значения>;
<описания_переменных>;
Begin
<Тело
подпрог
раммы
>;
End;
```

Как и у процедуры, заголовок функции обязателен.

Функция предполагает обязательную передачу информации из подпрограммы в программу через имя функции. Поэтому раздел операторов обязательно должен содержать хотя бы один оператор, в котором имени функции присваивается значение результата. В противном случае функция не возвратит результат (вернее возвратит произвольный результат).

#### Порядок выполнения работы:

- 1) Запустите программу ABCPascal.
- 2) Определить значение выражения  $t(a, b, c) + t(b, c, d) + t(c, d, a)$  для вещественных  $a, b, c, d$ , где  $t(x, y, z)$  - функция определения минимального из трёх чисел.  
Откомпилируйте программу. Проверьте правильность решения задачи.
- 3) Составить программу нахождения выражения  $x^{10} + 2x^9 - 5x^7 + 4$ , используя функцию.  
Откомпилируйте программу. Проверьте правильность решения задачи на любом примере.

4) Даны три квадратных уравнения  $ax^2 + bx + c$ ,  $dx^2 + ex + f$ ,  $px^2 + qx + s$ .

*Сколько из них имеют вещественные корни?*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

5) *Вычислить периметр треугольника по его координатам.*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

6) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

7) Выйдите из программы ABCPascal.

***Использование рекурсивных подпрограмм.***

**Цель:** научиться составлять программы решения задач с использованием рекурсивных процедур и функций.

**Теоретические сведения:**

Рекурсия - это способ описания функции или процессов через самих себя (когда процедура или функция сама себя вызывает). В Паскале можно пользоваться именами лишь тогда, когда в тексте программы этому предшествует их описание. Рекурсия является единственным исключением из этого правила. Имя рекурсивной функции можно использовать сразу же, не закончив его описания.

В Паскале возможно применения рекурсии в процедурах и функциях.

**Порядок выполнения работы:**

1) Запустите программу ABCPascal.

2) *Составить программу нахождения факториала, используя рекурсивную функцию.*

Откомпилируйте программу. Проверьте правильность решения задачи. {Подсказка: Факториал  $a_n = n!$  означает  $n! = 1 * 2 * 3 * 4 * 5 * 6 * \dots * n$ . а)  $a_1 = 1$ ; б)  $a_n = n * a_{n-1}$  }

3) *Написать программу вычисления членов геометрической прогрессии.*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

{Подсказка: Геометрическая прогрессия: а)  $a_1 = a_0$ ; б)  $a_n = a_{n-1} * q$ . При  $a_0 = 2$ ,  $q = 2$  имеем степенной ряд 2,4,8,16,32,...; }

4) *Написать программу вычисления членов арифметической прогрессии.*

Откомпилируйте программу. Проверьте правильность решения задачи на конкретном примере.

{Подсказка: Арифметическая прогрессия: а)  $a_1 = a_0$ ; б)  $a_n = a_{n-1} + d$ . При  $a_0 = 1$ ,  $d = 1$  имеем натуральный ряд 1,2,3,... }

5) *Составить программу вычисления всех чисел Фибоначчи, используя рекурсивную функцию.*

Откомпилируйте программу. Проверьте правильность решения задачи на любом примере.



{Подсказка: Числа Фибоначчи. Один из наиболее ярких примеров применения рекурсии дают числа Фибоначчи. Они определяются следующим образом:  $x_1=x_2=1$  ,  $x_n=x_{n-1}+x_{n-2}$  при  $n > 2$

Каждый элемент ряда Фибоначчи является суммой двух предшествующих элементов, т.е.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55,... }

6) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

7) Выйдите из программы ABCPascal.

***Использование процедур и функций: графические возможности.***

**Цель:** научиться работать в графическом режиме программы ABCPascal.

**Теоретические сведения:**

Система программирования ABCPascal снабжена несколькими модулями или «библиотеками», в которых хранятся все её процедуры и функции.

Библиотека, содержащая процедуры и функции для поддержки графического режима носит имя GRAPH.TPU.

Для того, чтобы компилятор «узнавал» названия процедур и функций, содержащихся в библиотеке GRAPH.TPU, необходимо после заголовка программы разместить строчку следующего вида: *Uses Graph;*

До сих пор экран всегда находился в текстовом режиме, поэтому можно было видеть только символы. Для рисования прямых, окружностей и пр. необходимо перевести экран в графический режим.

Для включения графического режима используется процедура *InitGraph(Gd, Gm: integer; Path: string)* три параметра: Gd является кодом графического адаптера, позволяющего использовать несколько графических режимов, отличающихся количеством цветов и разрешающей способностью, Gm предназначен для того, чтобы указать какой из режимов следует включить, Path является строкой, содержащей путь к файлу, в котором содержится драйвер, необходимый для работы мониторов в графическом режиме.

*CloseGraph* – выключение графического режима.

Для того, чтобы можно было нарисовать что-то на экране, нужно уметь задавать положение этой фигуры. Для этого с экраном связывается система координат. Точка (0,0) находится в левом верхнем углу.

**Описание некоторых графических процедур и функций**

*Line(x1,y1,x2,y2: integer)* – рисует на экране отрезок, соединяющий точки (x1,y1) и (x2,y2).

По умолчанию, цвет линий белый. Для изменения текущего цвета используется процедура

*SetColor(Color: word)* – нужно указать число, являющееся кодом цвета (0..15).

*SetBkColor(Color: word)* – устанавливает текущий цвет фона;

*PutPixel(x,y: integer; Pixel:word)* – рисует точку с координатами (x,y) и цветом Pixel.

*Rectangle(x1,y1,x2,y2: integer)* – строит контур прямоугольника из линий текущего цвета.

*Circle(x,y: integer; r:word)* – строит окружность с центром в точке (x,y) и радиусом r.

*Arc(x,y: integer; StA,EndA,Radius:word)* – рисует дугу окружности текущим цветом в центром в точке (x,y) и радиусом Radius, где StA,EndA – начальный и конечный угол в градусах.

*Ellipse (x,y: integer; StA,EndA:word; xr,yr:word)* – рисует дугу эллипса текущим цветом в центром в точке (x,y), где StA,EndA – начальный и конечный угол в градусах, xr,yr – горизонтальная и вертикальная полуоси эллипса.

*SetLineStyle(Ln, P, T)* – изменение параметров линий, где Ln – стиль линии (0- сплошная, 1 – пунктирная, 2 – штрихпунктирная, 3 – штриховая), T – толщина линии (1-нормальная, 3- толстая), P – шаблон.

### **Порядок выполнения работы:**

1) Запустите программу ABCPascal.

2) *Нарисуйте разноцветную «ромашку», используя рисования окружности и эллипса.* Откомпилируйте программу. Проверьте правильность решения задачи.

3) *Нарисуйте олимпийский флаг.*

Откомпилируйте программу. Проверьте правильность решения задачи.

4) *Изобразите год вашего рождения и инициалы.*

Откомпилируйте программу. Проверьте правильность решения задачи.

5) *Нарисуйте три окружности с заключенными в них треугольником, квадратом и звездой. Все фигуры должны быть разного цвета.*

Откомпилируйте программу. Проверьте правильность решения задачи.

6) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

7) Выйдите из программы ABCPascal.

## Лабораторная работа №15

### *Работа с типизированными файлами последовательного доступа.*

**Цель:** научиться работать с файлами в программе ABCPascal.

#### **Теоретические сведения:**

Файловый тип переменной — это структурированный тип, представляющий собой совокупность однотипных элементов, количество которых заранее не определено.

Структура описания файловой переменной:

Var <имя переменной>: File Of <тип элемента>;

где <тип элемента> может быть любым, кроме файлового.

Для того чтобы начать запись в файл, его следует открыть для записи. Это обеспечивает процедура Rewrite (FV); где FV — имя файловой переменной. При этом указатель устанавливается на начало файла. Если в файле есть информация, то она исчезает.

Запись в файл осуществляется процедурой Write (FV, V); где V — переменная того же типа, что и файл FV. Запись происходит туда, где установлен указатель.

Для чтения элементов файла с его начала следует открыть файл для чтения. Это делает процедура Reset (FV). В результате указатель устанавливается на начало файла. При этом вся информация в файле сохраняется.

Чтение из файла осуществляется процедурой Read (FV,V); где V — переменная того же типа, что и файл FV. Значение текущего элемента файла записывается в переменную V; указатель смещается к следующему элементу.

Для организации связи между файловой переменной и внешним файлом в Турбо Паскале используется процедура назначения: Assign (<имя файловой переменной>, <внешней файл>);

Здесь <внешней файла> — строковая величина;

Работа с файлом в программе завершается его закрытием с помощью процедуры

Close (<имя файловой, переменной>).

Для определения размера файла используется функция: FileSize (<имя файловой переменной>);

#### **Порядок выполнения работы:**

1) Запустите программу ABCPascal.

2) Составить программу, которая создает файл, состоящий из  $N$  значений целого типа. Прочитать файл и вывести только четные элементы. Откомпилируйте программу. Проверьте правильность решения задачи.

3) Сформировать файл  $F$ , компонентами которого являются целые числа. Записать в файл  $C$  все четные числа файла  $F$ , а в файл  $N$  - все нечетные. Порядок следования чисел сохраняется. Откомпилируйте программу. Проверьте правильность решения задачи.

4) Заполнить файл последовательного доступа  $N$  действительными числами, полученными с помощью датчика случайных чисел. Найти максимальный элемент этого файла. Откомпилируйте программу. Проверьте правильность решения задачи.

5) Записать в файл  $f$  последовательного доступа  $N$  целых чисел. Получить в другом файле последовательного доступа все компоненты файла  $f$ , кроме тех, которые кратны  $K$ . Откомпилируйте программу. Проверьте правильность решения задачи.

6) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

7) Выйдите из программы ABCPascal.

## Лабораторная работа № 16

### *Работа с текстовыми файлами.*

**Цель:** научиться работать с текстовыми файлами в программе ABCPascal.

#### **Теоретические сведения:**

Текстовый файл можно рассматривать как последовательность символов, разбитую на строки длиной от 0 до 256 символов. Для описания используется стандартный тип Text:

**Var F: text;** {F - файловая переменная}. Каждая строка завершается маркером конца строки.

Вызов **Read(F, Ww)**, где Ww – переменная типа word, осуществляет чтение из файла F последовательности цифр, которая затем интерпретируется в число, значение которого и будет присвоено переменной Ww. В случае если вместо последовательности цифр идет любая другая последовательность символов, использование такого оператора приводит к ошибке выполнения программы. Открытие текстового файла можно произвести двумя стандартными способами:

- поставить в соответствие файловой переменной имя файла (процедура Assign), открыть новый текстовый файл (процедура Rewrite);
- поставить в соответствие файловой переменной имя файла (процедура Assign), открыть уже существующий файл (процедура Reset).

Текстовый файл в силу своей специфики во время работы допускает только один вид операции: чтение или запись. В связи с этим для работы с текстовыми файлами используется еще одна процедура открытия файла:

**Append(var F : text);** Эта процедура открывает уже существующий файл и позиционирует указатель обработки на конец файла. После этого в текстовый файл можно только добавлять информацию, причем только в конец файла. Для обработки текстовых файлов используются процедуры Read и Write, обеспечивающие соответственно чтение и запись одной строки и более в текстовый файл. Использование специальных разделителей строк позволило ввести в состав языковых средств еще две процедуры: Readln, выполняющую те же действия, что и Read, и дополнительно – чтение до маркера конца строки и переход к новой строке; Writeln, обеспечивающую запись всех величин с обязательной установкой маркера конца строки в файл.

Процедура Read обеспечивает ввод данных общим потоком из одной строки, а Readln приводит к обязательному переходу к следующей строке текстового файла, т. е. ввод данных осуществляется из различных строк. Все вышесказанное в равной мере относится к операциям записи с помощью процедур Write и Writeln.

При организации операций ввода-вывода используются специальные языковые средства в виде функций **Eoln**, **Eof**, **SeekEoln**, **SeekEof**.

Функция **Eoln(var F: text)** возвращает булевское значение True, если текущая файловая позиция находится на маркере конца строки или вызов Eof(F) вернул значение True. Во всех остальных случаях значение функции будет False.

Функция **Eof(var F: text)** возвращает булевское значение True, если указатель конца файла находится сразу за последним компонентом, и False – в противном случае.

### **Порядок выполнения работы:**

1) Запустите программу ABCPascal.

2) *Дано 10 слов. Записать их в файл, расположив каждое слово на отдельной строке.*

Откомпилируйте программу. Проверьте правильность решения задачи.

3) *Имеется текстовый файл. Переписать в другой файл те его строки, в которых содержится более 30-ти символов.* Откомпилируйте программу. Проверьте правильность решения задачи.

4) *Имеется текстовый файл. Удалить из него первую строку, оканчивающуюся вопросительным знаком.* Откомпилируйте программу. Проверьте правильность решения задачи.

5) *Имеется текстовый файл. Найти количество строк, начинающихся на букву «А». Напечатать самую длинную строку (если таких строк несколько, вывести первую из них).* Откомпилируйте программу. Проверьте правильность решения задачи.

6) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

7) Выйдите из программы ABCPascal.

### ***Программирование данных типа «запись».***

**Цель:** научиться составлять программы с использованием структурированного типа данных в программе ABCPascal.

#### **Теоретические сведения:**

Комбинированный тип данных — это структурированный тип, состоящий из фиксированного числа компонент (полей) разного типа. Комбинированный тип имеет еще и другое название — запись.

Обычно запись содержит совокупность разнотипных атрибутов, относящихся к одному объекту. В Паскале эта информация может храниться в одной переменной типа Record (запись).

К каждому элементу записи можно обратиться, используя составное имя, которое имеет следующую структуру: <имя переменной>.<имя поля>.

В программе могут использоваться массивы записей.

Любая обработка записей, в том числе ввод и вывод, производится поэлементно.

Осуществлять доступ к полям записи можно с помощью оператора присоединения. Его общий вид: **With <имя записи> Do <оператор>**. Внутри оператора к компонентам записи можно обращаться только с помощью имени соответствующего поля.

Элементы файла пронумерованы в порядке их занесения в файл, начиная с нуля. Задав номер элемента файла, можно непосредственно установить на него указатель. После этого можно читать или перезаписывать данный элемент. Установка указателя на нужный элемент файла производится процедурой Seek(FV,n). Здесь FV — имя файловой переменной, n — порядковый номер элемента.

#### **Порядок выполнения работы:**

- 1) Запустите программу ABCPascal.
- 2) Создать программу "Телефонный справочник". Справочник содержит следующую информацию: Ф.И.О. абонента, домашний адрес, номер телефона. Программа осуществляет поиск номера телефона по фамилии, по имени и отчеству абонента. Откомпилируйте программу. Проверьте правильность решения задачи.
- 3) Для каждого из 25 студентов группы известны фамилия и оценка (в баллах) по пяти дисциплинам. Требуется вычислить среднюю оценку



*каждого студента и выбрать человека, имеющего максимальный средний балл. Откомпилируйте программу. Проверьте правильность решения задачи.*

*4) Ввести сегодняшнюю дату следующим образом: Первая цифра - это число, вторая - месяц, третья – год. Определить дату завтрашнего дня. Откомпилируйте программу. Проверьте правильность решения задачи.*

*5) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.*

*6) Выйдите из программы ABCPascal.*

### ***Программирование модулей.***

**Цель:** научиться составлять программы с использованием модулей в программе ABCPascal.

#### **Теоретические сведения:**

*Модуль* — это набор ресурсов (функций, процедур, констант, переменных, типов и т.д.), разрабатываемых и хранимых независимо от использующих их программ. Модуль может содержать достаточно большой набор процедур и функций, а также других ресурсов для разработки программ. Существуют стандартные модули Турбо Паскаля (system, crt, graph и т.д.).

Модуль имеет следующую структуру:

```
Unit <имя модуля>; {заголовок модуля}
Interface {интерфейсная часть}
Implementation {раздел реализации}
Begin
{раздел инициализации модуля}
End.
```

После служебного слова ***Unit*** записывается имя модуля, которое должно совпадать с именем файла, содержащего данный модуль.

В разделе ***Interface*** объявляются все ресурсы, которые будут в дальнейшем доступны программисту при подключении модуля.

В разделе ***Implementation*** описываются все подпрограммы, которые были ранее объявлены. Кроме того, в нем могут содержаться свои константы, переменные, типы, подпрограммы и т.д., которые носят вспомогательный характер и используются для написания основных подпрограмм. В отличие от ресурсов, объявленных в разделе ***Interface***, все, что дополнительно объявляется в ***Implementation***, уже не будет доступно при подключении модуля. При описании основной подпрограммы достаточно указать ее имя, а затем записать тело подпрограммы.

Раздел инициализации содержит операторы, которые должны быть выполнены сразу же после запуска программы, использующей модуль.

#### **Порядок выполнения работы:**

- 1) Запустите программу ABCPascal.
- 2) Создать программу: *Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над обыкновенными дробями вида  $P/Q$  ( $P$  — целое,  $Q$  — натуральное):*
  - 1) сложение;
  - 2) вычитание;
  - 3) умножение;
  - 4) деление;
  - 5) сокращение

дроби;

б) возведение дроби в степень  $TV$  ( $7V$  — натуральное);

7) функции, реализующие операции отношения (равно, не равно, больше или равно, меньше или равно, больше, меньше).

Дробь представить следующим типом: **Type Frac=Record**

*P: Integer;*

*Q: 1..32767; End;*

Используя этот модуль, решить задачу:

1. Дан массив  $A$ , элементы которого — обыкновенные дроби. Найти сумму всех элементов и их среднее арифметическое; результаты представить в виде несократимых дробей.

2. \*Дан массив  $A$ , элементы которого — обыкновенные дроби.

Отсортировать его в порядке возрастания.

Откомпилируйте программу. Проверьте правильность решения задачи.

3) Создать программу: Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над векторами:

1) сложения;

2) вычитания;

3) скалярного умножения векторов;

4) умножения вектора на число;

5) нахождения длины вектора.

Вектор представить следующим типом:

**Type Vektor=Record**

*X, Y: Real;*

**End;**

Используя этот модуль, решить задачу:

Дан массив  $A$  — массив векторов. Отсортировать его в порядке убывания длин векторов.

Откомпилируйте программу. Проверьте правильность решения задачи.

4) Запишите полученные результаты в тетрадь. Оформите отчет о проделанной работе, который должен содержать тему, цель работы, формулировки задач с решениями.

5) Выйдите из программы ABCPascal.





y:=0; FOR x := 1 TO 5 do Y := Y \* x; Writeln (y);

- а) 0;                      б) 12345;                      в) 120;                      г) 00000.

**21.** Тело цикла - это...

- а) группа команд, не входящих в циклическую структуру;
- б) группа команд, повторяющихся некоторое число раз;
- в) произвольная группа команд;
- г) команды, заключенные в операторные скобки.

**22.** Какой из перечисленных заголовков циклов не содержит ошибок?

- а) FOR X:=3 TO 12 DO;
- б) FOR K = 1 TO 5.5 DO;
- в) FOR K = 2 DOWNT0 10 DO;
- г) FOR J := 10 TO -2 DO.

**23.** Что определяет индекс массива?

- а) Индекс определяет положение элемента массива данных относительно его конца;
- б) Индекс определяет положение элемента массива данных относительно друга;
- в) Индекс определяет положение элемента массива данных относительно его начала.

**24.** Укажите правильное описание массива

- а) Var a:array[1..1000] of integer;
- б) Var A: ARRAY [1..50 OF REAL];
- в) Var A, B, C: ARRAY [1 ..50] OF REAL.

**25.** Что производит следующий фрагмент программы?

```
s:=0; for i:=1 to n do
for j:=1 to n do s: =s+a[i, j]; end;
writeln('s=',s);
```

- а) Находит сумму элементов каждой строки и выводит их на экран;
- б) Находит сумму всех элементов двумерного массива и сумму элементов каждой строки и выводит их на экран;
- в) Находит сумму всех элементов двумерного массива и выводит их на экран.

## **СПИСОК ЛИТЕРАТУРЫ**

### **Литература основная:**

1. Основы программирования: Учебник для сред. проф. образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2003. – 432 с.
2. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: учеб. пособие. – 3-е изд., испр. и доп. – М.: ФОРУМ, 2008. – 432 с.

### **Литература дополнительная:**

1. Канцедал С.А. Алгоритмизация и программирование: учебное пособие. – М.: ИД «ФОРУМ»: ИНФРА-М, 2008. – 352 с.
2. Голицына О.Л., Партыка Т.Л., Попов И.И. Языки программирования: учеб. пособие. – М.: ФОРУМ: ИНФРА-М, 2008. – 400 с.

## СОДЕРЖАНИЕ

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| Введение.....                                                                              | 3  |
| Лабораторная работа №1. Программирование линейных алгоритмов                               | 5  |
| Лабораторная работа №2. Программирование разветвляющихся алгоритмов                        | 8  |
| Лабораторная работа №3. Программирование алгоритмов с использованием оператора выбора..... | 10 |
| Лабораторная работа №4. Программирование циклических алгоритмов: цикл с параметром.....    | 12 |
| Лабораторная работа №5. Программирование циклических алгоритмов: цикл с предусловием.....  | 14 |
| Лабораторная работа №6. Программирование циклических алгоритмов: цикл с постусловием.....  | 16 |
| Лабораторная работа №7. Обработка одномерных и двумерных масс                              | 17 |
| Лабораторная работа №8. Сортировка массивов. Поиск элемента массива                        | 20 |
| Лабораторная работа №9. Работа со строковым типом данных.....                              | 22 |
| Лабораторная работа №10. Работа с множественным типом данных.....                          | 25 |
| Лабораторная работа №11. Организация процедур.....                                         | 27 |
| Лабораторная работа №12. Организация функций.....                                          | 30 |
| Лабораторная работа №13. Использование рекурсивных подпрограмм.....                        | 32 |
| Лабораторная работа №14. Использование процедур и функций: графические возможности.....    | 34 |
| Лабораторная работа №15. Работа с типизированными файлами последовательного доступа.....   | 36 |
| Лабораторная работа №16. Работа с текстовыми файлами.....                                  | 38 |
| Лабораторная работа №17. Программирование данных типа «запись».....                        | 40 |
| Лабораторная работа №18. Программирование модулей.....                                     | 42 |
| Контрольное тестирование.....                                                              | 44 |
| Список литературы.....                                                                     | 47 |