

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшее профессиональное образование
«Оренбургский государственный университет»

Колледж электроники и бизнеса ОГУ

Л.А.ДЕЛЬ, А.А.ГУРЬЯНОВ

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

ЧАСТЬ 2

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРЫМ РАБОТАМ

Рекомендовано к изданию Редакционно-издательским советом
государственного образовательного учреждения
высшего профессионального образования
«Оренбургский государственный университет»

Оренбург 2008

УДК 681.3.06(075.32)

ББК 22.18я73

Д-29

Рецензент

Заместитель директора по НМР Кузюшин С.А.

Д-29 **Дель Л.А.**
Основы алгоритмизации и программирования [Текст]: методические указания к лабораторным работам: в 2 ч /Л.А.Дель, А.А.Гурьянов. – Оренбург: ГОУ ОГУ, 2008. - 55 с.

Методические указания предназначены для выполнения лабораторных работ, обеспечивающих учебный процесс по дисциплине “Основы алгоритмизации и программирования” часть 2 в колледже электроники и бизнеса ОГУ для студентов 2 курса в 4 семестре специальности 220105.51 “программное обеспечение вычислительной техники и автоматизированных систем” очной формы обучения.

Методические указания составлены с учетом Государственного образовательного стандарта среднего профессионального образования по направлению подготовки дипломированных специалистов – утвержденного 8.02.2002 Министерством Образования Российской Федерации.

ББК 22.18я73

© Дель Л.А., 2008

© Гурьянов А.А., 2008

© ГОУ ОГУ, 2008

Содержание

Введение	5
1 Лабораторная работа № 1. Составление и отладка программ с использованием процедур и функций	6
1.1 Ход работы	6
1.2 Содержание отчета	6
1.3 Варианты заданий	6
1.4 Методические указания к лабораторной работе № 1	11
1.5 Вопросы к защите лабораторной работы № 1	12
2 Лабораторная работа № 2. Составление и отладка программ со строковыми типами данных(Строки)	13
2.1 Ход работы	13
2.2 Содержание отчета	13
2.3 Варианты заданий	13
2.4 Методические указания к лабораторной работе № 2	16
2.5 Вопросы к защите лабораторной работы № 2	17
3 Лабораторная работа № 3. Составление и отладка программ с использованием множества	18
3.1 Ход работы.	18
3.2 Содержание отчета	18
3.3 Варианты заданий	18
3.4 Методические указания к лабораторной работе № 3	21
3.5 Вопросы к защите лабораторной работы № 3	26
4 Лабораторная работа № 4. Составление и отладка программ с комбинированными типами данных(Записи)	26
4.1 Ход работы	26
4.2 Содержание отчета	27
4.3 Методические указания к лабораторной работе № 4	27
4.4 Варианты заданий	31
4.5 Пример программы	36
4.6 Вопросы к защите лабораторной работы №4	37
5 Лабораторная работа № 5. Назначение, открытие и закрытие файлов	37
5.1 Ход работы	37
5.2 Содержание отчета	38
5.3 Методические указания к лабораторной работе № 5.	38
5.4 Вопросы к защите лабораторной работы № 5	48
6 Лабораторная работа № 6. Процедуры и функции модуля System	48
6.1 Ход работы	49
6.2 Содержание отчета	49
6.3 Методические указания к лабораторной работе № 6	49
6.4 Вопросы для допуска к лабораторной работе № 6	50
7 Лабораторная работа № 7. Работа с временными и внешними файлами	50
7.1 Ход работы	51

7.2 Содержание отчета	51
7.3 Варианты заданий к лабораторной работе №7	51
7.4 Вопросы к защите лабораторной работы № 7	53
8 Вопросы к экзамену по предмету основы алгоритмизации и программирования	54
Список использованных источников	55

Введение

Для успешного выполнения лабораторных работ на первом занятии студент должен получить у преподавателя график выполнения лабораторных работ на текущий семестр.

Перед посещением лаборатории изучить теорию вопроса, предполагаемого к исследованию, ознакомиться с руководством по соответствующей работе.

Оформление отчетов должно производиться после окончания работы непосредственно в лаборатории, при наличии свободного времени или дома.

Для подготовки к защите отчета следует проанализировать разработанные программные средства, сопоставить их с целевым назначением задания, обобщить результаты исследований в виде лаконичных выводов по работе.

В данных методических указаниях рассмотрены особенности применения подпрограмм в виде процедуры или функций, примеры для усвоения темы.

Структурированные типы данных: записи, строки., файлы. Организация доступа к файлам, процедуры и функции для работы с файлами, текстовые типизированные и не типизированные файлы.

В таблице 1 представлены этапы и календарные сроки выполнения лабораторных работ

Таблица 1- Этапы выполнения лабораторных работ

№ занятия	Наименование разделов и тем	Кол-во часов практика	Календарные сроки изучения тем
1	2	3	4
1	Лабораторная работа №1 «Составление и отладка программ с использованием процедур и функций»	4/4	2 неделя
2	Лабораторная работа №2 «Составление и отладка программ со строковыми типами данных».	6/10	4 неделя
3	Лабораторная работа №3 «Составление и отладка программ с использованием множеств».	4/14	6 неделя
4	Лабораторная работа №4 «Составление и отладка программ с комбинированными типами данных».	8/22	8 неделя
5	Лабораторная работа №5 «Назначение, открытие и закрытие файлов».	4/26	9 неделя
6	Лабораторная работа №6 «Процедуры и функции модуля System».	4/30	11 неделя
7	Лабораторная работа №7 « Работа с временными и внешними файлами».	10/40	13 неделя

1 Лабораторная работа №1. «Составление и отладка программ с использованием процедур и функций»

Цель работы: «Получение навыков в написании программ с использованием процедур и функций»

1.1 Ход работы:

- 1 Составить блок-схему по заданию своего варианта
- 2 Запустить интегрированную среду программирования Turbo Pascal
- 3 Набрать текст программы
- 4 Откомпилировать программу
- 5 Сохранить файл с программой на диске A: под именем lab1.pas
- 6 Распечатать текст программы и её результат
- 7 Составить отчет

1.2 Содержание отчёта:

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задачи
- 5 Блок-схема
- 6 Листинг программы
- 7 Результаты программы

1.3 Варианты заданий:

1 Напишите программу, которая по значениям двух катетов вычисляет гипотенузу, площадь, радиусы вписанной и описанной окружностей прямоугольного треугольника?

2 Напишите программу, состоящую из трёх процедур и основной программы. Первая процедура организует ввод двух целых чисел X и Y, вторая вычисляет их сумму, третья выводит результат.

3 Напишите программу вычисления площади поверхности и длины экватора на основе известного радиуса планет солнечной системы. Форму планет будем считать шаром. Вычисление площади и длины экватора оформите отдельными функциями.

4 Составить программу поиска большего из четырёх чисел с использованием подпрограммы поиска большего из двух.

5 Даны координаты вершин многоугольника($X_1, Y_1, X_2, Y_2, \dots, X_{10}, Y_{10}$). Определить его периметр(вычисление расстояния между вершинами оформить подпрограммой).

6 Вычислить сумму: $1!+2!+3!+\dots+n!$, используя функцию вычисления факториала числа k.

7 Вычислить сумму простых и сверхпростых чисел, не превосходящих заданного числа N.

8 Составьте программу вычисления сочетания из N по M. Число сочетаний определяется по формуле: $\frac{N!}{M!(N-M)!}$. Используйте подпрограмму вычисления факториала.

9 Определить НОД трёх натуральных чисел.

10 Даны действительные числа S, t. Составить программу вычисления выражения: $f(t, -2S, 1.17) + f(2/2, t, S-1)$, где $f(a, b, c) = (2a - b - \sin(c)) / (5 + |c|)$

11 Составить программу перевода двоичной записи натурального числа в десятичную.

12 Составить программу сокращения дроби $\frac{M}{N}$, где M и N натуральные числа

13 Составить программу вычисления суммы квадратов простых чисел, лежащих в интервале (M..N).

14 Составьте программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр.

15 Составьте программу нахождения наименьшего натурального N-значного числа X ($x \geq 10$), равного утроенному произведению своих цифр

16 Составить программу сравнения площадей и периметров двух параллелограммов

17 Написать программу сравнения площадей и периметров двух прямоугольных треугольников, если заданы их катеты.

18 Вычислить значение выражения:

$y := a_1 x^4 + a_2 x^3 + a_3 x^2 + a_4 x + a_5$, где коэффициенты a_1, a_2, a_3, a_4, a_5 и x – это числа, вводимые с клавиатуры.

19 Упорядочить значения трех переменных a, b и c в порядке их возрастания, используя процедуру перемены местами значений двух переменных.

20 Даны координаты трех вершин треугольника. Найти длины всех его сторон.

21 Дано натуральное число. Найти все его делители. Подсчитать их количество.

22 Даны два натуральных числа. Определить, является ли первое число перевертышем второго?

23 Даны координаты трех вершин треугольника ABC и даны координаты четвертой точки D. Определить, является ли эта точка внутренней точкой треугольника.

24 Составить программу нахождения наибольшего общего делителя нескольких чисел, используя функцию нахождения НОД двух чисел (алгоритм Евклида).

25 Даны действительные числа s, t. Получить $F(t, -2s, 1.17) + F(2.2, t, s - t)$,

Где $F(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}$.

26 Даны действительные числа s, t. Получить $g(1.2, s) + g(t, s) - g(2s, t)$,

Где $n(a, b)$

27 Даны действительные числа a, b, c.

Получить $\max(a, a+b) + \max(a, b+c)$

$$1 + \max(a+bc, 15)$$

28 Даны действительные числа a , b .

Получить $u = \min(a, b)$, $v = \min(ab, a+b^2)$, $\min(u + v, 3.14)$.

29 Дано натуральное число n . Среди чисел $1, 2, \dots, n$ найти все те, которые можно представить в виде суммы квадратов двух натуральных чисел. (Определить процедуру, позволяющую распознавать полные квадраты.)

30 Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Найти периметр десятиугольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. (Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами).

31 Даны действительные числа a, b, c, d . Найти площадь пятиугольника. (Определить процедуру вычисления площади треугольника по трем его сторонам).

32 Даны натуральное число n , действительные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Найти площадь n -угольника, вершины которого при некотором последовательном обходе имеют координаты $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

(Определить процедуру вычисления площади треугольника по координатам его вершин).

33 Дано натуральное число n . Выяснить, имеются ли среди чисел $n, n+1, \dots, 2n$ близнецы, т.е. простые числа, разность между которыми равна двум. (определить процедуру, позволяющую распознавать простые числа).

34 Дано натуральное число n ; найти $n!$. Использовать программу, включающую рекурсивную процедуру вычисления $n!$. Чем эта программа хуже не рекурсивной программы вычисления программы вычисления $n!$?

35 Даны натуральные числа n, m ; найти НОД (n, m). Использовать программу, включающую рекурсивную процедуру вычисления НОД, основанную на соотношении НОД, основанную на соотношении НОД (n, m)=НОД (m, r), где r - остаток от деления n на m). Чем эта программа хуже не рекурсивной программы вычисления НОД (n, m)?

36 Даны натуральные числа a, c, m . Получить $f(m)$, где

$$f(n) = \begin{cases} n, & \text{если } 0 \leq n \leq 9 \\ g(n) f(n-1-g(n)) + n & \text{в противном случае} \\ g(n) = \text{остаток от деления } an + c \text{ на } 10 \end{cases}$$

Использовать программу, включающую рекурсивную процедуру вычисления $f(n)$.

37 Даны неотрицательные целые числа n, m ; вычислить $A(n, m)$,

где

$$A(n, m) = \begin{cases} m+1, & \text{если } n=0, \\ A(n-1, 1), & \text{если } n \neq 0, m=0 \\ A(n-1, A(n, m-1)), & \text{если } n > 0, m > 0 \end{cases}$$

Вычислить z -сумму значения функций $z=f(a, b)+f(a^2, b^2)+f(a^2-1, b)+f(a-b, b)+f(a^2+b^2, b^2-1)$,

$$f(u, t) = \begin{cases} u^2 + t^2, & \text{если } u > 0, t > 0; \\ u + t^2, & \text{если } u \leq 0, t \leq 0; \\ u - t, & \text{если } u > 0, t \leq 0; \\ u + t, & \text{если } u \leq 0, t > 0; \end{cases}$$

где

а) $a=2,5$; $b=-7,3$;

б) $a=-0,5$; $b=4,2$;

в) $a=-0,2$; $b=-0,42$;

г) $a=23,7$; $b=41,2$;

38 Вычислить z-сумму значения функций

$$z = f(\sin \alpha, a) + f(\cos \alpha, a) + f(\sin^2 \alpha, a-1) + f(\sin \alpha - \cos \alpha, a^2-1) + f(\sin^2 \alpha - 1, \cos \alpha + a),$$

$$f(u, t) = \begin{cases} u + \sin(t), & \text{если } u > 0; \\ u + t, & \text{если } u \leq 0; \end{cases}$$

где

а) $\alpha = \pi/18$; $a = -2,1$; б) $\alpha = 2,3$; $a = 21,3$;

в) $\alpha = -\pi/14$; $a = -0,2$; г) $\alpha = -\pi/10$; $a = 31,2$;

39 Вычислить z-сумму значения функций

$$z = f(\sqrt{|x|}, y) + f(a, b) + f(\sqrt{|x|} + 1, -y) + f(|x| - |y|, x) + f(x + y, a + b),$$

$$f(u, t) = \begin{cases} u + 2t, & \text{если } u \geq 0; \\ u + t, & \text{если } u \leq -1; \\ u + t, & \text{если } -1 < u < 0; \end{cases}$$

где

а) $x=2,31$; $y=4,2$; $a=3,1$; $b=0,02$;

б) $x=-4,21$; $y=-31,2$; $a=1,2$; $b=-3,2$;

в) $x=0,34$; $y=17,2$; $a=-4,6$; $b=-0,44$;

г) $x=-14,2$; $y=0,31$; $a=7,2$; $b=4,7$;

40 Вычислить z-сумму значений функций

$$z = f(\sin(x) + \cos(y), x + y) + f(\sin(x), \cos(y)) + f(x - y, x) + f(\sin^2(x) - 2, a) + f(a + 3, b + 1),$$

$$f(u, t) = \begin{cases} u + t, & \text{если } u > 1; \\ u - t, & \text{если } 0 \leq u \leq 1; \\ t - u, & \text{если } u < 0; \end{cases}$$

где

а) $x=3,14$; $y=0,41$; $a=0,1$; $b=-2,1$;

б) $x=0,32$; $y=3,14/10$; $a=-0,21$; $b=4,2$;

в) $x=19,2$; $y=0,48$; $a=-4,3$; $b=-6,1$;

г) $x=0,62$; $y=2/3$; $a=17,1$; $b=0,2$;

41 Даны действительные числа $a_1, \dots, a_n, b_1, \dots, b_n$. В последовательности a_1, \dots, a_n , и в последовательности b_1, \dots, b_n все элементы, следующие за элементом с наибольшим значением(за первым по порядку, если их несколько), заменить на 0.5.

42 а) Написать программу, в ходе выполнения которой компоненты файла F1 переписываются в файл F2, а компоненты файла F2 - в файл F1. Использовать файл H как вспомогательный. Компоненты всех файлов имеют тип real. В Паскале не разрешены операторы присваивания вида: $F:=G?$ где F и G имена файлов, поэтому следует описать процедуру присваивания присв(F, G);

б) С помощью процедуры присв(F, G) (см. задание а)) написать программу, в ходе выполнения которой файлы F1, F2, F3, F4, F5 обмениваются компонентами в соответствии со следующей схемой:

F1 F2 F3 F4 F5
F3 F4 F5 F3 F1

т.е. компоненты файла F1 переписываются в файл F3, компоненты файла F2 переписываются в файл F4 и т.д. Разрешается использовать только один дополнительный файл.

43 Дано: натуральное n , действительные a_1, \dots, a_{3n} . Получить $x+y^2+z^2$, где

$x=a_1*a_2*\dots*a_n,$
 $y=a_{n+1}*a_{n+2}*\dots*a_{2n},$
 $z=a_{2n+1}*a_{2n+2}*\dots*a_{3n}.$

44 Даны натуральные k, m . Требуется вывести на экран рамку из звездочек

```
*****  
*      *  
*      *  
*      *  
*****
```

высота которой – k строк, ширина – m знаковых позиций. Полезно описать процедуру печати (s, n), обращение к которой дает вывод n символов s.

44 Дано: натуральное n , целые неотрицательные a_1, \dots, a_n . Рассмотреть отрезки последовательности a_1, \dots, a_n (идущих подряд членов), состоящей из:

- а) полных квадратов;
- б) степеней пятерки;
- в) простых чисел;
- г) совершенных чисел.

В каждом случае получить наибольшую из длин рассматриваемых отрезков. (Описать процедуры, позволяющие распознавать полные квадраты, степени пятерки, простые числа, совершенные числа.)

45 Даны действительные $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Найти периметр десятиугольника, вершины которого имеют, соответственно, координаты $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Описать процедуру вычисления расстояния между двумя точками, заданными своими координатами.

46 Даны целые $u_1, u_2, v_1, v_2, w_1, w_2$ ($u_2, v_2, w_2 \neq 0$). Вычислить $2u + \frac{3uw}{2+v^2} - 7$,

где u, v, w - рациональные числа $u_1/u_2, v_1/v_2, w_1/w_2$. Ответ получить в виде двух взаимно простых чисел p_1, p_2 – числителя и знаменателя дроби, являющейся значением выписанного выражения. Описать процедуру полного сокращения рационального числа, заданного числителем и знаменателем x, y ($y > 0$). Описать также процедуры сложения и умножения рациональных чисел.

1.4 Методические указания к лабораторной работе №1

1.4.1 Процедуры и функции. Структура процедур и функций.

В турбопаскале имеются средства деления программы на части. На верхнем уровне это модули, на нижнем уровне деления это процедуры и функции.

Для того чтобы процедура или функция начала работать её необходимо вызвать. Процедура или функция состоит из операторов, локальных данных и внутренних процедур и функций.

Процедура и функция – это часть программы с одним входом и выходом. Основной файл программы содержит главный блок – основная программа, который с необязательным заголовком, который может обращаться к вложенным в неё блокам. Каждый из подблоков подчиняется главному или старшему блоку, в котором содержится его вызов.

Блочной называется программа, состоящая из двух блоков. Если результатом блока является одно значение, блок оформляется как функция:

Function имя функции(список формальных параметров):тип результата;
Label }
Const } описание локальных меток, констант, переменных, типов.
Var }

Procedure } .
Function } описание внутренних процедур и функций

Begin }
. . } операторы, среди которых должен быть хотя бы один,
End } который присваивает имени функции значение результата.

Аргументы функции называют формальными параметрами, и содержатся в списке формальных параметров. Результат функции присваивается её имени. Результаты блока могут быть так же указаны в списке параметров, в этом случае используется не функция а процедура:

Procedure: имя процедуры (список формальных параметров);

Label
Const
Var } описание локальных меток, констант, переменных, типов.

Procedure
Function } описание внутренних процедур и функций.

Begin
.
.
.
End } операторы.

Имя процедуры не используется для обозначения результата, а в списке формальных параметров имени результата предшествует слово Var.

Если результаты имеют разный тип, то в списке формальных параметров каждому имени результата предшествует слово Var.

Обращение к процедуре является предложением языка, оно не может быть использовано в выражениях в операторах присвоения и вывода.

1.5 Вопросы для защиты лабораторной работе №1

1 Что понимается под структурным программированием? Зачем оно применяется?

2 Что называется программой?

3 В чём различие между стандартными и определёнными пользователями программами? Привести пример.

4 Что такое подпрограмма – процедура, подпрограмма – функция? Для чего они предназначены?

5 Сходство и отличие процедур и функций в языке Turbo Pascal?

6 Как в программе объявляется процедура, функция?

7 Как в программе вызывается процедура, функция?

8 Что называется параметром, и каково его назначение?

9 Каковы отличия параметров-значений от параметров переменных, особенности их описания и применения?

10 Чем отличаются локальные и глобальные параметры? Какова область их действия?

2 Лабораторная работа №2. Составление и отладка программ со строковыми типами данных (Строки).

Цель работы: Получение навыков в организации ввода/вывода значений строковых типов данных. Получение практических навыков программирования задач с использованием строк.

2.1 Ход работы:

- 1 Составить и отладить программу используя действия со строками и функции `Contact`, `Length`
- 2 Составить и отладить программу используя `Pos`, `Delete`, `Insert`
- 3 Выполнить индивидуальное задание по данной теме
- 4 Составить блок-схему по заданию своего варианта
- 5 Запустить интегрированную среду программирования Turbo Pascal набрать текст программы
- 6 Откомпилировать программу
- 7 Сохранить файл с программой на диске A: под именем `lab3.pas`
- 8 Распечатать текст программы и результат её работы
- 9 Составить отчёт

2.2 Содержание отчёта:

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задачи
- 5 Блок-схемы к каждой задаче
- 6 Листинг программы
- 7 Результат выполнения программы

2.3 Варианты заданий:

1 Напишите программу, подсчитывающую количество букв во введенном с клавиатуры слове. Ввод осуществляйте в цикле `while do`. Выход из программы – строка '999'.

2 Напишите программу, подсчитывающую количество вхождений заданной вами буквы в введенной строке.

3 Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на 1 символ до тех пор, пока в строке не останется 1 символ.

4 Напишите программу, определяющую число слов в строке. Одно слово от другого отделяется 1 пробелом.

- 5 Составьте программу, определяющую, является ли введенное слово числом.
- 6 Введите 2 целых числа. Преобразуйте числа в две строки, объедините их в одну строку и выведите на экран результат.
- 7 Напишите программу, которая удаляет из введенной строки любой требуемый введенный с клавиатуры символ. Процесс удаления выделите в отдельную процедуру DelChInString, строку и символ определите как глобальные переменные. В результате у вас должна получиться программа-модель работы одного из режимов любого текстового редактора.
- 8 Составьте программу, удаляющую все пробелы из введенной строки. Для удаления постройте отдельную функцию NewSt и примените в ней оператор Repeat и функцию Pos.
- 9 Напишите программу, сортирующую символы введенной с клавиатуры строки в порядке возрастания их номеров в ASCII-таблице. Например, если введено: 'CBA', в результате надо получить 'ABC'.
- 10 Вычислите длину самого короткого слова в предложении из трех слов, разделенных пробелами.
- 11 Выясните, какая из букв первая или последняя встречается в заданном слове чаще.
- 12 Задано существительное первого склонения, оканчивающееся на "а". Напечатайте это слово во всех падежах.
- 13 Сколько букв "у" в слове стоит на четных местах?
- 14 Замените в заданном слове все буквы "о" пробелами.
- 15 Даны два слова. Поменяйте местами буквы этих слов, занимающие одинаковые позиции.
- 16 Заданы фамилия, имя и отчество учащегося, разделенные пробелом. Напечатайте его фамилию и инициалы.
- 17 Вычеркните i-ю букву слова.
- 18 Дан текст, в котором слова разделены пробелами.
 - а) сколько слов в тексте?
 - б) найдите самое длинное слово текста (длина текста 100 символов).
- 19 Задан текст, состоящий из слов, которые разделены одним или несколькими пробелами. Сформируйте новый текст, включив в него слова заданного, разделенные только одним пробелом.
- 20 Сложное слово состоит из двух частей одинаковой длины и соединительной гласной. Найдите обе части этого слова.
- 21 Вычеркните из заданного слова все буквы, совпадающие с его последней буквой.
- 22 Вычеркните из слова X те буквы, которые встречаются в слове Z.
- 23 Подсчитайте число различных букв в слове.
- 24 Составьте программу подсчета сколько раз в тексте встречается заданный фрагмент (цепочка символов). Например, в тексте "банан упал на барабан" фрагмент "ба" встречается 2 раза.
- 25 Даны два слова. Составьте программу, определяющую можно или нет из букв слова A составить слово B.
- 26 Составьте программу, вычеркивающую каждую третью букву слова X.

27 Составьте программу, выясняющую, на гласную или согласную букву оканчивается слово X.

28 Составьте программу вычисления суммы мест, на которых в слове x стоят буквы "в" и "п".

29 Составьте программу шифрования текстового сообщения. Можно использовать такой способ шифрования. Шифровальщик задает ключ шифровки — целое число, который определяет величину смещения букв русского алфавита, например: ключ = 3, тогда в тексте буква "а" заменяется на "г" и т. д. Используются все буквы русского алфавита. Е считается дважды.

30 Составьте программу дешифрования текстового сообщения, зашифрованного программой из задачи 29.

31 Дано слово. Определить, одинаковы ли второй и четвертый символы в нем. Вывести на экран его последний символ. Определить количество символов в нем.

32 Даны два слова. Верно ли, что первое слово начинается на ту же букву, которой оканчивается второе слово? Определить количество символов в каждом слове. Вывести на экран слова в обратном порядке.

33 Дано слово, состоящее из четного числа букв. Вывести на экран его первую половину. Вывести слово в обратном порядке в середине экрана.

34 Из слова "программа" путем вырезок и склеек его букв получить слова "ром" и "рампа". Из последних двух слов вывести на экран те буквы, которые стоят на четных местах.

35 Из слова "клоун" путем вырезок и склеек его букв получить слова "уклон", "кулон" и "колун". Из последних трех слов вывести на экран те буквы, которые стоят на нечетных местах.

36 Дано слово из четного числа букв. Поменять местами его половины. Вывести на экран слова в обратном порядке, в середине экрана.

37 Дано слово из 12 букв. Поменять местами его трети следующим образом: первую треть слова разместить на месте третьей, вторую треть — на месте первой, третью треть — на месте второй. Удалить из получившегося слова каждый третий символ.

38 Дано слово s1, Получить слово s2, образованное нечетными буквами слова s1. Вывести на экран слова в обратном порядке.

39 Дано предложение. Вывести все буквы «м» и «н» в нем и подсчитать их количество. Сколько слов в вашем предложении?

40 Даны два предложения. Напечатать слова, которые встречаются в двух предложениях только один раз. Какое из предложений содержит большее число символов?

41 Дано предложение. Определить: а) число вхождений в него буквосочетания "до"; б) число вхождений в него некоторого буквосочетания из двух букв; в) число вхождений в него некоторого буквосочетания.

42 Дано предложение. Все его символы, стоящие на четных местах, заменить на букву "ы". Вывести в середине экрана первое и последнее слово предложения. Какое из выведенных слов длиннее другого?

2.4. Методические указания к лабораторной работе №2

Строка – последовательность символов, кодовой таблицы ПК. Длина строки динамически изменяется от 0 до 255.

Переменные строки должны быть описаны предложениями:

Type <имя типа>=String[макс длина строки];

Var имя:<имя типа>;

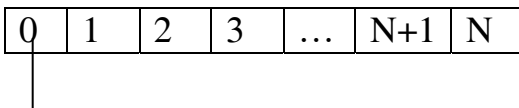
Const;

Строка трактуется как цепочка символов, для строки из N символов, отводится N+1 байт.

N байт – для хранения символов в строке;

1 байт - для значения текущей длины строки.

К любому символу строки можно обратиться, указав его номер:



Текущая длина строки.

M – макс длина строки

L – текущая длина

A – ячейка памяти

A+1 – первый символ строки

A+L – последний действительный символ

A+L+1 ; A+M – незанятые ячейки памяти

Действия со строками:

1 Сложение (конкатенация строк изображается знаком «+», при его выполнении начало строки идущей после знака «+» подсоединяется к концу строки, указанной до этого знака, аналогичная операция Concat)

2 Сору(строка, номер, количество) – возвращает подстроку, заданной строки, начинающуюся в позиции, указанной номером и содержащую заданное количество символов.

3 Delete(St, Poz, N) – удаление N- символов и строки St начиная с позиции Poz.

4 Insert(Str1, Str2, Poz) – замена строки Str1, строкой Str2, в указанной позиции Poz.

5 Val(St, x, code) – преобразует значение St в величину целочисленного или вещественного типа и помещает результат x, значение St не должно содержать незначащих пробелов в начале и в конце. Если во время операции преобразования ошибка не обнаружена, значение code равно нулю, если ошибка обнаружена code будет содержать номер первого ошибочного символа, а значение x не определено.

6 Str(x, st) – преобразование числового значения величины x и помещение результата в строку St, после x может записаться формат, аналогичный формату

вывода, если в формате указано недостаточное для вывода количество разрядов, после вывода расширяется автоматически.

7 Pos(str1,str2) – обнаруживает первое появление в строке Str2 подстроки str1, результат имеет целочисленный тип и равен номеру той позиции, где находится первый символ подстроки Str1, если в Str2, подстроки Str1 не найдено, результат равен нулю.

8 Length(St) – вычисляет текущую длину символов в строке St, результат имеет целочисленный тип.

9 UpCase(ch) – преобразует строчную букву в прописную. Параметр и результат имеет литерный результат, обрабатывает буквы только латинского алфавита.

Порядковые типы данных.

Если нужно по коду получить символ, используется функция Chr(код), который записывается справа от «#».

Каждое значение порядкового типа связано с порядковым номером I, узнать значение I можно с помощью функции Ord - I:=ord(z)

Говорят, что z1 и z2 порядкового типа

Z1> если ord(z1)>ord(z2) в этом смысле 'd'>'a'

Для целых чисел значение функции равно аргументу Ord(5)=5

Succ(z) – даёт следующее по отношению к Z значение

Pred(z) - предыдущее

2.5 Вопросы для защиты лабораторной работы №2

1 Что такое строка?

2 Каким идентификатором определяются данные строкового типа?

3 Какова максимально возможная длина строки? Как определить текущую длину строки?

4 Какие выражения называют строковыми?

5 Какие операции допустимы над строковыми данными?

6 Каким образом производится сравнение строк?

7 Как можно обратиться к отдельным символам строки?

8 Как производится сцепление строк?

9 Назначение специальных процедур и функций обработки данных строкового типа. Привести примеры.

3 Лабораторная работа №3 Составление и отладка программ с использованием множества

Цель работы: Получение навыков в написании программ с использованием множества»

3.1 Ход работы:

- 1 Составить блок-схему по заданию своего варианта
- 2 Составить программу по заданию своего варианта
- 3 Запустить интегрированную среду программирования Turbo Pascal
- 4 Набрать текст программы
- 5 Откомпилировать программу
- 6 Сохранить файл с программой на диске A: под именем lab3.pas
- 7 Распечатать текст программы и её результат
- 8 Составить отчет

3.2 Содержание отчёта:

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задачи
- 5 Блок-схема
- 6 Листинг программы
- 7 Результаты программы

3.3 Варианты заданий:

1 Известны сорта роз, выращиваемых тремя цветоводами: «Анжелика», «Виктория», «Гагарин», «Ave Maria», «Катарина», «Юбилейная». Определить те сорта, которые имеются у каждого из цветоводов, которые есть хотя бы у одного из цветоводов, которых нет ни у одного из цветоводов.

2 Заданы имена девочек. Определить, какие из этих имен встречаются во всех классах данной параллели, которые есть только в некоторых классах и какие из этих имен не встречаются ни в одном классе.

3 Задан некоторый набор товаров. Определить для каждого из товаров, какие из них имеются в каждом из n магазинов, какие товары есть хотя бы в одном магазине и каких товаров нет ни в одном магазине.

4 Имеется список класса (все имена различны). Определить, есть ли в классе человек, который побывал в гостях у всех. (Для каждого ученика составить множество побывавших у него в гостях друзей, сам ученик в это множество не входит.)

- 5 Имеется множество, содержащее натуральные числа из не которого диапазона. Сформировать два множества, первое из которых содержит все простые числа из данного множества, а второе — все составные.
- 6 На трех участках возделывают сельскохозяйственные культуры. Известны виды культур, выращиваемых на каждом из участков. Определить виды тех культур, которые возделывают на каждом из участков; возделывают хотя бы на одном из участков; не возделывают ни на одном участке. (Культуры: картофель, укроп, морковь, горох, капуста, редис.)
- 7 Известны марки машин, изготавливаемых в данной стране и импортируемых за рубеж. Даны некоторые N стран. Определить для каждой из марок, какие из них были:
- доставлены во все страны;
 - доставлены в некоторые из стран;
 - не доставлены ни в одну страну.
- 8 В озере водится несколько видов рыб. Три рыбака поймали рыб, представляющих некоторые из имеющихся видов. Определить:
- какие виды рыб есть у каждого рыбака;
 - какие рыбы есть в озере, но нет ни у одного из рыбаков.
- 9 В N колхозах выращивают некоторые сельскохозяйственные культуры из имеющегося перечня. Определить культуры:
- возделываемые во всех колхозах;
 - возделываемые только в некоторых колхозах.
- 10 Есть список игрушек, некоторые из которых имеются в N детских садах. Определить игрушки из списка:
- которых нет ни в одном из детсадов;
 - которые есть в каждом из детсадов.
- 11 Составить программу, которая вычисляет сумму тех элементов двумерного массива, номера строк и столбцов которых принадлежат соответственно непустым множествам S_1 и S_2 .
- 12 Задано некоторое множество M и множество T того же типа. Подсчитать, сколько элементов из множеств T и M совпадает.
- 13 Из диапазона целых чисел $m \dots n$ выделить:
- 1) множество чисел, делящихся без остатка или на k , или на $(k, I$ — простые);
 - 2) множество чисел, делящихся на k и I без остатка.
- 14 Дан текст замените текст из цифр и строчных латинских букв, за которыми следует точка. Определить, каких букв — гласных ($я, е, у, ю, и, а, о, и$) или согласных — больше в этом тексте.
- 15 Подсчитать количество различных цифр в десятичной записи натурального числа.
- 16 Напечатать в возрастающем порядке все цифры, не входящие в запись данного натурального числа.
- 17 Дан текст из строчных латинских букв, за которыми следует точка. Напечатать все буквы, входящие в текст не менее двух раз.
- 18 Дан текст из строчных латинских букв, за которыми следует точка. Напечатать все буквы, входящие в текст по одному разу.

19 Дан текст, за которым следует точка. В алфавитном порядке напечатать все строчные русские гласные буквы (*а, е, и, о, у, ы, э, ю, я*), входящие в этот текст.

20 Дан текст на русском языке. Напечатать в алфавитном порядке все гласные буквы, которые входят в каждое слово.

21 Дан текст на русском языке. Напечатать в алфавитном порядке все согласные буквы, которые не входят ни в одно слово.

22 Дан текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят в каждое не четное слово и не входят ни в одно четное слово.

23 Дан текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят хотя бы в одно слово.

24 Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые не входят хотя бы в одно слово.

25 Дан текст на русском языке. Напечатать в алфавитном порядке все согласные буквы, которые входят только в одно слово.

26 Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые не входят только в одно слово.

27 Дан текст на русском языке. Напечатать в алфавитном порядке все звонкие согласные буквы, которые входят более чем в одно слово.

28 Дан текст на русском языке. Напечатать в алфавитном порядке все гласные буквы, которые не входят более чем в одно слово.

29 Дан текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые входят в каждое не четное слово и не входят хотя бы в одно четное слово.

Задачи повышенной сложности:

1 Дан текст, являющийся русскими словами. Выполнить разделение каждого его слова на части для переноса. Далее правила переноса русских слов.

а) Две идущие подряд гласные можно разделить, если первой из них предшествует согласная, а за второй идет хотя бы одна буква (буква *й* при этом рассматривается вместе с предшествующей гласной как единое целое).

б) Две идущие подряд согласные можно разделить, если первой из них предшествует гласная, а в той части слова, которая идет за второй согласной, имеется хотя бы одна гласная (буквы *ъ, ь* вместе с предшествующей согласной рассматриваются как единое целое).

с) Если не удастся применить пункты а, б, то следует попытаться разбить слова так, чтобы первая часть содержала более чем одну букву и оканчивалась на гласную, а вторая содержала хотя бы одну гласную. Вероятность правильного разбиения увеличивается, если предварительно воспользоваться хотя бы неполным списком приставок, содержащих гласные, и попытаться, прежде всего, выделить из слова такую приставку.

2 Написать программу раздачи карт при игре в дурака, количество игроков задается с клавиатуры.

3 Для произвольного символьного множества сгенерировать все подмножества.

Примечание: Гласные буквы - а,е,и,о,у,ы,э,ю,я (ё обычно не входит в литерный тип); согласные - все остальные буквы, кроме ь, ъ; звонкие согласные - б,в,г,д,ж,з,й,л,м,н,р; глухие согласные - к,п,с,т,ф,х,ц,ч,ш,щ.

3.4 Методические указания к лабораторной работе №3

Множества -это структурированный тип данных, представляющий собой набор взаимосвязанных по какому – либо признаку или группе признаков объектов, которые можно рассматривать как единое целое.

Каждый объект в множестве называется элементом множества, все элементы множества должны принадлежать одному типу. Этот тип называется базовым типом множества, он задаётся диапазоном, или перечислением, область значений типа множества, это набор всевозможных подмножеств, составленных из элементов базового типа, они указываются в квадратных скобках. Если множество не имеет элементов, оно называется пустым и обозначается пустыми квадратными скобками [], количество элементов множества называется его мощностью. В Turbo Pascal в качестве базовых могут использоваться порядковые типы, мощность которых не превышает 256 значений. Порядковые значения, верхние и нижние границы базового типа не должны выходить за пределы диапазона от 0 до 255. в качестве базового типа не рекомендуется использовать тип LongInt, Integer, Word, ShortInt.

Формат записи:

1) type имя типа = Set of элемент1, ... элементN;

Var идентификатор: имя типа;

2) Var идентификатор Set of элемент1, ...;

Операции над множествами:

In – используется для проверки принадлежности какого – либо значения к указанному множеству, применяется в условных операторах if. При использовании операции in на принадлежность значения и множества в квадратных скобках не обязательно предварительно описывать в разделе описания. Операция in наглядно производит сложные проверки условий значения десяти других операций:

if (a=1) or (a=2) or (a=3) or (a=4) then if a in [1..4] then

часто операция in пытаются записать с отрицанием, это ошибка, правильно будет not (x in M) нахождение простых чисел по методу Сито по определению простым является нарицательное число $x > 1$, которое не разложено на множители, кроме множителей L и x. простые числа, исключая 2, могут быть получены по формуле $x = 2 * L + 1$, где L – основа это любое число, не присутствующее ни в одной из следующих бесконечных арифметических прогрессий, их число так же бесконечно:

1) 4,7,10,13,16,19,22...

2) 12,17,22,27,32,37...

3)2,31,38,45,52,59,66,73...

$A=2k*(k+1); a+2r, a+3r...$

$R=2k+1$ K – номер прогрессии

Если поставлена задача найти всё кроме числа 2, простые числа, не превосходящие нуля, заданного числа n , нужно выявить все основы для диапазона $[1...n \text{ div } 2]$. Метод решения, строим полное множество $[1...n \text{ div } 2]$ исключаем из него числа, состоящие из прогрессий, оставшиеся числа используем как основу L , и печатаем это значение $2L+1$.

Переменная множественного типа может принимать как все значения множества, так и ни одного. Любой множественный тип может принимать значение [], которое называется пустым множеством.

Пример:

type Digits = set of 0..9; Letters = set of 'A'..'Z';

В таблице 2 представлены операции над множествами.

Таблица 2-Операции над множествами

Операция	Действие	Результат
+	Объединение	Порядковое значение s содержится в $a+b$ только тогда, когда он содержится в a или в b .
-	Разность	Порядковое значение s содержится в $a-b$ только тогда, когда оно содержится в a и не содержится в b .
*	Пересечение	Порядковое значение s содержится в $a*b$ только тогда, когда он содержится в обоих множествах a и b .

Если наименьшим порядковым значением, которое является членом результата операций над множествами, является a , а наибольшим - b , то типом результата будет множество $a..b$.

Напишем программу использующую множества. На примере программы реализующей детскую считалку.

```
program kinder;
const nmax = 20;
var n, {Число игроков} m, {Число слов в считалке}
    i, j: integer; krug: set of 1..nmax;
begin
    Writeln ('Введите значения n и m'); readln(n); readln(m);
    krug:=[1..n];{В круге стоит n детей} i:=n;
    repeat for j:=1 to m do
        repeat i:= i mod n+1; until i in krug; writeln(i:3); krug:=krug-[i];
    until krug=[];
end.
```

Программирование с использованием множеств

В отличие от массивов к элементам множества нет прямого доступа (по индексам этих элементов, как в массивах). Поэтому ввод-вывод множеств производится с использованием операций объединения (при вводе) и проверки принадлежности (при выводе). Под мощностью множества понимают количество элементов, содержащихся в данном множестве.

Перед выполнением работы необходимо ознакомиться с правилами описания и использования переменных типа множество, типизированных констант типа множество, переменных, заданных перечислением, изучить допустимые операции над переменными этих типов.

Пример1: Дан текст. Определить каких букв больше - гласных или согласных.
Этапы решения задачи: на рисунке 1. Составлена блок-схема к программе



Рисунок 1- Блок-схема к программе

На рисунке 2..Описан подробнее блок "Подсчитываем количество гласных и согласных букв"

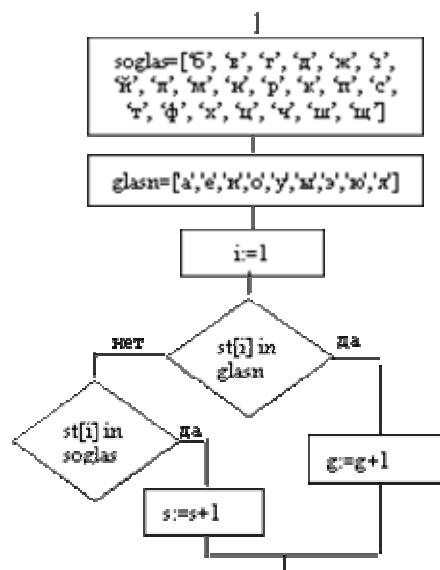


Рисунок 2 -Блок –схема "Подсчитывает количество гласных и согласных букв"

На рисунке 3. Рассмотрен блок "Печатает соответствующее сообщение"

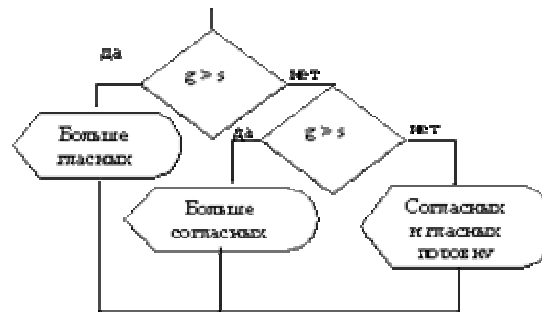


Рисунок 3 - Блок –схема "Печатает соответствующее сообщение"

На рисунке 4 показана блок-схема целиком

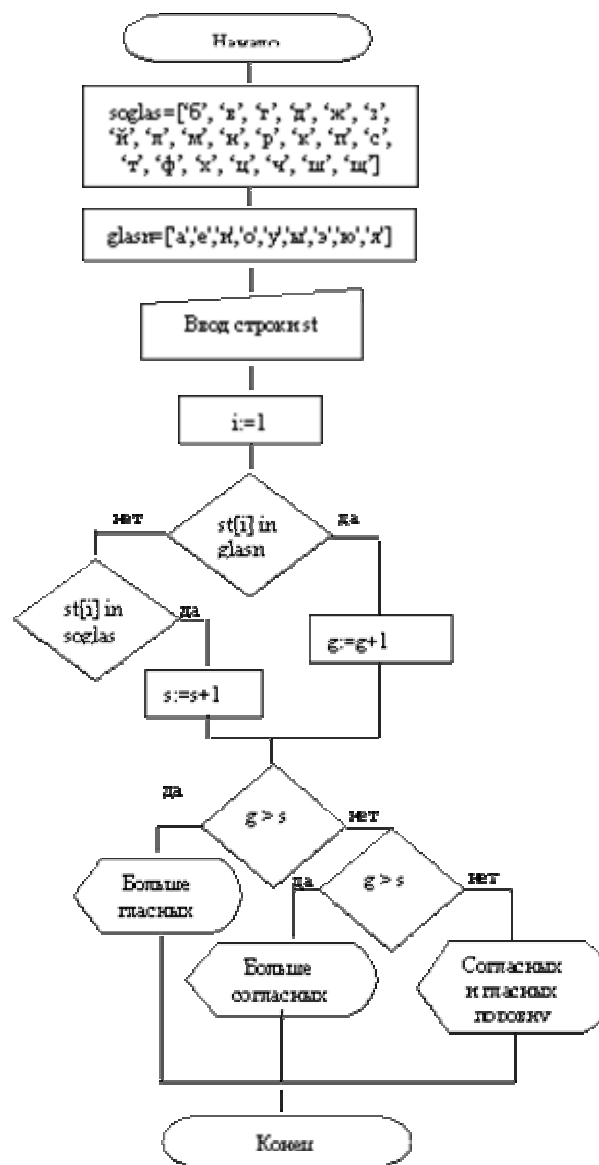


рисунок 4- Блок-схема целиком

2. Переведем алгоритм на язык Паскаль

```
program example1;
const  glasn=['a','e','и','o','y','ы','э','ю','я'];
       soglas=['б','в','г','д','ж','з','й','л','м','н','р','к','п','с','т','ф','х','ц','ч','ш','щ'];
var    st: string;
       g,s,i:integer;
begin
  write('Введите строку> '); readln(st);
  g:=0; s:=0;
  for i:= 1 to length(st) do
    if st[i] in glasn then inc(g) else if st[i] in soglas then inc(s);
    if g> s then writeln('Гласных больше')
    else if g< s then writeln('Согласных больше')
    else writeln('Согласных и гласных букв поровну');
  readln;
end.
```

Пример 2. Составить программу, которая вырабатывает и выводит на экран дисплея наборы случайных чисел для игры в "Спортлото 5 из 36".

Для заполнения каждой карточки спортлото необходимо получить набор из пяти псевдослучайных чисел. К этим числам предъявляются два требования:

- числа должны находиться в диапазоне 1..36;
- числа не должны повторяться.

Program Lotto;

```
  Var  nb, k: Set of 1..36;  kol, l, i, n: Integer;
begin
  Randomize;
  WriteLn('ВВЕДИ kol');  ReadLn(kol);
  nb:=[1..36];
  for i:=1 to kol do
    begin  k:=[];
  for l:=1 to 5 do
    begin
      repeat
n:=Random(36)
      until (n in nb) and not (n in k);
k:=k+[n];  Write(n:4)
    end;
  WriteLn
    end
end.
```

3.5 Вопросы для защиты лабораторной работы №3

- 1 Что такое множество? Каким требованиям должны удовлетворять все элементы множества? Преимущества использования типа множество.
- 2 Что такое базовый тип множества? Как он задается?
- 3 Какое множество называется пустым, как оно обозначается?
- 4 Как описываются типизированные константы типа множество?
5. Как осуществляется ввод-вывод значений переменных типа множество?
- 6 Какие типы данных используются в качестве базовых при объявлении типа множество?
- 7 Какие операции определены над множествами?
- 8 Как определить новый тип данных с использованием перечисления?
- 9 Какие операции допустимы над переменными, заданными перечислением?
- 10 Чем похожи и чем отличаются множества и массивы?
- 11 Какое значение у выражений: а) $x \text{ in } [x]$; б) $[] \leq [x, y, z]$; в) $[x] \lt \gt [x, x, x]$
- 12 Какие операции допустимы над множествами? Каков тип результатов выражений с применением операций над множествами?
- 13 Какие множества считаются равными, неравными? Имеет ли значение для сравниваемых множеств порядок следования элементов?
- 14 Для чего применяются операции "больше или равно", "меньше или равно"? В чем их отличие?
- 15 Для чего применяется операция in ? Особенности ее применения.
- 16 Что называется объединением множеств?
- 17 Что называется пересечением множеств?
- 18 Что называется разностью множеств?

4 Лабораторная работа №4 Составление и отладка программ с комбинированными типами данных (записи)

Цель работы: Получение навыков в организации ввода/вывода значений комбинированных типов данных. Получение практических навыков программирования задач с использованием записей.

4.1 Ход работы:

- 1 Набрать предложенную программу и разобраться в её работе
- 2 Составить блок-схему программы по заданию №1 своего варианта
- 3 Запустить интегрированную среду программирования Turbo Pascal
- 4 Набрать текст программы №1
- 5 Откомпилировать программу
- 6 Сохранить файл с программой на диске А: под именем lab2.pas
- 7 Распечатать текст программы и результат её работы
- 8 Ответить на вопрос второго задания
- 9 Составить отчёт

4.2 Содержание отчёта:

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задачи №1
- 5 Блок-схема задания №1
- 6 Листинг программы №1
- 7 Постановка задачи №2
- 8 Результат задания №2

4.3 Методические указания к лабораторная работа №4

Компьютеры широко используются в различных информационно-поисковых системах (адресное бюро, телефонная справочная служба и т. д.). В реальных информационных системах приходится обрабатывать и хранить большие объемы данных.

При решении научно-технических и экономических задач обработки совокупности больших объемов значений используются массивы. Но при работе с массивами основное ограничение заключается в том, что каждый элемент массива должен иметь один и тот же тип данных.

******* В работе «Поиск в массиве по критерию» мы использовали процедуру Val. Мы все элементы массива объявляли типа String (строка), а затем, с помощью процедуры Val, значение возраста учащихся из типа String преобразовывали в величину целочисленного типа (Integer).

Решение задачи получилось громоздким, затрачивалось время на преобразование типа.

Иногда для решения задач, в которых возникает необходимость хранить и обрабатывать совокупности данных различного типа, используются отдельные массивы для каждого типа данных, а для установления соответствия между ними вводятся соответствующие индексы.

Итак, реальные данные об объектах часто описываются величинами разных типов. Например, товар на складе описывается следующими величинами: наименование, количество, цена, наличие сертификата качества и т. д. В этом примере наименование – величина типа String, количество – Integer цена – Real, наличие сертификата качества можно описать величиной типа Boolean.

Для записи комбинации объектов разных типов в Паскале применяется комбинированный тип данных запись.

Определение: Запись – это структурированный тип данных, состоящий из фиксированного числа компонентов одного или нескольких типов.

! т. е. В отличие от массива, запись может содержать элементы, принадлежащие разным типам.

! Определение типа записи начинается зарезервированным словом Record (запись) и заканчивается зарезервированным словом end (конец). Между ними

заклучен список компонентов, называемых полями, с указанием идентификаторов полей и типа каждого поля.

Итак: Элементы, образующие запись, называются полями.

Формат записи:

```
    Type <имя типа> = Record
<идентификатор поля>: <тип компонента>;
...
<идентификатор поля>: <тип компонента>;
end;
var <идентификатор, ...>: <имя типа>;
```

Пример:

```
    Type
    Car = record
        Number: Integer;
        Fio: String[40];
        Adress: String[60];
        Salary: Real;
    End;
```

имена полей { тип поля { номер } { фамилия, инициалы } { адрес } { зарплата }

Var M, V: Car;

Доступ к полям записи осуществляется через переменные M и V типа Car.

Замечания:

- 1 После зарезервированного слова Record точка с запятой не ставится.
- 2 Описания отдельных полей (или групп полей, принадлежащих одному типу) завершается точкой с запятой.
- 3 Перед словом End (конец) ‘;’ не ставится.
- 4 Описания полей записи похожи на описание обычных переменных.
- 5 Поля могут принадлежать любым типам – как простым, так и структурированным, как стандартным, так и определенным пользователем.
- 6 Допустимы записи, поля которых представляет собой также записи или массивы.

Идентификатор поля должен быть уникален только в пределах записи, однако во избежании ошибок лучше делать их уникальными в пределах всей программы, т. е. Имена полей в пределах записи не должны повторяться, чтобы они не использовались. Следует стремиться, чтобы имена полей соответствовали смыслу информации в том или ином поле, а информация, в свою очередь, должна соответствовать своему полю.

Организация данных о людях – один из наиболее типичных случаев применения записей. Например, если существует информация о какой-то группе людей, объединенных в производственный коллектив, то эту информацию лучше всего организовывать в виде набора записей, где для каждого члена коллектива предусмотрена своя запись. При этом все записи будут принадлежать одному типу, структуру которого определяет характер данных. Вот как может выглядеть объявление подобного типа в разделе описания программы:

```

Type
  Employee = record
    ID: Word; {идентификатор (личный номер)}

    FirstName, SecondName, SurName: String[20]; {Имя, Отчество, Фамилия}
    Standing: Word; {стаж}
    Salary: Real {зарплата}

  End;

  Var Assistant: Employee;

```

Описанный выше комбинированный тип Employee включает шесть полей, три из них (имя, отчество, фамилия) представляют собой строки по 20 символов каждая. Для остальных полей ID, стаж, зарплата – выбраны типы, подходящие для соответствующей информации.

Содержание одной из записей, принадлежащих типу Employee, выглядит так:

ID	SurName	FirstName	SecondName	Standing	Salary
14285	Петров	Иван	Кузьмич	15	1000

Обращение к значению поля осуществляется с помощью идентификатора переменной (имя записи) и идентификатора поля (имя поля), разделенных точкой. Такая комбинация называется составным именем.

Например: Чтобы получить доступ к полям записи Car, надо записать: M.Number; M.Fio; M.Adress.

Составное имя можно использовать везде.

```

Пример1:  M.Number:= 14285;
           M.Fio:= 'Петров И. К.';
           M.Adress:= 'Ул. Советская 35-5';
           M.Salary:= 1000;

```

Допускается применение оператора присваивания и к записям в целом, если они имеют один и тот же тип: V:=M;

После выполнения этого оператора значения полей записи V станут равны значениям соответствующих полей записи M.

Составные имена можно использовать в операторах ввода-вывода:

```

Write(Assistant.FirstName[1]) (вывод на экран первой буквы имени)
Read(M.Number, M.Fio, M.Adress, M.Salary);
Writ (M.Number: 4, M.Fio: 7, M.Adress: 12, M.Salary: 25);

```

```

Пример 2: Write('Петров');
           Write(Assistant.FirstName[1],'.');

```

```

Write(Assistant.SecondName[1],'.');

```

Результат Петров И. К.

В ряде задач удобно пользоваться массивами из записей. Их можно описать:

```

Type
  Person = record
    Fio: String[20];
    Age: 1..99; {возраст}

```

```
Prof: String[30] {профессия}
End;
Var List: array[1..50] of Person;
```

Оператор With.

Указывать каждый раз составное имя – занятие не из самых увлекательных и к тому же текст программы получается громоздким, особенно если в операторах используются длинные имена. Ситуация еще больше ухудшится, если несколько подобных операторов окажутся сосредоточены в одном месте программы.

Для решения этой проблемы в языке Паскаль предназначен оператор With(в переводе «с»):

```
With <переменная типа запись> do <оператор>
```

Один раз указав переменную типа запись в операторе With, можно работать с именем полей как с обычными переменными, т. е. Без указания перед идентификатором поля имени переменной, определяющей запись.

Пример 3: Присвоить значение полям записи Car с помощью With.

```
With M do begin
  Number:= 14285;
  Fio:= 'Петров И. К.';
  Adress:= 'Ул. Советская 35-5';
  Salary:= 1000;
End;
```

Здесь имя переменной M употребляется всего один раз – после оператора With. Затем все действия над полями выполняются только с указанием имен полей.

Вложенные (иерархические) записи.

Возможно существование записей, отдельные поля которых представляют собой также записи => оператор With тоже может быть вложенным.

Выводы:

- так как поле записи трактуется как переменная, оно может фигурировать в выражениях (обратиться к полю записи – достаточно указать имя записи и имя поля). Набор операций над полями записей соответствует операциям, допустимым для типа, которому принадлежит данное поле.

- значения переменных, принадлежащих комбинированному типу, можно присваивать другим переменным того же типа.

- нельзя использовать с записями операции сравнения (неверно While x=y do ...). Если такая необходимость существует, можно сравнивать по очереди каждое поле одной записи с полями другой.

- совершенно неприменимы к записям (в целом) арифметические и логические операции.

- к записям нельзя применить стандартные процедуры Read и Write. Однако можно организовать считывание и вывод на экран каждого поля записи в отдельности.

Вопросы к самостоятельной работе по теме «Записи».

- 1 Что такое запись?
- 2 Отличие записи от массива?
- 3 Что из себя представляют элементы записи?
- 4 Формат записи?
- 5 Для чего нужен оператор With?

4.4 Варианты заданий

Постановка задачи №1: Составить список учётной группы, включающей 25 человек. Для каждого учащегося указать дату рождения, год поступления в колледж, курс, группу, оценки каждого года обучения. Информацию о каждом учащемся оформить в программы в виде записи. Совокупность записей объединить в массив.

Составить программу, которая обеспечивает ввод полученной информации, распечатку её в виде таблицы, а также распечатку информации, согласно конкретному варианту.

Индивидуальные задания №1

- 1 Распечатать анкетные данные студентов-отличников
- 2 Распечатать анкетные данные студентов, успевающих на 4 и 5
- 3 Распечатать анкетные данные студентов, получивших одну оценку 3 за всё время обучение
- 4 Распечатать анкетные данные студентов, получивших в последнюю сессию оценки 2
- 5 Распечатать анкетные данные студентов, получивших в первую сессию все оценки 5
- 6 Распечатать анкетные данные студентов, получивших за всё время обучения одну оценку 4, а остальные – 5
- 7 Распечатать список студентов, фамилии которых начинаются с буквы А, и их оценки за всё время обучения
- 8 Распечатать список студентов, фамилии которых начинаются с буквы Б, и их даты рождения
- 9 Распечатать оценки в последнюю сессию студентов, фамилии которых начинаются с букв В и Г
- 10 Распечатать фамилии и даты рождения студентов, не получивших ни одной оценки 3 за всё время обучения
- 11 Упорядочить список студентов по среднему баллу и распечатать его

12 Упорядочить список студентов по среднему баллу последней сессии и распечатать его

13 Вычислить средний балл группы и распечатать список студентов, имеющих средний балл выше среднего балла группы

14 Вычислить средний балл группы и распечатать список студентов, имеющих средний балл ниже среднего балла группы

15 Вычислить средний балл группы в последнюю сессию и распечатать список студентов, имеющих средний балл, равный среднему баллу группы

16 Упорядочить список студентов по году рождения и распечатать его

17 Распечатать список студентов, упорядоченный по алфавиту

18 Распечатать список студентов, упорядоченный по месяцу рождения

19 Распечатать список отличников, упорядоченный по году рождения

20 Распечатать список студентов, упорядоченный по дате рождения

Индивидуальные задания №2

1 Опишите запись с именем типа `Karta`, содержащую следующие поля:

- Номер измерения (тип `integer`)
- Значение (тип `Real`)

Переменную, определяющую запись, назовите `Z`

2 Опишите запись с именем типа `Karta`, содержащую следующие поля:

- Номер измерения (тип `integer`)
- Значение (тип `Real`)

Переменную, определяющую запись, назовите `S`

3 Опишите запись с именем типа `Tovar`, содержащую следующую информацию о хранящемся на складе товаре:

- Код товара (тип `integer`)
- Наименование товара (тип `string`)
- Цену (тип `Real`)

Переменную, определяющую запись, назовите `Tov`

4 Опишите запись с именем типа `Graf`, содержащую следующие данные необходимые для построения графика из 40 точек:

- Название графика (тип `string`)
- 40 значений (тип `integer`)

Переменную, определяющую запись, назовите `X`

5 Опишите запись с именем типа `Vaza`, содержащую информацию для школьной базы данных:

- Личный номер ученика (тип `integer`)
- ФИО (тип `string`)
- Год рождения (тип `integer`)
- Адрес (тип `string`)

Переменную, определяющую запись, назовите `Inf`

6 Опишите запись с именем типа `Systema`, содержащую информацию о планетах солнечной системы:

- Номер планеты по удалению от солнца (тип `integer`)
- Название планеты (тип `string`)
- Объём (тип `Real`)
- Диаметр (тип `Real`)
- Удалённость от земли (тип `Real`)

Переменную, определяющую запись, назовите `Planeta`

7 Опишите запись с именем типа `Sport`, содержащую информацию о лучших спортивных достижениях школы по лёгкой атлетике:

- Название вида (тип `string`)
- Фамилия рекордсмена (тип `string`)
- Дата установления рекорда (запись `Dat`, состоящая из полей `Day`, `Month`, `Year`)

- Сообщение о результате (тип `Real`)

Переменную, определяющую запись, назовите `Rec`

8 Опишите запись с именем типа `Geometr`, содержащую информацию об оценках учеников вашего класса по геометрии:

- ФИО (тип `string`)
- Оценки за девять месяцев `max` по 20 оценок в месяц

Переменную, определяющую запись, назовите `Dig`

9 Опишите запись с именем типа `Rasp`, содержащую информацию о движении электропоездов из вашего города:

- Направление (тип `string`)
- Время отправления электропоездов (тип `real`)

Переменную, определяющую запись, назовите `R`

10 Опишите запись с именем типа `Post`, содержащую информацию в почтовой базе данных о подписчиках на газеты и журналы:

- ФИО (тип `string`)
- Адрес (тип `string`)
- 10 строк с названиями газет и журналов

Переменную, определяющую запись, назовите `G`

11 Опишите запись с именем типа `Voln`, содержащую информацию в медицинской базе данных о стационарных больных:

- ФИО (тип `string`)
- Возраст (тип `integer`)
- Адрес (тип `string`)
- Дата поступления (тип `string`)
- Диагноз (тип `string`)
- ФИО лечащего врача (тип `string`)

Переменную, определяющую запись, назовите B

12 Опишите запись с именем типа `Tovar`, содержащую информацию о хранящемся на складе товаре:

- Код товара (тип `integer`)
- Наименование товара (тип `string`)
- Дата поступления (тип `string`)
- Цену (тип `real`)

Переменную, определяющую запись, назовите `Tov`. Без помощи `with` присвойте значение (10, 'туфли женские', 45200.00) полям одной из записей

13 Опишите запись с именем типа `Data`, содержащую информацию о средней температуре в хранилище за 30 дней:

- Номер месяца (тип `integer`)
- Температура (тип `real`)

Переменную, определяющую запись, назовите `Zamer`. Без помощи `with` Присвойте записи начальное значение: месяц «июль» и температура для первого дня 9,5

14 Опишите запись с именем типа `Graf`, содержащую данные, необходимые для построения графика из 40 точек:

- Название графика (тип `string`)
- 40 значений (тип `integer`)

Переменную, определяющую запись, назовите `X`. Без помощи `with` присвойте записи начальное значение: название графика « $Y=f(T)$ », значения первых трёх точек: 5, 7, 9

15 Опишите запись с именем типа `Post`, содержащую информацию в почтовой базе данных о подписчиках на газеты и журналы:

- ФИО (тип `string`)
- Адрес (тип `string`)
- 10 строк с названиями газет и журналов

Переменную, определяющую запись, назовите `G`. Без помощи `with` присвойте записи начальное значение: 'Петров И.В.', 'г.Москва, ул.Горького, 5', 'Московский Комсомолец', 'Спорт'

16 Опишите запись с именем типа `Chempion`, содержащую информацию о чемпионах по разным видам спорта в разное время:

- ФИО (тип `string`)
- Вид спорта (тип `string`)
- Год завоевания медали (тип `word`)
- Призовой фонд (тип `real`)

Переменную, определяющую запись, назовите `W`

17 Опишите запись с именем типа `Rescпт`, содержащую информацию о рецепте приготовления блюда:

- Название блюда (тип `string`)

- Инструменты (тип string)
- Когда подаётся (запись Vremya, состоящая из полей Zavtrak, Obed, Ugin) Переменную, определяющую запись, назовите Eda

Дополнительные задания:

1 Составьте программу, которая описывает массив записей – телефонный справочник одноклассников – и обеспечивает ввод данных, поиск номера телефона по фамилии, подсчёт и вывод списка всех абонентов по критерию «увлечение компьютерными играми». В записи о каждом однокласснике содержатся следующие сведения: фамилия, имя, телефон, хобби.

2 Составьте программу, которая описывает таблицу химических элементов, отображая следующую информацию: название, символическое обозначение, массу атома, заряд атомного ядра, перечень основных химических свойств. Программа должна выполнять вывод данных о химическом элементе по указанному символическому обозначению, находить элемент с самой большой массой, с самым маленьким зарядом ядра.

3 Составьте программу, которая описывает массив записей жильцов дома, отображая в нём следующую информацию о каждом: номер квартиры, фамилия, имя, возраст, для лиц старше 18 лет в зависимости от рода занятий – записать места учебы/работы, трудового стажа, для пенсионеров – год выхода на пенсию. Программа должна обеспечить ввод данных, поиск квартиры с максимальным числом жильцов, поиск самого юного и самого пожилого жильца, поиск студентов, пенсионеров.

4 Опишите, используя структуру записи, вступительные экзамены, на которых абитуриенты сдавали три экзамена, а для поступления надо было набрать 12 баллов. Составьте программу, считывающую с клавиатуры результаты всех вступительных экзаменов и выводящую на экран следующую информацию:

- A.Список абитуриентов, сдавших все три экзамена на 5
- B.Список абитуриентов, потерпевших неудачу на экзаменах
- C.Список абитуриентов, зачисленных в институт

5 Опишите, используя структуру записи, школьный журнал. Предусмотрите в записи поля для хранения информации о фамилии учащегося, предмете, оценке. Составьте программу, считывающую с клавиатуры данные об успеваемости учащихся класса и выводящую на экран сведения об отличниках класса, о средней успеваемости учащихся класса.

6 Опишите, используя структуру записи, школьный класс (фамилия, инициалы, дата рождения, месяц рождения, год рождения). Составьте программу, считывающую с клавиатуры данные об учащихся и выводящую на экран данные о днях рождения учащихся по месяцам, например:

Январь 12 Петров И.В.

4.5 Пример программы:

Программа Zap вводит информацию о пяти студентах: фамилию, год рождения, год поступления в техникум и оценки последней сессии. Результатом работы программы является список отличников. В программе показаны возможности организации работы с записями с использованием оператора присвоения.

```
Program Zap;
Uses crt;
Type Student = record
    FIO:string[10];
    GodRo:1965..1975;
    GodPostTex:1985..1988;
    Ocenki : record
        Alg, Istor, Inform:1..5;
    end;
end;
Var Gruppa: array[1..5]of Student;
    S1:Student;  i,j:Integer;

Begin
    clrscr;
With S1.Ocenki do
    begin
for i:=1 to 5 do
    begin
Write('=> ');
    ReadLn(FIO);
    ReadLn(GodRo,GodPostTex,Alg,Istor,Inform);
    Gruppa[i]:=S1;
    end;
    end;
Clrscr;
WriteLn('Список отличников:');
    for i:=1 to 5 do
        with Gruppa[i].Ocenki do
if (Alg=5) and (Istor=5) and (Inform=5) then
    WriteLn(FIO,'Год рождения: ',GodRo,'Год поступления: ',GodPostTex:4);
end.
```

4.6 Вопросы к защите лабораторной работы №4:

- 1 Что такое запись?
- 2 Отличие записей от массивов?
- 3 Почему запись называют комбинированным типом данных?
- 4 Как определяется тип записи? Формат записи?
- 5 Что называется полем записи?
- 6 Как осуществляется обращение к записи поля?
- 7 Какие требования предъявляются к идентификаторам поля в записи?
- 8 Чем определяется объём памяти, требуемый для размещения записи?
- 9 Какие действия можно производить с полями и записями в целом?
- 10 Что такое составное имя поля записи? Из каких частей оно состоит и как записывается?
- 11 Как применяется оператор With?
- 12 Что такое вложение записей? Какой максимально допустимый уровень вложения?

5 Лабораторная работа №5 Назначение, открытие и закрытие файлов»

Цель работы : Выработать навыки работы с текстовыми файлами в языке Т.Р. Научиться создавать и открывать файлы, считывать информацию из файлов, записывать информацию в файл, работать с элементами файла, закрывать и удалять файлы.

5.1 Ход работы:

- 1 Составить полную цепочку команд для создания простого текстового файла.
- 2 Вычислить факториал чисел 1, 2, 3 ... 10 и вывести от введённого значения (PRN или CON).
- 3 Используя строковую процедуру Val(s,t,code) для преобразования текстовой информации составить программу получения звуковых сигналов.
- 4 Вычислить сумму элементов файла myfile.dat .
- 5 Составить программу проверки существования файла на диске.
- 6 Распечатать программы и результаты работы составленных программ.
- 7 Составить отчёт.

5.2 Содержание отчёта :

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задач
- 5 Листинг программ

5.3. Методические указания к лабораторной работе №5

Типы данных, с которыми мы до сих пор имели дело, предназначены для манипулирования информацией, содержащейся в оперативной памяти компьютера. Однако, как известно, оперативная память способна хранить данные только временно – пока компьютер включён. Представим, что в течении дня производился какой-то сложный расчёт, к результатам которого придётся обращаться в последствии – иногда много дней спустя. Как сохранить эти результаты, чтобы они были в целости и сохранности, постоянно под рукой и при этом иметь возможность выключать компьютер, когда необходимо? Очевидно, что для эффективной работы нужен какой-то способ продолжительного хранения информации, чтобы ею можно было воспользоваться после выключения и включения компьютера снова. И такой способ существует. Для долговременного хранения информация из оперативной памяти переносится в файлы.

Понятие физического и логического файла в Turbo Pascal.

Файл – это именованная область внешней памяти, содержащая какую – либо информацию, это физический файл.

Файл – структура данных, используемая в программировании, это логический файл. Он существует только в логическом представлении, при написании программы.

Целесообразность применения файла.

1) Вывод больших объёмов данных, подлежащих обработке. Очень удобно создать отдельный файл данных, который подготавливается заранее и применяется неоднократно.

2) Файл данных может быть подготовлен другой программой, затем связывать между собой различные задачи, а так же не связывать программу с внешними средствами.

3) Программа использующая данные из файлов не требует присутствия пользователя в момент фактического исполнения.

Классификация файлов:

1 по типу (логической структуре)

а) Var <имя файловой переменной>:Text ; - текстовый файл

б) Var <имя файловой переменной>:file of <тип компонентов>; - типизированный файл

в) Var <имя файловой переменной>:file – не типизированный файл

где : - f – файловая переменная

- file и of – зарезервированные слова, имеющие смысл файл и из.

- Text- это не зарезервированное слово , а идентификатор стандартного типа данных, как Real или Integer.

На рисунке 5 представлена схема классификации файлов.

2 по методу доступа к элементам файла

а) Файл последовательного доступа – доступ к элементам обеспечивается в такой же последовательности, в какой они записывались.

б) Файл прямого доступа – доступ к элементу осуществляется по адресу элемента.



Рисунок 5- Классификация файлов

Допустимость применения методов доступа к каждой разновидности файлов по типу показано на рисунке 6

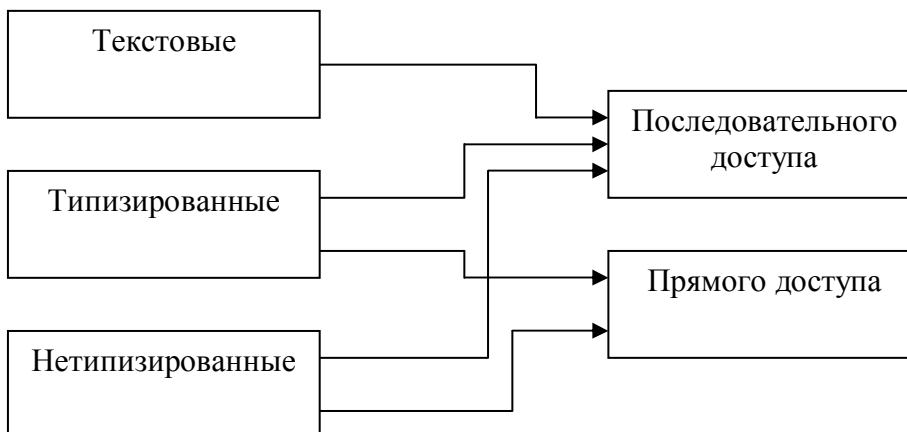


Рисунок 6- Применение методов доступа к файлам

В программах файлы представляются файловыми переменными определенного типа. В качестве внешнего носителя информации могут служить разного рода диски (жесткие, гибкие, магнитооптические, компакт-диски, zip- и jaz-диски и т.д.), а также магнитные ленты. Также, в качестве этого «условного» внешнего носителя могут рассматриваться виртуальные диски, которые можно создавать (с помощью команды VDISK) в оперативной памяти компьютеров, работающих под управлением операционной системы MSDOS.

Кроме “долговременности” у файлов имеется ещё одна отличительная особенность: их неограниченный объём (или длина). Если для каждого из прочих структурированных типов всегда точно определено, сколько элементов содержит, то или иное значение, то сколько элементов должно быть в файле, при объявлении файлового типа не указывается. Максимальная длина файла ограничивается только свободным местом на диске, и это является основным отличием файлов от массивов.

Целесообразность применения файлов (Для каких целей используются файлы)

1 Ввод больших объёмов данных, подлежащих обработке. Очень удобно создать отдельный файл данных, который подготавливается заранее и применяется неоднократно.

2 Файл данных может быть подготовлен другой программой, затем связывать между собой разные задачи, а также связывать программу с внешними средстами.

3 Программа использующая данные из файлов, не требует присутствия пользователя на момент фактического исполнения.

Каждый файл на диске имеет обозначение, которое состоит из двух частей: имени и расширения. В имени файла может быть от 1 до 8 символов. Расширение начинается с точки, за которой следует от 1 до 3 символов

Пример: autotext.exe
Com.pas

Имя файла – это любое выражение строкового типа, которое строится по правилам определения имени в MSDOS.

Любой файл имеет три характерных особенности:

1 У него есть имя, что даёт возможность программе работать сразу с несколькими файлами.

2 Файл содержит компоненты одного типа. Таким компонентом может быть любой тип Турбопаскаля, кроме файлового. Например допускается файл записи или файл строк, но нельзя создать “файл файлов”.

3 Длина вновь создаваемого файла никак не оговаривается при его объявлении и ограничивается только емкостью устройств внешней памяти.

Всё это позволяет считать файлы одной из фундаментальных структур данных в турбопаскале.

Пример объявления файловых переменных:

```
type
  arrays: array of integer;
  Date=record
    Day: 1..31;      {число}
    Month:1..12;    {месяц}
    Year: 1900..2000 {год}
  end;
```

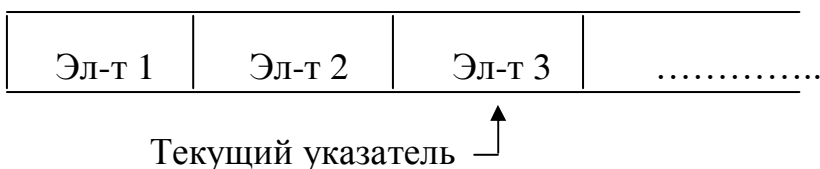

var

F1:file;	}	Пять Ф.П. F1 – нетип. файл F2 – текстовый F3,F4,F5 – тип. Файлы
F2:text;		
F3:file of integer;		
F4:file of arrays;		
F5:file of date;		

Причём если среди тип. Файлов элементы F3 относятся к стандартному типу (Integer), то элементы файлов F4 и F5 – к типам, объявленным пользователем (тип Arrays - массив целых чисел, тип Date - запись).

В файле в каждый момент может быть доступен только один элемент. Доступ к элементам файла обычно осуществляется последовательно, путём их поочерёдного перебора. Иными словами, для того чтобы “добраться” до последнего элемента, прежде придётся обработать (считать или записать) все предыдущие. Для типизированных и не типизированных файлов возможен переход к определённой элементу.

Файл можно представить как потенциально бесконечный список значений одного и того же (базового) типа. Все элементы файла считаются пронумерованными, начальный элемент имеет нулевой номер.



В любой момент времени программе доступен только один элемент файла, на который ссылается текущий указатель.

Как правило, все действия с файлом (чтение из файла, запись в файл) производятся поэлементно, причём в этих действиях участвует тот элемент файла, который обозначает текущим показателем. В результате совершения операции текущий указатель может перемещаться, настраиваясь на тот или иной элемент файла.

Назначение, открытие и закрытие файлов.

Для работы с физическим файлом необходимо связать его с файловой переменной, с помощью которой будет осуществляться доступ к этому физическому файлу.

Такое связывание осуществляется с помощью процедуры

Assign (var f;filename:string);

F – имя файловой переменной любого типа

filename – полное имя файла

1 случай Assign (f,'Lab.dat');

2 случай file_name:='A:\Prak\Lab.dat'; - полный путь.

Assign (f,File_name);

Процедура Assign всегда предшествует другим процедурам работы с файлами т.к. ставит в соответствие конкретному файлу файловую переменную, к которой в последствии будут обращаться все другие файловые процедуры. Недопустимо использовать Assign для уже открытого файла. Прежде чем использовать эту процедуру файл необходимо закрыть.

Открытие файла.

Для работы с файлом прежде всего необходимо его открыть. Открыть файл в Т.Р. можно для чтения, для записи или одновременно. До открытия файл должен быть связан с Ф.П. При открытии файла ищется уже существующий файл на диске либо создаётся новый (если файл открывается для записи).

Для открытия файлов предназначены процедуры Reset, Rewrite и Append. Первые две предназначены для открытия файла любого вида. Третья для открытия текстового файла.

Процедура Reset (var F:file ; RecSize:word);

Если эта процедура применяется к текстовому файлу, то он будет открыт для чтения при последовательном доступе к элементам файла. Если применяется к тип. или нетип. файлам, то такой файл будет открыт как для чтения, так и для записи. При открытии указатель текущей позиции файла устанавливается в его начало. Эта процедура применяется для уже существующих файлов. Если она применяется для файлов отсутствующих на диске будет ошибка при выполнении программы. F – файловая переменная. RecSize – необяз. параметр (только для нетипизированных. файлов) устанавливает длину записи для функций обмена с файлом.

Стандартное открытие файла. Assign (F,' '); Reset(F);

При назначении файловой переменной пустой строки происходит автоматическая ссылка на стандартный файл ввода, что в модуле System соответствует устройству CON. С открытием такого файла появляется возможность ввода данных с клавиатуры.

Процедура Rewrite (var F:file; RecSize:word);

Если эта процедура применяется к текстовому файлу, то она открывает его только для записи. Если применяем для тип. и нетип. файлов, то они открываются, как для чтения так и для записи.

Использование этой процедуры требует особого внимания. При попытке создать и открыть новый файл с именем уже существующим на диске набора данных эта процедура удалит этот набор и создаст новый пустой файл с тем же именем.

Отличие между процедурами Reset и Rewrite.

1 При применении этих процедур для текстовых файлов, процедура Reset открывает его только для чтения, а Rewrite только для записи.

2 При применении этих процедур для тип. и нетип. файлов:

3 Если Reset применить к несуществующему файлу будет ошибка.

Rewrite в данном случае создаст новый файл.

4 Если Reset применить к существующему файлу. Файл будет открыт как для чтения так и для записи

Процедура Close (var f:file); - закрывает файл.

Пример 1: Полная цепочка команд для создания простого текстового файла с именем Work.txt

.....

```
Var F:text;
```

```
begin
```

```
Assign (F,'Work.txt');
```

```
Rewrite (F);
```

```
Write (F,'Простой текстовый файл');
```

```
Close(F);
```

```
end.
```

Процедура Rename (var F; NewName: string); - переименовывает неоткрытый внешний файла любого типа.

где F – файловая переменная

NewName – новое имя файла.

Файлу, ассоциированному с переменной F, присваивается новое имя, которое указывается при вызове процедуры как, например NewName. С этого момента с Ф.П. ассоциируется файл с новым именем.

Процедура Erase (var F); - удаляет неоткрытый внешний файл любого типа.

Пример 2: Удаление или переименование файла.

```
Var F:file; ch:char; St:string;
```

```
begin
```

```
Write ('Введите имя файла: ');
```

```
ReadLn (St);
```

```
Assign (F,St); { Назначаем имя файловой переменной }
```

```
Write ('Удалить файл (У), Переименовать (П), выход (В)');
```

```
ReadLn(ch);
```

```
Case ch of
```

```
'У', 'у' : Erase (F); { Удаляем файл }
```

```
'П', 'п' : begin
```

```
Write('Введите новое имя файла: ');
```

```
ReadLn(St);
```

```
Rename (F,St); { переименовываем файл }
```

```
end;
```

```
'В', 'в' : Halt (1);
```

```
end;
```

```
end.
```

В предыдущем примере выбор действий целиком зависел от того, что будет введено с клавиатуры. Этот вариант программы не позволяет обрабатывать ошибочные ситуации в случае, если файл с именем st не существует на диске.

Для того чтобы файловые операции выполнялись чётко и без ошибок используют специальную функцию.

Функция IOResult : integer; (без параметров)

Используется эта функция в программах в том случае, если на время выполнения файловых операций отключена стандартная проверка операций ввода – вывода.

Для этих целей используется либо специальная опция в интегрированной системе, либо директива компилятора {\$I}, которая может задаваться внутри текста программы.

Пример 3: Программа для проверки существования файла на диске.

```
Var F:file;
    St: string;
begin
Write ('Введите имя файла: ');
ReadLn (St);
Assign (F, St); {назначаем имя файловой переменной}
{$I-}           {отключаем стандартную обработку ошибок}
Reset (F);     {открываем файл}
{$I+}         {включаем стандартную обработку ошибок}
if IOResult=0 then begin
    WriteLn ('Файл существует и нормально открыт');
    Close (F);
end
else WriteLn ('Файла с именем '+St+' на диске нет.');
```

end.

После корректного выполнения операции ввода-вывода функция Ioresult возвращает значение, равное нулю, в остальных функция возвращает соответствующий код ошибки.

Устройства с символическими именами.

В качестве параметра процедуры Assign вместо имени файла файла с диска можно использовать зарезервированное (за некоторыми устройствами) имена. В MSDOS основным аппаратным средствам П.К. присвоены символические имена в MSDOS являются зарезервированными, т.е. не удастся создать файл с именем, соответствующим зарезервированному за одним из устройств.

Символические имена	Устройство
1. CON	Консоль.
2. PRN, LPT1, LPT2, LPT3	Принтеры.
3. AUX, COM1, COM2	Коммуникационные каналы
4. NUL	Фиктивное устройство

1 CON - Под консолью понимается одновременно и клавиатура, и монитор П.К. Логический имя CON можно указывать для процедуры Assign вместо имени файла, когда информация требуется ввести с клавиатуры или

2 вывести на экран. Система сама определяет с клавиатурой или с монитором ей следует иметь дело в том или ином случае. Она ориентируется по выполняемой операции. Вывести данные можно только на экран, а ввести только с клавиатуры.

3 PRN – Указывают в качестве параметра, когда требуется распечатать информацию на принтере. Если к одному компьютеру подключены несколько принтеров, то в MSDOS предусмотрено ещё несколько символических имён LPT1, LPT2, LPT3, причём имена PRN и LPT1 – взаимно заменяемые (или синонимы).

4 AUX – коммуникационный канал предназначен для обмена информацией с другими компьютерами либо устройствами, подходящими для подключения к этому каналу. Имя AUX можно указать в качестве параметра, когда требуется передать информацию на другой компьютер (или устройство) либо принять извне. Обычно у П.К. более одного коммуникационного канала поэтому в MSDOS определены ещё два символических имени COM1 и COM2, причём COM1 и AUX – синонимы.

5 NUL Предположим, идёт процесс отладки некоторой программы, оперирующей объёмными файлами. В этой программе среди прочих имеются операции вывода данных. Однако место на диске ограничено, а осуществлять вывод на экран навсегда удобно, и главное, в данный момент интересует не сама информация, а процесс её вывода. В этой ситуации очень удобно в качестве устройства вывода указать фиктивное устройство, чтобы ненужная информация в процессе отладки отправлялась “в никуда”

Пример 4: Вычислить факториал чисел 1, 2, 3 ... 10 и вывести их на принтер или экран.

```

.....
Uses CRT;
Var j:word; p:Real; Name:String[3];
      F:Text;
begin
  WriteLn ('Введите значение Prn (Принтер) или CON (дисплей)');
  ReadLn (Name);
  Assign (f, Name);
  Rewrite (f);
  p:=1;

```

```

for j:=1 to 10 do begin p:=p*j; write(f,p); end;
                WriteLn;
Close (f);
                ReadLn;
end.

```

Примечание: Если по ошибке будет введено непредусмотренное имя (например CON с цифрой 0), оно будет воспринято как имя файла. Если диск незащищён от записи, то на нём будет создан текстовый файл CON содержащий 10 значений

Ориентирование в файле.

Функция Eof (var f) :Boolean; - для тип. и нетип. файлов.

Данная функция возвращает значение TRUE, если указатель текущей позиции находится за последним элементом файла, либо если файл пуст. В противном случае функция возвращает значение False.

Пример 5:

```

Program diman;
    Uses CRT;
    Var Date:Text; S:String;
Function Terc (S:String):Integer;
    Var Code, T: Integer;
    Begin
        Val (S,T,Code); Terc:=T;
    end;
begin
    Assign (Dete, 'A:\sound.dat');
    Reset (Dete);
    Repeat
        If Eof (Date) then Reset (Date);
        ReadLn (Date, S); Sound (Terc(S)); Delay (100);
        NoSound; Delay (25);
    Until keypressed;
End.

```

Процедура Seek (var f; n: longInt); Эта процедура позиционирует указатель на определённый элемент в файле. Файл должен быть открыт.

F – файловая переменная любого типа, за исключением Text

N – номер элемента в файле.

Пример: Seek (f; 3) – указатель переместить к третьему элементу.

Процедура Truncate (var f); Применяется к открытому файлу и удаляет его содержимое от текущей позиции и до конца.

F – файловая переменная любого типа кроме Text.

Пример 6:

```
Var f:file of char;
    I:integer; j:char;
Procedure Out;
Begin
    Reset (f);
    While not Eof ()f do
    Begin
        Read (f,j); Write (j);
    end;
end;
begin
    Assign (f, 'File.txt');
    Rewrite (f);
For j:='A' to 'F' do write (f,j); {выводим буквы ABCDEF}
    WriteLn('Файл до усечения: ');
Out; {Выводим содержимое файла}
    Reset (f);
for i:=1 to 3 do
    Read (f,j); {Считывание трёх записей}
    Truncate (f); {усечение файла}
    WriteLn;
    WriteLn ('Файл после усечения: ');
Out; {вывод усечённого файла}
    Close (f);
    Erase (f);
end.
```

В этой программе файловая переменная *f* ассоциируется с файлом *File.txt* (имя выбрано произвольно; такого файла в данный момент ещё нет). После этого файл открывается с помощью процедуры *Rewrite*. Мы знаем, что если файла с таким именем нет в каталоге, то эта процедура создаёт новый пустой файл. Вопрос: о каком каталоге идёт речь? Если при связывание ф.п. с файлом каталог не указывается, то по умолчанию используется текущий каталог. А текущим является *BIN*. Итак файл *File.txt* окажется в каталоге *BIN*. В принципе ничего страшного в этом нет т.к. в конце программы процедура *Erase* удаляет созданный файл. Однако на всякий случай лучше указать другой каталог.

Поле того как файл создан и открыт в него вводится 6 элементов. Символы *A*, *B*, *C*, *D*, *E*, *F* и содержимое файла выводиться на экран. Затем организуется считывание трёх элементов файла и проводится усечение файла. Затем содержимое файла вновь выводиться на экран, чтобы продемонстрировать последствия использования процедуры *Truncate*.

Наконец созданный файл закрывается и удаляется.

Результат: Файл до усечения:

ABCDEF

Файл после усечения:

ABC.

Функция FileSize (var f): Longint;

Возвращает значение текущей длине файла (т.е. числу содержащихся в нём элементов). Файл должен быть открыт. Если файл пуст (в нём нет ни одного элемента), функция вернёт значение равное нулю.

f – ф.п. за исключением Text.

Функция FilePos (var f): longint; Возвращает значение, соответствующее положению указателя текущей позиции в файле. Файл должен быть открыт.

Пример 7:

```
.....  
Var    f:file of char;  
        X:longint;  
  
begin  
  Assign (f, 'a.pas');  
  Reset (f);  
    X:= fileSize (f);  
  WriteLn ('размер файла ',x,'байт');  
    Seek (f, x div 2); {указатель в середину файла}  
  WriteLn ('Позиция в середине: ', filePos(f));  
    Close (f);  
end.
```

В этой программе типизированная файловая переменная ассоциируется с файлом a.pas из каталога BIN. Этот файл с текстом программы на Т.Р., который имеется в составе интегрированной среду программирования и содержится в указанном каталоге. Этот каталог является текущим. Т.е. при связывание файла с ф.п. можно даже не указывать путь. После данный файл открывается, определяется его размер и указатель становится в середину файла. Затем файл закрывается.

5.4 Вопросы к защите к лабораторной работе №5

- 1 Что такое файл ?
- 2 Для каких целей применяются файлы ?
- 3 Характерные особенности файлов ?
- 4 Какие устройства компьютера Т.Р. использует наряду с файлами ? Каковы логические имена этих устройств ?
- 5 Затем используется специальная файловая переменная ?
- 6 Какие процедуры служат для открытия файлов ? Их общие черты и отличие?
- 7 Затем применяется процедура Close ?
- 8 Для каких целей используется специальная функция IOResult ?
- 9 Дать определение текстового файла ? Из чего состоит содержимое текстового файла ?
- 10 Назначение процедуры Append Назначение и отличительные особенности процедур Read и Write от Readln и Writeln.

6 Лабораторная работа №6 Процедуры и функции модуля System

Цель работы : Познакомиться с процедурами работы с каталогами. Научиться создавать, переименовывать и удалять каталоги, помещать файл в нужный каталог, переносить файл из одного каталога в другой.

6.1 Ход работы :

- 1 Изучить теоретический материал по теме данной работы.
- 2 Составить программу работы с каталогами в модуле System, используя процедуры ChDir , Mkdir , Rmdir , GetDir .
- 3 В созданный каталог занести простой текстовый файл.
- 4 Переименовать созданный в каталог файл.
- 5 Переименовать сам каталог содержащий данный файл.
- 6 Удалить созданный файл и каталог.
- 7 Запустить интегрированную среду программирования Т.Р. и набрать текст составленной программы.
- 8 Откомпилировать программу.
- 9 Распечатать программу и результат её работы.
- 10 Сохранить программу на диске А.
- 11 Составить отчёт о выполненной работе.

6.2 Содержание отчёта:

- 1 Тема
- 2 Цель работы
- 3 Ход работы
- 4 Постановка задачи
- 5 Листинг программы
- 6 Результаты работы программы

6.3 Методические указания к лабораторной работе №6

Процедуры работы с каталогами

Процедура ChDir (St:String); Изменяет текущий каталог на каталог который указывается в данной процедуре.

St – выражение типа String, содержащее путь к новому каталогу.

Процедура Mkdir (St:String); Создает новый подкаталог. Имя должно быть индивидуальное т.е. не совпадать с именем других существующих каталогов.

St – содержит путь и имя нового каталога.

Процедура Rmdir (St:String); Удаляет пустой каталог.

St – содержит путь и имя удаляемого каталога. Каталог должен быть пустым, должен существовать и не быть текущим.

Процедура GetDir (d:Byte; var S:String);

Выясняет, какой каталог на указанном диске текущий

D – выражение типа Word, определяющее диск. Если D=0 => подразумевается устройство по умолчанию, если D=1 – диск A, D=2 – диск B, D=3 – диск C и т.д.

Пример 1: Процедуры работы с каталогами.

```
.....  
Uses CRT;  
Var S:String;  
begin  
    ClrScr;  
    ChDir ('B:\');           {установка корневого каталога}  
    GetDir (0,S);  
    WriteLn ('Текущий диск и каталог: ', s);  
    Mkdir ('Grup1p');       {Создаём подкаталог Grup1p}  
    ReadLn;  
    WriteLn ('Перейти в подкаталог Grup1p1');  
    ChDir ('\');            {Установим текущий каталог диска в}  
    WriteLn ('Удалить подкаталог Grup1p');  
    Rmdir ('Grup1p');  
    GetDir (0, S);  
    WriteLn ('Текущий диск и каталог: ',S);  
end.
```

6.4 Вопросы для защиты лабораторной работе №6

1 Какие процедуры применяются для работы с каталогом? Какое действие выполняет каждая из процедур?

2 Какие процедуры применяются для переименования и удаления файлов? Каковы особенности их использования?

3 Можно ли одно и то же время считать из файла и записывать в него? Почему?

4 Если не переписывать файл заново, то значения таких элементов можно изменять : только первого, только последнего, каких угодно, никаких? Какие элементы можно удалять из файла?

5 В какое место файла можно добавлять новые элементы : в начало, в конец, в середину, куда угодно, никуда? Ответ поясните.

6 Можно ли, считав из файла пятый элемент, затем сразу же считать второй элемент?

7 Лабораторная работа №7 Работа с временными и внешними файлами

Цель работы : Изучение файлового типа данных в языке турбопascal. (типизированные и не типизированные файлы). Выработать навыки работы с файловым типом данных. Научиться решать задачи с использованием файлов.

7.1 Ход работы :

- 1 Изучить теоретический материал по теме данной работы.
- 2 Составить программы по заданиям.
- 3 Запустить интегрированную среду программирования Т.Р. и набрать тексты составленных программ.
- 4 Откомпилировать программы.
- 5 Сохранить программы на диске А.
- 6 Распечатать тексты программы и результаты их работ.
- 7 Составить отчёт о выполненной работе.

7.2 Содержание отчёта :

- 1 Тема.
- 2 Цель работы.
- 3 Ход работы.
- 4 Постановка задач.
- 5 Листинг программ.
- 6 Результаты работы программ.

7.3 Варианты заданий к лабораторной работе №7

Постановка задачи задание №1 Работа с временными файлами

Подготовить анкетные данные своих ближайших родственников. Оформить сведения о каждом родственнике в виде записи, в которой определены следующие поля :

- 1 Фамилия, имя , отчество.
- 2 Степень родства (мать, сестра и т.д.)
- 3 Город проживания.
- 4 Год рождения.
- 5 Признак – работает или нет.

Организовать ввод данных с терминала и записать их во временный файл. Вычислить в программе средний возраст родственников. Организовать чтение созданного файла и вывод на терминал информации о тех родственниках, у которых возраст меньше среднего.

Постановка задачи задание №2 Работа с внешними файлами

Подготовить данные об абитуриентах, поступающих в техникум. Информацию о каждом абитуриенте оформить в виде записи, содержащей следующие поля :

- 1 Фамилия, имя , отчество.
- 2 Год рождения.
- 3 Год окончания школы.
- 4 Оценки в аттестате.
- 5 Признак – нуждается ли в общежитии.

6 Оценки вступительных экзаменов.

Используя внешний файл, содержащий исходные данные, добавить N записей и распечатать :

- 1 Список абитуриентов, имеющих в аттестате оценки только 5; N = 2.
- 2 Список абитуриентов, имеющих в аттестате одну оценку 4, а остальные оценки 5 ; N=3.
- 3 Список абитуриентов, имеющих средний балл больше 4,5 ; N=4.
- 4 Список абитуриентов, имеющих средний балл меньше 4; N=1.
- 5 Список абитуриентов, не нуждающихся в общежитии; N=2.
- 6 Список абитуриентов, нуждающихся в общежитии; N=3.
- 7 Список абитуриентов, сдавших вступительные экзамены только на оценки 4 и 5; N =2.
- 8 Список абитуриентов, сдавших вступительные экзамены на оценки 4; N=2.
- 9 Список абитуриентов, сдавших экзамены с двумя оценками 4 и остальными оценками 5; N=3.
- 10 Список абитуриентов, получивших на вступительных экзаменах одну оценку 3; N=4.

Дополнительные задания.

1 Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл, содержащий сведения о багаже каждого пассажира представляет собой запись в которой определены следующие поля :

Ф.И.О.,Количество вещей, Вес (в килограммах)

а) найти число пассажиров, имеющих более двух вещей и число пассажиров, и число пассажиров количество вещей которых превосходит среднее число вещей.

б)определить, имеется ли два пассажира, баланс которых совпадает по числу вещей и различаются по весу.

в)выявить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей и по весу.

г)выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом менее 30 кг.

2 Сведения о студенте состоят из его имени, фамилии, названия группы (номер и буква), года обучения. Дан файл F, содержащий сведения о студентах коллежа :

а) выяснить, имеются ли в колледже однофамильцы

б) выяснить, имеются ли однофамильцы в группе.

в) выяснить, имеются ли в группе студенты с одинаковыми именами.

г) выяснить, в каких группах насчитывается более n студентов

д) выяснить, в какая группа самая многочисленная.

3 Сведения о студенте состоят из его фамилии, года рождения, года поступления в колледж и количества баллов набранных на вступительных экзаменах. Дан файл f, содержащий сведения о студентах колледжа.

а) выяснить, кто из студентов самый старший

б) выяснить, кто родился раньше 1986 года

в) выяснить, кто из студентов набрал большее количество баллов на вступительных экзаменах

г) выяснить средний балл всех студентов

д) Найти среди всех студентов, тех у кого количество баллов набранных на вступительных экзаменах выше(ниже) среднего количества баллов всех студентов.

4 Сведения об автомобиле состоят из его марки, номера и фамилии владельца.

Дан файл содержащий сведения о нескольких автомобилях. Найти :

а) фамилии владельцев и номера автомобилей данной марки.

б) количество автомобилей каждой марки

в) автомобилей какой марки больше

5 Дан файл f, содержащий различные даты. Каждая дата – это число, месяц и год. Найти :

а) год с наименьшим (наибольшим) номером

б) все весенние даты

в) самую раннюю (позднюю) дату

6 Дан файл f, содержащий сведения о книгах. Сведения о каждой из книг – это фамилия автора, название и год издания.

а) Найти названия книг данного автора, изданных с 1960г.

б) Определить имеется ли книга с названием «Информатика».

в) Определить имеется ли книги изданные в один год и количество таких книг.

7 Дан файл f, содержащий сведения о кубках : размер каждого кубка (длина ребра в сантиметрах), его цвет (красный, зелёный, жёлтый или синий) и материал (деревянный, металлический, картонный). Найти:

а) количество кубиков каждого из перечисленных цветов.

б) количество деревянных кубиков с ребром 3 см.

в) количество металлических кубиков с ребром большим 5 см.

8 Дан файл f, содержащий сведения о веществах : указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор). Найти :

а) удельные веса и названия всех полупроводников.

б) вещество с максимальным (минимальным) удельным весом.

в) удельные веса и названия всех изоляторов.

7.4 Вопросы к защите лабораторной работы №7:

1 Назначение функций Eolr, Eof, SeekEolr, SeekEof.

2 Какие файлы относятся к типизированным ? Как они определяются ?

3 Как представлена информация в типизированных файлах ?

4 Назначение процедур Sizeof, Seek, Truncate, FilePos, FileSize .

5 Какие файлы называют нетипизированными? Как они определяются и каковы их особенности ?

6 Назначение процедур BlockRead и BlockWrite. Их особенность?

8 Вопросы к экзамену по предмету основы алгоритмизации и программирования

- 1 Определение алгоритма, свойства и виды алгоритмов. Примеры алгоритмов.
- 2 Алгоритм. Способы описания алгоритмов.
- 3 Базовые структуры алгоритмов.
- 4 Структура программы в Turbo Pascal.
- 5 Арифметика и алфавит Turbo Pascal. Типы данных. Преобразование типов и действия над ними.
- 6 Организация программ линейной структуры. Операторы ввода – вывода.
- 7 Условные конструкции. Оператор выбора.
- 8 Циклические конструкции (Оператор цикла с предусловием).
- 9 Циклические конструкции (Оператор цикла с постусловием).
- 10 Циклические конструкции (Оператор цикла по параметру).
- 11 Процедуры и функции в Turbo Pascal. Определения. Особенности.
- 12 Процедуры и функции пользователя. Форматы описаний. Формальные и фактические параметры.
- 13 Оформление подпрограммы в виде процедуры.
- 14 Оформление подпрограммы в виде функции.
- 15 Структурированные типы данных. Определение «строки». Формат описания данных строкового типа.
- 16 Операции над строковыми данными.
- 17 Подпрограммы для работы со строками (Pos, Concat, Val, Delete).
- 18 Подпрограммы для работы со строками (Length, Copy, Insert, Str).
- 19 Данные типа «массив». Определение. Формат описания. Примеры.
- 20 Действия над массивами. Двумерные массивы. Особенность расположения в памяти. Пример описания.
- 21 Понятие сортировки. Виды сортировок.
- 22 Множественный тип данных. Операции над множествами.
- 23 Данные типа «запись». Определение. Формат описания. Пример.
- 24 Данные типа «запись». Обращение к полям. Два способа обращения.
- 25 Понятие физического и логического файла. Необходимость применения файлового типа.
- 26 Классификация файлов.
- 27 Открытие и закрытие файлов.
- 28 Назначение функций IOResult, Eof, Eoln.
- 29 Процедуры работы с каталогами.
- 30 Текстовые файлы.
- 31 Типизированные и Нетипизированные файлы.
- 32 Модуль CRT. Основные процедуры и функции.
- 33 Модуль GRAPH. Запуск графической системы.
- 34 Модуль GRAPH. Работа с текстом. Цвет и стиль текста.
- 35 Модуль GRAPH.. Управление экраном монитора. Окно в графическом режиме.

Список использованных источников

- 1 **Семагин И.Г.** Основы программирования [Текст] /И.Г.Семагин, А.П.Шестаков. –М.: Академия, 2004. –385 с.
- 2 **Попов В.Б.** TURBO PASCAL для школьников [Текст] /В.Б.Попов –М.: Финансы и статистика, 2007. -518 с.
- 3 **Житкова О.А.**Справочные материалы по программированию на языке Паскаль [Текст] /О.А.Житкова, Е.К.Кудрявцева–М. Интеллект-центр, 2002. -77 с.
- 4 **Меженный О.А.** TURBO PASCAL. Учись программировать [Текст] /О.А.Меженный. –М.: Диалектика, 2001.-88 с.
- 5 **Фаронов В.В.** TURBO PASCAL 7.0 [Текст] /В.В.Фаронов. –М.: Нолидж, 2007. -312 с.
- 6 **Марченко А.И.** Программирование в среде TurboPascal 7.0 [Текст] /А.И. Марченко. –М.: Бинوم Универсал, 2005. – 485с.
- 7 **Культин Н.** Delphi Программирование на Object Pascal [Текст] /Н.Культин. –М.: ВHV–Санкт-Петербург, 2005. –297с.
- 8 **Зубов В.С.** Программирование на языке TP [Текст] /В.С.Зубов. –М.: ТТО «Филинь», 2000. –301с.
- 9 **Васюокова Н.Д.** Практикум по основам программирования язык Паскаль [Текст] /Н.Д.Васюокова, В.В.Тюляева. –М.: Высшая школа, 2000. –с.
- 10 **Бондарев В.М.** Основы программирования [Текст] /В.М.Бондарев, В.И.Рублинецкий, Е.Г.Качко. –М.: ТТО «Филинь», 2004. –363с.