

Оглавление

Раздел 1. Паскаль.....	4
Лабораторная работа № 1 Линейные программы. Операторы ввода-вывода	4
Лабораторная работа № 2 Ветвления	7
Лабораторная работа № 3 Циклы.....	11
Лабораторная работа № 4 Массивы: одномерные и двумерные	15
Лабораторная работа № 5 Процедуры и функции.....	18
Лабораторная работа № 6 Символьные величины	22
Лабораторная работа № 7 Множества	25
Лабораторная работа № 8 Интегрированные структуры данных: файлы, записи	29
Лабораторная работа № 9 Модуль CRT	34
Лабораторная работа № 10 Графика.....	38
Раздел 2. Дельфи.....	43
Лабораторная работа № 1 Программирование кнопок.....	43
Лабораторная работа № 2 Ветвления и циклы	49
Лабораторная работа № 3 Файлы.....	64
Лабораторная работа № 4 Мультимедиа.....	70
Лабораторная работа № 5 Массивы.....	86
Лабораторная работа № 6 Графика.....	93
Приложение Тесты по курсу Дельфи	97

Раздел 1. Паскаль

Лабораторная работа № 1

Линейные программы. Операторы ввода-вывода

Цель работы:

Приобретение и закрепление практических навыков работы в интегрированной среде разработчика. Научиться использовать стандартные процедуры, функции и операции, определенные для каждого стандартного типа данных, организация ввода-вывода. Разработка и отладка линейных программ.

Темы для предварительной проработки:

- Интегрированная среда разработчика.
- Типы данных языка Паскаль.
- Стандартные процедуры и функции для простых типов данных.
- Структура программы на языке Паскаль.
- Объявление переменных.
- Оператор присваивания.
- Ввод/вывод данных.

Пример.

Написать программу, реализующую сложение двух чисел. Т.е. $C = A + B$.
Реализуем следующие шаги:

1. Напишем название программы;
2. Введем переменные, т.е. A, B, C; и зададим им тип;
3. Выполним сложение и поместим результат в C;

Программа будет выглядеть следующим образом:

```
Program First;  
var  
A, B, C: Integer;  
Begin  
Read(A,B);  
C := A + B;  
Write(C);  
end.
```

Задания для работы.

Номер варианта	Задание
Вариант 1.	Определить объём и площадь боковой поверхности цилиндра с заданными радиусом основания R и высотой H.
Вариант 2.	Найти произведение цифр заданного целого четырехзначного числа.
Вариант 3.	Вычислить длину окружности, площадь круга и объём шара одного и того же заданного радиуса.

Вариант 4.	Вычислить периметр и площадь прямоугольного треугольника по длинам двух его катетов.
Вариант 5.	Возвести положительное число “х” в степень “n”
Вариант 6.	Рассчитать площадь треугольника по формуле Герона.
Вариант 7.	Рассчитать высоты, медианы и биссектрисы треугольника по заданным сторонам.
Вариант 8.	Рассчитать радиус окружности, вписанной в треугольник и описанной вокруг треугольника, зная его стороны.
Вариант 9.	Вычислить периметр и площадь равнобедренной трапеции по заданным сторонам.
Вариант 10.	Треугольник задан координатами вершин. Вычислить биссектрису W_a и радиус вписанной окружности.
Вариант 11.	Рассчитать сумму "S" первых N членов арифметической прогрессии A_1, A_2, \dots, A_N , где A_1, A_N, N – исходные данные.
Вариант 12.	Рассчитать сумму "S" первых N членов ряда геометрической прогрессии B_1, B_2, \dots, B_N , где B_1, Q, N – исходные данные.

Семестровые задания.

Номер варианта	Задание
Вариант 1.	Ромб задан координатами трех вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Вычислить площадь и периметр ромба.
Вариант 2.	Вычислить время падения тела с высоты H с начальной скоростью V.
Вариант 3.	Найти площадь кольца, внутренний радиус которого равен 20, внешний - заданному числу $R > 20$.
Вариант 4.	Известно, что точки с координатами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ являются тремя вершинами некоторого параллелограмма. Найти координаты четвертой вершины.
Вариант 5.	Вычислить длину окружности, площадь круга и объем шара одного радиуса.
Вариант 6.	По длинам двух сторон треугольника и углу между ними найти длину третьей стороны и площадь треугольника.
Вариант 7.	Определить время свободного падения физического тела с высоты H.
Вариант 8.	Вычислить значение функции $f(x)=3,5x+\cos x/\sin 2x+e^x$, где $x=0,1$. Округлить значение $f(x)$ к ближайшему целому.
Вариант 9.	Вычислить площадь поверхности и объем правильной пирамиды, в основании которой квадрат со стороной a и высота

	h.
Вариант 10.	По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали.
Вариант 11.	Ввести 2 числа и вычислить сумму, разность, произведение и частное от деления первого введенного числа на второе.
Вариант 12.	Смешаны V1 литр воды температуры T1 с V2 литрами воды температуры T2. Написать программу вычисления объема и температуры воды.

Контрольные вопросы:

1. Для чего предназначен идентификатор и правила записи идентификатора?
2. Дать определение константам и переменным.
3. Выражения в ТР. Правила записи арифметических выражений в ТР, операции и их приоритеты.
4. Структура программы на Паскале, содержание каждой части.
5. На что указывает тип данных, и какие типы существуют в ТР?
6. Какие простые стандартные типы используются в Паскале?
7. Как объявляется тип переменной в Паскале?
8. Перечислить простые операторы, их назначение, формат, действие.
9. Стандартные процедуры ввода/вывода, их назначение, формат, действие.

Тесты:

- 1: Процедура write устанавливает курсор в...
 - ... конец строки
 - ... начало следующей строки
 - ... конец выведенного текста
 - ... начало текущей строки
- 2: Использование writeln устанавливает курсор в...
 - ... конец строки
 - ... начало следующей строки
 - ... конец выведенного текста
 - ... начало текущей строки
- 3: Каждое предложение программы заканчивается...
 - ... новой строкой
 - ... точкой с запятой
 - ... оператором stop
 - ... запятой
- 4: Если в программу заносятся данные с клавиатуры, то первая строка будет
 - program NAME (output);

- program NAME (input);
- program (input, output) NAME;
- program NAME (input)

5: Если в программу заносятся данные с клавиатуры, а потом выводятся на экран, то первая строка будет

- program NAME (output);
- program NAME (input);
- program (input, output) NAME;
- program NAME (input, output);

Лабораторная работа № 2 Ветвления

Цель работы: Ознакомиться с составлением программ разветвляющейся структуры.

Темы для предварительной проработки:

- Оператор перехода;
- Условный оператор.

Пример.

Решить квадратное уравнение: $ax^2+bx+c=0$.

PROGRAM XXX;

VAR

A,B,C,D:INTEGER;

X,X1:REAL;

BEGIN

WRITE('Задайте A=');

READ(A);

WRITE('Задайте B=');

READ(B);

WRITE('Задайте C=');

READ(C);

D:=SQR(B)-4*A*C;

IF D<0 THEN WRITELN('Решений нет')

ELSE

BEGIN

IF D=0 THEN

BEGIN

X:=-B/(2*A);

WRITELN(X);

END

ELSE

```

BEGIN
  X1:=(-B-SQRT(D))/(2*A);
  X2:=(-B+SQRT(D))/(2*A);
  WRITELN('X1=',X1);
  WRITELN('X2=',X2);
END;
END;
  READLN;
END.

```

Задания для работы.

Номер варианта	Задание
Вариант 1.	Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a , радиус круга r . Вывести на экран название и значение площади большей фигуры.
Вариант 2.	Ввести 2 числа. Если первое введенное число больше второго, то вычислить разность этих чисел, в противном случае - их сумму.
Вариант 3.	Определить вид функции в зависимости от значения аргумента: $y=7$ при $x < -7$, $y=-x$ при $-7 \leq x < 0$, $y=x$ при $0 < x \leq 5$, $y=5$ при $x > 5$. Программа запрашивает ввод значения "x" и выводит на экран вид функции "y".
Вариант 4.	Рассчитать значение функции $Y = x^N$ в зависимости от значения "x" с выбором вида формулы: для случая аргумента $x > 0$ функция $Y = \exp(N * \ln(x))$, при $x < 0$ преобразовать "x" в положительное число $x = \text{abs}(x)$, при $x = 0$ функция $Y = 1$.
Вариант 5.	Идентифицировать треугольник: остроугольный, тупоугольный, прямоугольный, равнобедренный, равносторонний по трем элементам - углам (сторонам). Причем, имя равнобедренного треугольника может быть составным. Контролировать допустимые значения элементов при вводе данных.
Вариант 6.	Если $a \leq b \leq c$, то числа заменить квадратами; если $a > b > c$, то заменить наибольшим из них; в остальных случаях сменить знак чисел.
Вариант 7.	Даны три точки на плоскости. Определить, какая из них ближе к началу координат.
Вариант 8.	Две прямые описываются уравнениями

	$a_1 x + b_1 y + c_1 = 0;$ $a_2 x + b_2 y + c_2 = 0.$ Напечатать координаты точки пересечения этих прямых, либо сообщить, что эти прямые совпадают, не пересекаются или вовсе не существуют.
Вариант 9.	Определите, имеется ли среди заданных целых чисел А, В, С хотя бы одно чётное.
Вариант 10.	Решите линейное уравнение $ax = b.$
Вариант 11.	Определите, лежит ли заданная точка на одной из сторон треугольника, заданного координатами своих вершин.
Вариант 12.	Заданы координаты двух точек. Определите, лежат ли они на одной окружности с центром в начале координат.

Семестровые задания:

Ввести значение x . Вычислить функцию y .

Номер варианта	Задание	Номер варианта	Задание
Вариант 1.	$y = \begin{cases} x^2 - 1, & x < 0, \\ \cos x, & x = 0, \\ \sin(x^3 + 2), & x > 0. \end{cases}$	Вариант 7.	$y = \begin{cases} 5x, & x < -3, \\ \lg x , & x = -3, \\ x^2 + 2x - 1, & x > -3. \end{cases}$
Вариант 2.	$y = \begin{cases} x - 7, & x < 1, \\ \operatorname{tg} x, & x = 1, \\ \ln x, & x > 1. \end{cases}$	Вариант 8.	$y = \begin{cases} x - 1 , & x < 0, \\ \arcsin x, & x = 0, \\ \sqrt{x + 7}, & x > 0. \end{cases}$
Вариант 3.	$y = \begin{cases} \sin x + 3, & x < 0, \\ \operatorname{arctg} x, & 0 \leq x \leq 3, \\ x^3 + 2x^2 - x + 8, & x > 3. \end{cases}$	Вариант 9.	$y = \begin{cases} 2x^2 - 3, & x < 5, \\ \sqrt[3]{x}, & x = 5, \\ \arccos x, & x > 5. \end{cases}$
Вариант 4.	$y = \begin{cases} x, & x \leq -1, \\ \cos(x + \pi), & -1 < x \leq 1, \\ \log_3(x^2 + 3), & x > 1. \end{cases}$	Вариант 10.	$y = \begin{cases} 3x^2 + 5x - 2, & x < 0, \\ x, & 0 \leq x \leq 4, \\ x + 1 + \sqrt{x + 1}, & x > 4. \end{cases}$
Вариант 5.	$y = \begin{cases} x^3 + 4x^2 + x - 3, & x < 0, \\ x^3 + 6, & x = 0, \\ x^3 + 2x + 4, & x > 0. \end{cases}$	Вариант 11.	$y = \begin{cases} 4x^2 - 5x + 1, & x \leq 7, \\ \cos(\sin x), & 7 \leq x \leq 9, \\ 2x^3 + 2x^2 + x + 1, & x > 9. \end{cases}$
Вариант 6.	$y = \begin{cases} x + 1, & x < 0, \\ \operatorname{ctg} x, & 0 \leq x \leq 3, \\ \sin(x - 1), & x > 3. \end{cases}$	Вариант 12.	$y = \begin{cases} x, & x < 0, \\ x^5, & 0 \leq x < 5, \\ \sqrt[4]{x + 2}, & x \geq 5. \end{cases}$

Контрольные вопросы:

1. Как называется программа, в которой выполнение операторов происходит в зависимости от поставленного условия?
2. Что такое оператор перехода и условный оператор?
3. Какой вид записи имеет оператор перехода?
4. Куда передается управление с помощью оператора перехода?

Тесты:

1. Какой из операторов отношений в Pascal неправильный?
 - +
 - <>
 - <
 - >
2. Написать предложение в Pascal, которое сравнивает целую переменную sum с константой 10, и если это так, то печатает строку "Good guess"
 - if sum = 10 then writeln("Good guess");
 - if (sum == 10) then writeln('Good guess');
 - if (sum == 10) writeln('Good guess');
 - if sum = 10 then writeln('Good guess');
3. Написать предложение в Pascal, которое сравнивает символьную переменную letter с символьной переменной chinput, и если они не равны, то печатается значение переменной letter.
 - if letter <> chinput then writeln('letter');
 - if letter < chinput writeln("letter");
 - if letter <> chinput then writeln(letter);
 - if letter > chinput then writeln(letter);
4. Написать предложение в Pascal, которое сравнивает символьную переменную letter с символьной константой 'A', и если она меньше, то печатается строка "Too low", в обратном случае печатается "Too high"
 - if letter > 'Z' then writeln("Too low") else writeln("Too high");
 - if letter < 'A' then writeln("Too low"); else writeln("Too high");
 - if letter <> 'A' then writeln("Too low") else writeln("Too high");
 - if letter < 'A' then writeln("Too low") else writeln("Too high");
5. Какое ключевое слово не относится к разветвляющейся программе?
 - If
 - While
 - Then
 - Else

Лабораторная работа № 3

Циклы

Цель работы: ознакомиться с составлением программ циклической структуры.

Темы для предварительной проработки:

- Цикл с предусловием;
- Цикл с постусловием;
- Цикл с параметром.

Пример: Вычислить $Z=x^3-4x+y$ при изменении аргумента x от 1 до 9. Программа, составленная на языке Pascal:

```
PROGRAM CHIKL;
VAR
  Z:REAL;
  X,Y:INTEGER;
BEGIN
  WRITE('Задайте Y=');
  READ(Y);
  FOR X:=1 TO 9 DO
  BEGIN
    Z:=SQR(X)*X-4*X+Y;
    WRITELN(Z);
  END;
  READLN;
END.
```

Задания для работы.

Номер варианта	Задание
Вариант 1.	Расчет конечной суммы последовательности: $1^2 + 3^2 + 5^2 + \dots + (2*N-1)^2$; $S_N = N*(4*N^2-1)/3$.
Вариант 2.	Расчитать конечные суммы последовательности: $1^3 + 2^3 + 3^3 + \dots + N^3$; $S_N = N^2*(N+1)^2/4$;
Вариант 3.	Расчитать конечные суммы последовательности: $1^3 + 3^3 + 5^3 + \dots + (2*N-1)^3$; $S_N = N^2*(2*N^2-1)$;
Вариант 4.	Расчитать конечные суммы последовательности: $(X^3+1)/\cos(1) + (X^3+2)/\cos(4) +$ $(X^3+3)/\cos(9)+(X^3+N)/\cos(N^2)$;
Вариант 5.	Расчитать конечные суммы последовательности: $(X^2+1)/\sin(1) + (X^2+2)/\sin(2) + (X^2+3)/\sin(3) + (X^2+N)/\sin(N)$;
Вариант 6.	Расчет функции $y = A*\sin(x) - \cos(x)/A$; при изменении аргумента "x" в диапазоне от 0 до π с шагом $\pi/100$ и при из-

	менении параметра "А" в диапазоне от 1 до 3 с шагом 0.5.
Вариант 7.	Рассчитать значения функции $y = \sin(x) / (x+a)^2$ при изменении аргумента "x" в диапазоне от 0 до $\text{Pi}/2$ с шагом $\text{Pi}/80$ и при изменении параметра "А" в диапазоне от 1 до 2 с шагом 0,2.
Вариант 8.	Рассчитать значения функции $y=x^a/a^3$ при изменении аргумента "x" в диапазоне от 1 до 10 с шагом 0,2 и при изменении параметра "А" в диапазоне от 1 до 5 с шагом 1.
Вариант 9.	Рассчитать все целые числа (a, b, c), удовлетворяющие условию: $a^2 + b^2 = c^2$; где $c < 101$.
Вариант 10.	Рассчитать все целые числа (a, b, c, d), удовлетворяющие условию: $a^3 + b^3 + c^3 = d^3$; где $d < 21$.
Вариант 11.	Рассчитать все трехзначные числа, сумма цифр которых делится нацело на тринадцать.
Вариант 12.	Рассчитать номера всех счастливых четырехзначных билетов (сумма первых двух цифр равна сумме последних двух цифр)

Семестровые задания.

Номер варианта	Задание
Вариант 1.	Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 - 1/2^2 + 1/3^2 - 1/4^2 + \dots$; $(-1)^{(N-1)} / N^2$; $S = \text{Pi}^2 / 12$; $N = 1, 2, 3, \dots$
Вариант 2.	Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 - 1/3 + 1/5 - 1/7 + \dots$; $(-1)^N / (2*N+1)$; $S = \text{Pi} / 4$; $N = 0, 1, 2, \dots$
Вариант 3.	Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 / (1*2) + 1 / (2*3) + 1 / (3*4) + \dots$; $1 / (N*(N+1))$; $S = 1$; $N = 1, 2, 3, \dots$
Вариант 4.	Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 / (1*3) + 1 / (3*5) + 1 / (5*7) + \dots$; $1 / ((2*N-1)*(2*N+1))$; $S = 1/2$; $N = 1, 2, 3, \dots$
Вариант 5.	Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 - 1/1! + 1/2! - 1/3! + \dots$; $(-1)^N / N!$; $S = 1/e$; $N = 0, 1, 2, \dots$

Вариант 6.	<p>Определить число членов ряда, необходимых для расчета с заданной погрешностью суммы членов ряда: $1 + 1/1! + 1/2! + 1/3! + \dots; 1/N!; S = e; N = 0, 1, 2, \dots$</p>
Вариант 7.	<p>Рассчитать значение функции $y=f(x)$ при некотором значении "x" с использованием представления функции в виде ряда: $1 - x^2/2! + x^4/4! - \dots; (-1)^N * x^{(2*N)} / (2*N)!; N = 0, 1, \dots$ $y = \cos(x); x$- любое.</p>
Вариант 8.	<p>Рассчитать значение функции $y=f(x)$ при некотором значении "x" с использованием представления функции в виде ряда: $1 + x^1/1! + x^2/2! + \dots; x^N/N!; N = 0, 1, \dots; y = e^x; x$- любое.</p>
Вариант 9.	<p>Рассчитать значение функции $y=f(x)$ при некотором значении "x" с использованием представления функции в виде ряда: $x - x^3/3 + x^5/5 - \dots; \frac{(-1)^N * x^{(2*N+1)}}{2N+1}; N = 0, 1, \dots; y = \arctg(x); X < 1.$</p>
Вариант 10.	<p>Рассчитать значение функции $y=f(x)$ при некотором значении "x" с использованием представления функции в виде ряда: $\frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \dots; \frac{(x-1)^N}{N * x^N}; N = 1, 2, \dots; y = \ln(x); X > 0.5$</p>
Вариант 11.	<p>Рассчитать значение функции $y=f(x)$ при некотором значении "x" с использованием представления функции в виде ряда: $x - x^2/2 + x^3/3 - \dots; (-1)^N * x^N / N; N = 1, 2, \dots; y = \ln(1+x); -1 < X \leq 1$</p>
Вариант 12.	<p>Рассчитать значение функции $y = \sin x$ при некотором значении "x" с использованием представления функции в виде ряда: $x - x^3/3! + \dots + (-1)^{(N+1)} * x^{(2*N+1)} / (2*N+1)! + \dots;$ $a_0 = x, a_N = k * a_{N-1}, k = (-x^2) / (2*N * (2*N+1)), N = 0, 1, 2, \dots$</p>

Контрольные вопросы:

1. Какая программа называется циклической?
2. Формат оператора цикла с параметром.
3. Чем отличается цикл «пока» от цикла «до»?

Тесты:

1. Написать фрагмент программы, выводящий следующее:

1 2 3 4 5 6 7 8 9 10

- for loop := 1 to 10 do write(loop, ' ');
- for loop := 1 to 10 do write('loop ');
- for loop := 1 to 9 do write(loop, ' ');
- for loop = 1 to 10 do write(loop, ' ');

2. Написать фрагмент программы, выводящий следующее:

1
22
333
4444
55555

- for loop = 1 to 5
begin
 for loop1 = 1 to loop do write(loop1);
 writeln
end
- for loop := 1 to 5 do
begin
 for loop1 := 1 to loop do write(loop);
 writeln
end
- for loop := 1 to 5 do
begin
 for loop1 := 1 to loop do write(loop1);
 writeln
end

3. Написать фрагмент программы, выводящий следующее:

A B C D E F

- loop := 'A';
while loop <= 'F' do
begin
 write(loop, ' ');
 loop := loop + 1
end;
- loop := 'F';
while loop > 'A' do

```
begin
    write( loop, ' ');
    loop := loop - 1
end;
```

- loop := A;
while loop < F do
begin
 write(loop, ' ');
 loop := loop + 1
end;

4. Какое ключевое слово не относится к циклической программе?

- If
- While
- For
- Until

5. Какой оператор цикла с предпроверкой условия?

- Repeat
- While
- For
- Until

Лабораторная работа № 4

Массивы: одномерные и двумерные

Цель работы: Ознакомиться с составлением программ, включающих массивы. Моделирование представления в памяти векторов и таблиц.

Прорабатываемые темы.

- Простейшие статические структуры данных;
- Векторы как одномерные массивы;
- Матрицы как двумерные массивы.

Пример: Составить программу для вычисления таблицы умножения.

Программа, составленная на языке Pascal:

```
PROGRAM MASSIV;  
    VAR C: ARRAY[1..10,1..10] OF INTEGER;  
        A,B:INTEGER;  
BEGIN  
    FOR A:=1 TO 10 DO  
    BEGIN  
        FOR B:=1 TO 10 DO  
        BEGIN
```

```

C[A,B]:=A*B;
WRITELN(A, ' ',B, ' ',C[A,B]);
END;
END;
  READLN;
  END.

```

Задания для работы.

Номер варианта	Задание
Вариант 1.	Составить программу заполнения одномерного массива, так чтобы его i -ый элемент был равен $a[i]=(i*i+1)/\sin(i)$.
Вариант 2.	Составить программу определения количества элементов одномерного массива, значение элементов которых меньше заданного действительного числа t .
Вариант 3.	Нахождение среди значений элементов, находящихся на главной диагонали матрицы, наибольшего и наименьшего.
Вариант 4.	Составить программу суммирования значений элементов одномерного массива, больших заданного действительного числа t .
Вариант 5.	Составить программу определения количества элементов одномерного массива, значения элементов которых больше заданного действительного числа t .
Вариант 6.	Подсчитать число и сумму положительных, число и произведение отрицательных элементов заданного массива $A(N)$.
Вариант 7.	Элементы заданного массива $B(N)$ переписать в новый массив $A(N)$ в обратном порядке.
Вариант 8.	В заданном массиве $X(N)$ заменить нулями все отрицательные компоненты, непосредственно предшествующие его максимальной компоненте (первой по порядку, если их несколько).
Вариант 9.	Вычислить сумму квадратов всех элементов заданного массива $X(N)$, за исключением элементов, кратных пяти.
Вариант 10.	В заданном массиве $A(N)$ поменять местами наибольший и наименьший элементы.
Вариант 11.	Вывести на печать номера элементов заданного массива $Y(N)$, удовлетворяющих условию $0 < y[i] < 1$.
Вариант 12.	В заданном массиве $A(N)$ вместо $a[1]$ записать наибольший элемент массива, а вместо $a[N]$ - наименьший элемент массива.

Семестровые задания.

Номер варианта	Задание
Вариант 1.	В заданном массиве $A(N)$, все элементы которого попарно различны, найти наибольший элемент из отрицательных.
Вариант 2.	В заданном массиве $A(N)$, все элементы которого попарно различны, найти наименьший элемент из положительных.
Вариант 3.	В заданном массиве $A(N)$ положительные элементы уменьшить вдвое, а отрицательные заменить на значения их индексов.
Вариант 4.	В заданном массиве $A(N)$ вычислить среднее геометрическое и среднее арифметическое значения для положительных элементов.
Вариант 5.	Дана матрица $A(N,M)$. Найти её наибольший элемент (первый по порядку, если их несколько) и номера строки и столбца, на пересечении которых он находится.
Вариант 6.	В каждой строке заданной матрицы $A(N,M)$ вычислить сумму, количество и среднее арифметическое положительных элементов.
Вариант 7.	Для заданной целочисленной матрицы $A(N,M)$ определить, является ли сумма её элементов чётным числом.
Вариант 8.	Дана матрица $A(N,M)$. Найти количество элементов этой матрицы, больших среднего арифметического всех её элементов.
Вариант 9.	В заданной матрице $A(N, M)$ поменять местами столбцы с номерами P и Q .
Вариант 10.	Найти наибольший элемент побочной диагонали заданной матрицы $A(N,N)$ (первый по порядку, если их несколько) и вывести на печать всю строку, в которой он находится.
Вариант 11.	Для заданной матрицы $A(N,N)$ найти сумму элементов, расположенных в строках с отрицательным элементом на главной диагонали.
Вариант 12.	Дана матрица $A(N,M)$. Определить число ненулевых элементов в каждой строке матрицы.

Контрольные вопросы:

1. Что такое массив?
2. Какой массив характеризует матрицу ?
3. В каком формате располагается значение элементов в памяти машины?
4. Какие существуют виды массивов?

Тесты:

1. Написать предложение, определяющее числовой массив из целых чисел в количестве 20.

- type numbers = ARRAY[1..20] of int;
- type numbers = ARRAY[1..20] of integer;
- type numbers := ARRAY[1..20] of integer;
- var numbers = ARRAY[0..19] of int;

2. Написать предложение, присваивающее число 20 четвертому элементу массива mynumbers.

- mynumbers[3] = 20;
- mynumbers[20] := 4;
- mynumbers[4] = 20;
- mynumbers[4] := 20;

3. Массив – это:

- запись множества переменных разного типа
- неупорядоченная совокупность отличных друг от друга однотипных элементов
- последовательность, состоящая из фиксированного числа однотипных элементов
- тип одномерных величин

4. Как обозначается массив?

- Array
- Dim
- Procedure
- Set

5. Как называется такая таблица? а 1 2 3 4 5 6 7

- квадратная
- кубическая
- прямоугольная
- линейная

Лабораторная работа № 5 **Процедуры и функции**

Цель работы: Ознакомиться с составлением программ с использованием подпрограмм.

Темы для предварительной проработки:

- Процедуры;
- Нестандартные функции пользователя.

Пример: Функция для расчета высоты треугольника по заданным значениям его сторон.

Program TR;


```

Var a, b, c, ha, hb, hc: real;
Function H_TR(a, b, c: real): real;    { a, b, c - Стороны треугольника }
  Var p, s: real;
Begin
  If (a<0) or (b<0) or (c<0) Then begin
    Writeln('Стороны треугольника >0 ?'); readln; Halt end;

  If (a>(b+c)) or (b>(a+c)) or (c>(a+b)) Then begin
    Writeln('a<(b+c), b<(a+c), c<(a+b) ?'); readln; Halt end;
  p:= (a+b+c)/2;                { полупериметр }
  s:= Sqrt(p*(p-a)*(p-b)*(p-c)); { площадь }
  H_TR:= (2*s)/a; { Присвоение функции значения }
End;
Begin
  Writeln('Введите значения сторон треугольника a,b,c');
  Readln(a,b,c);
  ha:= H_TR(a, b, c); hb:= H_TR(b, a, c); hc:= H_TR(c, b, a);
  Writeln('Высоты треугольника:');
  Writeln('ha=',ha:-10:4, 'hb=',hb:-10:4, 'hc=',hc:-10:4); Readln
End.

```

В программе трижды вызывается функция расчета высоты треугольника для различных комбинаций фактических параметров, что позволяет вычислить все высоты треугольника.

Задания для работы.

Номер варианта	Задание
Вариант 1.	Рассчитать площадь треугольника по известным координатам вершин с использованием формулы $S = 0.5 * \text{abs}(y1 * (x3 - x2) + y2 * (x1 - x3) + y3 * (x2 - x1))$
Вариант 2.	Разработать функцию расчета площади треугольника по формуле Герона (значения сторон вычисляются внутри функции).
Вариант 3.	Вывести на экран сообщения о типе треугольника: остроугольный, прямоугольный, тупоугольный (задаются координаты вершин). Для расчета углов использовать теорему косинусов, например: $c_a = (b*b + c*c - a*a) / (2*b*c)$; где c_a – косинус угла, противоположного стороне "a".
Вариант 4.	Вычислить значение выражения $y = a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5$, используя процедуру для a^n .
Вариант 5.	Упорядочить значения a, b, c по возрастанию, используя процедуру смены местами двух чисел.

Вариант 6.	Подсчитать число сочетаний без повторений из n элементов по k элементов.
Вариант 7.	Найти количество и сумму делителей числа.
Вариант 8.	Вычислить факториал числа с помощью рекурсии.
Вариант 9.	Найти НОД трех чисел с помощью алгоритма Евклида.
Вариант 10.	Найти НОК четырех чисел.
Вариант 11.	Заданы 3 одномерных массива A , B и C из n чисел. Вычислить среднее арифметическое наименьших элементов этих массивов.
Вариант 12.	Вычислить среднее арифметическое действительных корней квадратных уравнений $2x^2-3x+1=0$ и $3x^2+2x-1=0$.

Семестровые задания.

Номер варианта	Задание
Вариант 1.	Подсчитать количество отрицательных элементов в двумерном массиве 7×8 , используя процедуру их подсчета в одномерном массиве.
Вариант 2.	Вычислить $Z = (x + y) / (k * n)$, где x и k – сумма и количество положительных элементов массива $A(40)$, где y и n – сумма и количество отрицательных элементов массива $B(50)$. Определение суммы и количества положительных и отрицательных элементов выполнить в подпрограмме.
Вариант 3.	В трех целочисленных массивах $A(40)$, $B(30)$, $C(60)$ найти все элементы, кратные 3. Поиск элементов однородного массива, кратных некоторому числу P , оформить в виде функции.
Вариант 4.	В массиве $A(12, 15)$ найти произведение столбцов: второго на десятый, третьего на девятый. Вычисление произведения двух столбцов матрицы оформить в виде функции.
Вариант 5.	В целочисленном массиве $Z(10, 15)$ определить номер столбца, содержащего максимальное число нечетных элементов. Подсчет числа нечетных элементов оформить в виде функции.
Вариант 6.	Вычислить: $S = N! + K! + (I + K)!$, где $N = 3$, $K = 5$, $I = 8$. Вычисление факториала оформить в виде функции.
Вариант 7.	Даны действительные числа s, t . Составить программу вычисления выражения $f(t, -2s, 1.17) + f(2.2, t, s-t)$, где $f(a, b, c) = (2a - b - \sin(c)) / (5 + c)$.
Вариант 8.	Даны координаты вершин многоугольника $(x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10})$. Напишите программу для вычисления его периметра (вычисление расстояния между вершинами оформить подпрограммой).

Вариант 9.	Определить количество положительных, отрицательных и нулевых элементов в трех заданных матрицах: $A(8, 10)$, $B(4, 8)$, $C(3, 7)$. Подсчет указанных чисел в матрице оформить в виде функций.
Вариант 10.	Даны действительные числа a_0, \dots, a_6 . Получить для $x = 1, 2, 3, 4$ значения $P(x + 1) - P(x)$, где $P(y) = a_6y^6 + a_5y^5 + \dots + a_0$. Вычисление $P(y)$ оформить в виде функции.
Вариант 11.	Определить полусумму длин двух векторов: $A(1,5; 2,5; -0,3)$ и $B(-11,7; 9,3; 2,5; 3,7; -1,2)$. Вычисление длины вектора оформить в виде функции.
Вариант 12.	В массиве $T(33, 19)$ найти произведение строк: первой на последнюю, второй на предпоследнюю. Вычисление произведения двух строк матрицы оформить в виде функции.

Контрольные вопросы:

1. Для чего оформляются вычисления в виде программных модулей (процедур)?
2. Где определяется процедура?
3. Общая структура записи процедуры.
4. Чем отличается процедура от функции?
5. Почему необходимо использовать подпрограммы (процедуры и функции пользователя) в программе на Паскале?
6. Что общего и в чем отличие процедуры от основной программы, если рассматривать их оформление (запись на языке Паскаль)?

Тесты:

1. Процедуры и функции – это:
 - операторы
 - подпрограммы
 - имена
 - переменные
2. Рекурсивная функция – это:
 - повторение выполнения функции или процедуры внутри себя
 - оператор
 - цикл
 - метод определения функции или процедуры
3. Обращение к функции в программе имеет вид:
 - $\langle \text{имя функции} \rangle (\langle \text{список глобальных параметров} \rangle : \text{тип параметров}) : \text{тип значения};$
 - $\langle \text{оператор функции} \rangle ;$
 - $\langle \text{имя функции} \rangle := \text{тип значения};$
 - $\langle \text{имя функции} \rangle ;$

4. Обращение к процедуре в программе имеет вид:

- <имя процедуры>(<список глобальных параметров>);
- <оператор процедуры>;
- <имя процедуры>:= тип значения;
- <имя процедуры>;

5. Определить результат вычисления функции $akk(1,k)$; если задано ее описание:

```
function akk(m,n:integer):integer;
```

```
begin
```

```
if m=0
```

```
then akk:=n+1
```

```
else
```

```
if n=0
```

```
then akk:=akk(m-1,1)
```

```
else akk:=akk(m-1, akk(m,n-1));
```

```
end;
```

- $2*k$
- $k+3$
- $2*k=3$
- $k+2$

6. В некоторой программе описаны переменные:

```
var i:integer; r:real; c:char;
```

Какая из данных процедур или функций может быть вызвана из этой программы с помощью оператора $c:=f(i, 5, r, 5*i, c, "r")$;

- Function $f(\text{var } a:\text{integer}; b:\text{integer}; \text{var } c:\text{real}; d:\text{real}; \text{var } e:\text{char}; g:\text{char})$;
- Function $f(\text{var } a:\text{integer}; b:\text{integer}; \text{var } c:\text{real}; d:\text{real}; \text{var } e:\text{char}; g:\text{char}):\text{char}$;
- procedure $f(\text{var } a:\text{integer}; b:\text{integer}; \text{var } c:\text{real}; d:\text{real}; \text{var } e:\text{char}; g:\text{char})$;
- Function $f(\text{var } a:\text{integer}; \text{var } b:\text{integer}; \text{var } c:\text{char}; d:\text{real}; \text{var } e:\text{char}; g:\text{char}):\text{char}$;

Лабораторная работа № 6

Символьные величины

Цель работы: Приобретение и закрепление знаний о представлении данных типа строка символов и о строковых операциях.

Прорабатываемые темы: Строки: представление строк, операции над строками. Средства обработки строк в языках программирования.

Пример программы вывода на экран символов (букв) от A до Z:

```
program Simbol; {Данные символьного типа}
var i:Char;
```

```

begin
writeln('Вывод на экран букв от A до Z:');
for i:='A' to 'Z' do
write(' ',i);
writeln;
writeln('Выход-любая клавиша. ');
readkey;
end.

```

Задания для работы.

Номер варианта	Задание
Вариант 1.	Определить количество слов в заданном тексте.
Вариант 2.	Определить, является ли заданное слово "перевёртышем".
Вариант 3.	В заданном тексте одно заданное слово везде заменить на другое заданное слово такой же длины.
Вариант 4.	Заданную последовательность слов переупорядочить в алфавитном порядке.
Вариант 5.	Подсчитайте количество запятых в заданном тексте.
Вариант 6.	Определите долю пробелов в заданной строке.
Вариант 7.	В заданном тексте везде букву "a" замените на букву "б", а букву "б" — на букву "a".
Вариант 8.	Удвойте каждую букву в заданном тексте.
Вариант 9.	Вычеркните из заданного слова все буквы "a".
Вариант 10.	Выясните, верно ли, что в заданном предложении <i>P</i> есть все буквы, входящие в заданное слово <i>S</i> .
Вариант 11.	Определите количество предложений в заданном тексте (предложение заканчивается либо точкой, либо вопросительным или восклицательным знаком).
Вариант 12.	Определите долю гласных букв в заданном тексте на английском языке.

Семестровые задания.

Номер варианта	Задание
Вариант 1.	Отредактируйте заданное предложение, удаляя из него все слова с чётными номерами.
Вариант 2.	Дан текст из цифр и строчных латинских букв, за которым следует точка. Определите, каких букв — гласных (<i>A, E, I, O, U</i>) или согласных — больше в этом тексте.
Вариант 3.	Найдите самое длинное и самое короткое слово в заданном предложении.

Вариант 4.	Для каждого из слов заданного предложения укажите, сколько раз оно встречается в предложении.
Вариант 5.	Сколько раз во втором слове встречается первая буква первого слова?
Вариант 6.	Дано натуральное число n и символы s_1, s_2, \dots, s_n . Выяснить, каких символов больше + или *.
Вариант 7.	Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на 1 символ до тех пор, пока в строке не останется 1 символ.
Вариант 8.	Напишите программу, определяющую, какая из букв первая или последняя встречаются в заданном слове чаще.
Вариант 9.	Напишите программу, определяющую, сколько букв «о» стоит в заданном слове на четных местах.
Вариант 10.	Заданы фамилия, имя и отчество учащегося, разделенные пробелами. Напишите программу, печатающую фамилию ученика и его инициалы.
Вариант 11.	Напишите программу, подсчитывающую количество различных букв в заданном слове.
Вариант 12.	Определить и вывести на экран номера позиций и количество повторений заданного символа в строке, введенной с клавиатуры.

Контрольные вопросы:

1. Что такое данные символьного типа?
2. Что такое данные строкового типа?
3. Что такое конкатенация?
4. В чем различия и сходства данных типа char и string?

Тесты:

1. В типе string количество символов одной строки не должно превышать:

- 256
- 255
- 1024
- 2400

2. Каков будет результат выполнения программы:

```
var s1,s2,s3:string;
begin
s1:='паро'; s2:='воз';
s3:=concat(s1,s2);
writeln(s3);
end.
```

- Пар и воз

- Парвз
- 7
- паровоз

3. Каков будет результат выполнения программы:

```
var s1,s2:string;
begin
s1:='информатика';
delete(s1,3,4);
writeln(s1);
end.
```

- Инатика
- Форма
- Инф
- Инфо

4. Каков будет результат выполнения программы:

```
var s1,s2:string;
begin
сору('крокодил',4,3);
writeln(s1);
end.
```

- Крок
- Одил
- Код
- кродил

5. Каков будет результат выполнения программы:

```
var s:string;
begin s:=length('каникулы') end.
```

- S=0
- S=1
- S=8
- S=true

6. Какая из данных программ на Паскале правильная:

- var r:string; begin r:=true; if r then halt; end.
- Var r:string; c:char; begin r:=4/c; end.
- Var r:string; begin r:='Hello'; writeln(r); end.
- Var r:string; begin r:=friend; write(r); end.

Лабораторная работа № 7 Множества

Цель работы: приобретение знаний о множественном типе данных.

Темы для предварительной проработки:

- Множества;
- Перечисляемый тип;
- Интервальный тип.

Пример. Составить программу выделения из множества целых чисел от 1 до 30 следующих подмножеств:

множества чисел, кратных 2;

множества чисел, кратных 3;

множества чисел, кратных 6;

множества чисел, кратных 2 или 3.

```
Program mnoj;
```

```
Const n=30;
```

```
Type mn=Set Of 1..n;
```

```
Var n2,n3, n6, n23:mn;
```

```
{n2 - множество чисел, кратных 2, n3 - множество чисел, кратных 3, n6 -  
множество чисел, кратных 6, n23 - множество чисел, кратных 2 или 3}
```

```
k: Integer;
```

```
procedure print1 (m:mn);
```

```
var i: Integer;
```

```
begin
```

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

```
if i In m then write(i:3);
```

```
writeln;
```

```
end;
```

```
begin
```

```
n2:=[]; n3:=[]; {начальное значение множеств}
```

```
for k:=1 to n do {формирование n2 и n3}
```

```
begin {если число делится на 2, то заносим его в n2}
```

```
if k mod 2=0 then n2:=n2+[k];
```

```
if k mod 3=0 then n3:=n3+[k]; {если число делится на 3, то заносим его в  
n3}
```

```
end;
```

```
{числа, кратные 6, кратны 2 и 3, поэтому это – пересечение двух мно-  
жеств; числа, кратные 2 или 3, - это объединение этих множеств}
```

```
n6:=n2*n3; n23:=n2+n3;
```

```
writeln('2'); {вывод множеств}
```

```
print1(n2);
```

```
writeln('3');
```

```
print1(n3);
```

```
writeln('6');
```

```
print1(n6);
```

```
writeln('2 3');
```



```
print1(n23);
readln;
end.
```

Задания для работы:

Номер варианта	Задание
Вариант 1.	Дано натуральное число n . Вывести все цифры, не входящие в десятичную запись этого числа в порядке возрастания.
Вариант 2.	Найти простые числа в промежутке $[1..n]$. Число n вводится с клавиатуры. (Решето Эратосфена).
Вариант 3.	Дана непустая последовательность символов. Вывести множества, элементами которых являются: цифры от «0» до «9» и знаки арифметических операций.
Вариант 4.	Дана непустая последовательность символов. Вывести множества, элементами которых являются: буквы от «A» до «F» и от «X» до «Z»
Вариант 5.	Дана непустая последовательность символов. Вывести множества, элементами которых являются: знаки препинания и буквы от «E» до «N»
Вариант 6.	Подсчитать общее количество цифр и знаков «+», «-», «*» в строке, введенной с клавиатуры.
Вариант 7.	Сформировать множество строчных латинских букв, входящих в произвольную введенную строку, и подсчитать количество знаков препинания в ней.
Вариант 8.	Подсчитать количество цифр в заданной строке и вывести их.
Вариант 9.	Вывести в алфавитном порядке все буквы текста, оканчивающегося точкой, входящих в него не менее двух раз.
Вариант 10.	Вывести в алфавитном порядке все буквы текста, оканчивающегося точкой, входящих в него не более двух раз.
Вариант 11.	Вывести в алфавитном порядке все буквы текста, оканчивающегося точкой, входящих в него более двух раз.
Вариант 12.	Вывести все символы заданного текста, входящих в него по одному разу.

Семестровые задания:

Номер варианта	Задание
Вариант 1.	Подсчитать число гласных и согласных букв в тексте и определить, каких букв больше. Учесть, что в строке могут

	быть другие символы, кроме букв.
Вариант 2.	Вывести все первые вхождения в данный текст строчных латинских букв, сохраняя их взаимный порядок.
Вариант 3.	Найти и вывести в порядке убывания всех простых чисел из промежутка [2..201], используя метод «решета Эратосфена».
Вариант 4.	Задано множество вычислительных машин. Известен набор машин, имеющихся в каждом из 10 техникумов города. Построить и вывести множества, включающие в себя вычислительные машины, которыми обеспечены все техникумы.
Вариант 5.	Задано множество вычислительных машин. Известен набор машин, имеющихся в каждом из 10 техникумов города. Построить и вывести множества, включающие в себя вычислительные машины, которые имеет хотя бы один техникум.
Вариант 6.	Задано множество вычислительных машин. Известен набор машин, имеющихся в каждом из 10 техникумов города. Построить и вывести множества, включающие в себя вычислительные машины, которых нет ни в одном техникуме.
Вариант 7.	Задать множество, элементы которого – случайные числа из диапазона целых чисел до 50. Вывести все элементы этого множества и подсчитать их количество.
Вариант 8.	Заполнить карточку спортлото 6 из 49, т.е. сформировать множество с заданным числом элементов.
Вариант 9.	Найти наименьший элемент множества из случайных чисел от 1 до 200.
Вариант 10.	Найти наибольший элемент множества из случайных чисел от 1 до 200.
Вариант 11.	Ввести строку символов, состоящую из латинских букв, цифр и пробелов. Осуществить проверку правильности введенных символов.
Вариант 12.	Подсчитать количество слов в заданном тексте.

Контрольные вопросы:

1. Что такое тип-множество?
2. Каково максимальное число элементов в множестве?
3. Какие зарезервированные слова используются для задания множества?
4. Какие операции можно проводить над множествами?
5. Что такое конструктор множеств?

Тесты:

1. Каков будет результат выполнения программы:

```
type digits=set of 0..9;
var d1, d2, d3: digits;
begin
```

```
d1:=[1,3,5];
d2:=[0,4,5];
d3:=d1*d2;
write(d3);
end.
```

- d3=[0, 1, 3, 4, 5]
- d3=[0, 1, 3, 4]
- d3=[5]
- d3=[1, 3, 5, 0, 4, 5]

2. Какой из представленных типов является перечисляемым?

- m: Integer;
- type n=(...);
- s:string;
- g:char;

3. Что нельзя делать с множествами?

- Суммировать
- Вычитать
- Вводить элемент
- Умножать

4. Какое максимальное количество элементов в множестве?

- 10
- 255
- 355
- 100

5. Чему будет равно c: a:=[6,19,0,5]; b:=[19,3]; c:=a+b;

- 19
- [6, 0, 5]
- [6, 19, 0, 5, 3]
- [0, 6]

Лабораторная работа № 8

Интегрированные структуры данных: файлы, записи

Цель работы: Освоить основные приемы работы с записями и файлами.

Прорабатываемые темы:

Простейшие статические структуры данных: записи. Типичные операции. Текстовый ввод-вывод (консольный и файловый).

Пример: Составить программу с использованием записи, включая следующие поля: ФИО студента, дату рождения, адрес, курс, группа.

Программа, составленная на языке Pascal:

```

PROGRAM ZAPIS;
TYPE
MEN=RECORD
  FIO:STRING[30];
  DATE:STRING[10];
  ADRES:STRING[60];
  GR:STRING[5];
  KURS:1..4;
END;
VAR
  R:MEN;
BEGIN
  WITH R DO
  BEGIN
    FIO:='Иванов Василий Петрович';
    DATE:='15.07.1995';
    ADRES:='Los Angeles, Loves bich, 49-10';
    GR:='RO-11';
    KURS:=1;
    WRITE(FIO,' ',DATE,' ',ADRES,' ',GR,' ',KURS);
  END;
  READLN;
END.

```

Задания для работы:

Постановка задачи

Для заданной прикладной области разработать описание объектов этой области. Разработать процедуры, реализующие базовые операции над этими объектами, в том числе:

- текстовый ввод-вывод (консольный и файловый);
- присваивание;
- задание константных значений;
- сравнение (не менее 2-х типов).

Процедуры и описания данных должны составлять отдельный модуль (модуль типа данных).

Подготовить на магнитном носителе файл исходных данных. Используя процедуры и описания модуля типа данных, разработать программу, обеспечивающую ввод исходных данных из первого файла данных в память и хранение их в массиве, сортировку массива по алфавитному и по числовому параметру.

Для каждой области перечислены параметры объекта. Среди параметров обязательно есть ключевое алфавитное поле (например, фамилия), которое идентифицирует объект, у каждого объекта имеется также одно или не-

сколько числовых полей, по которым вероятны обращения к объекту. Набор характеристик может быть расширен.

Для работы с типом данных принимается следующий набор операций:

- ввод значения объекта из текстового файла;
- ввод значения объекта с консоли;
- вывод значения объекта в текстовый файл;
- вывод значения объекта на консоль;
- форматный вывод значения объекта на терминал;
- присваивание объекту константных значений;
- присваивание объектов;
- сравнение объектов.

Задания для работы:

№	Прикладная область	Атрибуты информации
1	Отдел кадров	фамилия сотрудника, имя, отчество, должность, стаж работы, оклад
2	Красная книга	вид животного, род, семейство, место обитания, численность популяции
3	Производство	обозначение изделия, группа к которой оно относится, год выпуска, объем выпуска, расход металла
4	Персональные ЭВМ	фирма-изготовитель, тип процессора, тактовая частота, емкость ОЗУ, емкость жесткого диска
5	Библиотека	автор книги, название, год издания, цена, количество в библиотеке
6	Радиодетали	обозначение, тип, номинал, количество на схеме, обозначение возможного заменителя
7	Текстовые редакторы	наименование, фирма-изготовитель, количество окон, количество шрифтов, стоимость
8	Телефонная станция	номер абонента, фамилия, адрес, наличие блокиратора, задолженность
9	Быт студентов	фамилия студента, имя, отчество, факультет, размер стипендии, число членов семьи
10	Спортивные соревнования	фамилия спортсмена, имя, команда, вид спорта, зачетный результат, штрафные очки
11	Соревнование факультетов по успеваемости	факультет, количество студентов, средний балл по факультету, число отличников, число двоечников
12	С/х работы	фамилия студента, имя, отчество, факультет, вид работ, заработок

Семестровые задания.

№	Прикладная область	Атрибуты информации
1	Спутники планет	название, название планеты-хозяина, год открытия, диаметр, период обращения
2	Сельхозработы	наименование с/х предприятия, вид собственности, число работающих, основной вид продукции, прибыль
3	Сведения о семье	фамилия студента, имя, отчество, факультет, специальность отца, специальность матери, количество братьев и сестер
4	Скотоводство	вид животных, количество особей в стаде в возрасте до 1 года, количество особей 1 - 3 лет, свыше 3 лет, смертность в каждой группе, рождаемость
5	Микросхемы памяти	обозначение, разрядность, емкость, время доступа, количество на схеме, стоимость
6	Описание изображения	тип фигуры (квадрат, окружность и т.п.), координаты на плоскости, числовые характеристики (длина стороны, радиус и т.п.).
7	Лесное хозяйство	наименование зеленого массива, площадь, основная порода, средний возраст, плотность деревьев на кв.км
8	Городской транспорт	вид транспорта, номер маршрута, начальная остановка, конечная остановка, время в пути
9	Деканат	фамилия студента; имя; отчество; год рождения; обозначение группы; средний балл в сессию.
10	Записная книжка	фамилия, имя, отчество; адрес: улица, дом; телефон; E-mail;
11	Пункт проката	фамилия, имя, отчество клиента; адрес (улица, дом, квартира); что взял (один предмет)
12	Расписание поездов	номер поезда, название (откуда-куда), время прибывания на станцию, время отправления.

Контрольные вопросы:

1. Что такое запись?
2. Определение типа записи.
3. Что такое поля и идентификаторы полей?
4. Что используется для обращения к полям записи?
5. Формат оператора обращения к записи.
6. Что такое файл?

7. Какие зарезервированные слова используются для задания типа-файла?
8. Как определяются файловые переменные?
9. Перечислить стандартные процедуры и функции для работы с файлами.
10. Какие виды файлов используются в Паскале?

Тесты:

1. При помощи какой функции можно определить конец файла?
 - Eoln
 - Eof
 - Odd
 - End
2. Открытие файла для перезаписи:
 - append
 - read
 - rewrite
 - write
3. При помощи какой процедуры можно связать программное имя файла с дисковым именем файла?
 - Append
 - Close
 - Assign
 - Name
4. Как объявляется текстовый файл?
 - a: text;
 - a: tfile;
 - a: file;
 - a: textf;
5. Каким образом можно поставить маркер конца строки в текущем окне?
 - Readln
 - Writeln
 - End
 - Read
6. Сколько уровней вложения разрешено использовать в операторе with?
 - 3
 - 5
 - 9
 - 15

Лабораторная работа № 9 Модуль CRT

Цель работы: Освоить основные приемы работы с модулями Паскаль, в частности рассмотреть модуль вывода текста.

Прорабатываемые темы:

- Режимы работы монитора;
- Процедуры и функции модуля;
- Координаты курсора.

Пример программы, выводящей на экран в различных текстовых режимах надпись в виде ступеньки с заданным шагом "dx" по оси "x", в каждой строке "y", начиная с позиции (1, 1).

```
PROGRAM FAM;
```

```
Uses CRT;
```

```
var N : word; f, dx, x, y, i, j, xm, ym : byte;
```

```
BEGIN
```

```
for i:=0 to 9 do begin { режимы работы монитора }
```

```
if i<4 then N:=i else N:=256+i-4; if i=9 then N:=7;
```

```
textMode(N);
```

```
xm:=lo(WindMax)+1; ym:=hi(WindMax)+1;
```

```
write('xm=',xm, '_ym=',ym, '_N=',N, '_Нажмите Enter'); readln;
```

```
TextBackGround(1); clrscr; TextColor(14); x:=1;
```

```
f:=8; dx:=3; { f - длина фамилии + курсор, dx - приращение от-  
ступа }
```

```
for j:=1 to ym-1 do begin y:=j;
```

```
if (xm-x-f)<0 then x:=1; { контроль выхода надписи за экран }
```

```
gotoXY(x, y); write('ФАМИЛИЯ'); x:= x + dx; end;
```

```
Writeln; write('Нажмите Enter'); readln end;
```

```
TextMode(3)
```

```
END.
```

Задания для работы:

Номер варианта	Задание
Вариант 1.	Вывести на экран в различных текстовых режимах надпись в виде ступеньки с заданным шагом "dx" по оси "x", в каждой строке "y", начиная с позиции (1, Ym) с направлением вверх, вправо.
Вариант 2.	Вывести на экран в различных текстовых режимах надпись в виде ступеньки с заданным шагом " dx<0 " по оси " x ", в каждой строке " y ", начиная с позиции: (Xm-f, Ym) с направлением вверх, влево.

Вариант 3.	В режимах N=1 и N=3 вывести на экран окна разного цвета с уменьшающимся размером (окно в окне). В первой позиции окон выводить номер окна. Использовать оператор цикла с условием ограничения размеров наименьшего окна, например: $(x_2 - x_1 = 2) \text{ or } (y_2 - y_1 = 0)$.
Вариант 4.	В режимах N=1 и N=3 вывести на экран окна разного цвета с координатами, определяемыми функцией Random с ограничением по размеру экрана, например: $x_1 := \text{Random}(X_m - 1) + 1$; $x_2 := x_1 + \text{Random}(X_m - x_1)$; Окна выводятся в операторе цикла с условием: до нажатия любой клавиши.
Вариант 5.	Создать на экране несколько рядов разворачивающихся по горизонтали, либо по вертикали окон разного цвета с надписями.
Вариант 6.	Вывести на экране несколько рядов карточек со случайными номерами цветов, в различных диапазонах для каждого ряда. В центре экрана вывести окно с сообщением о количестве рядов с одинаковым цветом карточек.
Вариант 7.	Создать на экране несколько рядов окон с чередованием двух орнаментов. (Для построения орнаментов в текстовом режиме удобно использовать массив $N_1 \times N_2$ значений из нулей и единиц, записанных в файле).
Вариант 8.	Построить горизонтальную и вертикальную гистограммы столбиками разного цвета для 20-ти значений функций: $Y = \text{Sin}(x) * x$ и $Y = \text{Sin}(x) + x$; в диапазоне "x" от 0 до $2 * \text{Pi}$, с выводом значений "x" и "Y".
Вариант 9.	Разделить экран на четыре равные части. Создать в первой части окно и ввести с клавиатуры надпись: Программу разработал "Фамилия". Создать окно 4 и скопировать эту надпись в окно 4.
Вариант 10.	Создать четыре окна в порядке 1, 2, 3, 4. В центре первого окна ввести с клавиатуры свою фамилию, в других – любые надписи, введенные с клавиатуры в позициях, определяемых функцией Random (без использования массивов).
Вариант 11.	В режиме TextMode(1); создать в центре экрана окно размером 6 строк и 30 колонок. Вывести в нижней строке окна надпись и сделать прокрутку надписи вверх, меняя яркость символов (нормальная, повышенная, пониженная) и цвет фона (без оператора ClrScr;) после каждой прокрутки.
Вариант 12.	Разделить экран по горизонтали на две части. В первой части создать окно и вывести все символы от 0 до 255 с

	указанием номеров, во втором окне в двух строчках вывести два символа, бегающих в противоположные стороны.
--	--

Семестровые задания.

Номер варианта	Задание
Вариант 1.	Разделить экран по вертикали на две части. В первой части создать окно и вывести все символы от 0 до 255 с указанием номеров, во втором окне в двух колонках вывести два символа, бегающих в противоположных направлениях.
Вариант 2.	"Падающие символы" для цифр от 0 до 9, имеющих коды с номерами от 48 до 57. При нажатии на клавишу с "падающей цифрой", генерируется другая "падающая цифра" и т. д. до нажатия клавиши ESC.
Вариант 3.	Создать "падающие символы" для прописных букв латинского алфавита с номерами кодов от 65 до 90. (Введите счетчики правильно и неправильно нажатых клавиш. В конце программы выведите на экран результат работы счетчиков).
Вариант 4.	Создать "падающие символы" для строчных букв латинского алфавита с номерами кодов от 97 до 122. (Введите счетчики правильно и неправильно нажатых клавиш. В конце программы выведите на экран результат работы счетчиков).
Вариант 5.	Создать "бегущую строку". Строку выводить в окне, размером $n+1$ (n – число символов). Цвет окна и символов изменять после каждого вывода строки. Режим монитора TextMode(1).
Вариант 6.	Создать "бегущую строку". Выводить одновременно две надписи одинаковой длины в разных строках одного окна. Цвет окна и символов изменять после каждого вывода строки. Режим монитора TextMode(1).
Вариант 7.	Создать "бегущую строку" с использованием функции копирования. Строку вводить оператором Read(S); с пробелом в конце строки. Длина строки $n := \text{Length}(s)$; Выводить строку оператором Write(Sc); где $Sc := \text{Copy}(S, L1, L2)$.
Вариант 8.	Создать "бегущую строку" с использованием функции копирования при " $j \leq n$ " $Sc := \text{Copy}(S, L1, L2)$; а после "else" с процедурой удаления Delete(Sc, L3, L4);. Строку вводить оператором Read(S); с пробелом в конце строки. Выводить оператором Write(Sc); Длина строки $n := \text{Length}(s)$.
Вариант 9.	Создать "бегущую строку" со звучанием мелодии. После

	вывода каждой группы символов вместо процедуры <code>delay(200)</code> ; ставятся операторы: <code>Sound(F[k])</code> ; <code>Delay(N[k])</code> ; <code>NoSound</code> ; где <code>F</code> , <code>N</code> – массивы частот и длительности звучания нот, определенные в начале программы согласно исполняемой мелодии. Параметр "k" наращивается до числа нот в мелодии, затем обнуляется и т. д.
Вариант 10.	Создать мелодию с частотой "F" и длительностью звучания нот "N", изменяющихся по выбранным формулам: а) $F=1000*\sin(x)+100$, б) $N=\text{random}(2)+1$, где $x=x+0.01$ от $x=0$ до $x=Pi$.
Вариант 11.	Перемещать курсор по экрану с запоминанием выводимых символов и их координат. После окончания "рисования" на экране символов запрограммировать "бесконечный процесс домино" – последовательное исчезновение символов с первого до последнего и восстановление в обратном порядке.
Вариант 12.	Перемещать курсор по экрану с запоминанием выводимых символов и их координат. После окончания "рисования" на экране символов в виде замкнутой фигуры запрограммировать "бесконечный процесс бегущих" символов – последовательное исчезновение символа, задержка и его восстановление и т. д. с первого символа до последнего.

Контрольные вопросы:

1. Чем отличается цвет фона от цвета символов? Какие процедуры управляют этими цветами?
2. Что произойдет, если после вызова процедуры `Sound` не вызвать процедуру `NoSound`?
3. Чем отличается `Readkey` от `Readln`?
4. Что такое курсор?
5. Куда перемещается курсор после выполнения процедуры `ClrScr`?
6. Что называют модулем в Паскаль-программе?

Тесты:

1. Модуль `CRT` позволяет:
 - подключать «мышь»
 - инициализировать графику
 - управлять курсором
 - форматировать дискеты
2. Процедура `Window`:
 - устанавливает текстовый режим
 - рисует прямоугольник
 - создает графическое окно
 - задает текстовое окно на экране.

3. Функция readkey:
 - очищает буфер клавиатуры
 - анализирует нажатие клавиши
 - возвращает символ нажатой клавиши
 - переназначает клавиши
4. Установка текстового режима:
 - textmode
 - modetext
 - text
 - mode
5. Выключение звука:
 - NoSound
 - Sound
 - Delay
 - NotSound

Лабораторная работа № 10 Графика

Цель работы: освоить основные приемы работы с компьютерной графикой

Пример. Составить программу, рисующую голову робота. Вывести под рисунком строку “Это робот”.

Программа составленная на языке Pascal:

```
PROGRAM robot;  
USES GRAPH;  
VAR DRIVER,MODE: INTEGER;  
BEGIN  
  DRIVER:=DETECT;  
  INITGRAPH (DRIVER,MODE,'C:\TP\BGI');  
  SETCOLOR (BLACK);  
  SETBKCOLOR (WHITE);  
  RECTANGLE (100,100,300,300);  
  CIRCLE (150,170,30);  
  CIRCLE (250,170,30);  
  ARC (150,170,45,135,40);  
  ARC (250,170,45,135,40);  
  ELLIPSE (200,250,0,359,10,20);  
  RECTANGLE (130,280,270,290);  
  MOVETO (100,300);  
  LINETO (50,350);
```

LINETO (350,350);
 LINETO (300,300);
 PUTPIXEL (150,170,GREEN);
 PUTPIXEL (250,170,GREEN);
 OUTTEXTXX(195,400,'ЭТО РОБОТ');
 READLN;
 CLOSEGRAPH;
 END.

Задания для работы:

Номер варианта	Задание
Вариант 1.	Нарисовать домик с покатой крышей. В программе вводятся координаты конька крыши.
Вариант 2.	Нарисовать квадратную спираль, причем звенья, идущие слева направо и сверху вниз, имеют нечетную длину, а звенья, идущие справа налево и снизу вверх, имеют четную длину. Предлагается вариант программы, когда для одного значения параметра цикла рисуется полный оборот спирали. Спираль рисуется по часовой стрелке из точки, координаты которой задаются с клавиатуры.
Вариант 3.	Нарисовать диагональную спираль, длина каждого следующего звена которой увеличивается на единицу. Вновь используется прием построения полного витка спирали при одном значении параметра цикла, когда длина следующего звена возрастает за счет увеличения на единицу коэффициентов $(i*a, (i+1)*a, (i+2)*a, (i+3)*a$).
Вариант 4.	Нарисовать лист из нотной тетради. Вычерчивается 12 полос по пять линий, причем сначала рисуется нижняя (пятая) линия в каждой полосе, а в последнюю очередь - верхняя.
Вариант 5.	Подготовить лист в косую линейку, как в тетради-прописи для младших школьников (с полями).
Вариант 6.	Нарисовать зигзаг из линий. В программе задаются координаты начальной точки, а также расстояние между зубцами по горизонтали и высота зубцов зигзага.
Вариант 7.	Провести диагональную штриховку квадрата: из левого нижнего угла в правый верхний проводится линия, а остальные линии проводятся параллельно на расстоянии, задаваемом как шаг. Число линий штриховки определяется как результат целочисленного деления стороны квадрата на шаг штриховки.
Вариант 8.	Нарисовать два заштрихованных треугольника, расположенных на одной диагонали квадрата.

Вариант 9.	Нарисовать горизонтальную бабочку из линий штриховки путем вращения одной из диагоналей треугольника вокруг горизонтальной оси квадрата.
Вариант 10.	Нарисовать цветок, лепестками которого будут служить прямые линии. Если изменять значения радиуса и коэффициенты перед функциями \sin и \cos , то размер лепестков и их количество будут меняться.
Вариант 11.	Нарисовать пять квадратов, лежащих на одной прямой и соприкасающихся друг с другом, причем сторона каждого следующего квадрата в два раза меньше стороны предыдущего. С клавиатуры вводятся координаты точки, из которой проводится линия, служащая основанием всех пяти квадратов.
Вариант 12.	Нарисовать цепочку из квадратов, расположенных на диагонали экрана. С клавиатуры вводятся значение ребра квадрата и значение ребра малого квадрата, получающегося при наложении квадратов друг на друга.

Семестровые задания:

Номер варианта	Задание
Вариант 1.	Нарисовать в цикле семейство окружностей, расположенных как на олимпийской эмблеме. Раскрасить окружности в разные цвета.
Вариант 2.	Нарисовать гроздь винограда. С клавиатуры задается число рядов ягод и их радиус.
Вариант 3.	Нарисовать конусный волчок из горизонтальных эллипсов, уменьшающихся в размере к вертикальным вершинам волчка. Каждый новый эллипс рисуется с меньшими радиусами по обеим осям с новыми координатами по оси y .
Вариант 4.	Нарисовать новогоднюю картину - елка, снеговик, падающий снег. Елка и снеговик (шары снега) создаются с использованием процедур пользователя. Падающий снег можно создать двумя способами: используя символ "*" или используя белые точки, задаваемые процедурой PutPixel.
Вариант 5.	Нарисовать цветик-семицветик, у которого поочередно после нажатия клавиши улетает лепесток (исчезает с экрана).
Вариант 6.	Написать программу, в которой Колобок (шарик) поднимается по ступенькам вверх, движется по площадке, падает вниз, разделяется на два новых колобка, которые начинают движение в разные стороны.

Вариант 7.	Нарисовать циркового клоуна, у которого из стороны в сторону движутся глаза. В программе используются переменные n и $n1$ для контроля направления движения зрачков левого и правого глаза.
Вариант 8.	Нарисовать тучу, закрывающую солнце. Когда туча наполовину закрывает солнце, начинается дождь.
Вариант 9.	Определить три орнамента заполнения: "снежинка", "иголки" с наклоном влево и вправо. Нарисовать из треугольников пять елок и заполнить их орнаментом "иголки". Заполнить экран орнаментом "снежинка".
Вариант 10.	Нарисовать расходящийся из центра экрана по спирали прямоугольный лабиринт шириной 5 пикселей из отрезков вертикальных и горизонтальных прямых. Очищать экран и менять толщину и форму линии.
Вариант 11.	Нарисовать узор из 30 – ти эллипсов с центром узора в середине экрана. Радиусы каждого эллипса (R_x , R_y) и расстояние от "центра" эллипсов до центра узора увеличивать на один пиксел.
Вариант 12.	Составить программу рисования самолета (задавать координаты узлов, которые соединяются прямыми линиями). Нарисовать самолет, движущийся вокруг центра узора по эллиптической траектории ($L_x \times L_y$). Самолет должен поворачиваться вокруг своего "центра" в соответствии с траекторией ($\cos(A) = F_x/L$, $\sin(A) = F_y/L$, где $L = \sqrt{F_x^2 + F_y^2}$).

Контрольные вопросы:

1. Какие графические адаптеры и режимы вы знаете?
2. Перечислить основные графические примитивы.
3. Каков результат выполнения процедуры GetPalette?
4. Каков результат выполнения функции GraphResult?

Тесты:

1. Какой модуль нужно подключить для работы с графикой?
 - Crt
 - Printer
 - Dos
 - Graph
2. Какая процедура вычерчивает круг?
 - Putpixel
 - Arc
 - Circle
 - Rectangle

3. Какому цвету соответствует цифра 0?
 - Черный
 - Белый
 - Синий
 - Красный
4. Какая функция возвращает текущий цвет рисунка?
 - GetBkColor
 - GetColor
 - FloodFill
 - Graph
5. Какая процедура перемещает текущий указатель в точку (x,y)?
 - LineTo
 - MoveTo
 - OutText
 - DoTo

Раздел 2. Дельфи

Лабораторная работа № 1 Программирование кнопок

Пример 1. Создание простейшего Windows-приложения с заданным заголовком окна и цветом формы.

1. Создать папку для сохранения разработанных приложений
2. Запустить Delphi
3. Изменить заголовок окна формы с Form1 на Привет:
в окне инспектора объектов (Object Inspector) установить для свойства Caption значение Привет.
4. Изменить цвет формы со стандартного на другой:
в окне инспектора объектов установить для свойства Color значение clAqua.
5. Выполнить приложение:
 - 5.1. Запустить приложение - меню Run, Run или F9 или кнопка на панели инструментов.
 - 5.2. Изменить размеры окна.
 - 5.3. Поэкспериментировать со стандартными кнопками минимизации и максимизации окна.
 - 5.4 Закончить работу приложения, закрыв его окно.
6. Сохранить форму и проект на диске:
Меню File, Save All, установить свою папку, создать новую папку (с именем ПРИЛОЖЕНИЕ 1), открыть ее, ввести имя проекта.

Пример 2. Создание Windows-приложения, которое содержит текст "Моя первая программа!" и кнопки, позволяющие изменять размер шрифта и двигать текст.

1. Поместить объект Label в окно формы Form1:
2. Переместить объект Label1 на желаемое место в форме.
3. Изменить свойства объекта Label1:
В окне инспектора объектов (Object Inspector) установить следующие значения для свойств объекта:

Объект	Свойство	Значение
Label1	Caption	Моя первая программа!
	Font	12 p., красный
	Alignment	taCenter
	Color	желтый (Yellow)
	AutoSize	False



4. Выполнить приложение: меню Run, Run или F9.
5. Сохранить форму и проект на диске:
Меню File, Save All, установить свою папку, ввести имя Приложение2.
6. Поместить объект Button (командная кнопка) в окно Form1. Он по умолчанию получит имя Button1. Изменить его размеры.
7. Установить свойство Caption объекта Button1 в значение "Увеличение".
8. Написать код для события Click на объекте Button1:
Два раза щелкнуть по объекту Button1 в форме.
Между словами Begin и End написать следующий код: `Label1.Font.Size := Label1.Font.Size + 2;`
9. Выполнить программу. Обратит внимание на то, что происходит при нажатии кнопки с надписью "Увеличение".
10. Сохранить форму и проект на диске: Меню File, Save.
11. Создать объект "командная кнопка" для уменьшения размера шрифта в тексте.
12. Создать объект "командная кнопка" для того, чтобы двигать текст.
Код: `Label1.Left := Label1.Left + 10;`
`Label1.Top := Label1.Top + 10;`
13. Создать объект "командная кнопка" для того, чтобы сделать текст невидимым.
Код: `Label1.visible := false;`
14. Создать объект "командная кнопка" для выхода из работы программы.
Код: `Close;`
15. Сохранить форму и проект.

Пример 3. Создание Windows-приложения, в котором при щелчке на радио-кнопке с названием цвета на светофоре загорается соответствующий цвет.

1. Поместить компоненты Label, Panel, GroupBox, RadioButton (страница Standard) в форму.
2. Установить следующие свойства объектов, используя Инспектор объектов:

Label1	Caption	Светофор
Panel1,2,3	Caption	Цвет
GroupBox1	Caption	Красный
RadioButton1	Caption	Желтый
RadioButton2	Caption	Зеленый
RadioButton3	Caption	



3. Записать код для процедуры обработки события Click (щелчок мыши) на объекте RadioButton1:

```
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
  Panel1.Color := clRed;
  Panel2.Color := clWhite;
  Panel3.Color := clWhite; end;
```

Самостоятельно записать код для процедур:

```
TForm1.RadioButton2Click и TForm1.RadioButton3Click
```

4. Добавить печать информации "Стойте", "Внимание", "Идите" на панели с соответствующим сигналом белым цветом шрифта жирным начертанием 12п.

Пример 4. Создание Windows-приложения, в котором работают цифровые часы с разной скоростью.

1. Поместить компоненты Label (вкладка Standard) и Timer (System) в форму Form1.

2. Установить следующие свойства объектов

Объект	Свойство	Значение
Form1	Name	Clock
Label1	Caption	00:00:00
Label1	Color	clYellow
Label1	Font.Size	24
Label1	Font.Color	Красный

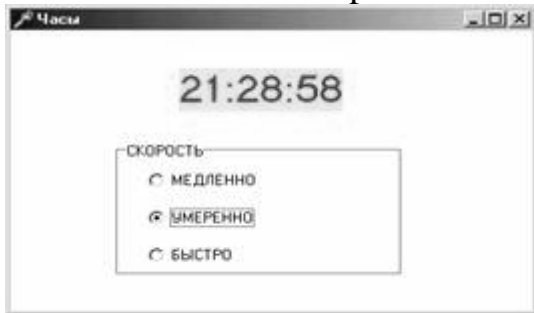
3. Записать код обновления времени для процедуры TClock.Timer1Timer: Label1.Caption:=TimeToStr(Time);

4. Добавление кнопок регулирования скорости обновления времени.

4.1. Добавить в форму компоненты GroupBox и RadioButton:

4.2. Установить следующие свойства объектов:

GroupBox1	Caption	Скорость
RadioButton1	Caption	Медленно
RadioButton2	Caption	Умеренно
RadioButton3	Caption	Быстро



4.3. Записать код для процедуры TForm1.RadioButton3Click:

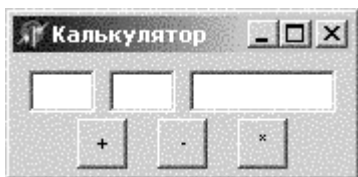
```
Timer1.Interval := 1000;
```

Самостоятельно записать код для процедур:

```
TForm1.RadioButton1Click (3000) и TForm1.RadioButton2Click (2000)
```

Пример 5. Калькулятор.

Для начала поместите на форму три компонента TEdit. В свойстве "Name" для первого компонента напишите "Chislo1", у второго - "Chislo2", у третьего - "Summa". Также сотрите везде свойство "Text". Поместите три кнопки. У первой свойству "Name" присвойте значение "Plus", у второй - "Minus", у третьей - "Umnoj". Свойство "Caption" поставьте соответственно. Скомпонуйте все это и в результате должно получиться примерно следующее:



Вот пример процедуры подсчета :

```
procedure TForm1.PlusClick(Sender: TObject);  
Var  
S: Integer;  
begin  
S := StrToInt(Chislo1.Text)+StrToInt(Chislo2.Text);  
Summa.Text := IntToStr(S);  
end;
```

Тип "Integer" определяет целое число. В нашем случае сумма.

Для того, чтобы перевести строку в число служит оператор "StrToInt". Оператор "IntToStr" обратно переводит целое число в строку. В нашем случае он пишет сумму. Тоже самое, но с разными знаками надо проделать для каждой из кнопок.

Щелкните по свободному от кнопок месту на форме и в инспекторе объектов найдите свойство "BorderStyle" и присвойте ему значение "bsSingle", найдите свойство "BorderIcons" и щелкните по плюсику рядом с названием. В открывшихся подсвойствах свойству "biMaximize" придайте значение "False".

У формы в инспекторе объектов найдите свойство "Position" и придайте ему значение "poScreenCenter" ("По центру экрана").

Вот код этой программы:

```
unit Kalc;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Chislo1: TEdit;
Chislo2: TEdit;
Plus: TButton;
Minus: TButton;
Umnoj: TButton;
Summa: TEdit;
procedure PlusClick(Sender: TObject);
procedure MinusClick(Sender: TObject);
procedure UmnojClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.PlusClick(Sender: TObject);
Var
S: Integer;
begin
S := StrToInt(Chislo1.Text)+StrToInt(Chislo2.Text);
```

```

Summa.Text := IntToStr(S);
end;
procedure TForm1.MinusClick(Sender: TObject);
Var
S: Integer;
begin
S := StrToInt(Chislo1.Text)-StrToInt(Chislo2.Text);
Summa.Text := IntToStr(S);
end;
procedure TForm1.UmnojClick(Sender: TObject);
Var
S: Integer;
begin
S := StrToInt(Chislo1.Text)*StrToInt(Chislo2.Text);
Summa.Text := IntToStr(S);
end;
end.

```

Пример 6. Создание Windows-приложения для вычисления различных математических функций.

Ввести значение x и вычислить значения следующих математических функций:

Cos(x) Ln(x) Power(x,k) Sqr(x) Sqrt(x) Sin(x) Log10(x)



Exp(x)

1. Поместить компоненты Label и Edit в окно формы Form1 и установить их свойства.

2. Записать код для обработки события OnClick на объекте Button1:

```

procedure TForm1.Button1Click(Sender: TObject);
Var x, y1, y2 : real;
begin
x := StrToFloat(Edit1.Text);
y1 := cos(x);
y2 := ln(x);

```

```
Edit2.text := FloatToStr(y1);  
Edit3.text := FloatToStr(y2);  
end;
```

4. Добавить возможность вычисления остальных функций. Учтите, что для использования функций Log10 и Power необходимо в строке Uses добавить в перечень модуль Math.

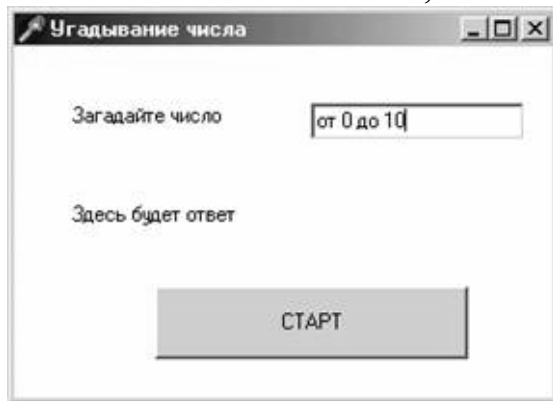
Контрольные вопросы и задания:

1. Покажите в тексте программы: спецификацию программы, заголовок программы, тело программы, секцию определения переменных программы.
2. В чем отличие операторов Write и WriteLn? В чем отличие операторов Read и ReadLn?
3. Укажите порядок выполнения операций в выражении (расставьте скобки): $A \bmod B + C \operatorname{div} D * E$.

Лабораторная работа № 2

Ветвления и циклы

Пример 1. Создание Windows-приложения для вычисления случайного числа и сравнения его с числом, введенным пользователем. По результатам сравнения выдается сообщения: "Вы угадали", "Число меньше искомого", "Число больше искомого", "Число вне диапазона".



Код:

```
var  
  Form1: TForm1; m : integer;          { Случайное число }  
...  
procedure TForm1.Button1Click(Sender: TObject);  
Var n : integer; s : string;  
begin  
  n := StrToInt(Edit1.Text);  
  If (n < 0) Or (n > 10) Then s := 'Число вне диапазона'
```

```

    Else if n > m Then s := 'Число больше искомого'
    Else If n < m Then s := 'Число меньше искомого!'
    Else If n = m Then s := 'Вы угадали';
    Label2.Caption := s;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    Randomize; m := Random (10);
end;

```

Пример 2. Подсчитать среднее арифметическое значение чётных цифр, входящих в запись числа типа LongInt.

Текст программы:

```

unit congo2;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, Mask;
type
    TForm1 = class(TForm)
        ME1: TMaskEdit;
        Label1: TLabel;
        Button1: TButton;
        Button2: TButton;
        Label2: TLabel;
        Label3: TLabel;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
    private { Private declarations }
    public { Public declarations }
    end;
var
    Form1: TForm1;
    n,k:byte;
    s:extended;
    m:int64;
implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
    s:=0;
    k:=0;

```



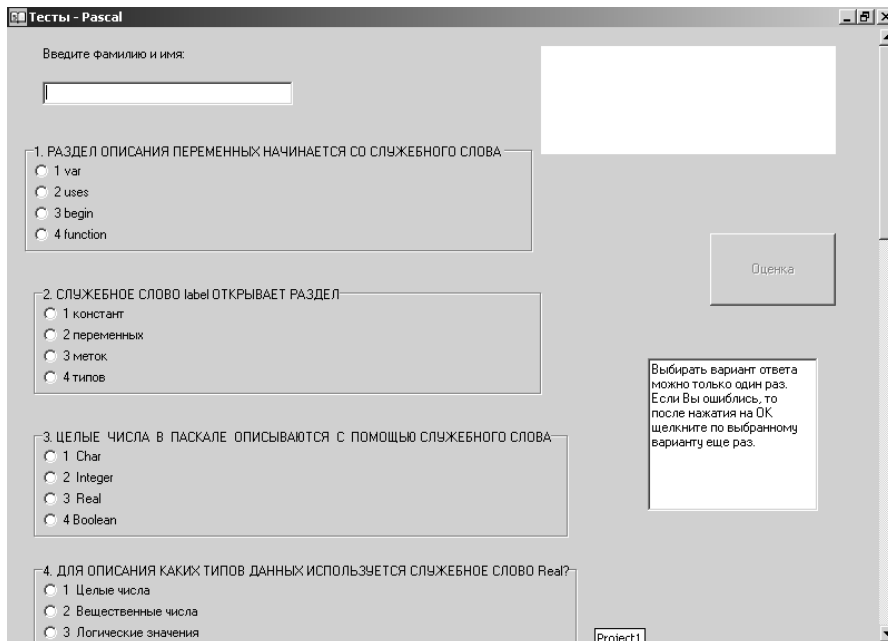
```

if trim(me1.text) <> "" then begin
m:=StrToInt(trim(me1.text));
m:=abs(m);
while m <> 0 do begin
n:=m mod 10;
if not(odd(n)) then begin      { стандартная функция odd(n) возвращает
значение TRUE, если число нечетное }
k:=k+1;
s:=s+n; end;
m:=m div 10 ;
end;
s:=s/k ;
label3.caption:=FloatToStr(s);
end;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
halt;
end;
end.

```

Пример 3. Составить тестирующую программу о языке программирования Паскаль.

1. Написать 14 вопросов по 4 варианта ответов в каждом.
2. Вопросы выводятся в компоненте объединения радио-кнопок, а варианты ответов – щелчки по радио-кнопке.
3. Реализовать защиту от повторного нажатия.
4. Продумать систему оценивания.
5. Обязательно поле для ввода фамилии, без которого не нажметя кнопка «оценка».



Компоненты формы и свойства объектов:

object Form1: TForm1

Left = 195

Top = 100

Width = 498

Height = 495

HorzScrollBar.Position = 271

Caption = 'Тесты - Pascal'

Color = clBtnFace

Font.Charset = DEFAULT_CHARSET

Font.Color = clWindowText

Font.Height = -11

Font.Name = 'MS Sans Serif'

Font.Style = []

OldCreateOrder = False

Position = poDesktopCenter

PixelsPerInch = 96

TextHeight = 13

object Label1: TLabel

Left = 209

Top = 16

Width = 265

Height = 97

AutoSize = False

Color = clWhite

Font.Charset = DEFAULT_CHARSET

Font.Color = clWindowText

```

Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentColor = False
ParentFont = False
end
object Label2: TLabel
  Left = -239
  Top = 16
  Width = 128
  Height = 13
  Caption = 'Введите фамилию и имя:'
end
object RadioGroup1: TRadioGroup
  Left = -255
  Top = 104
  Width = 457
  Height = 97
  Caption = '1. РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ НАЧИНАЕТСЯ СО
СЛУЖЕБНОГО СЛОВА '
  Items.Strings = (
    '1 var'
    '2 uses'
    '3 begin'
    '4 function')
  TabOrder = 0
  OnClick = RadioGroup1Click
end
object RadioGroup2: TRadioGroup
  Left = -247
  Top = 232
  Width = 457
  Height = 97
  Caption = '2. СЛУЖЕБНОЕ СЛОВО label ОТКРЫВАЕТ РАЗДЕЛ'
  Items.Strings = (
    '1 констант'
    '2 переменных'
    '3 меток'
    '4 типов')
  TabOrder = 1
  OnClick = RadioGroup2Click
end

```

```

object Button1: TButton
  Left = 361
  Top = 184
  Width = 113
  Height = 65
  Caption = 'Оценка'
  Enabled = False
  TabOrder = 2
  OnClick = Button1Click
end
object Edit1: TEdit
  Left = -239
  Top = 48
  Width = 225
  Height = 21
  TabOrder = 3
  OnChange = Edit1Change
end
object RadioGroup3: TRadioGroup
  Left = -247
  Top = 360
  Width = 481
  Height = 97
  Caption =
    '3. ЦЕЛЫЕ ЧИСЛА В ПАСКАЛЕ ОПИСЫВАЮТСЯ С ПОМО-
    ЩЬЮ СЛУЖЕБНОГО ' +
    'СЛОВА'
  Items.Strings = (
    '1 Char'
    '2 Integer'
    '3 Real'
    '4 Boolean')
  TabOrder = 4
  OnClick = RadioGroup3Click
end
object RadioGroup4: TRadioGroup
  Left = -247
  Top = 480
  Width = 489
  Height = 97
  Caption =

```

```

'4. ДЛЯ ОПИСАНИЯ КАКИХ ТИПОВ ДАННЫХ ИСПОЛЬЗУЕТСЯ
СЛУЖЕБНОЕ СЛОВО ' +
'Real?'
Items.Strings = (
'1 Целые числа'
'2 Вещественные числа'
'3 Логические значения'
'4 Символы')
TabOrder = 5
OnClick = RadioGroup4Click
end
object RadioGroup5: TRadioGroup
Left = -239
Top = 600
Width = 465
Height = 97
Caption = '5. ДЛЯ ВОЗВЕДЕНИЯ ЧИСЛА X В КВАДРАТ МОЖНО ИСПОЛЬЗОВАТЬ ФУНКЦИЮ '
Items.Strings = (
'1 Round(X)'
'2 Sqrt(X)'
'3 Sqr(X)'
'4 Abs(X)')
TabOrder = 6
OnClick = RadioGroup5Click
end
object RadioGroup6: TRadioGroup
Left = -239
Top = 712
Width = 465
Height = 97
Caption =
'6. ИЗВЕСТНО, ЧТО A=10, B=3. ЧЕМУ РАВНО ЗНАЧЕНИЕ ВЫРАЖЕНИЯ A DIV ' +
'B?'
Items.Strings = (
'1 30'
'2 0,3'
'3 3,3'
'4 3')
TabOrder = 7
OnClick = RadioGroup6Click

```

```

end
object RadioGroup7: TRadioGroup
  Left = -239
  Top = 824
  Width = 417
  Height = 89
  Caption = '7. Язык Паскаль является:'
  Items.Strings = (
    '1 компилятором'
    '2 интерпретатором'
    '3 транслятором'
    '4 редактором')
  TabOrder = 8
  OnClick = RadioGroup7Click
end
object RadioGroup8: TRadioGroup
  Left = -247
  Top = 928
  Width = 385
  Height = 89
  Caption = '8. Переменная типа "char" способна хранить:'
  Items.Strings = (
    '1 один символ '
    '2 два символа '
    '3 три символа '
    '4 до 255 символов')
  TabOrder = 9
  OnClick = RadioGroup8Click
end
object RadioGroup9: TRadioGroup
  Left = -247
  Top = 1032
  Width = 425
  Height = 89
  Caption = '9. Какая процедура обеспечивает ввод данных в Паскале ? '
  Items.Strings = (
    '1 begin '
    '2 input '
    '3 read или readln '
    '4 print ')
  TabOrder = 10
  OnClick = RadioGroup9Click

```

```

end
object RadioGroup10: TRadioGroup
  Left = -247
  Top = 1136
  Width = 449
  Height = 97
  Caption = '10. Чем отличается readln от read ? '
  Items.Strings = (
    ' ничем '
    '2 переводом курсора в readln к началу следующей строки '
    '3 в readln все данные набираются в одной строке '
    '4 в read все данные набираются в отдельной строке ')
  TabOrder = 11
  OnClick = RadioGroup10Click
end
object RadioGroup11: TRadioGroup
  Left = -247
  Top = 1248
  Width = 473
  Height = 89
  Caption =
    '11. Какая процедура выводит числовые данные, символы, строки и б' +
    'улевские значения? '
  Items.Strings = (
    '1 write или writeln '
    '2 print '
    '3 read или readln '
    '4 output ')
  TabOrder = 12
  OnClick = RadioGroup11Click
end
object RadioGroup12: TRadioGroup
  Left = -247
  Top = 1352
  Width = 449
  Height = 89
  Caption = '12. Символ-разделитель операторов в Паскале.'
  Items.Strings = (
    '1 точка '
    '2 точка с запятой '
    '3 запятая '
    '4 пробел ')

```

```

TabOrder = 13
OnClick = RadioGroup12Click
end
object RadioGroup13: TRadioGroup
Left = -247
Top = 1456
Width = 561
Height = 89
Caption =
'13. В каких случаях применяется оператор безусловного перехода g' +
'oto: "перейти к"? '
Items.Strings = (
'1 если нужно продолжить работу программы дальше '
'2 если нужно выйти из цикла '
'3 когда после выполнения оператора надо выполнить не следующую
щй п' +
'о порядку, а какой-либо другой'
'4 если в программе использовался оператор if')
TabOrder = 14
OnClick = RadioGroup13Click
end
object RadioGroup14: TRadioGroup
Left = -247
Top = 1560
Width = 449
Height = 89
Caption = '14. Какое по типу выражение вычисляется в условии оператора if? '
Items.Strings = (
'1 числовое выражение'
'2 неравенство '
'3 равенство '
'4 логическое выражение')
TabOrder = 15
OnClick = RadioGroup14Click
end
object Memo1: TMemo
Left = 305
Top = 296
Width = 153
Height = 137
Lines.Strings = (

```



```

    'Выбирать вариант ответа'
    'можно только один раз.'
    'Если Вы ошиблись, то'
    'после нажатия на ОК'
    'щелкните по выбранному '
    'варианту еще раз.')
```

ReadOnly = True
TabOrder = 16
end
end

Текст программы:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    RadioGroup1: TRadioGroup;
    RadioGroup2: TRadioGroup;
    Label1: TLabel;
    Button1: TButton;
    Label2: TLabel;
    Edit1: TEdit;
    RadioGroup3: TRadioGroup;
    RadioGroup4: TRadioGroup;
    RadioGroup5: TRadioGroup;
    RadioGroup6: TRadioGroup;
    RadioGroup7: TRadioGroup;
    RadioGroup8: TRadioGroup;
    RadioGroup9: TRadioGroup;
    RadioGroup10: TRadioGroup;
    RadioGroup11: TRadioGroup;
    RadioGroup12: TRadioGroup;
    RadioGroup13: TRadioGroup;
    RadioGroup14: TRadioGroup;
    Memo1: TMemo;
    procedure RadioGroup1Click(Sender: TObject);
    procedure RadioGroup2Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure RadioGroup3Click(Sender: TObject);
```

```

procedure RadioGroup4Click(Sender: TObject);
procedure RadioGroup5Click(Sender: TObject);
procedure RadioGroup6Click(Sender: TObject);
procedure RadioGroup7Click(Sender: TObject);
procedure RadioGroup8Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure RadioGroup9Click(Sender: TObject);
procedure RadioGroup10Click(Sender: TObject);
procedure RadioGroup11Click(Sender: TObject);
procedure RadioGroup12Click(Sender: TObject);
procedure RadioGroup13Click(Sender: TObject);
procedure RadioGroup14Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
  a,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14:integer;
  m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14:integer;
  s2,s3,s4:string;
  k:real;
implementation
{$R *.DFM}
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
if Form1.RadioGroup1.ItemIndex=0 then a1:=1;
m1:=m1+1;
if (m1>1) and (a1=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a1:=0;
m1:=0;
end;
end;
procedure TForm1.RadioGroup2Click(Sender: TObject);
begin
if Form1.RadioGroup2.ItemIndex=2 then a2:=1;
m2:=m2+1;
if (m2>1) and (a2=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a2:=0;

```

```

m2:=0;
end;
end;
procedure TForm1.Edit1Change(Sender: TObject);
begin
if Edit1.Text<>" then Button1.Enabled:=true;
end;
procedure TForm1.RadioGroup3Click(Sender: TObject);
begin
if Form1.RadioGroup3.ItemIndex=1 then a3:=1;
m3:=m3+1;
if (m3>1) and (a3=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a3:=0;
m3:=0;
end;
end;
procedure TForm1.RadioGroup4Click(Sender: TObject);
begin
if Form1.RadioGroup4.ItemIndex=1 then a4:=1;
m4:=m4+1;
if (m4>1) and (a4=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a4:=0;
m4:=0;
end;
end;
procedure TForm1.RadioGroup5Click(Sender: TObject);
begin
if Form1.RadioGroup5.ItemIndex=2 then a5:=1;
m5:=m5+1;
if (m5>1) and (a5=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a5:=0;
m5:=0;
end;
end;
procedure TForm1.RadioGroup6Click(Sender: TObject);
begin
if Form1.RadioGroup6.ItemIndex=3 then a6:=1;
m6:=m6+1;
if (m6>1) and (a6=1) then begin

```

```

ShowMessage('Выбирать вариант можно только один раз!');
a6:=0;
m6:=0;
end;
end;
procedure TForm1.RadioGroup7Click(Sender: TObject);
begin
if Form1.RadioGroup7.ItemIndex=0 then a7:=1;
m7:=m7+1;
if (m7>1) and (a7=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a7:=0;
m7:=0;
end;
end;
procedure TForm1.RadioGroup8Click(Sender: TObject);
begin
if Form1.RadioGroup8.ItemIndex=0 then a8:=1;
m8:=m8+1;
if (m8>1) and (a8=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a8:=0;
m8:=0;
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
a:=a1+a2+a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14;
Str(a,s2);
k:=(a/14)*100;
if k<=50 then s4:='2';
if (k>50) and (k<=75) then s4:='3';
if (k>75) and (k<=95) then s4:='4';
if k>95 then s4:='5';
Str(k:3:1,s3);
Label1.Caption:='Количество правильных ответов: '+s2+#13+'Ваш уровень
знаний: '+s3+'%'+#13+'Ваша оценка: '+s4;
end;
procedure TForm1.RadioGroup9Click(Sender: TObject);
begin
if Form1.RadioGroup9.ItemIndex=2 then a9:=1;
m9:=m9+1;

```

```

if (m9>1) and (a9=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a9:=0;
m9:=0;
end;
end;
procedure TForm1.RadioGroup10Click(Sender: TObject);
begin
if Form1.RadioGroup10.ItemIndex=1 then a10:=1;
m10:=m10+1;
if (m10>1) and (a10=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a10:=0;
m10:=0;
end;
end;
procedure TForm1.RadioGroup11Click(Sender: TObject);
begin
if Form1.RadioGroup11.ItemIndex=0 then a11:=1;
m1:=m1+1;
if (m11>1) and (a11=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a11:=0;
m11:=0;
end;
end;
procedure TForm1.RadioGroup12Click(Sender: TObject);
begin
if Form1.RadioGroup12.ItemIndex=1 then a12:=1;
m12:=m12+1;
if (m12>1) and (a12=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a12:=0;
m12:=0;
end;
end;
procedure TForm1.RadioGroup13Click(Sender: TObject);
begin
if Form1.RadioGroup13.ItemIndex=2 then a13:=1;
m13:=m13+1;
if (m13>1) and (a13=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');

```

```

a13:=0;
m13:=0;
end;
end;
procedure TForm1.RadioGroup14Click(Sender: TObject);
begin
if Form1.RadioGroup14.ItemIndex=3 then a14:=1;
m14:=m14+1;
if (m14>1) and (a14=1) then begin
ShowMessage('Выбирать вариант можно только один раз!');
a14:=0;
m14:=0;
end;
end;
end.

```

Лабораторная работа № 3 Файлы

Пример 1. Создание поля текстового редактора, в который можно загрузить файл, отредактировать его и сохранить.

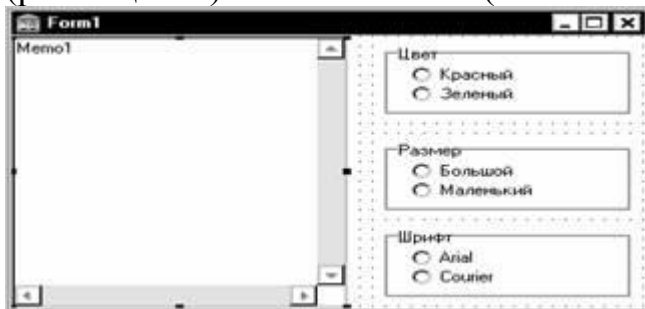
1. Вызвать текстовый редактор Блокнот и создать в нем текстовый файл My_text.txt с содержанием:

Button	кнопка
RadioButton	переключатель
Label	надпись
Edit	строка текста
Memo	редактор текста

2. Сохранить файл в папку MEMO.

3. Начать новый проект и сразу сохранить его в папке Мемо.

4. Поместить компонент Мемо в форму и установить для свойства ScrollBars (линейки прокрутки) значение ssBorth, а для свойства Align (размещение) значение alLeft (левая часть формы).



5. Записать код для загрузки файла при создании формы:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
```

```
    Memo1.Lines.LoadFromFile('My_text.txt');
```

6. Записать код, позволяющий сохранить файл при закрытии формы:

```
procedure TForm1.FormClose...
```

```
begin
```

```
    Memo1.Lines.SaveToFile('My_text.txt');
```

7. Запустить программу. Добавить что-нибудь в появившийся текст. Закрыть программу.

8. Запустить ее еще раз и убедиться, что загружается откорректированный текст.

9. Добавить группу переключателей (RadioButton) для выбора цвета шрифта и записать соответствующий код.

10. Добавить группу переключателей для выбора размера шрифта.

11. Добавить группу переключателей для выбора вида шрифта.

Пример 2. Создание простого текстового редактора.

1. Поместить компонент Bevel (на странице Additional) на новую форму и присвойте его свойству Align значение alTop.

2. На компонент Bevel поместить четыре кнопки. Озаглавьте их &Open..., &Save, Save &As... и E&xit.

Многоточие в конце имени представляет собой секретный код Windows, означающий, что щелчок на кнопке вызовет еще один диалог, а не приведет к немедленному выполнению некоторого действия.

3. Поместите на форму компонент Мемо и установите следующим образом его свойства:

<i>Свойство</i>	<i>Значение</i>
Align	alClient
ScrollBars	ssBoth
Lines	(пусто)

4. Изменить заголовок формы на My Editor. Затем поместите на форму компонент OpenFileDialog и присвойте его свойствам следующие значения:

<i>Свойство</i>	<i>Значение</i>
Default Ext	TXT
Options.ofHideReadOnly	True
Options.ofFileMustExist	True
Options.ofNoReadOnlyReturn	True
Title	Open Text File

5. Для того чтобы открыть окно Filter Editor (Редактор фильтра), дважды щелкните на свойстве Filter.

6. Заполнить две первые строки в колонке Filter Name (Имя фильтра) названиями фильтров Text files и All files, а в колонке Filter (Фильтр) ввести соответствующие фильтры: *.TXT и *.*.

7. Поместить на форму компонент SaveDialog и установить для его свойств такие значения:

<i>Свойство</i>	<i>Значение</i>
Default Ext	TXT
Options.ofHideReadOnly	True
Options.ofOverwritePrompt	True
Options.ofNoReadOnlyReturn	True
Title	Save Text File

8. Ввести тот же набор фильтров, что и для диалога открытия файла.

9. Создать обработчик события OnClick для кнопки Exit, содержащий одну команду Close;

10. Для обработчика события OnClick кнопки Open используйте следующие строки:

```
WITH OpenFileDialog1 DO
  IF Execute THEN
    BEGIN
      Memo1.Lines.LoadFromFile(Filename);
      HistoryList.Add(Filename);
      Caption:='My Editor-'+
ExtractFilename(Filename);
      SaveDialog1.Filename:=Filename;
      Filename:='';
    End;
```

11. Дважды щелкнуть на кнопке Save и ввести следующую строку в качестве ее обработчика события OnClick:

```
Memo1.Lines.SaveToFile( SaveDialog1.Filename);
```

12. Наконец, для кнопки Save As укажите следующее:

```
WITH SaveDialog1 DO
  IF Execute THEN
    BEGIN
      Memo1.Lines.SaveToFile(Filename);
      Caption:='My Editor - ' +
      ExtractFilename(Filename);
    End;
```

Этот текстовый редактор позволяет загружать, редактировать и сохранять небольшие и средние файлы. Возможности этого редактора приблизительно такие же, как и у приложения Notepad (Блокнот) Windows. Можно вырезать, копировать и вставлять текст, используя стандартные комбинации клавиш Windows. Имя каждого открытого файла добавляется в список

диалога открытия файла на случай, если вам потребуется открыть его снова. Сохраним проект. Назовем модуль MYEDITU.PAS, а главный проект — MYEDIT.PRJ.

Диалог выбора шрифта.

Загрузите опять проект MyEdit и поместите на форму компонент FontDialog. Переместите кнопку Exit вправо и вставьте кнопку с заголовком &Font между кнопками Save As и Exit. В обработчик события OnClick новой кнопки поместите следующий код:

```
WITH FontDialog1 DO
```

```
  IF Execute THEN
```

```
    BEGIN
```

```
      Memo1.Font:= Font;
```

Следующую строку поместите в обработчик события OnApply компонента FontDialog:

```
  Memo1.Font:= FontDialog1.Font;
```

Запустите программу и загрузите файл. С помощью кнопки Font выберите какой-нибудь шрифт. Используя все возможности, создайте самый необычный вариант. Содержимое файла будет отображаться выбранным вами шрифтом с применением указанного размера, стиля и цвета. Опять откройте диалог выбора шрифта и переместите его так, чтобы он не закрывал самого окна редактора. Внесите несколько изменений в окне диалога и щелкните на кнопке Apply (Применить). Изменения станут действительными немедленно, а диалог Font (Шрифт) не закроется. Это упрощает испытание различных возможностей. Обратите внимание, что в этом примере программы, после того как была использована кнопка Apply (Применить), вы никак не сможете отменить внесенные изменения. С другой стороны, это не такая уж и большая беда — если не устраивает только что выбранный шрифт, можно вернуть шрифт, использовавшийся раньше. Сохраните обновленный проект MyEdit, обладающий теперь возможностью изменять шрифты.

Компонент Memo не обладает способностью отображать больше одного шрифта сразу. Для этого существуют другие компоненты.

Добавим в свою программу диалог выбора цвета.

1. Поместите на форму диалог выбора цвета (компонент ColorDialog) и присвойте его свойству Color значение clWhite.
2. Установите свойство Ctl3D формы в False, а в качестве константы цвета формы также выберите clWhite.
3. Поместите на форму новую кнопку, назовите её Color.

В обработчик события OnClick формы поместите следующие строки:

```
IF ColorDialog1.Execute THEN
```

```
Form1.Color := ColorDialog1.Color;
```

4. Запустите программу.

Диалог выбора принтера.

Загрузите опять проект MyEdit и поместите на него диалог настройки принтера компонент PrinterSetupDialog1. Компонент Bevel расположите таким образом, чтобы осталось место для второго ряда кнопок, где разместите две новые кнопки, озаглавленные &Print... и P&rint Setup.... Создайте обработчик события OnClick кнопки Print Setup с единственной строкой кода:

```
PrinterSetupDialog1.Execute;
```

Запустите программу и щелкните на кнопке Print Setup (кнопка Print пока еще делать ничего не может). Появится диалог.

Подготовка к печати.

Диалог Print Setup (Настройка принтера) используется для выбора и конфигурации принтера. Когда дело дойдет до печати, в действие вступит диалог Print (Печать).

В проект MyEdit добавьте компонент PrintDialog, оставив все его свойства со значениями, установленными по умолчанию. Дважды щелкните на кнопке Print, и замените пару begin—end следующим кодом:

```
VAR  
  POutput: TextFile;  
  N      : LongInt;  
BEGIN  
  IF PrintDialog1.Execute THEN  
    BEGIN  
      AssignPrn (POutput);  
      Rewrite (POutput);  
      Printer.Canvas.Font: = Memo1.Font;  
      FOR N: = 0 to Memo1.Lines.Count - 1 DO  
        Writeln(POutput, Memo1.Lines [N]);  
        CloseFile (POutput);  
      END;  
    END;
```

Перейдите в верхнюю часть модуля и добавьте Printers в оператор uses. (Найдите строку в начале файла, начинающуюся со слова uses. После этого слова добавьте имя модуля и запятую.) Выберите на форме компонент FontDialog и присвойте его свойству Device значение fdBoth, чтобы ограничить выбор шрифтами, которые годятся и для экрана, и для принтера. Мы только что добавили в ваш редактор возможность печати. Этот редактор не позволяет печатать несколько копий или направлять их в подбор, но вы сможете распечатать содержимое редактора любым шрифтом. А с помощью диалога Print Setup (Выбор принтера) вы сможете выбрать принтер, ориентацию бумаги и параметры конфигурации, соответствующие вашему принтеру.

Диалог поиска и замены текста.

Загрузите проект MyEdit и установите свойство HideSelection компонента Мемо в False. На форме разместите компоненты FindDialog и ReplaceDialog и выберите их оба. Дважды щелкните на свойстве Options и установите frHideMatchCase, frHideWholeWord и frHideUpDown в True. Добавьте во второй ряд кнопки с заголовками Fi&nd... и &Replace... Для события OnClick кнопки Find используйте следующую строку:

```
FindDialog1.Execute;
```

а для кнопки Replace следующую строку:

```
ReplaceDialog1.Execute;
```

Выберите компонент FindDialog, перейдите на страницу Events и дважды щелкните на событии OnFind. Вставленную Delphi пару begin—end замените следующим блоком кода:

```
VAR
  Buff, P, FT: Pchar;
  BuffLen : Word;
begin
  WITH Sender AS TFindDialog DO
  BEGIN
    GetMem (FT, Length (FindText) + 1);
    StrPCopy (FT, FindText);
    BuffLen := Memo1.GetTextLen + 1;
    GetMem (Buff, BuffLen);
    Memo1.GetTextBuff (Buff, BuffLen);
    P := Buff + Memo1.SelStart + Memo1.SelLength;
    P := StrPos (P, FT);
    IF P = NIL THEN MessageBeep (0)
    ELSE
      BEGIN
        Memo1.SelStart := P-Buff;
        Memo1.SelLength := Length (FindText);
      END;
    FreeMem (FT, Length (FindText) + 1);
    FreeMem (Buff, BuffLen);
  END;
end;
```

Когда закончите, щелкните на компоненте ReplaceDialog, перейдите на страницу Events в окне Object Inspector и щелкните на кнопке раскрытия списка обработчика события OnFind. Из появившегося списка выберите только что созданный обработчик события для FindDialog. Теперь создайте обработчик события OnReplace со следующими строками:

```
WITH Sender AS TReplaceDialog DO
```

```

WHILE True DO
BEGIN
  IF Memo1.SelText <> FindText THEN
  FindDialog1Find (Sender);
  IF Memo1.SelLength = 0 THEN Break;
  Memo1.SelText: = ReplaceText;
  IF NOT (frReplaceAll IN Options) THEN Break;
END;

```

Сохраните программу, запустите ее и загрузите текстовый файл. Замените в нем все точки на восклицательные знаки и сохраните его.

Задание.

1. Создайте контекстно-зависимое меню, состоящее из пунктов: Шрифт, Поиск, Замена.
2. Создайте всплывающие подсказки для объектов формы.

Лабораторная работа № 4 Мультимедиа

Пример 1. Создадим "Лазерный проигрыватель", который будет проигрывать CD, опознавать CD, поддерживать специальную базу данных описаний всех CD.

1.1. Начните новый проект.

1.2. Установите свойства формы:

Свойство	Значение
BorderIcons.biMaximize	False
BorderStyle	bsSingle
Caption	CD-проигрыватель
Height	350
Name	MPForm
Width	370

1.3. Разместите на форме (почти под самым заголовком) компонент MediaPlayer (закладка System палитры компонентов).

1.4. Установите следующие свойства компонента Media Player:

Свойство	Значение
AutoOpen	True
DeviceType	dtCDAudio
Name	Player

1.5. Дважды щелкните по имени свойства VisibleButtons и установите в False значение следующих подсвойств: btNext, btPrev, btStep, btBack, btRecord. Тем самым вы уберете соответствующие кнопки в компоненте Media Player, размещенном на форме. Что касается кнопок "Следующий",

"Предыдущий", "Вперед" и "Назад", то они нужны только на панели классического CD-проигрывателя, но, используя Delphi, можно организовать быстрый доступ к нужной песне и более эффективным способом.

1.6. Дважды щелкните по имени свойства ColoredButtons и установите в False значения подсвойств btPause и btEject.

1.7. Сохраните проект на диске в отдельном каталоге, присвоив для модуля (PAS-файл) имя MainForm, а для файла проекта (DPR-файл) имя Player (или любое другое, которое, по вашему мнению, больше подходит для CD-плеера).

1.8. Возле правого края формы разместите кнопку (компонент Button страницы Standard). В ее свойстве Caption введите значение: &Закреть.

1.9. Щелкните по компоненту Media Player, а затем, нажав Shift, по кнопке, выбрав сразу два компонента.

1.10. Выберите пункт меню Edit > Align... и в окне Alignment, в разделе Vertical установите радиокнопку Tops. Щелкнув ОК, вы добьетесь выравнивания кнопки по верхнему краю.

1.11. Не отменяя выделения обоих компонентов, выберите пункт меню Edit > Size... и в разделе Height появившегося окна установите флажок Shrink to smallest. Нажав ОК, вы установите высоту панели компонента Media Player равной высоте кнопки (ради удовлетворения собственных эстетических представлений, "советующих" хоть как-то выравнивать компоненты друг относительно друга).

1.12. Дважды щелкните по кнопке "Закреть" на главной форме проекта и введите Close;

1.13. Нажмите F12 (чтобы вернуться к шаблону формы). Щелкните по вкладке Win32 палитры компонентов VCL, выберите компонент ProgressBar и разместите полосу индикатора под панелью проигрывателя, растянув ее почти во всю ширину формы (если что, сверьтесь с рисунком). Свойству Name присвойте значение ProgressBar.

1.14. Под компонентом ProgressBar разместите две метки (компонент Label вкладки Standard) - одна под другой. Первой дайте имя (свойство Name) Author, а второй - Album. Эти метки будут отображать информацию о названии диска и его авторе (или исполнителе).

Выделите обе метки и очистите свойство Caption - Delphi позволяет устанавливать общее свойство для нескольких компонентов, а свойство Caption имеется у обеих меток.

1.15. Откройте вкладку Win32 палитры компонентов, найдите на ней компонент Status Bar и дважды щелкните по нему. Delphi автоматически разместила строку статуса в нижней части формы. Ее свойству Name присвойте значение Status.

1.16. Строку статуса можно использовать для вывода какой-нибудь информации, поясняющей процесс работы программы. Например, выводить

информацию о номере проигрываемого трека, времени звучания, а также название композиции. Дважды щелкните по компоненту Status Bar, чтобы открыть диалоговую панель для манипуляций с этим компонентом (заголовок этого окна сейчас: Editing Status.Panels). Теперь три раза щелкните по кнопке "Add New" - в специальном окне этой панели появилось три строки: "0 - TStatusPanel", "1 - TStatusPanel" и "2 - TStatusPanel". Щелкните по строке "0 - TstatusPanel" и ее свойству Width присвойте значение 25. Для второй строки точно также задайте ширину в 40 пикселей. Можете закрыть окно Editing Status.Panels.

1.17. Теперь выберите компонент StringGrid - сетка строк (закладка Additional).

Разместите его на форме только что размещенными метками и установите значения для его следующих свойств:

Свойство	Значение
ColCount	3
Color	clInfoBk
DefaultRowHeight	18
FixedCols	0
Name	Songs
RowCount	10
ScrollBars	ssVertical

Тем самым вы установили трехколоночную сетку из 10 строк цвета стандартной контекстной подсказки Windows (clInfoBk). Высота строк равна 18 пикселей, ни одной колонки не зафиксировано, разрешен только вертикальный ползунок.

1.18. Измените ширину списка строк так, чтобы она равнялась ширине компонента ProgressBar (см. п. 1.13), а высоту - так, чтобы были видны все десять строк. Теперь измените ширину колонок, используя разделительные линии на первой фиксированной строке (при помощи технологии Drag&Drop).

После того, как вы добились нужной ширины колонок, присвойте свойству FixedRows значение 0 (ноль) - верхняя фиксированная строка превратилась в обычную полосу (если вы позднее захотите изменить ширину колонок, снова установите это свойство в 1 и выполните точную подгонку, а затем снова верните ноль).

1.19. Дважды щелкните по свойству Options компонента StringGrid и установите в False подсвойства goVertLine и goRangeSelect (отмена вертикальных разделяющих линий в списке строк и запрет одновременного выбора нескольких строк).

Подсвойству goRowSelect присвойте значение True. Это подсвойство отвечает за внешний вид активной строки списка. В данном случае выделяться будут все три колонки сразу.

1.20. И наконец, откройте вкладку System палитры компонентов и разместите в любом месте формы (симпатичнее всего - на списке строк) два компонента Timer. Они автоматически получат имена, соответственно: Timer1 и Timer2, которые можно оставить без изменения.

Свойство Interval (частота "тиканья" в миллисекундах) обоих компонентов сделайте равным 100. Для компонента Timer2 установите свойство Enabled в False, чтобы сделать его "не тикающим" в начале выполнения программы.

1.21. Сохраните проект на диске (на панели Delphi есть две кнопки сохранения - Save file и Save All. Первая сохраняет текущую форму, а вторая - проект в целом, то есть все редактируемые формы проекта).

Запуск проекта.

Итак, у нас имеется заготовка формы с девятью размещенными на ней компонентами: MediaPlayer, Button, два компонента Label, ProgressBar, StringGrid, StatusBar и два компонента Timer. А теперь нужно заставить их как-то взаимодействовать друг с другом.

2.1. Нажмите F12, чтобы перейти в окно Редактора кода (Code Editor).

2.2. Переместитесь в начало модуля и добавьте в секцию импортируемых модулей (uses) новый модуль: MMSystem.

2.3. Переместитесь к описанию формы и вставьте после служебного слова type перед описанием формы (класс TMPForm) следующую строку:

```
TPlayerStatus = (NoDisk, InsertingDisk, Played, Paused, Stopped);
```

Компонент MediaPlayer имеет свойство Mode, которое показывает его текущее состояние. Однако, проверяя это свойство, невозможно поймать тот момент, когда диск только-только вставлен в CD-ROM - это свойство фиксирует либо отсутствие диска, либо его присутствие (и одно из нескольких состояний: не проигрывается, проигрывается и т. п.).

А нам нужно точно поймать момент, когда диск вставлен, чтобы начать его сканирование и проверку: есть ли его описание в базе или это еще неизвестный программе диск. Вот для этого и понадобился собственный набор возможных состояний CD-проигрывателя.

2.4. В секции private описания формы вместо комментария { private declarations } введите строки кода:

```
private  
Trk, Min, Sec: word; // Трек и его длительность (мин., сек.)  
CD: TStringList; // Описание дисков  
CurStatus: TPlayerStatus; // Текущий статус CD-ROM  
procedure ClearSongsList;  
procedure EmptyIndicat;  
procedure InitProgressBar;
```

2.5. Перейдите в секцию реализации модуля, найдите строку и введите под директивой компилятору {\$R *.DFM} текст кода:

```

procedure TMPForm.ClearSongsList; {Очистить список строк}
begin
with Songs do
begin
RowCount:=1; Cells[0,0]:= ""; Cells[1,0]:= ""; Cells[2,0]:= "";
end;
Author.Caption:= ""; Album.Caption:= "";
end;
procedure TMPForm.EmptyIndicat; {Очистить строку статуса}
begin
Status.Panels [0].Text:= '00'; Status.Panels [1].Text:= '00:00';
Status.Panels [2].Text:= "";
ProgressBar.Position:= 0; ProgressBar.Max:= 100;
end;
procedure TMPForm.InitProgressBar; {Настройка индикатора звучания
трека}
var Len: longint;
begin
with Player, ProgressBar do
if ( StartPos > 0 ) AND ( StartPos <= Tracks ) then
begin
Len:= TrackLength [ StartPos ]; Position:= 0;
Max:= MCI_MSF_MINUTE ( Len ) * 60 + MCI_MSF_SECOND ( Len );
// Вывод названия песни в строке статуса
Status.Panels [ 2 ].Text:= Songs.Cells [ 2, StartPos - 1 ];
end;
end;

```

Процедуры ClearSongsList и EmptyIndicat требуются лишь для того, чтобы в момент замены диска очистить текстовые свойства компонентов, несущие информацию о текущем диске, наборе его треков, названии каждого трека, информацию о текущей исполняющейся композиции и пр.

StartPos, Tracks и TrackLength - свойства компонента Media Player (в данном проекте он называется Player), хранящие информацию соответственно о: текущем треке, количестве треков на диске и длине указанного трека.

Длина трека в свойстве TrackLength хранится в так называемом упакованном формате MSF (минуты, секунды, фреймы).

У компонента Media Player еще имеется свойство TimeFormat, которое устанавливает формат, описывающий каждый трек CD.

Чтобы распаковать этот формат и извлечь из него количество минут и секунд, надо воспользоваться функциями MCI_MSF_MINUTE и MCI_MSF_SECOND соответственно, описанными в модуле MMSystem (см. 2.2). Используя эти две функции, значение свойства Max компонента

ProgressBar устанавливается равным количеству секунд звучания текущего трека.

2.6. Компонент Timer обеспечивает возможность периодического выполнения какого-то однотипного действия с указанной в его свойстве Interval частотой. Таймер можно использовать для контроля текущего состояния CD-ROM (через повторяющееся с частотой 0,01 сек. обращение к компоненту Media Player).

Для упрощения кода лучше разделить две операции: проверку текущего состояния Media Player и вывод информации о текущем треке (индикатор и счетчик секунд в реале будут показывать текущее состояние выполняющейся композиции).

Дважды щелкните по компоненту Timer2 - он будет отвечать за индикацию состояния композиции, воспроизводящейся в данный момент - и введите между строками begin - end код, приведенный ниже.

```
with Player, Status do
begin
Trk:= MCI_TMSF_TRACK ( Position );
Min:= MCI_TMSF_MINUTE ( Position );
Sec:= MCI_TMSF_SECOND ( Position );
// Закончилась одна песня и началась другая?
if ( Min = 0 ) AND ( Sec = 0 ) AND ( Mode = mpPlaying ) then
begin InitProgressBar; Songs.Row:= Trk - 1; end;
ProgressBar.Position:= Min * 60 + Sec;
// Информация в панели состояния
Panels [ 0 ].Text:= Format ( '%.2d', [ Trk ] );
Panels [ 1 ].Text:= Format ( '%.2d:%.2d', [ Min, Sec ] );
end;
```

Свойство Position компонента Media Player хранит информацию о текущем звуке в несколько ином формате - TMSF (трек, минуты, секунды, фреймы) и для ее извлечения понадобились функции, аналогичные тем, что были использованы в процедуре InitProgressBar (см. 2.5).

Доступ к отдельным фрагментам панели состояния (StatusBar) осуществляется через массив Panels, одно из полей которого (Text) хранит отображаемую на панели состояния информацию.

2.7. Теперь опишем метод, который позволит загружать информацию о композициях CD (о его треках). В описании формы введите под строкой public объявление следующего метода: procedure AddSongsToList (Sng: TStringGrid);

```
Перейдите в конец модуля и перед финальной end. (с точкой) введите код:
procedure TForm.AddSongsToList ( Sng: TStringGrid );
var I: longint; M, S: word;
begin
```

```

with Sng do
begin
// Установить количество строк в Sng равным количеству треков CD
RowCount:= Player.Tracks;
// Для каждого трека:
for I:= 1 to Player.Tracks do
begin
// Получить число минут
M:= MCI_MSF_MINUTE ( Player.TrackLength [ I ] );
// Получить число секунд
S:= MCI_MSF_SECOND ( Player.TrackLength [ I ] );
// Первая колонка - номер трека
Cells [ 0, I - 1 ]:= Format ( '%.2d', [ I ] );
// Вторая - время звучания
Cells [ 1, I - 1 ]:= Format ( '%.2d:%.2d', [ M, S ] );
// Третья пока пустая
Cells [ 2, I - 1 ]:= "";
end;
// Позиционировать указатель на первую строку
Row:= 0;
end;
end;

```

2.8. А теперь потребуется создать механизм проверки каждого вновь вставленного в CD-ROM диска.

Поскольку опознавать диск придется при помощи таймера, то во избежание зависания при слишком долгой проверке, лучше всего воспользоваться механизмом сообщений Windows.

Переместитесь в верхнюю часть модуля и после секции uses (перед словом type) введите следующий текст:

```

const
WM_SCANCD = WM_USER + 1;
WM_UNKNOWNC = WM_USER + 2;

```

Таким способом вы объявили два внутренних сообщения: сообщение о начале сканирования диска и сообщение, которое программа пошлет системе в том случае, если просмотренный диск окажется ей неизвестным.

2.9. Теперь перейдите к описанию формы и в секции private после описания трех служебных методов (см. 2.4) введите объявление для двух обработчиков выше объявленных сообщений:

```

procedure ScanCD ( var Msg: TMessage ); message WM_SCANCD;
procedure UnknownCD ( var Msg: TMessage ); message WM_UNKNOWNC;

```

2.10. Теперь добавьте в конец секции реализации модуля текст:

```

procedure TMPForm.ScanCD ( var Msg: TMessage );

```

```
begin end;
procedure TMPForm.UnknownCD ( var Msg: TMessage );
begin end;
```

Самым верным решением будет сканирование CD в тот момент, когда он только вставлен в устройство чтения. Для этого надо осуществлять периодический опрос текущего состояния CD-ROM. И тут нам на помощь приходит таймер. Поскольку компонент Timer2 уже занят (см. 2.6), используем Timer1.

2.11. Нажмите F12, чтобы перейти к шаблону формы и дважды щелкните по размещенному на ней компоненту Timer1. В шаблоне обработчика между парой begin - end поместите код:

```
// Определить текущее состояние CD-ROM
with Player, Status do
case Mode of
mpOpen: CurStatus:= NoDisk; // Открыт (нет диска)
mpStopped: if CurStatus = NoDisk then // Диск есть, но не играет
CurStatus:= InsertingDisk
else if CurStatus <> Paused then
CurStatus:= Stopped;
mpPlaying: begin // Играет
CurStatus:= Played;
// Если список строк еще не заполнен
if Songs.RowCount = 1 then
AddSongsToList ( Songs );
end;
end;
// Активизировать второй таймер, если CD находится в устройстве чтения
Timer2.Enabled:= CurStatus <> NoDisk;
// Выполнить действия для каждого из текущих состояний диска
case CurStatus of
NoDisk: begin
ClearSongsList;
EmptyIndicat;
end;
InsertingDisk: begin
AddSongsToList ( Songs );
// Послать сообщение о начале сканирования
SendMessage ( Handle, WM_SCANCD, 0, 0 );
end;
end;
```

Первоначально проверяется свойство Mode компонента Player (CD-проигрыватель), и в зависимости от его значения устанавливается значения переменной CurStatus.

Если оказывается, что диск находится в устройстве чтения, а устройство готово (состояние mpStopped), то проверяется предыдущее значение CurStatus. Если оно равно NoDisk, то программа приходит к очень логичному выводу, что диск только что вставлен и устанавливает CurStatus в InsertingDisk. Вот оно - мы поймали момент, когда диск только вставлен. В зависимости от текущего состояния CD-ROM выполняются те или иные действия. Если диск только-только вставлен, ядру ОС посылается сообщение WM_SCANCD - о начале проверки диска на предмет поиска информации о нем.

Сообщение посылается при помощи стандартной функции SendMessage. Первым параметром этой функции является параметр Handle. Это ссылка на экземпляр программы, пославшей сообщение, которая очень нужна Windows, чтобы точно знать, кому переслать ответное сообщение в случае, коли в том будет нужда.

Второй параметр - само сообщение; третий и четвертый параметры могли бы нести какую-нибудь информацию (и эту информацию можно было бы получить в обработчике, проанализировав структуру TMessage), поэтому помещаем туда нули.

Теперь сохраните проект и запустите его. Несмотря на то, что мы еще не довели дело до конца, программа уже вполне работоспособна: как только вы вставите аудиодиск в приемник CD-ROM, сетка строк программы сразу же заполнится информацией о композициях: их количестве и времени звучания каждой.

Кнопками можно запустить воспроизведение, выключить или выбросить диск из CD-ROM и, что характерно, в строке статуса будет отображаться информация о композиции, которая воспроизводится в текущий момент.

2.12. К нашему проекту необходимо добавить еще одну форму, которая будет использоваться для ввода информации о диске.

2.12.1. Выберите пункт File > New Form (или щелкните соответствующую кнопку на toolbar) и сохраните новый модуль на диске под именем CDdescript.pas.

2.12.2. Для новой формы установите следующие свойства:

BorderIcons.byMinimize = False,

BorderIcons.byMaximize = False,

BorderStyle = bsDialog,

Caption = Описание диска,

Height = 305,

Name = DescriptionForm,

Width = 376.

2.12.3. Разместите на форме одну под другой три метки (компонент Label, страница Standard), назвав их, соответственно, (изменив их свойства Caption): "&Автор", "&Название" и "&Композиции" .

2.12.4. Справа от первых двух меток разместите два компонента Edit (на той же вкладке), свойству Name первого присвойте значение CDAuthor, а второго - CDName и очистите у обоих свойство Text.

2.12.5. Под меткой "Композиции" разместите компонент StringGrid (найдя его на странице Additional). Присвойте следующие значения его свойствам:

```
ColCount = 3,  
Color = clInfoBk,  
DefaultRowHeight = 18,  
FixedCols = 2,  
FixedRows = 0,  
Name = List,  
Options.goFixedVertLine = False,  
Options.goVertLine = False,  
Options.goRangeSelect = False,  
Options.goDrawFocusSelect = True,  
Options.goEditing = True,  
RowCount= 1,  
ScrollBars = ssVertical.
```

Прежде чем устанавливать свойство FixedRows, измените ширину колонок (см. п. 1.18). Свойство Options.goEditing устанавливает в True для того, чтобы иметь возможность непосредственно редактировать строку сетки. Поскольку свойство FixedCols равно 2, что делает две левых колонки недоступными, то редактировать можно будет лишь колонку названий композиций.

2.12.6. У меток "Автор", "Название" и "Композиции" свойству FocusControl присвойте значения (выбрав из раскрывающегося списка), соответственно: CDAuthor, CDName и List. Если эти свойства не установить - на работоспособности программы это не отразится, но будет потеряна возможность выбора строк редактирования при помощи клавиатурных эквивалентов (Alt+A и пр.).

Расширьте компоненты Edit, чтобы выровнять их по правому краю сетки строк.

2.12.7. Под списком строк разместите компонент Panel (страница Standard), очистите ее свойство Caption и установите следующие свойства (превратив, таким образом, в разделяющую бороздку):

```
BevelInner = bvRaised  
BevelOuter = bvLowered  
Height = 3
```

Установите ширину Panel равной ширине сетки строк (см. пункты 1.10, 1.11).

2.12.8. Под "бороздкой" разместите две кнопки (компонент Button страницы Standard). Свойству Caption левой кнопки присвойте значение "&Сохранить", второй - "&Отмена". Для свойства ModalResult кнопки "Сохранить" из раскрывающегося списка выберите значение mrOk, для второй - mrCancel. Наконец, свойство Cancel кнопки "Отмена" установите в True (теперь отменить действие можно будет и нажатием на Esc).

2.12.9. Щелкните один раз по любому участку форму, не занятому компонентами. В Инспекторе объектов (Object Inspector) перейдите к закладке Events и дважды щелкните по свойству OnActivate (только щелкать надо по пустой строке ввода). В шаблон созданного обработчика введите между строками begin и end текст :

```
CDAuthor.Text:= ""; CDName.Text:= ""; List.RowCount:= 1;  
List.Cells [0,0]:= ""; List.Cells [1,0]:= ""; List.Cells [2,0]:= "  
// Заполнить сетку строк информацией о треках  
MPForm.AddSongsToList ( List );
```

2.12.10. Переместитесь к строке {\$R *.DFM} (директива компилятору для подключения внешнего описания формы) и под ней на новой строке введите:

```
uses MainForm;
```

2.12.11. Щелкните по закладке MainForm Редактора Кода (Code Editor).

Здесь также найдите строку {\$R *.DFM} и под ней введите:

```
uses CDdescript;
```

Таким способом вы осуществили взаимную ссылку одного модуля на другой (а того, в свою очередь, на первый). Но делать это можно лишь в секции реализации модуля (под строкой {\$R *.DFM}).

Теперь о методе AddSongsToList главной формы MPForm. Дело в том, что этот метод нам понадобился для заполнения сетки строк второй формы (см.2.12.9). Если бы его описание было помещено в секции private главной формы, то вторая форма его бы не "увидела", ибо в эту секцию помещаются внутренние методы и свойства, недоступные извне.

Для этого, кстати, в модуле CDdescript и понадобилось подключение модуля MainForm. Обычно этого не требуется.

2.12.12. Выровняйте все компоненты второй формы, сверившись с рисунком и сохраните проект.

2.13. Идентифицировать CD (то есть отличить его от других) не так уж сложно. Можно сделать следующее утверждение: количество треков на диске и время звучания каждого из них (именно в данной последовательности) - величина постоянная и уникальна для каждого диска.

Другой вопрос - как именно хранить информацию о CD? Лучшим будет способ хранения в обычном ASCII-файле (текстовом файле) в виде после-

довательности строк - каждая новая строка описывает очередной CD. Чтобы программно можно было вычлнить из строки нужное информационное поле, все поля в строке должны отделяться одно от другого специальными символами разделителями. Для этого потребуется специальная функция, которая будет возвращать из общей строки ограниченную разделителями подстроку указанного номера.

В модуле MainForm переместитесь на самый верх к секции объявления констант (const) и под объявлением сообщений введите такой текст:

```
CH_TILDA = '~';  
CH_ZERO = '^';  
CH_VERT = '|';  
FILE_NAME = 'cd.txt';
```

Константа FILE_NAME - это имя дискового файла, в котором будет храниться информация о CD.

Точно также, кстати, можно ввести и другие символы-разделители (только выбирать нужно такие, которые не встретятся в названиях дисков и композиций).

2.14. Переместитесь в секцию реализации и сразу после строки uses CDdescript; введите код:

```
function FoundPos ( St, Dlm: string ): integer;  
var StPtr, DlmPtr: PChar;  
begin  
St:= St + #0;  
Dlm:= Dlm + #0;  
StPtr:= @St [ 1 ];  
DlmPtr:= @Dlm [ 1 ];  
DlmPtr:= StrPos ( StPtr, DlmPtr );  
Result:= DlmPtr - StPtr + 1;  
end;  
function GetSubString ( S, Delim: string; Num: integer ): string;  
var I, P1, P2: integer;  
begin  
Result:= ''; P1:= FoundPos ( S, Delim );  
if P1 > 0 then  
if Num = 1 then  
Result:= Copy ( S, 1, P1 - 1 )  
else  
begin  
// Найти левый обрамляющий символ  
I:= 1;  
while ( I <= Num - 1 ) AND ( P1 > 0 ) do  
begin
```

```

P1:= FoundPos ( S, Delim );
if P1 > 0 then S:= Copy ( S, P1 + 1, Length ( S ) );
Inc ( I );
end;
if P1 > 0 then
begin
// Найти правый обрамляющий символ
P2:= FoundPos ( S, Delim );
if P2 > 0 then S:= Copy ( S, 1, P2 - 1 );
if Length ( S ) > 0 then Result:= S
else Result:= "";
end
else Result:= "";
end
else
if Num = 1 then Result:= S
else Result:= "";
end;

```

Функция FoundPos носит вспомогательное значение и используется в GetSubString для поиска в строке позиции символа-разделителя. Функция GetSubString возвращает из строки S подстроку номер Num, обрамленную разделителем Delim.

2.15. Договоримся о структуре строки, хранящейся в файле. Пусть она состоит из пяти полей, разделенных символом CH_TILDA:

1. Общая длина диска (свойство Length компонента MediaPlayer).
2. Количество треков (свойство Tracks компонента MediaPlayer).
3. Название диска (присваивается пользователем).
4. Автор/исполнитель (присваивается пользователем).
5. Секция описания композиций.

В секции описания композиций одно описание от другого отделяется символом CH_ZERO и состоит из трех, разделенных символом CH_VERT полей:

- 1) длина трека (в формате "минуты:секунды");
- 2) название композиции;
- 3) зарезервировано (вдруг вы доработаете проект и захотите хранить еще какую-нибудь информацию о композиции, например, ее текст. Между прочим, почему бы нет?).

Теперь у нас имеется все, чтобы описать текст обработчиков сообщений WM_SCANCD и WM_UNKNOWNC. Найдите в модуле MainForm заглушки обработчиков (см. п. 2.10). Вместо строк begin - end метода TMPForm.ScanCD введите код:


```

var I, J, L, Len, TotSongs: integer; Found: boolean; S, S2, S3: string;
begin
Found:= false; I:= 0; L:= CD.Count - 1;
// Просмотреть все строки
while ( NOT Found ) AND ( I <= L ) do
begin
S:= CD [ I ]; Len:= StrToInt ( GetSubString ( S, CH_TILDA, 1 ) );
TotSongs:= StrToInt ( GetSubString ( S, CH_TILDA, 2 ) );
if ( Len = Player.Length ) AND ( TotSongs = Player.Tracks ) then
begin
S2:= GetSubString ( S, CH_TILDA, 5 ); Found:= true; J:= 1;
while Found AND ( J <= TotSongs ) do
begin
S3:= GetSubString ( S2, CH_ZERO, J );
Found:= ( Player.TrackLength [ J-1 ] =
StrToInt ( GetSubString ( S3, CH_VERT, 1 ) ) );
Inc ( J );
end;
end;
Inc ( I );
end;
if Found then
begin
Author.Caption:= GetSubString ( S, CH_TILDA, 3 );
Album.Caption:= GetSubString ( S, CH_TILDA, 4 );
ShowSongsName ( TotSongs, S2 );
end
else
if MessageDlg ( 'Неизвестный CD. Добавить описание CD в базу?',
mtConfirmation, [mbYes, mbNo], 0 ) = mrYes then
SendMessage ( Handle, WM_UNKNOWNCD, 0, 0 );
end;

```

В этом методе организован цикл WHILE DO по всем строкам, загруженным в список CD (см. п. 2.4). В каждой очередной строке получается информация (с помощью функции GetSubString) о длине CD и количестве композиций. Если она совпадает с той, что в данный момент хранится в компоненте MediaPlayer (Player), организуется еще один - внутренний цикл WHILE DO, в котором также с помощью GetSubString получается информация о времени звучания каждой композиции и сверяется со значением массива TrackLength, который хранит информацию о композициях текущего диска.

В конечном итоге - после всех этих циклов - переменная Found будет содержать True, если тестируемый диск с успехом выдержал испытание (в этом случае в главной форме выводится информация о нем - описание метода ShowSongsName см. п. 2.18) и False, если диск опознан, как неизвестный. В этом случае программа с помощью стандартной функции MessageDlg информирует пользователя о ситуации и спрашивает его решение добавить описание диска в базу.

2.16. Перейдите в главной форме к заглушке TMPForm.UnknownCD и введите вместо строк begin - end код:

```
begin
with DescriptionForm do
// Пользователь нажал ОК?
if ShowModal = mrOk then
begin
CD.Add ( DescriptionCD ( CDAuthor.Text, CDName.Text, List ) );
CD.SaveToFile ( FILE_NAME );
SendMessage ( Handle, WM_SCANCD, 0, 0 );
end;
end;
```

Метод ShowModal - стандартный метод любой формы, предназначенный для вывода ее на экран. Метод возвращает mrOk, если пользователь нажал кнопку Ok (кнопка, свойство которой ModalResult, равна mrOk - см. 2.12.8) в вызываемой форме.

Если это произошло, новое описание сохраняется в списке строк (CD.Add), затем весь список сохраняется в файле на диске (CD.SaveToFile) и вновь посылается сообщение (SendMessage) WM_SCANCD, в результате чего в главной форме будет выведена только что введенная информация. При формировании сохраняемой строки используется метод DescriptionCD.

2.17. Введите в разделе private описания главной формы следующие строки:

```
procedure ShowSongsName
( TotSongs: word; S: string );
function DescriptionCD
( Name, Album: string; Song: TStringGrid ): string;
Затем переместитесь к концу модуля и введите текст:
function TMPForm.DescriptionCD ( Name, Album: string; Song: TStringGrid ):
string;
var I: word;
begin
Result:= Format ( '%d', [Player.Length] ) + CH_TILDA;
Result:= Result + Format ( '%d', [Player.Tracks] ) + CH_TILDA;
Result:= Result + Name + CH_TILDA;
```

```

Result:= Result + Album + CH_TILDA;
with Song do
// Просмотреть все строки сетки
for I:= 1 to RowCount do
begin
Result:= Result + Format ('%d',[Player.TrackLength[I-1]])+CH_VERT;
// Добавить название композиции и пустой зарезервированный блок
Result:= Result + Cells [2,I-1] + CH_VERT + CH_VERT + CH_ZERO;
end;
end;

```

Здесь формируется структурированная строка с использованием всех тех разделителей.

2.18. Начиная с новой строки введите текст метода "Заполнение сетки" :

```

procedure TMPForm.ShowSongsName;
var I: word; Sng: string;
begin
with Songs do
for I:= 1 to TotSongs do
begin
Sng:= GetSubString ( S, CH_ZERO, I );
Cells [ 2, I - 1 ]:= GetSubString ( Sng, CH_VERT, 2 );
end;
end;

```

Этот метод используется для заполнения сетки строк главной формы названиями композиций.

2.19. Нажмите F12, чтобы перейти к шаблону, и сделайте активной саму форму. На странице Events Инспектора объектов дважды щелкните по событию OnCreate и введите следующий текст в созданном шаблоне (между строками begin и end):

```

CurStatus:= NoDisk;
CD:= TStringList.Create;
// Загрузить описание дисков из файла
CD.LoadFromFile ( FILE_NAME );
// Очистить индикаторы
EmptyIndicat;

```

Вновь перейдите к Инспектору Объектов, создайте шаблон для события OnClose и введите следующую строку:

```

CD.Free;

```

2.20. Несмотря на имеющийся список треков, до сих пор еще нельзя запускать произвольную композицию. Чтобы можно было двойным щелчком по любой строке (или нажатием Enter) запускать нужную песню, перейдите к шаблону главной формы, сделайте активным сетку строк и создайте шаб-

лон для обработчика события OnDbClick (двойной щелчок мышью). Введите в шаблоне код :

```
with Player do
begin
// Позиционировать согласно указанной строки
StartPos:= ( Sender AS TStringGrid ).Row + 1;
// Эмулировать нажатие кнопки Play
Play;
end;
InitProgressBar;
```

Теперь для этой же сетки строк создайте обработчик события OnKeyPress и введите:

```
// Если нажаты Enter или пробел
if Key IN [ #13, #32 ] then
SongsDbClick ( Sender );
```

Теперь снова перейдите к шаблону главной формы и дважды щелкните по компоненту MediaPlayer.

В созданном шаблоне введите следующее:

```
// Кнопка Play не нажата
if Button <> btPlay then
DoDefault:= true
else
SongsDbClick ( Songs );
```

2.21. Выберите пункт меню File > New. В появившемся окне New Items на странице New дважды щелкните по пиктограмме Text. Выберите пункт File > Save As и сохраните пустой текстовый файл в том же каталоге, что и весь проект под именем, присвоенном константе FILE_NAME.

Сохраните проект. Перед этим изменить пиктограмму проекта.

Лабораторная работа № 5

Массивы

Пример 1. Программа, которая вводит одномерный массив действительных чисел и вычисляет двумерный массив.

Даны действительные числа x_1, \dots, x_8 . Получить квадратную матрицу:

$$\begin{bmatrix} x_1 & x_2 & \dots & x_8 \\ x_1^2 & x_2^2 & \dots & x_2^8 \\ \dots & \dots & \dots & \dots \\ x_1^8 & x_2^8 & \dots & x_8^8 \end{bmatrix}$$

Для ввода исходного массива и вывода полученной матрицы использовать компоненты StringGrid.

Развитие задачи:

1. Ввести исходный массив из файла.
2. Полученный массив и исходную информацию вывести в файл.

Пример 2. Дан массив целых чисел, содержащий 15 элементов. Известно, что в массиве всего два одинаковых элемента. Найти эти элементы и их номера.

Текст программы:

```
unit congo1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Grids;
type
  TForm1 = class(TForm)
    Gd1: TStringGrid;
    Label1: TLabel;
    Button1: TButton;
    procedure Gd1KeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);
    procedure poisk;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private { Private declarations }
  public { Public declarations }
  end;
const a:array[0..14] of integer=
  (12,23,14,12,18,-14,-89,65,89,-212,3,56,47,-12,36);
var
  Form1: TForm1;
  n,m:byte;
implementation
{$R *.DFM}
procedure TForm1.poisk;
begin
  label1.caption:="";
  for n:=0 to 13 do
  for m:=n+1 to 14 do begin
  if a[n]=a[m] then begin label1.caption:='повторяются элементы '+
  IntToStr(a[n])+' с номерами '+inttostr(n+1)+' '+inttostr(m+1);
  break; break;
  end; end; end;
  procedure TForm1.Gd1KeyUp(Sender: TObject; var Key: Word;
```

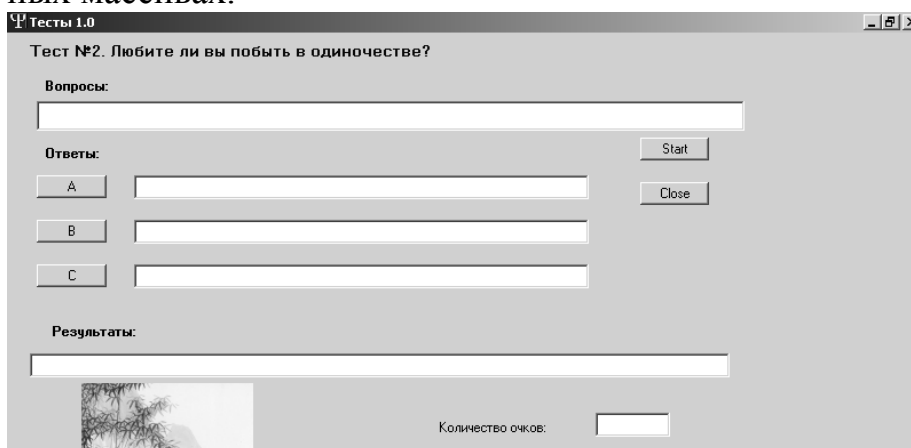
```

    Shift: TShiftState);
begin
if (trim(gd1.Cells[gd1.Col,gd1.row])<>'-' )
and(trim(gd1.Cells[gd1.Col,gd1.row])<>"") then begin
a[gd1.Col]:=StrToInt(trim(gd1.Cells[gd1.Col,gd1.row]));
poisk; end; end;
procedure TForm1.FormCreate(Sender: TObject);
begin
for n:=0 to 14 do
gd1.Cells[n,0]:=inttostr(a[n]);
poisk; end;
procedure TForm1.Button1Click(Sender: TObject);
begin
halt;
end;
end.

```

Пример 3. Написать программу для составления психологических тестов.

1. В поле вводится один вопрос, причем ввод самого первого вопроса активизируется кнопкой «старт».
2. После появления на экране вопроса автоматически выводятся 3 варианта ответа в соответствующих полях напротив кнопок с буквами А, В и С.
3. При выборе варианта ответа в процедуре высчитывается сумма баллов, а на экране появляется следующий вопрос с вариантами ответов.
4. Ниже приведено поле для вывода подсчитанных баллов и их интерпретации.
5. Вопросы и ответы, а также баллы за каждый ответ, хранятся в символьных массивах.



Текст программы:

```

unit Unit2;
interface
uses

```

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls;

type

```
TForm3 = class(TForm)
  EditQues: TEdit;
  EditA: TEdit;
  EditB: TEdit;
  EditC: TEdit;
  ButtonA: TButton;
  ButtonB: TButton;
  ButtonC: TButton;
  Label1: TLabel;
  Start1: TButton;
  Close1: TButton;
  EditR: TEdit;
  Edit1: TEdit;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Image1: TImage;
  {procedure FormCreate(Sender: TObject); }
  procedure WriteAll(p1,p2,p3,p4:string);
  procedure ButtonAClick(Sender: TObject);
  procedure ButtonBClick(Sender: TObject);
  procedure ButtonCClick(Sender: TObject);
  procedure Start1Click(Sender: TObject);
  procedure Close1Click(Sender: TObject);
  procedure Result(p5: string);
  private
    { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form3: TForm3;
k,l,m,i,j:Integer;
obr:real;
s1,s2,s3,s4:string;
Ar1: array[1..7]of string;
Ar2:array[1..3,1..7]of string;
Ark:array[1..7]of Integer;
```

```

Ar1:array[1..7]of integer;
Arm:array[1..7]of integer;
Ar3:array[1..3]of string;
Arf:array[1..3,1..7]of integer;
implementation
{$R *.DFM}
procedure TForm3.WriteAll(p1,p2,p3,p4:string);
begin
Ar1[1]:='Как вы отреагируете, если вам придется провести ночь одной в
гостиничном номере чужого города?';
Ar1[2]:='Что в общении с людьми для вас важнее всего?';
Ar1[3]:='Составляете ли вы для себя с вечера список завтрашних дел?';
Ar1[4]:='Ужиная дома в одиночестве, чем вы параллельно занимаетесь?';
Ar1[5]:='От какого излишества вы с наибольшей легкостью могли бы отка-
заться?';
Ar1[6]:='Часто ли вы ходите одна на концерты, в кино или в театр?';
Ar1[7]:='Что вам приходит в голову, когда вы думаете о природе?';
Ar2[1,1]:='Без проблем';
Ar2[2,1]:='Буду нервничать и испытывать чувство раздражения и страха';
Ar2[3,1]:='Приложив усилие, справлюсь с этой ситуацией';
Ar2[1,2]:='Независимость';
Ar2[2,2]:='Дружба';
Ar2[3,2]:='Чувство защищенности';
Ar2[1,3]:='Регулярно';
Ar2[2,3]:='Если многое предстоит и я боюсь что-то упустить из виду';
Ar2[3,3]:='Никогда';
Ar2[1,4]:='Читаю или смотрю телевизор';
Ar2[2,4]:='Работаю над бумагами, прихваченными из офиса';
Ar2[3,4]:='Ничем. Просто наслаждаюсь трапезой';
Ar2[1,5]:='От алкоголя';
Ar2[2,5]:='От курения';
Ar2[3,5]:='От сладостей';
Ar2[1,6]:='Никогда, хожу только с друзьями';
Ar2[2,6]:='Тогда, когда надоедает сидеть дома';
Ar2[3,6]:='Часто';
Ar2[1,7]:='Прогулка вдвоём по одинокому пляжу';
Ar2[2,7]:='Гроза после жаркого летнего дня';
Ar2[3,7]:='Лес, полный зверей';
Arf[1,1]:=1;
Arf[1,2]:=1;
Arf[1,3]:=3;
Arf[1,4]:=3;

```



```

Arf[1,5]:=1;
Arf[1,6]:=1;
Arf[1,7]:=2;
Arf[2,1]:=3;
Arf[2,2]:=3;
Arf[2,3]:=2;
Arf[2,4]:=1;
Arf[2,5]:=2;
Arf[2,6]:=2;
Arf[2,7]:=1;
Arf[3,1]:=2;
Arf[3,2]:=2;
Arf[3,3]:=1;
Arf[3,4]:=2;
Arf[3,5]:=3;
Arf[3,6]:=3;
Arf[3,7]:=3;
p1:=Ar1[i];
p2:=Ar2[1,i];
p3:=Ar2[2,i];
p4:=Ar2[3,i];
EditQues.Text:=p1;
EditA.Text:=p2;
EditB.Text:=p3;
EditC.Text:=p4;
i:=i+1;
end;
procedure TForm3.Start1Click(Sender: TObject);
begin
i:=1;
k:=0;
l:=0;
m:=0;
WriteAll(EditQues.Text,EditA.Text,EditB.Text,EditC.Text);
j:=1;
end;
procedure TForm3.Result(p5: string);
begin
obr:=k+l+m;
Ar3[1]:='Вы любите уединиться и проводить время вдали от людей. Вы не
доверяете окружающим';

```

```

Ar3[2]:='Вы хорошо переносите одиночество и не считаете себя покину-
той. Скучать вам не приходится';
Ar3[3]:='Вы не в состоянии надолго оставаться одна. Вы хотите иметь все-
гда кого-то рядом';
if (obr>=7) and (obr<=11) then p5:=Ar3[1];
if (obr>=12)and (obr<=16) then p5:=Ar3[2];
if (obr>=17) and (obr<=21) then p5:=Ar3[3];
EditR.Text:=p5;
Str(obr:2:0,s1);
Str(k,s2);
Str(l,s3);
Str(m,s4);
Edit1.Text:=s1;
end;
procedure TForm3.ButtonAClick(Sender: TObject);
begin
if i<8 then
begin
WriteAll(EditQues.Text,EditA.Text,EditB.Text,EditC.Text);
k:=k+Arf[1,j];
j:=j+1;
end
else
begin
k:=k+Arf[1,7];
Result(EditR.Text);
end;
end;
procedure TForm3.ButtonBClick(Sender: TObject);
begin
if i<8 then
begin
WriteAll(EditQues.Text,EditA.Text,EditB.Text,EditC.Text);
l:=l+Arf[2,j];
j:=j+1;
end
else
begin
l:=l+Arf[2,7];
Result(EditR.Text);
end;
end;
end;

```

```

procedure TForm3.ButtonCClick(Sender: TObject);
begin
if i<8 then
begin
WriteAll(EditQues.Text,EditA.Text,EditB.Text,EditC.Text);
m:=m+Arf[3,j];
j:=j+1;
end
else
begin
m:=m+Arf[3,7];
Result(EditR.Text);
end;
end;
procedure TForm3.Close1Click(Sender: TObject);
begin
close;
end;
end.

```

Лабораторная работа № 6 Графика

Пример 1. Нарисовать Олимпийский флаг.

Компоненты формы и свойства объектов:

```

object Form1: TForm1
  Left = 281
  Top = 174
  BorderStyle = bsSingle
  ClientHeight = 217
  ClientWidth = 277
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  OnPaint = FormPaint
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel

```

```

Left = 8
Top = 176
Width = 257
Height = 33
AutoSize = False
WordWrap = True
end
end

```

Текст программы:

```

unit olf_;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    procedure FormPaint(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
// рисует Олимпийский флаг
procedure TForm1.FormPaint(Sender: TObject);
begin
  with Canvas do
  begin
    // полотнище
    Canvas.Pen.Width := 1;
    Canvas.Pen.Color := clBlack;
    Canvas.Brush.Color := clCream;
    Rectangle(30,30,150,115);
    // кольца
    Pen.Width := 2;
    Brush.Style := bsClear; // область внутри круга не закрашивать
    Pen.Color := clBlue;
  end;
end;

```

```

    Ellipse(40,40,80,80);
    Pen.Color := clBlack;
    Ellipse(70,40,110,80);
    Pen.Color := clRed;
    Ellipse(100,40,140,80);
    Pen.Color := clYellow;
    Ellipse(55,65,95,105);
    Pen.Color := clGreen;
    Ellipse(85,65,125,105);
end;
end;
end.

```

Пример 2. Нарисовать звезды.

Компоненты формы и свойства объектов:

```

object Form1: TForm1
  Left = 198
  Top = 105
  Width = 350
  Height = 212
  Caption = 'StarLine'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  OnMouseDown = FormMouseDown
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 8
    Top = 168
    Width = 324
    Height = 13

```

Текст программы:

```

unit Stars_;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

```

Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure StarLine(x0,y0,r: integer); // рисует звезду
  end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
  // рисует звезду
  procedure TForm1.StarLine(x0,y0,r: integer);
    // x0,y0 - координаты центра звезды
    // r - радиус звезды
  var
    p : array[1..11] of TPoint; // массив координат лучей и впадин
    a: integer; // угол между осью OX и прямой, соединяющей
      // центр звезды и конец луча или впадину
    i: integer;
  begin
    a := 18; // строим от правого гор. луча
    for i:=1 to 10 do
      begin
        if (i mod 2 = 0) then
          begin // впадина
            p[i].x := x0+Round(r/3*cos(a*2*pi/360));
            p[i].y:=y0-Round(r/3*sin(a*2*pi/360));
          end
        else
          begin // луч
            p[i].x:=x0+Round(r*cos(a*2*pi/360));
            p[i].y:=y0-Round(r*sin(a*2*pi/360));
          end;
        a := a+36;
      end;
    p[11].X := p[1].X; // чтобы замкнуть контур звезды
  end;

```

```

    p[11].Y := p[1].Y;
    Canvas.Polyline(p); // начертить контур звезды
end;
// нажатие кнопки мыши
procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    if Button = mbLeft // нажата левая кнопка?
    then Canvas.Pen.Color := clBlack
    else Canvas.Pen.Color := clRed;
    StarLine(x, y, 30);
end;
end.

```

Примеры для самостоятельной работы:

1. Нарисовать кораблик.
2. Построить график функции $y=x^2$ по точкам.
3. Нарисовать домик с покатой крышей.

Приложение **Тесты по курсу Дельфи**

1. Какие операторы осуществляют обработку исключительных ситуаций в Delphi?
 - try
 - except
 - finally
 - все
2. Какие типы данных из перечисленных ниже существуют в Delphi?

указательные

 - файловые
 - оба типа
 - нет таких
3. Как наиболее точно охарактеризовать SQL?
 - язык управления данными
 - структурированный язык запросов
 - язык определения данных
4. Какие из перечисленных ниже являются действительными типами данных в Delphi?
 - Currency

- Real
 - ни один
 - оба
5. Какие технологии включает в себя ActiveX?
- библиотеки ActiveX и серверы автоматизации
 - элементы управления и формы ActiveForm
 - все
 - ни одно
6. Что позволяет определять свойство Border Style?
- вид заголовка формы
 - отображение строки меню и кнопки в заголовке формы
 - оба
 - ни один
7. Какой компонент связывает воздействия со стороны пользователя и реакции на них?
- TActionQuery
 - TActionList
 - TAction
 - ни один
8. Что общего у компонентов TQuery и TTable?
- оба компонента осуществляют SQL-запросы
 - ничего
 - предназначены для создания и управления данными
9. Какие из перечисленных ниже принадлежат к структурным типам данных в Delphi?
- записи
 - объекты
 - оба
 - ни один
10. Какие свойства определяют высоту и ширину формы?
- Top и Bottom
 - Widht и Height
 - все
 - ни одно
11. Какие из перечисленных ниже - целые типы данных в Delphi?
- LongStr
 - Smalint
 - оба типа
 - ни один

12. Какое свойство определяет размещение формы при запуске приложения?

- Position
- Style
- Place
- первое и второе

13. В каких случаях свойство Active родительского MDI-окна равно TRUE?

- после выполнения метода FormShow
- после выполнения метода ChildOpen
- никогда
- всегда

14. Какое свойство определяет размещение дочерних форм?

- TitleMode
- FormStyle
- TitleStyle
- ни одно