

**Министерство образования Московской области
Государственное бюджетное профессиональное образовательное
учреждение Московской области
«Воскресенский колледж»**

**Методические рекомендации по выполнению лабораторных работ
ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Наименование специальности
09.02.07 «Информационные системы и программирование»

Квалификация выпускника

Программист

Методические рекомендации по выполнению лабораторных работ по дисциплине разработаны на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее – СПО)

09.02.07 «Информационные системы и программирование»

Организация разработчик: Государственное бюджетное профессиональное образовательное учреждение Московской области «Воскресенский колледж»

Разработчики:

Вострякова А.В., преподаватель компьютерных дисциплин

Рабочая программа профессионального модуля рассмотрена на заседании предметной (цикловой) комиссией компьютерных дисциплин

«___»_____2020г.

Председатель цикловой комиссии _____/Рязанцева О.В./

Утверждена зам директора по УР _____/Куприна Н.Л./

«___»_____2020 г.

Содержание

Введение	4
Лабораторная работа № 1 Разработка линейных программ	5
Лабораторная работа № 2 Разработка программ с разветвляющейся структурой. 7	
Лабораторная работа №3 Разработка программ с использованием операторов цикла	9
Лабораторная работа № 4,5 Разработка программы содержащей процедуры и функции. Разработка модуля содержащей процедуры и функции, и использование его в программе.....	12
Лабораторная работа № 6 Разработка программ с использованием одномерных и двумерных массивов.	15
Лабораторная работа № 7 Обработка файловых структур данных.....	19
Лабораторная работа №8	24
Создание графического изображения с помощью стандартных функций.	24
Приложение 1 Справка по работе с основным набором визуальных компонентов Delphi, требуемых при выполнении лабораторных работ	26
Список использованной литературы.....	39

Введение

Требования к уровню подготовки выпускников по специальности 09. 02. 03 «Программирование в компьютерных системах» определены Государственным образовательным стандартом среднего профессионального образования. Стандартом предусмотрено изучение дисциплины «Основы программирования», дающей учащемуся необходимый минимум знаний в области программирования. Лабораторный практикум проводится с обязательным использованием компьютера. Одновременное рассмотрение возможностей системы программирования и конструкций языка программирования позволяет учащимся буквально с первых занятий получить представление о технологии разработки приложений. По окончании курса учащийся может приобрести все необходимые навыки и умения для проектирования программ достаточно высокой сложности. В качестве языка программирования используется язык Delphi и C++ входящий в комплект языков среды программирования EmbarcaderoRAD Studio. Реализация этого языка в сочетании с дружественным интерфейсом среды делает работу студента весьма комфортной, что не может не сказаться на результатах его работы. В пособии предложено 8 лабораторных работ. В приложении даны необходимые сведения о тех компонентах, которые использованы при изложении материала. В списке литературы приведены наиболее удачные, по мнению автора, учебники, позволяющие обучающемуся самостоятельно увеличить объем знаний в области программирования в операционной системе Windows. Материал пособия может быть полностью или частично использован преподавателем для организации лабораторных работ в соответствии с отведенным объемом часов.

Лабораторная работа № 1

Разработка линейных программ

Цель работы — овладение практическими навыками разработки и программирования вычислительного процесса линейной структуры и навыками по отладке и тестированию программ.

Задания для самостоятельной подготовки

1. Изучить:

- запись констант, переменных, стандартных функций;
- правила записи арифметических выражений;
- арифметический оператор присваивания;
- организацию простейшего ввода-вывода данных.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тестовый вариант исходных данных и вычислить для них вручную или с помощью микрокалькулятора значения вычисляемых в программе величин.

Задание к работе

Вычислить на ЭВМ значения переменных, указанных в табл. 1.1 (вариант задается преподавателем), по заданным расчетным формулам и наборам исходных данных. На печать вывести значения вводимых исходных данных и результаты вычислений, сопровождая вывод наименованиями выводимых переменных.

Таблица 1.1

Вариант задания	Расчетные формулы	Значения исходных данных
1	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$ $b = 1 + \frac{z^2}{3 + z^2/5}$	x = 1.426 y = -1.220 z = 3.5
2	$\gamma = \left x^{y/x} - \sqrt[3]{y/x} \right $ $\psi = (y - x) \frac{y - z/(y - x)}{1 + (y - x)^2}$	x = 1.825 y = 18.225 z = -3.298

Таблица 1.1 Окончание

Вариант задания	Расчетные формулы	Значения исходных данных
3	$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$ $\psi = x(\sin x^3 + \cos^2 y)$	$x = 0.335$ $y = 0.025$
4	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$ $s = b \sin(at^2 \cos 2t) - 1$	$a = -0.5$ $b = 1.7$ $t = 0.44$
5	$w = \sqrt{x^2 + b} - b^2 \sin^3(x + a) / x$ $y = \cos^2 x^3 - x / \sqrt{a^2 + b^2}$	$a = 1.5$ $b = 15.5$ $x = -2.9$
6	$s = x^3 \operatorname{tg}^2(x + b)^2 + a / \sqrt{x + b}$ $Q = \frac{bx^2 - a}{e^{ax} - 1}$	$a = 16.5$ $b = 3.4$ $x = 0.61$
7	$R = x^2(x + 1) / b - \sin^2(x + a)$ $s = \sqrt{xb/a} + \cos^2(x + b)^3$	$a = 0.7$ $b = 0.05$ $x = 0.5$
8	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$ $z = \frac{x^2}{a} + \cos(x + b)^3$	$a = 1.1$ $b = 0.004$ $x = 0.2$
9	$f = \sqrt[3]{m \cdot \operatorname{tg} t + c \cdot \sin t }$ $z = m \cos(bt \sin t) + c$	$m = 2$ $c = -1$ $t = 1.2$ $b = 0.7$
10	$y = b \operatorname{tg}^2 x - \frac{a}{\sin^2(x/a)}$ $d = a e^{-\sqrt{a}} \cos(bx/a)$	$a = 3.2$ $b = 17.5$ $x = -4.8$

Контрольные вопросы

1. Какие типы величин используются в языке программирования?
2. Указать диапазон значений величин целого и действительного типов.
3. Какие имена переменных допустимы в программе? Как задать тип переменной в программе?
4. Указать имена стандартных функций для вычисления

\sqrt{x} , e^x , $\sin x$, $\ln x$, $|x|$.

5. Можно ли в качестве операнда в арифметическом выражении использовать: а) имя массива; б) имя стандартной функции, например $\text{SIN}(Y)$; в) имя символьной переменной или переменной логического типа?

6. Назвать последовательность действий при выполнении арифметического оператора присваивания. Допустимо ли использование величин разных типов в арифметическом выражении?

7. Указать старшинство выполнения операций при вычислении арифметического выражения.

8. Указать средства, имеющиеся в языке программирования для управления размещением данных на строке. Как организовать вывод значений, сопровождая выводимое числовое значение наименованием переменной? Как организовать пропуск одной, двух строк при выводе?

9. Как выбрать значения исходных данных для тестового варианта счета?

Лабораторная работа № 2

Разработка программ с разветвляющейся структурой.

Цель работы — овладение практическими навыками разработки, программирования вычислительного процесса разветвляющейся и циклической структур, получение дальнейших навыков по отладке и тестированию программы.

Задания для самостоятельной подготовки

Задание А

1. Изучить возможности языка программирования для реализации:
 - условной и безусловной передачи управления;
 - вычислительного процесса разветвляющейся структуры
2. Разработать алгоритм решения в соответствии с заданием.
3. Составить программу решения задачи.
4. Подготовить тесты (число тестов равно числу ветвей вычислительного процесса) для проверки правильности функционирования программы.

Задание Б.

1. Изучить возможности языка программирования для реализации:
 - вычислительных процессов циклической структуры с известным числом повторений в цикле;
 - приема программирования — табулирования функции от одного аргумента (вычисление значения функции при изменении значения аргумента в заданном диапазоне с шагом Δx).
2. Разработать алгоритм табулирования функции, определенной в задании А.
3. Составить программу табулирования функции.

Задание к работе

Задание А.

1. Вычислить значение функции, заданной в табл. 2.1 (в соответствии с вариантом задания). Осуществить вывод значений вводимых исходных данных и результат вычисления значения функции, сопровождая вывод наименованиями переменных.

Таблица 2.1

Вариант	Функция	Условие	Исходные данные	Диапазон и шаг изменения аргумента
1	$y = \begin{cases} at^2 \ln t \\ 1 \\ e^{at} \cos bt \end{cases}$	$\begin{cases} 1 \leq t \leq 2 \\ t < 1 \\ t > 2 \end{cases}$	$\begin{cases} a = -0.5 \\ b = 2 \end{cases}$	$\begin{cases} t \in [0; 3] \\ \Delta t = 0.15 \end{cases}$
2	$y = \begin{cases} \pi x^2 - 7/x^2 \\ ax^3 + 7\sqrt{x} \\ \lg(x + 7\sqrt{x}) \end{cases}$	$\begin{cases} x < 1.3 \\ x = 1.3 \\ x > 1.3 \end{cases}$	$a = 1.5$	$\begin{cases} x \in [0.8; 2] \\ \Delta x = 0.1 \end{cases}$
3	$w = \begin{cases} ax^2 + bx + c \\ a/x + \sqrt{x^2 + 1} \\ (a + bx)/\sqrt{x^2 + 1} \end{cases}$	$\begin{cases} x < 1.2 \\ x = 1.2 \\ x > 1.2 \end{cases}$	$\begin{cases} a = 2.8 \\ b = -0.3 \\ c = 4 \end{cases}$	$\begin{cases} x \in [1; 2] \\ \Delta x = 0.05 \end{cases}$
4	$y = \begin{cases} \pi x^2 - 7/x^2 \\ ax^3 + 7\sqrt{x} \\ \ln(x + 7\sqrt{ x+a }) \end{cases}$	$\begin{cases} x < 1.4 \\ x = 1.4 \\ x > 1.4 \end{cases}$	$a = 1.65$	$\begin{cases} x \in [0.7; 2] \\ \Delta x = 0.1 \end{cases}$
5	$y = \begin{cases} 1.5 \cos^2 x \\ 1.8ax \\ (x-2)^2 + 6 \\ 3 \operatorname{tg} x \end{cases}$	$\begin{cases} x < 1 \\ x = 1 \\ 1 < x < 2 \\ x > 2 \end{cases}$	$a = 2.3$	$\begin{cases} x \in [0.2; 2.8] \\ \Delta x = 0.2 \end{cases}$
6	$w = \begin{cases} x\sqrt[3]{x-a} \\ x \sin ax \\ e^{-ax} \cos ax \end{cases}$	$\begin{cases} x > a \\ x = a \\ x < a \end{cases}$	$a = 2.5$	$\begin{cases} x \in [1; 5] \\ \Delta x = 0.5 \end{cases}$
7	$Q = \begin{cases} bx - \lg bx \\ 1 \\ bx + \lg bx \end{cases}$	$\begin{cases} bx < 1 \\ bx = 1 \\ bx > 1 \end{cases}$	$b = 1.5$	$\begin{cases} x \in [0.1; 1] \\ \Delta x = 0.1 \end{cases}$

Таблица 2.1 Окончание

Вариант	Функция	Условие	Исходные данные	Диапазон и шаг изменения аргумента
8	$y = \begin{cases} \sin x \lg x \\ \cos^2 x \end{cases}$	$x > 3.5$ $x \leq 3.5$		$x \in [2; 5]$ $\Delta x = 0.25$
9	$f = \begin{cases} \lg(x+1) \\ \sin^2 \sqrt{ ax } \end{cases}$	$x > 1$ $x \leq 1$	$a = 20.3$	$x \in [0.5; 2]$ $\Delta x = 0.1$
10	$z = \begin{cases} (\ln^3 x + x^2) / \sqrt{x+t} \\ \sqrt{x+t} + 1/x \\ \cos x + t \sin^2 x \end{cases}$	$x < 0.5$ $x = 0.5$ $x > 0.5$	$t = 2.2$	$x \in [0.2; 2]$ $\Delta x = 0.2$

Контрольные вопросы

1. Перечислить действия, реализуемые при выполнении условного оператора.
2. Какие действия выполняются оператором перехода?
3. Что такое вычислительный процесс разветвляющейся структуры? Как организовать разветвление вычислений: а) на две ветви; б) на три ветви?
4. Составить последовательность операторов для вычисления величины $z = 0$, если $x < -2$; $z = 1$, если $-2 \leq x \leq 2$; $z = -1$, если $x > 2$.
5. Зачем необходимо при отладке программы тестировать **все** ветви алгоритма?
6. Указать последовательность действий, выполняемых при организации циклических участков программы с заданным числом повторений.
7. Указать назначение и правила организации цикла.
8. Перечислить возможные способы организации цикла с заданным числом повторений в изучаемом языке программирования.

Лабораторная работа №3

Разработка программ с использованием операторов цикла

Цель работы — овладение практическими навыками разработки и программирования алгоритмов итерационной циклической структуры; приобретение дальнейших навыков по отладке и тестированию программ.

Задания для самостоятельной подготовки

1. Изучить:
 - организацию итерационных циклов;
 - возможности языка программирования для организации таких циклов;

— приемы программирования — вычисление суммы членов бесконечного ряда, накопления суммы.

2. Разработать алгоритм решения задачи для задания.
3. Составить программу решения задач и для задания.
4. Вычислить предел суммы членов ряда, указанного в задании.

Задание к работе

Вычислить на ЭВМ значение суммы членов бесконечного ряда (табл. 3.1) с заданной точностью ϵ . На печать вывести значение суммы и число членов ряда, вошедших в сумму.

Сравнить полученное на ЭВМ значение суммы членов ряда со значением, вычисленным вручную.

Таблица 3.1

Вариант	Сумма членов ряда	Значение	Точность вычисления
1	$s = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!} + \dots$	0,20	10^{-5}
2	$s = x - \frac{x^3}{3} + \dots + (-1)^n \frac{x^{2n-1}}{2n-1} + \dots$	0,10	$0.5 \cdot 10^{-4}$
3	$s = \frac{x^3}{5} - \frac{x^5}{17} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 + 1} + \dots$	0,15	10^{-3}
4	$s = 1 + \cos \frac{\pi}{4} \cdot \frac{x}{1!} + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n + \dots$	0,12	10^{-4}
5	$\operatorname{ch} x \cong s = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$	0,70	10^{-4}
6	$\pi \cong s = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + \dots + (-1)^n \frac{1}{2n-1} + \dots\right)$		10^{-4}
7	$\operatorname{arctg} x \cong s = \frac{1}{x} - \frac{1}{3x^3} + \frac{1}{5x^5} - \dots + (-1)^n \frac{1}{(2n+1)x^{2n+1}}$	1,5	$0.5 \cdot 10^{-3}$

Таблица 3.1 Окончание

Вариант	Сумма членов ряда	Значение	Точность вычисления
8	$\cos \frac{\pi}{6} \cong s = 1 - \frac{\left(\frac{\pi}{6}\right)^2}{2!} + \frac{\left(\frac{\pi}{6}\right)^4}{4!} - \dots + (-1)^n \frac{\left(\frac{\pi}{6}\right)^{2n}}{(2n)!} + \dots$		$0.5 \cdot 10^{-4}$
9	$\operatorname{ch} x \cong x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$	1,7	10^{-3}
10	$\sin \frac{\pi}{3} \cong s = \frac{\pi}{3} - \frac{\left(\frac{\pi}{3}\right)^3}{3!} + \frac{\left(\frac{\pi}{3}\right)^5}{5!} - \dots + (-1)^n \frac{\left(\frac{\pi}{3}\right)^{2n+1}}{(2n+1)!} + \dots$		$0.5 \cdot 10^{-4}$
11	$s = 1 + \frac{x^2}{2!} - \frac{3x^4}{4!} + \dots + (-1)^n \frac{2n-1}{(2n)!} x^{2n} + \dots$	0,75	$0.5 \cdot 10^{-3}$
12	$s = \frac{2}{3} \sin 2x - \frac{3}{8} \sin 3x + \dots + (-1)^n \frac{n}{n^2 - 1} \sin nx$	0,62	10^{-4}
13	$s = 1 + \frac{\cos x}{1!} + \frac{\cos 2x}{2!} + \dots + \frac{\cos nx}{n!} + \dots$	0,20	10^{-4}
14	$s = \frac{x^3}{3} - \frac{x^5}{15} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1} + \dots$	0,30	10^{-5}
15	$s = \frac{x \cos \frac{\pi}{3}}{1} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2} + \dots + \frac{x^n \cos n \frac{\pi}{3}}{n} + \dots$	0,25	10^{-4}

Контрольные вопросы

1. Указать отличия в организации циклов с заданным числом повторении и итерационных.
2. Какие средства языка целесообразно использовать для организации циклов с заданным числом повторений?
3. Указать, какие операторы составляют тело цикла.
4. В чем состоят преимущества использования операторов цикла в программах?

Лабораторная работа № 4,5

Разработка программы содержащей процедуры и функции. Разработка модуля содержащей процедуры и функции, и использование его в программе.

Цель работы — овладение навыками алгоритмизации и программирования задач с использованием подпрограмм пользователя различных видов, овладение навыками написания подпрограмм и обращения к ним, выбора параметров подпрограмм.

Задания для самостоятельной подготовки

1. Изучить:

- правила записи подпрограмм различных видов и способов обращения к ним;
- способы передачи параметров в подпрограмму;
- правила записи программ, использующих подпрограммы различных видов;
- порядок выполнения программ, использующих подпрограммы.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тестовый вариант программы и исходных данных.

Задание к работе

Задание А.

Выполнить на ЭВМ программу, использующую подпрограмму-функцию, в соответствии с номером параметра, указанным в табл. 6.1.

Таблица 6.1

Вариант	Условия задачи	Примечания
1	<p>Вычислить наибольшие корни квадратных уравнений</p> $x^2 - ax + b = 0$ $cy^2 - dx - f = 0$	Все корни действительные
2	<p>Подсчитать число точек, находящихся внутри круга радиусом r с центром в начале координат; координаты заданы массивами X(100), Y(100)</p>	<p>Расстояние точки от начала координат вычислять в подпрограмме</p>
3	<p>Определить периметры треугольников, заданных координатами их вершин</p> <p>XA (5), XB (5), XC (5) YA (5), YB (J), YC (5)</p>	<p>Длину стороны треугольников вычислять в подпрограмме</p>
4	<p>Подсчитать число точек, находящихся внутри круга радиусом r с центром в точке с координатами (1, 1); координаты заданы массивами X (80), Y (80)</p>	<p>Расстояние точки от центра круга определять в подпрограмме</p>

Таблица 6.1 Окончание

Вариант	Условия задачи	Примечания
5	Вычислить $z = \frac{v_1 + v_2 + v_3}{3}$, где v_1, v_2, v_3 – объемы шаров с радиусами r_1, r_2, r_3	v_1 вычислять в подпрограмме
6	Вычислить суммы положительных элементов массивов X (N), Y (M), Z (K)	$N \leq 60$ $M \leq 60$ $K \leq 70$
7	Вычислить среднее арифметическое положительных элементов для массивов A (N1), B (N2), C (N3)	$N1 \leq 100$ $N2 \leq 100$ $N3 \leq 100$
8	Подсчитать количество отрицательных элементов матриц X (10, 15) и Y (20, 12)	Количество отрицательных элементов матрицы вычислять в
9	Вычислить суммы положительных элементов каждой строки для матриц A (10, 12) и B (15, 10)	Сумму положительных элементов строки
10	Вычислить $z = \frac{x_{m1} + x_{m2}}{2}$, где x_{m1} и x_{m2} – наименьшие элементы массивов X1 (70), X2 (80)	Наименьший элемент одномерного массива вычислять в подпрограмме

2. Проверить правильность выполнения программы с помощью тестового варианта.

Задание Б.

1. Выполнить на ЭВМ программу, использующую подпрограмму-процедуру в соответствии с номером, указанным в табл. 6.2.

Таблица 6.2.

Вариант задания	Условие задачи	Примечания
1	Вычислить $z = \frac{s_1 + s_2}{k_1 k_2}$, где s_1 и k_1 – сумма и количество положительных элементов массива X(N); s_2 и k_2 – сумма и количество отрицательных элементов массива Y (M)	$M \leq 100$ $N \leq 100$
2	Вычислить $z = \frac{e^{s_1} + e^{s_2}}{k_1 k_2}$, где s_1 и k_1 – сумма и количество положительных элементов массива X(100); s_2 и k_2 – сумма и количество отрицательных элементов массива Y (80)	Вычислять в одной подпрограмме s и k

3	Вычислить и запомнить суммы положительных элементов каждой строки матрицы A (10, 20), B (15, 10)	
4	Вычислить $z = (x_1 + y_1)/(x_2 - y_2)$, где x_1 и x_2 – корни уравнения $2x^2 + x - 4 = 0$, y_1 и y_2 – корни уравнения $ay^2 + 2y - 1 = 0$	Все корни действительные
5	Найти наибольшие элементы и их порядковые номера массивов X (N) и Y (M)	$N \leq 80$ $M \leq 70$
6	Переписать положительные элементы массива X(100) и Y (80) в массив Z подряд	Запись в массив Z осуществлять в подпрограмме
7	Найти наименьшие элементы и номера строк и столбцов, в которых они расположены, для матриц A (10, 15) и B (15, 12)	
8	Вывести на печать элементы целочисленных матриц N (5, 8) и M (10, 6), кратные трем	
9	Вычислить $z = \frac{\sum_{i=1}^{40} \sin x_i + \sum_{i=1}^{50} \cos y_i}{\sum_{i=1}^{40} x_i }$, где x_i и y_i заданы массивами	Все суммы вычислять с использованием подпрограммы
10	Вычислить $z = \frac{x_{\max} - y_{\min}}{2}$, где x_{\max} – максимальный элемент массива X (50); y_{\min} – минимальный элемент массива Y (40)	x_{\max} и y_{\min} вычислять в подпрограмме

2. Проверить правильность выполнения программы с помощью тестового варианта.

Контрольные вопросы

1. Указать, при каких условиях целесообразно использование подпрограмм, какие выгоды они предоставляют пользователю.
2. Указать, в чем отличие различных видов подпрограмм пользователя.
3. Указать способы обращения к подпрограммам пользователя.
4. Указать способы передачи параметров в подпрограмму.
5. Указать, как организовать подпрограмму без параметров.
6. Перечислить, как согласуются формальные и фактические параметры.
7. Указать конструкции, которые могут быть формальными и фактическими параметрами.

Лабораторная работа № 6

Разработка программ с использованием одномерных и двумерных массивов.

Цель работы — овладение практическими навыками работы с массивами, особенностями их ввода и вывода, приобретение дальнейших навыков по организации программ циклической структуры с использованием приемов программирования.

Задание для самостоятельной подготовки

1. Изучить:

- способы описания размеров массивов на языке программирования;
- способы ввода и вывода массивов;
- реализацию на конкретном языке программирования приемов накопления суммы или произведения, запоминания результатов, нахождения наибольшего и наименьшего.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тест для проверки программы.

Задание к работе

1. Обработать на ЭВМ массив в соответствии с вариантом задания, указанного в табл. 4.1.

2. Проверить правильность выполнения программы с помощью тестового варианта.

Таблица 4.1

Вариант задания	Массив	Действия	Условия и ограничения
1	X(100)	Вычислить сумму и количество элементов массива X	$0 \leq x_i \leq 1$
2	A (80)	Вычислить среднее арифметическое значение элемента массива A	$a_i > 0$
3	X(70)	Переписать элементы массива X в массив Y и подсчитать их количество	$-1 \leq x_i \leq 1$
4	B (50)	Определить максимальный элемент массива B и его порядковый номер	$x_i > 0$
5	C (40)	Вычислить минимальный элемент массива C и его номер	$x_i < 0$
6	D (80)	Найти максимальный и минимальный элементы массива D и поменять их местами	
7	Y(20)	Вычислить среднее геометрическое элемента массива Y	$y_i > 0$

8	Z(30)	Расположить в массиве R сначала положительные, а затем отрицательные элементы массива Z	
---	-------	---	--

Таблица 4.1 Окончание

Вариант задания	Массив	Действия	Условия и ограничения
9	N(50)	Определить сумму элементов массива N, кратных трем	$n_i / 3 * 3 = n_i$
10	X(N)	Вычислить сумму и количество элементов массива X	$x_i > 0, N \leq 30$
11	A(N)	Найти среднее геометрическое элементов массива A	$a_i > 0, N \leq 50$
12	X(N)	Переписать в массив Y подряд положительные элементы массива X	$x_i > 0, N \leq 40$
13	X(N)	Переписать подряд в массив Y положительные и в массив Z отрицательные элементы массива X	$N \leq 40$
14	B(K)	Определить максимальный элемент массива B и его порядковый номер	$x_i > 0, K \leq 40$
15	C(K)	Определить минимальный элемент массива C и его порядковый номер	$-1 \leq x_i \leq 1, K \leq 20$

Контрольные вопросы

1. Что такое массив как структура данных?
2. Что указывается в операторе объявления статического массива?
3. Какой вид цикла, как правило, применяют при обработке элементов массива?
4. Как организовать работу со статическим массивом переменной размерности?

Обработка матриц

Цель работы — овладение навыками алгоритмизации и программирования структур с вложенными циклами, навыками использования приемов программирования во вложенных циклах, способами ввода и вывода матриц.

Задания для самостоятельной подготовки

1. Изучить:

- правила организации вложенного цикла с учетом порядка перебора элементов матрицы;
- правила использования приемов программирования в структурах с вложенными циклами;
- способы ввода и вывода матриц, имеющиеся в языке программирования.

2. Разработать алгоритм решения в соответствии с заданием.
3. Составить программу решения задачи.
4. Подготовить тестовый вариант программы и исходных данных.

Задание к работе

1. Обработать на ЭВМ матрицу в соответствии с вариантом задания, указанного в табл. 5.1. Вывести на печать результаты и исходную матрицу в общепринятом виде.
2. Проверить правильность выполнения программы с помощью тестового варианта.

Таблица 5.1

Вариант задания	Имя матрицы и размеры	Действия	Условия ограничения
1	A (10, 15)	Вычислить и запомнить сумму и число положительных элементов каждого столбца матрицы. Результаты отпечатать в виде двух строк	$a_{ij} > 0$
2	A (N, M)	Вычислить и запомнить суммы и числа элементов каждой строки матрицы. Результаты отпечатать в виде двух	$N \leq 20$ $M \leq 15$
3	B (N, N)	Вычислить сумму и число элементов матрицы, находящихся под главной	$N \leq 12$
4	C (N, N)	Вычислить сумму и число положительных элементов матрицы, находящихся над главной диагональю	$c_{ij} > 0$ $N \leq 12$
5	D (K, K)	Записать на место отрицательных элементов матрицы нули и вывести ее на печать в общепринятом виде	$K \leq 10$
6	D(10, 10)	Записать на место отрицательных элементов матрицы нули, а на место положительных — единицы. Вывести на печать нижнюю треугольную матрицу в	
7	F (N, M)	Найти в каждой строке матрицы максимальный и минимальный элементы и поместить их на место первого и последнего элемента строки соответственно. Матрицу напечатать в обще-	$N \leq 20$ $M \leq 10$
8	F(10,8)	Транспонировать матрицу и вывести на печать элементы главной диагонали и диагонали, расположенной под главной. Результаты разместить в двух строках	
9	N (10, 10)	Для целочисленной матрицы найти для каждой строки число элементов, кратных пяти, и наибольший из полученных	$n_{ij} / 5 * 5 = n_{ij}$

Таблица 5.1 Окончание

Вариант задания	Имя матрицы и размеры	Действия	Условия ограничения
10	$N(10, 10)$	Из положительных элементов матрицы N сформировать матрицу $M(10, KMAX)$, располагая их в строках матрицы подряд, где $KMAX$ — максимальное число положительных элементов строки матрицы N . Записать нули на место отсутствующих элементов. Отпечатать обе	
11	$P(N, N)$	Найти в каждой строке наибольший элемент и поменять его местами с элементом главной диагонали. Отпечатать полученную матрицу в общепринятом	$N \leq 15$
12	$R(K, N)$	Найти наибольший и наименьший элементы матрицы и поменять их местами	$K \leq 20$ $M \leq 10$
13	$S(25, 8)$	Ввести исходные данные в первые 24 строки и первые 7 столбцов. Вычислить среднее арифметическое значение элементов каждой строки и записать его в 8-й столбец, а также среднее арифметическое каждого столбца и записать его в 25-ю строку. Отпечатать	
14	$T(N, M)$	Найти строку с наибольшей и наименьшей суммой элементов. Вывести на печать найденные строки и суммы их	$N \leq 20$ $M \leq 15$
15	$V(15, 10)$	Упорядочить по возрастанию элементы каждой строки матрицы. Отпечатать полученную матрицу в общепринятом	

Контрольные вопросы

1. Указать основные правила организации вложенных циклов.
2. Указать способы выхода из внутреннего цикла.
3. Как организовать вывод матрицы в общепринятом виде?
4. Как организовать вывод нижней треугольной матрицы в общепринятом виде?
5. Как организовать ввод матрицы размером $N \times M$ элементов?

Лабораторная работа № 7

Обработка файловых структур данных

Цель работы — овладение навыками алгоритмизации и программирования файловых структур данных; проектирование структуры файла, вывод данных в файл, чтение данных из файла.

Задания для самостоятельной подготовки

1. Изучить:
 - основную терминологию, связанную с файловыми структурами данных: файл и его структура, физическая и логическая записи, методы доступа, форматные и бесформатные записи, запись конца файла для файлов с последовательным доступом;
 - возможности языка программирования по обработке файла с последовательной организацией: запись данных в файл, чтение из файла, добавление записей в файл, корректировка записей и т. п.
2. Разработать алгоритм решения в соответствии с заданием.
3. Составить программу решения задачи.
4. Подготовить тестовый вариант программы и исходных данных.

Задание к работе

А. Выполнить на ЭВМ программу в соответствии с вариантом задания, указанным в таблице 7.1 (п.п. А). Прочитать и напечатать созданный файл.

Таблица 7.1

Вариант задания	Условие задачи
1	<p>А. Создать файл, содержащий сведения о месячной заработной плате рабочих завода. Каждая запись содержит поля — фамилия рабочего, наименование цеха, размер заработной платы за месяц. Количество записей — произвольное.</p> <p>Б. Вычислить общую сумму выплат за месяц по цеху X, а также среднемесячный заработок рабочего этого цеха. Напечатать для бухгалтерии ведомость для начисления заработной платы рабочим этого цеха</p>

Таблица 7.1 продолжение

Вариант задания	Условие задачи
2	<p>А. Создать файл, содержащий сведения о количестве изделий, собранных сборщиками цеха за неделю. Каждая запись содержит поля: фамилия сборщика, количество изделий, собранных им ежедневно в течение шестидневной недели, т.е. отдельно — в понедельник, вторник и т. д. Количество записей — произвольное.</p> <p>Б. Написать программу, выдающую на печать следующую информацию: фамилию сборщика и общее количество деталей, собранное им за неделю; фамилию сборщика, собравшего наибольшее число изделий, и день, когда он достиг наивысшей производительности труда</p>
3	<p>А. Создать файл, содержащий сведения о количестве изделий 20и20 категорий A, B, C, собранных рабочим за месяц. Структура записи имеет поля: фамилия сборщика, наименование цеха, количество изделий по категориям, собранных рабочим за месяц. Количество записей — произвольное.</p> <p>Б. Считая заданными значения расценок S_A, S_B, S_C за выполненную работу по сборке единицы изделия категорий A, B, C соответственно, выдать на печать следующую информацию: — общее количество изделий категорий A, B, C, собранных рабочим цеха X; — ведомость заработной платы рабочих цеха X; — средний размер заработной платы работников этого цеха</p>
4	<p>А. Создать файл, содержащий сведения о телефонах абонентов. Каждая запись имеет поля: фамилия абонентов, год установки телефона, номер телефона. Количество записей произвольное.</p> <p>Б. Написать программу, выдающую информацию следующего вида: — по вводимой фамилии абонента выдается номер телефона; — определяется количество установленных телефонов с $XXXX$ года. Номер года вводится с терминала</p>
5	<p>А. Создать файл, содержащий сведения об ассортименте игрушек в магазине. Структура записи: название игрушки, цена, количество, возрастные границы, например 2—5, т. е. от 2 до 5 лет. Количество записей произвольно.</p> <p>Б. Написать программу, в результате выполнения которой выдаются следующие сведения: — названия игрушек, которые подходят детям от 1 до 3 лет; — стоимость самой дорогой игрушки и ее наименование; — название игрушки, которая по стоимости не превышает x руб. и подходит ребенку в возрасте от a до b лет. Значения x, a, b ввести с терминала</p>

6	<p>А. Создать файл, содержащий сведения о сдаче студентами I курса кафедры «ЭВМ» сессии. Структура записи: индекс группы, фамилия студента, оценки по пяти экзаменам, признак участия в общественной работе: «1» — активное участие, «0» — неучастие. Количество записей — 30.</p> <p>Б. Написать программу зачисления студентов группы X на стипендию. Студент, получивший все оценки «5» и активно участвующий в общественной работе, зачисляется на повышенную стипендию (доплата 50 %), не активно участвует — доплата 25 %. Студенты, получившие «4» и «5», зачисляются на обычную стипендию. Студент, получивший одну оценку «3», но активно занимающийся общественной работой, также зачисляется на стипендию, в противном случае зачисление не производится. Индекс группы вводится с терминала</p>
7	<p>А. Создать файл, содержащий сведения о сдаче студентами сессии. Структура записи: индекс группы, фамилия студента, оценки по пяти экзаменам и пяти зачетам («з» означает зачет, «н» — незачет). Количество записей—25.</p> <p>Б. Написать программу, выдающую следующую информацию: — фамилии неуспевающих студентов с указанием индексов групп и количества задолженностей; — средний балл, полученный каждым студентом группы X, и всей группой в целом</p>
8	<p>А. Создать файл, содержащий сведения о личной коллекции книголюбца. Структура записи: шифр книги, автор, название, год издания, местоположение (номер стеллажа, шкафа и т. п.). Количество записей произвольное.</p> <p>Б. Написать программу, выдающую следующую информацию: — местонахождение книги автора X названия Y. Значения X, Y ввести с терминала; — список книг автора Z, находящихся в коллекции; — число книг издания XX года, имеющееся в библиотеке</p>
9	<p>А. Создать файл, содержащий сведения о наличии билетов и рейсах Аэрофлота. Структура записи: номер рейса, пункт назначения, время вылета, время прибытия, количество свободных мест в салоне. Количество записей произвольное</p> <p>Б. Написать программу, выдающую информацию следующего вида: — время отправления самолетов в город X; — наличие свободных мест на рейс в город X с временем отправления Y. Значения X, Y вводятся по запросу с терминала</p>

Таблица 7.1 Продолжение

Вариант задания	Условие задачи
10	<p>А. Создать файл, содержащий сведения об ассортименте обуви в магазине фирмы. Структура записи: артикул, наименование, количество, стоимость одной пары. Количество записей произвольное. Артикул начинается с буквы Д для дамской обуви, М для мужской, П для детской.</p> <p>Б. Написать программу, выдающую следующую информацию: — о наличии и стоимости обуви артикула X; — ассортиментный список дамской обуви с указанием наименования и имеющегося в наличии числа пар каждой модели</p>
11	<p>А. Создать два файла, содержащих сведения о десяти нападающих хоккейных команд «Динамо» и «Спартак» соответственно: имена нападающих, число заброшенных ими шайб, сделанных голевых передач, заработанное штрафное время.</p> <p>Б. Написать программу, которая по данным, извлеченным из этих файлов, создает новый третий файл, содержащий имя, команду, сумму очков (голы + передачи) для шести лучших игроков обеих команд. Имена и показатели результативности хоккеистов вывести на экран</p>
12	<p>А. Создать файл, содержащий сведения о том, какие из пяти предлагаемых дисциплин по выбору желает слушать студент. Структура записи: фамилия студента, индекс группы, 5 дисциплин, средний балл успеваемости. Выбираемая дисциплина отмечается символом 1, иначе — пробел. Количество записей — 25.</p> <p>Б. Написать программу, которая печатает список студентов, желающих прослушать дисциплину X. Если число желающих превысит 8 человек, то отобрать студентов, имеющих более высокий средний балл успеваемости</p>
13	<p>А. Создать файл, содержащий сведения об отправлении поездов дальнего следования с Казанского вокзала. Структура записи: номер поезда, станция назначения, время отправления, время в пути, наличие билетов. Количество записей произвольное.</p> <p>Б. Написать программу, которая позволяет получить следующую справочную информацию: — время отправления поездов в город X во временном интервале от Л до б часов, — наличие билетов на поезд с номером XXX</p>

Таблица 7.1 Окончание

Вариант задания	Условие задачи
14	<p>А. Создать файл, содержащий сведения о сотрудниках института. Структура записи: фамилия работающего, название отдела, год рождения, стаж работы, должность, оклад. Количество записей произвольное.</p> <p>Б. Написать программу, которая позволяет получить следующую информацию: — список сотрудников пенсионного возраста на сегодняшний день с указанием стажа работы, — средний стаж работающих в отделе X</p>
15	<p>А. Создать файл, содержащий сведения о пациентах глазной клиники. Структура записи: фамилия пациента, пол, возраст, место проживания (город), диагноз. Количество записей произвольное.</p> <p>Б. Написать программу, выдающую следующую информацию: — количество иногородних, прибывших в клинику, — список пациентов старше X лет с диагнозом Y. Значения X и Y ввести с терминала</p>

Задание Б. Выполнить на ЭВМ программу обработки файла, созданного в п. 1, в соответствии с заданием, указанным в п.п. Б табл. 7.1 для заданного варианта.

Проверить правильность выполнения обеих программ с помощью тестового варианта исходных данных.

Контрольные вопросы

1. Объяснить, что означают следующие термины: файл, запись, метод доступа, структура записи?
2. Каково назначение операторов открытия и закрытия файлов на языке программирования Object Pascal?
3. Допустимы ли различные типы данных для элементов одной записи?
4. Указать, с помощью каких операторов выполняется запись данных в файл последовательного доступа, чтение из файла?
5. Привести примеры использования файлов последовательного доступа.
6. Как распознать конец файла данных? Как распознать файл на диске?

Лабораторная работа №8

Создание графического изображения с помощью стандартных функций.

Цель работы: изучение возможностей языка программирования в среде DELPHI для организации графики как с помощью графических примитивов, так и с помощью сервисных визуальных средств.

Домашнее задание:

1. Изучить для графических компонентов классов TForms, TImage, TPaintBox использование канвы (свойство Canvas) и процедур, реализующих отображение графических примитивов на канве.
2. Изучить свойства и методы визуального компонента класса TChart для вывода графиков функций.

Порядок выполнения работы:

1. Программно исследовать область определения заданной функции (см. вариант).
2. С учетом проведенных исследований (точек разрыва) протабулировать функцию на заданном отрезке изменения аргумента от $X_{\text{начального}}$ до $X_{\text{конечного}}$ ($X_{\text{н}}$ и $X_{\text{к}}$ вводить в процессе работы программы).
3. Сохранить полученную таблицу значений функции в текстовом файле (выбор имени файла осуществлять динамически с помощью компонент страницы VCL DIALOGS).
4. Построить график заданной функции:
 - а) с использованием визуального компонента класса TChart;
 - б) на канве (свойство Canvas) компонента класса TPaintbox или класса TImage.
5. Продемонстрировать все результаты работы преподавателю.
6. Оформить отчет и защитить лабораторную работу преподавателю.

Варианты заданий для выполнения работы:

1. $y = (x+3)/(x-2)$;
2. $y = 3 + 2/x + 3/(x*x)$;
3. $y = 1/(3*x*x + 2*x + 1)$;
4. $y = 1/(x*x + 2*x + 1)$;
5. $y = 1/(x*x + 3*x + 1)$;
6. $y = x/(3*x*x + 2*x + 1)$;
7. $y = x/(x*x + 2*x + 1)$;
8. $y = x/(x*x + 3*x + 1)$;
9. $y = x/(x*x - 2*x + 1)$;
10. $y = x/(x*x - 3*x + 1)$;

Замечание.

При построении графика функции на канве необходимо прочертить оси координат с удобным началом координат, поставить единичные отрезки на осях с учетом выбранного масштаба и затем на координатной плоскости нарисовать линию, соответствующую графику заданной функции в заданной области определения: от $X_{\text{начального}}$ до $X_{\text{конечного}}$.

Контрольные вопросы:

- 1.Расскажите о возможностях компонента класса TImage.
- 2.Визуальный компонент класса TPaintBox-- его назначение, свойства и методы.
- 3.Опишите возможности компонента класса TChart.

Приложение 1

Справка по работе с основным набором визуальных компонентов Delphi, требуемых при выполнении лабораторных работ

TLabel

Компоненты класса *TLabel* (метки) предназначены для размещения на форме различного рода текстовых надписей. С компонентом может быть связан оконный управляющий элемент, который выбирается при нажатии *Alt+Буква*, где *Буква* – выделенная подчеркиванием буква в тексте метки. Такие символы в терминологии *Windows* называются акселераторами.

Свойства компонента:

property AutoSize: Boolean;	Указывает, будет ли метка изменять свои размеры в зависимости от помещенного в ее свойство <i>Caption</i>
property FocusControl: TWinControl;	Содержит имя оконного компонента, который связан с меткой акселератором.
TTextLayout = (tlTop, tlCenter, tlBottom); property Layout: TTextLayout;	Определяет выравнивание текста по вертикали относительно границ метки: <i>tlTop</i> – текст располагается вверху; <i>tlCenter</i> – текст центрируется по вертикали; <i>tlBottom</i> – текст располагается внизу.
property ShowAccelChar: Boolean;	Если содержит <i>True</i> , символ <i>&</i> в тексте метки предшествует символу-акселератору.
property Transparent: Boolean;	Определяет прозрачность фона метки. Если <i>False</i> , фон закрашивается собственным цветом <i>Color</i> , в противном случае используется фон родительского компонента.
property Wordwrap: Boolean;	Разрешает/запрещает разрыв строки на границе слова. Для вывода многострочных надписей задайте <i>AutoSize = False</i> , <i>Wordwrap = True</i> и установите подходящие размеры метки.

TEdit

Компонент класса *TEdit* представляет собой однострочный редактор текста. С его помощью можно вводить и/или отображать достаточно длинные текстовые строки.

Свойства компонента:

property AutoSelect: Boolean;	Указывает, будет ли выделяться весь текст в момент получения компонентом фокуса ввода.
property AutoSize: Boolean;	Если <i>True</i> и <i>BorderStyle = bsSingle</i> , высота компонента автоматически меняется при изменении свойства <i>Font.Size</i> .
TBorderStyle = bsNone..bsSingle; property BorderStyle: TBorderStyle;	Определяет стиль обрамления компонента: <i>bsNone</i> - нет обрамления; <i>bsSingle</i> - компонент обрамляется одной линией.
TEditCharCase = (ecNormal, ecUpperCase, ecLowerCase); property CharCase: TEditCharCase;	Определяет автоматическое преобразование высоты букв: <i>ecNormal</i> – нет преобразования; <i>ecUpperCase</i> - все буквы заглавные; <i>ecLowerCase</i> - все буквы строчные. Правильно работает с кириллицей.
property HideSelection: Boolean;	Если <i>False</i> , выделение текста сохраняется при потере фокуса ввода.
property MaxLength: Integer;	Определяет максимальную длину текстовой строки. Если имеет значение 0, длина строки не ограничена.
property Modified: Boolean;	Содержит <i>True</i> , если текст был изменен.
property OnChange: TNotifyEvent;	Определяет обработчик события <i>OnChange</i> , которое возникает после любого изменения текста.
property OEMConvert: Boolean;	Содержит <i>True</i> , если необходимо перекодировать текст из кодировки <i>MS-DOS</i> в кодировку <i>Windows</i> и обратно.
property PasswordChar: Char;	Если символ <i>PasswordChar</i> определен, он заменяет собой любой символ текста при отображении в окне. Используется для ввода паролей.
property ReadOnly: Boolean;	Если содержит <i>True</i> , текст не может изменяться.
property SelLength: Integer;	Содержит длину выделенной части текста.
property SelStart: Integer;	Содержит номер первого символа выделенной части текста.
property Text: String;	Содержит текст.

Методы компонента:

procedure Clear;	Удаляет весь текст.
procedure ClearSelection;	Удаляет выделенный текст.
procedure CopyToClipboard;	Копирует выделенный текст в <i>Clipboard</i> .
procedure CutToClipboard;	Копирует выделенный текст в <i>Clipboard</i> , после чего удаляет выделенный текст из компонента.
function GetSelTextBuf(Buffer: PChar; BufSize: Integer): Integer;	Копирует не более <i>BufSize</i> символов выделенного текст в буфер <i>Buffer</i> .
procedure PasteFromClipboard;	Заменяет выделенный текст содержимым <i>Clipboard</i> , а если нет выделенного текста, копирует содержимое <i>Clipboard</i> в позицию текстового курсора.
procedure SelectAll;	Выделяет весь текст.
procedure SetSelTextBuf(Buffer:PChar);	Заменяет выделенный текст содержимым <i>Buffer</i> , а если нет выделенного текста, копирует содержимое <i>Buffer</i> в позицию текстового курсора.

TButton

Кнопки *TButton* широко используются для управления программами.

Свойства компонента:

property Cancel: Boolean;	Если имеет значение <i>True</i> , событие <i>OnClick</i> кнопки возникает при нажатии клавиши <i>Esc</i> .
property Default: Boolean;	Если имеет значение <i>True</i> , событие <i>OnClick</i> кнопки возникает при нажатии клавиши <i>Enter</i> .
TModalResult = Low(Integer)..High(Integer); property ModalResult: TModalResult;	Определяет результат, с которым было закрыто модальное окно (см. ниже пояснение).

В терминологии *Windows* модальными окнами называются такие специальные окна, которые, раз появившись на экране, блокируют работу пользователя с другими окнами вплоть до своего закрытия. Обычно с их помощью реализуется диалог, требующий от пользователя принятия некоторого решения. Для этого в состав модального окна включается несколько кнопок. Если у кнопки определено свойство *ModalResult*, нажатие на нее приводит к закрытию модального окна и возвращает в программу значение *ModalResult* как результат диалога с пользователем. В *Delphi* определены следующие стандартные значения *ModalResult*:

mrNone	Модальное	окно	не	mrIgnore	Была нажата кнопка <i>Ignore</i> .
--------	-----------	------	----	----------	------------------------------------

mrOk	закрывается. Была нажата кнопка <i>OK</i> .	mrYes	Была нажата кнопка <i>Yes</i> .
mrCancel	Была нажата кнопка <i>Cancel</i> .	mrNo	Была нажата кнопка <i>No</i> .
mrAbort	Была нажата кнопка <i>Abort</i> .	mrAll	Была нажата кнопка <i>All</i> .
mrRetry	Была нажата кнопка <i>Retry</i> .		

TCheckBox

Независимый переключатель *TCheckBox* используется для того, чтобы пользователь мог указать свое решение типа *Да/Нет* или *Да/Нет/Не знаю*. Это решение отражается в свойстве *State* компонента, доступном как для чтения, так и для записи. В составе диалогового окна может быть несколько компонентов *TCheckBox*. Состояние любого из них не зависит от состояния остальных, поэтому такие переключатели называются независимыми.

Свойства компонента:

<i>TLeftRight</i> = (<i>taLeftJustify</i> , <i>taRightJustify</i>); property <i>Alignment</i> : <i>TLeftRight</i> ;	Определяет положение текста: <i>taLeftJustify</i> – с левой стороны компонента; <i>taRightJustify</i> – с правой стороны.
property <i>AllowGrayed</i> : Boolean;	Разрешает/запрещает использование состояния <i>cbGrayed</i> (<i>Не знаю</i>).
property <i>Checked</i> : Boolean;	Содержит выбор пользователя типа <i>Да/Нет</i> . Состояния <i>cbUnchecked</i> и <i>cbGrayed</i> отражаются как <i>False</i> .
<i>TCheckBoxState</i> = (<i>cbUnchecked</i> , <i>cbChecked</i> , <i>cbGrayed</i>); property <i>State</i> : <i>TCheckBoxState</i> ;	Содержит состояние компонента: <i>cbUnchecked</i> – нет; <i>cbChecked</i> – да; <i>cbGrayed</i> – не знаю.

TRadioButton

В отличие от *TCheckBox*, компоненты *TRadioButton* представляют собой зависимые переключатели, предназначенные для выбора одного из нескольких взаимоисключающих решений. На форму (точнее, в компонент-контейнер) помещается по меньшей мере два таких компонента. Они могут находиться в одном из двух состояний, определяемом свойством *Checked*. Если в одном компоненте это свойство принимает значение *True*, во всех других компонентах, расположенных в том же контейнере, свойства *Checked* принимают значения *False*.

Помимо свойства *Checked* компонент *TRadioButton* имеет еще одно специфичное свойство – *Alignment*, аналогичное такому же свойству *TCheckBox*.

ТМето

Обойтись простым текстовым полем удастся не всегда. Если пользователь должен ввести большой объем информации (например полный почтовый адрес или произвольный комментарий), ему может понадобиться несколько строк текста. В таком случае следует использовать компонент ТМето.

При вводе текста для перехода на новую строку (к новому абзацу) обычно используется клавиша ENTER. Однако в диалоговых окнах Windows эта клавиша часто применяется для завершения ввода. Способ использования клавиши ENTER определяется значением свойства *WantReturns*. Если оно имеет значение *true*, то клавиша ENTER позволяет переходить к новой строке внутри текстовой области, в противном случае она служит для завершения ввода и перехода к следующему элементу управления, а для перехода к новой строке применяется комбинация клавиш CTRL+ENTER.

Главное свойство данного компонента – *Lines (Строки)*, имеющее тип TStrings. В нем хранится список строк, введенных пользователем. Эти строки можно обрабатывать всеми методами, доступными в классе TStrings, например сохранять в файле:

```
Memo1.Lines.SaveToFile('C:\Memo.TXT') ;
```

Наличие у текстовой области полос прокрутки задается в свойстве *ScrollBars*.

Значение	Вид текстовой области
ssNone	Полосы прокрутки отсутствуют
ssHorizontal	Имеется горизонтальная полоса прокрутки
ssVertical	Имеется вертикальная полоса прокрутки
ssBoth	Имеются две полосы прокрутки

Если включена горизонтальная полоса прокрутки, значение свойства *WordWrap* игнорируется. Это свойство определяет, будет ли выполняться автоматический перенос слов на новую строку при достижении правой границы области (при этом никаких символов новой строки в текст не добавляется – перенос отображается только на экране).

При выделении фрагмента текста в текстовой области в свойство *SelStart* записывается позиция первого выделенного символа, а в свойство *SelLength* – число выделяемых символов. Выделенный текст доступен через свойство *SelText* (тип *string*).

Для выделения всего текста применяется метод *SelectAll*, для удаления выделенного текста – метод *ClearSelection*.

Чтобы очистить содержимое текстовой области, используется метод *Clear*, чтобы отменить последние изменения – метод *Undo*, а чтобы очистить буфер, хранящий историю изменений, и сделать такую отмену невозможной – метод *ClearUndo*.

Группа методов предназначена для работы с буфером обмена Windows. Для копирования выделенного текста в буфер обмена применяется метод

CopyToClipboard, для вырезания текста – метод *CutToClipboard*, для вставки текста из буфера – метод *PasteFromClipboard*.

Когда в текстовой области происходит изменение текста, генерируется событие *OnChange*.

TListBox

С помощью компонента список (*TListBox*) пользователь может выбрать один или несколько его элементов. Если элементов много и они не умещаются в отведенной для них области, то автоматически добавляется вертикальная полоса прокрутки. Элементы списка можно добавлять или удалять динамически.

Свойство *Items* имеет тип *TStrings* и содержит список строк, выводимых на экран. Во время разработки программист может редактировать свойство *Items* с помощью редактора списка строк (*String List editor*). Во время выполнения элементы списка можно добавлять с помощью метода *Add ()* класса *TStrings*, а удалять – с помощью метода *Delete ()* этого же класса. Свойство *ItemIndex* содержит порядковый номер выделенного элемента, для первого элемента значение *ItemIndex* равно 0. Если ни один элемент не выделен, то значение *ItemIndex* равно – 1. Свойство *MultiSelect* определяет, может ли пользователь выделить более одного элемента списка. Если значение *MultiSelect* равно *True*, то свойство *ItemIndex* является индексом выделенного элемента, имеющего фокус. По умолчанию *ItemIndex* имеет значение 0. Свойство *SelCount* содержит количество выделенных элементов, а свойство *Selected* определяет, выделен ли конкретный элемент. Свойство *Sorted* определяет, будут ли элементы выведены в алфавитном порядке. Свойство *Style* задает внешний вид элементов. По умолчанию элементы выводятся в виде строк, однако, изменив свойство *Style*, их можно вывести в графическом виде или с разной высотой. Ниже приведен код, иллюстрирующий использование списка.

TComboBox

Компонент поле со списком (*TComboBox*) объединяет возможности поля ввода и прокручиваемого раскрывающегося списка. Пользователь может или выбрать элемент списка или ввести его непосредственно в поле ввода. Поэтому поле со списком обладает многими свойствами и методами, входящими в рассмотренные раньше компоненты *TEdit* и *TListBox*. В поле со списком свойство *Style* определяет внешний вид списка и наличие поля ввода

Заменим в предыдущем примере список полем со списком. В этом случае для добавления нового элемента пользователь должен ввести его название в поле ввода и нажать клавишу *Enter*. Приведенный ниже код иллюстрирует применение поля со списком.

Компонент Окно выбора файла (TOpenDialog)

Компонент предназначен для выбора файла с целью последующего открытия.

Свойства класса *TOpenDialog*:

Свойство	Назначение
DefaultExt	Расширение имени, используемое по умолчанию. Добавляется в конец выбранного пользователем имени файла, если расширение не указано явно
FileName	Выбранное пользователем имя файла вместе с полным путем поиска
Files	Список выбранных имен файлов. В свойстве <i>Options</i> должен быть включен флажок <i>ofAllowMultiSelect</i>
Filter	Набор масок, в соответствии с которыми отбираются имена файлов для отображения в диалоговом окне. Каждая маска состоит из двух частей: названия и шаблона, – разделенных символом . Одному названию могут соответствовать несколько шаблонов. Маски отделяются друг от друга символом
FilterIndex	Номер текущей маски. Нумерация начинается с 1
HistoryList	Список ранее выбранных файлов (тип <i>TStrings</i>)
InitialDir	Текущий каталог, содержимое которого отображается при первом открытии диалогового окна
Options	Набор флажков, определяющих работу окна выбора файлов
Title	Заголовок диалогового окна

Среди методов этого класса следует отметить функцию

function GetStaticRect: TRect;

Она возвращает координаты прямоугольной области диалогового окна (часть клиентской области), зарезервированной для нужд разработчика (например для отображения содержимого текущего выбранного файла).

События класса *TOpenDialog*:

Событие	Условие генерации
OnCanClose	Пользователь пытается закрыть диалоговое окно. Обработчик этого события позволяет проконтролировать правильность выбранного или введенного в соответствующее поле окна имени файла и разрешить или запретить закрытие
OnFolderChange	Пользователь переключился в другой каталог
OnIncludeItem	К текущему списку файлов в диалоговом окне будет добавлено новое имя. Обработчик данного события дает возможность отбирать допустимые имена по алгоритму, определяемому программистом

OnSelectionChange	Пользователь выбрал новое имя файла в диалоговом окне
OnTypeChange	Пользователь выбрал новую маску файлов (свойство Filter)

В следующем примере при щелчке на кнопке отображается диалоговое окно выбора имени файла (объект `OpenDialog1`), которое имеет заголовок Выбор нужного файла, а в списке отображаются все файлы, имеющие расширение .PAS. Это обеспечивается присвоением свойству `FilterIndex` значения 2.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  OpenDialog1.Filter := 'Все файлы (* . *) |*.*|Файлы Паскаля
  (*.pas)|*.PAS';
  OpenDialog1.Title := 'Выбор нужного файла';
  OpenDialog1.FilterIndex := 2;
  if OpenDialog1.Execute then
    begin
      AssignFile(F, OpenDialog1.FileName);
      // работа с файлом F
    end; end;

```

Компонент Окно сохранения файла (TSaveDialog)

Этот компонент практически ничем не отличается от компонента `TOpenDialog` за исключением некоторых настроек, специфичных для процесса сохранения файла.

TStringGrid

Использование многими пользователями электронных таблиц типа Excel стало практически неотъемлемой частью применения компьютеров. В системе Delphi 5 имеются два компонента, которые позволяют до некоторой степени симитировать работу электронной таблицы, оставляя при этом, конечно, основную работу по реализации конкретных функций такой таблицы программистам.

Первый компонент – это таблица строк, позволяющая работать с текстовой информацией в двумерной таблице, имеющей столбцы и строки (их размеры можно менять с помощью мыши). Дополнительно, к каждой ячейке таблицы можно «привязать» свой объект, характеристики которого программист представит в виде строки, расположенной в этой ячейке.

Основное свойство таблицы строк – это двумерный массив `Cells`, позволяющий обращаться к содержимому ячеек и изменять их содержимое. Первое измерение – это номер строки, второе – номер столбца.

Замечание: Нумерация элементов в таблице строк начинается с нуля.

Число столбцов задается в свойстве *ColCount*, число строк – в свойстве *RowCount*.

Следующий код программы изменяет размер таблицы, помещенной на форму в режиме проектирования (по умолчанию принят размер 5x5 элементов), на размер 10x10 ячеек и заполняет ячейки строками, содержащими их координаты.

```
procedure TForm1.FormCreate(Sender: TObject);
  var c,r: integer; begin
  StringGrid1.ColCount := 10; StringGrid1.RowCount := 10;
  for c := 0 to StringGrid1.ColCount-1 do for r := 0 to
  StringGrid1.RowCount-1 do StringGrid1.Cells[c,r] :=
  '(' + IntToStr(c) + ',' + IntToStr(r) + ')'; end;
```

Обратите внимание, что левый столбец и верхняя строка, хотя и содержат текстовую информацию, фактически являются заголовочными областями. Использовать их наравне с другими ячейками не совсем правильно.

Замечание: Число строк и столбцов, имеющих характер заголовка, задается свойствами *FixedCols* и *FixedRows*. Если таблица не содержит заголовочной информации, эти свойства должны принимать значение 0.

Можно получить доступ ко всем элементам одного столбца или одной строки. Соответствующие свойства *Col* и *Row* имеют тип *TStrings*, позволяющий обращаться к нужному элементу по номеру.

Чтобы привязать к ячейке объект (наследник класса *TObject*), надо использовать свойство *Objects*, представляющее собой такой же массив, как *Cells*, но содержащий не строки, а объекты. Эти объекты должны создаваться, а также уничтожаться программистом вручную, иными словами, весь контроль над состоянием этого массива полностью возлагается на разработчика. Свойство *Objects* предоставляет только доступ к нужному объекту.

Свойства, предназначенные для оформления таблицы строк:

Свойство	Назначение
BorderStyle	Стиль отрисовки ячейки. Значение этого свойства можно комбинировать со значением свойства Ctrl3D для получения оригинального вида ячеек
Ctrl3D	Таблица представляется в «трехмерном» стиле
ColWidths	Массив, хранящий ширину каждого столбца в пикселах
DefaultColWidth	Начальная ширина столбца по умолчанию
DefaultDrawing	Если значение свойства – <i>true</i> , производится автоматическая отрисовка содержимого каждой ячейки. В противном случае для таблицы необходимо определить обработчик события <i>OnDrawCell</i> , чтобы запрограммировать процесс отрисовки ячейки

DefaultRowHeight	Начальная высота строки по умолчанию
FixedColor	Цвет области строк и столбцов, служащих заголовком таблицы
GridHeight	Высота всей таблицы (в пикселах)
GridLineWidth	Ширина (в пикселах) линий, разделяющих ячейки таблицы
GridWidth	Ширина всей таблицы (в пикселах)
Options	Множество значений (тип <i>set of TGridOption</i>), позволяющее задавать различные режимы работы таблицы: выделение нескольких ячеек, способ использования линий прокрутки и прочие. В частности, чтобы разрешить изменение размеров строк и столбцов, надо установить значение true для свойств <i>goRowSizing</i> и <i>goColSizing</i> , вложенных в свойство <i>Options</i> . Чтобы разрешить редактирование содержимого ячеек, надо записать значение true в подсвойство <i>goEditing</i>
RowHeights	Массив, хранящий высоту каждой строки в пикселах
ScrollBars	Наличие полос прокрутки
VisibleColCount	Число видимых в таблице столбцов (без области заголовка)

TImage

Данный компонент активно используется во многих программах, причем не только для отображения статических картинок, но и для создания различных анимационных эффектов.

В большинстве случаев содержимое изображения загружается из файла на этапе проектирования. Для этого служит свойство *Picture* (класс *TPicture*), описывающее точечное изображение (.BMP), значок, графический метафайл Windows или другой пользовательский графический ресурс. Класс *TPicture* (рисунок) не является компонентом Delphi 5, он просто входит в состав библиотеки VCL как вспомогательный, но на его основе могут быть созданы полноценные компоненты.

Текущее содержимое экземпляра класса хранится в одном из свойств: *Bitmap* (класс *TBitmap*), *Icon* (значок, класс *TIcon*) или *Metafile* (класс *TMetafile*, формат графического метафайла Windows .EMF). Обратиться к любому из этих свойств для отображения графики можно через свойство *Graphic*. Ширина и высота изображения (в пикселах) задаются в свойствах *Width* и *Height*.

После размещения объекта *Image* на форме появится пунктирная рамка, которая задает (по умолчанию) размеры будущей картинке. Эти размеры желательно заранее указать в свойствах *Width* и *Height*.

Выбрав в Инспекторе объектов свойство *Picture*, можно вызвать специальный редактор, с помощью которого можно загрузить изображения в форматах .BMP, .ICO, JPG или в одном из форматов графического метафайла Windows.

TChart

Это очень мощный и богатый возможностями компонент, разработанный Давидом Бернеда (версия, включенная в систему Delphi 5, имеет номер 4.02). Он позволяет строить красивые двух- и трехмерные диаграммы на основе различных данных, является наследником класса *TPanel* и наследует все свойства панели.

Создать диаграмму можно двумя способами: визуально с помощью Мастера (без программирования) и непосредственно средствами Паскаля.

Мастер запускается командой *File > New > Business > TeeChart Wizard* (Файл > Создать > Деловые > Мастер диаграмм), после чего разработчику надо выполнить ряд уточнений. Сначала выбирается источник данных. Пусть он не расположен в файле, а генерируется программой – переключатель *Non Database Chart* (Не на основе базы данных). Затем выбирается внешний вид диаграммы. Она может быть двумерной или трехмерной что определяется переключателем *2D/3D*.

На следующем этапе работы Мастера флажок *Show Legend* (Отображать легенду) определяет наличие *легенды* – дополнительной панели, на которой указывается соответствие цветов частей диаграммы указанным значениям. Флажок *Show Marks* включает небольшие желтые подсказки у каждой из частей диаграммы.

На этом создание диаграммы заканчивается. После щелчка на кнопке *Finish* (Готово) в Проектировщике форм появится новая форма, на которой будет расположен объект *Chart1*. Он заполнен неким набором случайно сгенерированных значений.

Настройка диаграммы выполняется с помощью редактора, который вызывается двойным щелчком на объекте *Chart1*.

Параметры отображения диаграммы в окне определяются на вкладке *Chart* (Диаграмма), состоящей в свою очередь из набора дополнительных панелей.

- ✓ Панель *Series* (Ряд данных) очень важна. Она позволяет объединять несколько диаграмм на одном графике с помощью кнопки *Add* (Добавить). При этом над значениями рядов данных можно выполнять различные операции, задаваемые на вкладке *Functions* (Функции): сложение (*Add*), вычитание (*Subtract*), умножение (*Multiply*), деление (*Divide*), взятие наибольшего (*High*), наименьшего (*Low*) или среднего (*Average*) значения.
- ✓ Панель *General* (Общие) содержит элементы управления для:
 - экспорта изображения в файл – кнопка *Export* (Экспортировать);
 - установки (в процентах) сдвига границ изображения по отношению к границам объекта – поля *Margins* (Поля);
 - масштабирования – панель *Zoom* (Масштаб);

- прокрутки – панель *Allow Scroll* (Разрешить прокрутку).
- ✓ Средства панели *Axis* (Оси) отвечают за все, что касается определения координатных осей, их масштаба, заголовков, шага пунктирной сетки и так далее.
- ✓ Панель *Titles* (Заголовки) содержит средства для оформления заголовка.
- ✓ Панель *Legend* (Легенда) используется при оформлении внешнего вида и содержимого легенды.
- ✓ Средства панели *Panel* (Панель) описывают форму и визуальное представление панели-основы, на которой расположена диаграмма.
- ✓ Панель *Pages* (Страницы) служит для разделения диаграммы на страницы. Увеличивая число точек на странице с помощью поля *Points per Page* (Точки на страницу), можно подобрать оптимальное соотношение между наглядностью диаграмм и разумным числом страниц.
- ✓ Панель *Walls* (Границы) позволяет задать цвет и размеры границ диаграммы.
- ✓ Панель 3D описывает пространственное представление трехмерных диаграмм. С помощью нескольких движков проектируемую диаграмму можно вращать и масштабировать.

На вкладке *Series* (Ряды данных) в редакторе задаются конкретные параметры оформления каждого ряда данных (каждого графика, добавленного при помощи вкладки *Chart*). Выбор текущего ряда данных производится с помощью раскрывающегося списка *Area* (Область).

Здесь наиболее важна панель *Data Source* (Источник данных). С ее помощью можно задать для ряда случайные значения (*Random Values*), отказаться от генерации значений (*No Data*) или сформировать значения текущего ряда данных как результат применения некоторой функции (раскрывающийся список *Function*) к значениям выбранных рядов данных. Выбор рядов данных – занесение в список *Selected Series* (Выбранные ряды) осуществляется с помощью кнопки >.

Добавление новой точки к серии выполняется с помощью метода *Add*, заголовок которого выглядит следующим образом.

```
function AddXY(Const AXValue, AYValue: Double; Const
  AXLabel: String; AColor: TColor) : Longint;
```

Обработчик щелчка на кнопке, например, *Button1* запишется следующим образом.

```
procedure TForm1.Button1Click(Sender: TObject); begin
  Series1.AddXY(
    StrToFloat(Edit1.Text),
    StrToFloat(Edit2.Text),
    'Эксперимент A', clRed); end;
```

TMainMenu

Компонент класса *TMainMenu* определяет главное меню формы. На форму можно поместить сколько угодно объектов этого класса, но отображаться в полосе меню в верхней части формы будет только тот из них, который указан в свойстве *Menu* формы.

После установки компонента на форму необходимо создать его опции. Для этого следует дважды щелкнуть по компоненту левой кнопкой мыши либо нажать на нем правую кнопку и выбрать продолжение *Menu Designer* в появившемся вспомогательном меню, либо, наконец, щелкнуть по кнопке в правой половине строки *Items* Инспектора Объектов.

Создание опций не вызывает проблем. Перейдите в окно Инспектора Объектов и введите текст опции в строке *Caption*, после чего нажмите *Enter* – опция готова и можно переходить к следующей. Каждая опция главного меню может раскрываться в список подопций или содержать конечную команду. Для создания подопций щелкните мышью по строке ниже опции и введите первую подопцию. Продолжайте ввод, пока не будет создан весь список подопций, после чего щелкните по пустому прямоугольнику справа от первой опции и введите вторую опцию. Процесс гораздо сложнее описать, чем выполнить.

В названиях опций можно указать символ «&» перед тем символом, который определит клавиши быстрого выбора опции.

Для создания разветвленных меню, т.е. таких, у которых подопции вызывают новые списки подопций, щелкните по подопции и нажмите *Ctrl+Вправо*, где *Вправо* – клавиша смещения курсора вправо. Такого же эффекта можно добиться после щелчка правой кнопкой мыши на подопции и выборе продолжения *Create Submenu*.

Каждый элемент меню является объектом класса *TMenuItem*. Свойства этого класса описаны в следующей таблице. Термин «родительская опция» означает опцию, выбор которой приводит к раскрытию подменю с данной опцией.

property Break: <i>TMenuBreak</i> ;	Позволяет создать многоколончатый список подменю (см. ниже).
property Checked: Boolean;	Если <i>True</i> , рядом с опцией появляется галочка.
property Command: Word;	Используется при разработке приложений, обращающихся непосредственно к API-функциям <i>Windows</i> .
property Count: Integer;	Содержит количество опций в подчиненном меню, связанном с данным элементом (только для чтения).
property Default: Boolean;	Определяет, является ли данная опция подменю умалчиваемой (см. ниже).
property GroupIndex: Byte;	Определяет групповой индекс для зависимых опций (см. свойство <i>Radioltem</i>).
property Items[Index: Integer]: <i>TMenuItem</i> ;	Позволяет обратиться к любой опции подчиненного меню по ее индексу.

property MenuItemIndex: Integer;	Определяет индекс опции в списке <i>Items</i> родительской опции.
property RadioItem: Boolean;	Определяет, зависит ли данная опция от выбора других опций в той же группе <i>GroupIndex</i> . Только одна опция группы может иметь <i>True</i> в свойстве <i>Checked</i> . Рядом с такой опцией вместо галочки изображается круг.
property Shortcut: TShortcut	Задаёт клавиши быстрого выбора данной опции.

Тип *TMenuBar* определен следующим образом:

```
type TMenuBar = (mbNone, mbBarBreak, mbBreak);
```

Свойство *Break* по умолчанию имеет значение *mbNone*. Два других возможных значений этого свойства используются для создания многоколоночатых списков подменю. Значение *mbBarBreak* заканчивает предыдущую колонку и начинает новую. Данная опция меню будет открывать новую колонку, которая отделяется от предыдущей вертикальной чертой. Значение *mbBreak* также создает новую колонку, но не вставляет разделительную черту. Эти значения игнорируются в опциях самого первого уровня. В окне конструктора меню многоколоночатое меню отображается как обычное.

Если в свойстве *Default* указано значение *True*, такая опция выделяется цветом и выбирается двойным щелчком мыши на родительской опции. Только одна опция в подменю может быть умалчиваемой.

В отличие от других компонентов, строка *Hint* для опций меню задает только расширенное сообщение, которое отображается на панели статуса.

Для элемента меню определено единственное событие *OnClick*, которое возникает при щелчке на опции или при нажатии *Enter*, если в этот момент данная опция была выбрана (подсвечена). Обработчик события становится доступен после двойного щелчка на опции в окне конструктора меню.

Список использованной литературы

1. Фаронов В.В. Delphi 7. Учебный курс. СПб, Питер, 2005.
2. Бобровский С. Delphi 5. Учебный курс. СПб, Питер, 2000.
3. Амосов А.А., Дубинский Ю.А., Копчёнова Н.В. Вычислительные методы для инженеров. – М.: Высшая школа, 1994.
4. Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране. – М.: Мир, 1977.
5. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989.
6. Турчак Л.И. Основы численных методов. – М.: Наука, 1987.

