

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ  
«ЧЕРЕМХОВСКИЙ ГОРНОТЕХНИЧЕСКИЙ КОЛЛЕДЖ  
ИМ. М.И. ЩАДОВА»**

**РАССМОТРЕНО**

на заседании ЦК  
«Информатики и ВТ»  
«04» февраля 2025 г.  
«Протокол № 6  
Председатель: Коровина Н.С.

**УТВЕРЖДАЮ**

Зам. директора  
О.В. Папанова  
« 26» мая 2025 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по практическим занятиям студентов  
учебной дисциплины

***ОП. 06 Основы алгоритмизации и программирования***

***09.02.07 Информационные системы и программирование***

Разработал:  
Н.С. Коровина

2025 г.

## СОДЕРЖАНИЕ

	<b>СТР.</b>
1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ	4
3. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ ЗАНЯТИЙ	5
4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	97
5. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЁННЫХ В МЕТОДИЧЕСКИЕ УКАЗАНИЯ	99

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по практическим занятиям учебной дисциплины «**Основы алгоритмизации и программирования**» составлены в соответствии с учебным планом и рабочей программы дисциплины по специальности **09.02.07 Информационные системы и программирование**

Цель проведения практических занятий: формирование практических умений, необходимых в последующей профессиональной и учебной деятельности.

Методические указания практических занятий являются частью учебно-методического комплекса по учебной дисциплине и содержат:

- тему занятия (согласно тематическому плану учебной дисциплины);
- цель;
- оборудование (материалы, программное обеспечение, оснащение, раздаточный материал и др.);
- методические указания (изучить краткий теоретический материал по теме практического занятия);
- ход выполнения;
- форму отчета.

В результате выполнения полного объема заданий практических занятий студент должен **уметь**:

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

При проведении практических работ применяются следующие технологии и методы обучения: проблемно-поисковых технологий, тестовые технологии, метод проектов.

### **Оценка выполнения практических занятий**

**«Отлично»** - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

**«Хорошо»** - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

**«Удовлетворительно»** - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

**«Неудовлетворительно»** - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

В соответствии с учебным планом и рабочей программы дисциплины **«Основы алгоритмизации и программирования»** на практические (лабораторные) занятия отводится **76 часов**.

## 2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

№ п/п	Тема практических занятий	Количество часов
1.	Составление программ линейной структуры.	2
2.	Составление программ разветвляющейся структуры.	2
3.	Составление программ циклической структуры.	2
4.	Знакомство со средой программирования.	2
5.	Создание проектов с использованием условного оператора if/else.	2
6.	Создание проектов с использованием оператора выбора switch и перечислений.	2
7.	Создание проектов создание циклической структуры (for)	2
8.	Создание проектов создание циклической структуры (while)	2
9.	Создание проектов создание циклической структуры (do)	2
10.	Обработка одномерных массивов	2
11.	Обработка двумерных массивов.	2
12.	Работа со строками.	2
13.	Работа с данными типа множество.	2
14.	Работа с файлами.	2
15.	Типизированные и не типизированные файлы.	2
16.	Организация процедур.	2
17.	Организация функций.	2
18.	Применение рекурсивных функций.	2
19.	Программирование модуля. Создание библиотеки подпрограмм.	2
20.	Использование указателей для организации связанных списков.	2
21.	Объявление и использование классов.	2
22.	Объявление и использование классов с использованием свойств.	2
23.	Создание наследованных классов.	2
24.	Изучение интегрированной среды разработчика.	2
25.	Создание проекта с использованием компонентов для работы с текстом.	2
26.	Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.	2
27.	События компонентов (элементов управления), их сущность и назначение.	2
28.	Создание процедур на основе событий.	2
29.	Создание проекта с использованием кнопочных компонентов.	2
30.	Создание проекта с использованием компонентов стандартных диалогов и системы меню.	2
31.	Средства прорисовки контурного изображения.	2
32.	Создание анимации.	2
33.	Разработка интерфейса приложения, разработка функциональной схемы работы приложения.	2
34.	Разработка игрового приложения.	4

35.	Тестирование и отладка приложений.	2
36.	Перегрузка методов.	4

### 3. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

#### Практическое занятие № 1

**Тема:** Составление программ линейной структуры.

**Цель:** Изучение способов задания алгоритмов, приобретение практических навыков составления блок - схем для программ с линейной структурой.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Описать исходные, выходные и промежуточные данные следующих задач:

Задача 1. Даны стороны прямоугольника  $a$  и  $b$ . Найти его периметр  $p$ , и площадь  $s$  этого прямоугольника.

Задача 2. Скорость звездолета 100 км/час. Звездолет летит до некоторой звезды, свет от которой до Земли идет 14 минут. Определить, сколько времени потребуется звездолету, чтобы долететь до данной звезды.

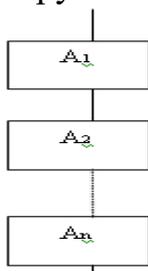
Задача 3. В треугольнике ABC известны длины сторон  $a$ ,  $b$ ,  $c$ . Вычислить длину высоты, проведенной из вершины A.

$$h = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a}$$

Задача 4. Студент сдал четыре экзамена  $a$ ,  $b$ ,  $c$ ,  $d$  и получил оценку по каждому из экзаменов по пятибалльной шкале. Определить средний балл студента.

**Задание 2.** Составить блок-схему в тетради для задач 1-5, используя основные алгоритмические конструкции: следование.

Структура следования - представляет собой последовательность размещенных блоков или групп блоков друг за другом.

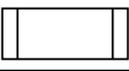


$A_i$  - функциональный блок

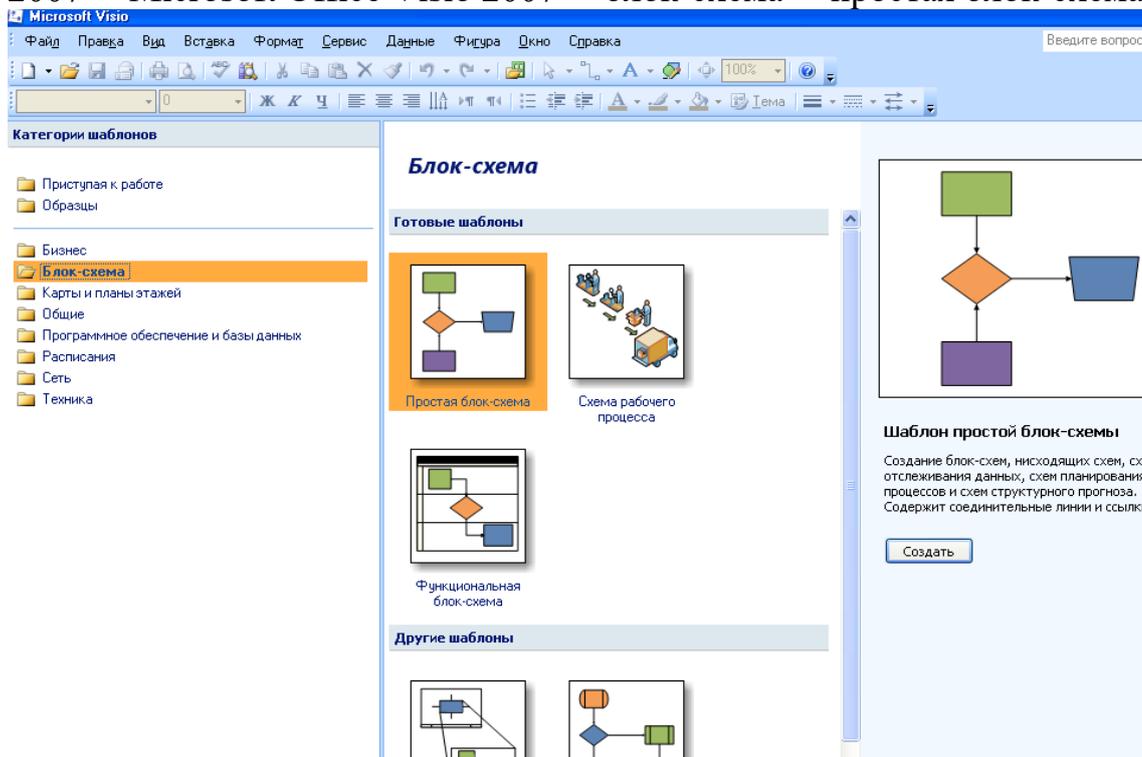
$i = \overline{1..n}$

Функциональный блок – это любая базовая структура или их комбинация

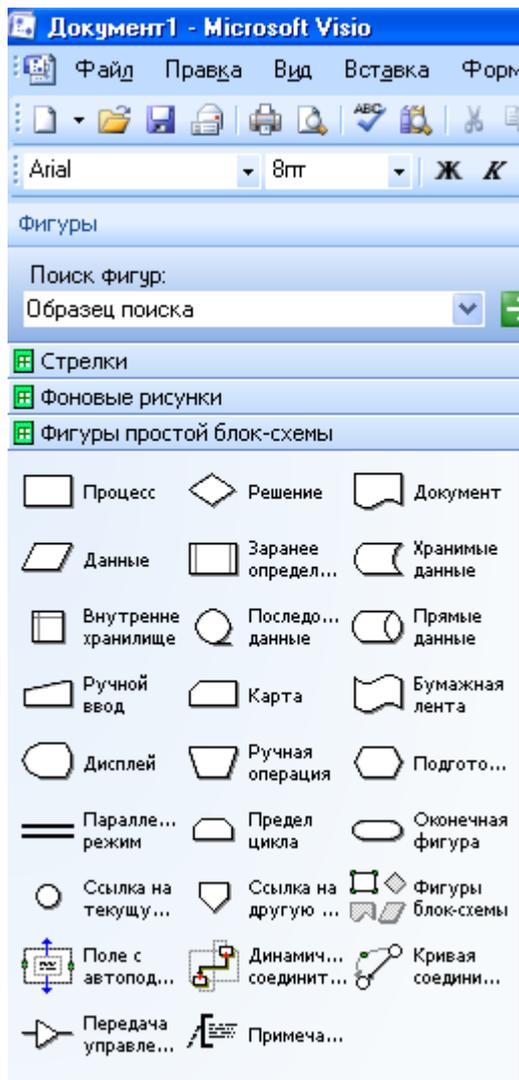
элемент	наименование	содержание
	терминатор	Начало или конец алгоритма
	данные	Общие обозначения ввода и вывода данных
	блок обработки (арифметический блок)	Вычислительные действия или последовательность действий
	логический	Выбор направления выполнения алгоритма в зависимости от некоторого условия

	Блок цикла с параметром (счетчик)	Функция выполняет действия, изменяющие пункты (например, заголовок цикла) алгоритма
	внутристраничный соединитель	Указание связи прерванными линиями между потоками информации в пределах одного листа
	межстраничные соединения	Указание связи между информацией на разных листах
	вспомогательный (подпрограмма)	Вычисление по стандартной программе или подпрограмме
---- ( )	комментарии	для внесения пометок

**Задание 3.** Запустить MS Visio 2007: пуск→все программы→ Microsoft Office 2007→ Microsoft Office Visio 2007→ блок-схема →простая блок-схема→создать.



**Задание 4.** Составить блок-схему решения для задач 1-5, используя основные алгоритмические конструкции в программе MS Visio.



Задача 1. Дана сторона квадрата  $a$ . Найти его периметр  $P = 4 \cdot a$ .

Задача 2. Дана сторона квадрата  $a$ . Найти его площадь  $S = a^2$ .

Задача 3. Даны стороны прямоугольника  $a$  и  $b$ . Найти его площадь  $S = a \cdot b$  и периметр  $P = 2 \cdot (a + b)$ .

Задача 4. Даны стороны треугольника  $a, b, c$ . Найти периметр треугольника

Задача 5. Дано два числа. Если они равны, то найти их сумму, иначе найти их произведение.

**Задание 5.** Описать исходные, выходные и промежуточные данные и составить блок-схему решения для задачи по варианту в программе MS Visio.

варианта	1-ая буква фамилии	задача
1	А, Б, В	<p>Дан диаметр окружности <math>d</math>.            Найти ее длину <math>L = \pi \cdot d</math>. В            качестве значения <math>\pi</math>            использовать 3.14.</p>

2	Г, Д, Е,	Дана длина ребра куба $a$ . Найти объем куба $V = a^3$ и площадь его поверхности $S = 6 \cdot a^2$ .
3	Ж, З, И	Даны длины ребер $a, b, c$ прямоугольного параллелепипеда. Найти его объем $V = a \cdot b \cdot c$ и площадь поверхности $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$ .
4	К, Л	Найти длину окружности $L$ и площадь круга $S$ заданного радиуса $R$ : $L = 2 \cdot \pi \cdot R$ , $S = \pi \cdot R^2$ .
5	Н, О, П	Даны два числа $a$ и $b$ . Найти их <i>среднее арифметическое</i> : $(a + b) / 2$ .
6	Р, С, Т	Даны два неотрицательных числа $a$ и $b$ . Найти их <i>среднее геометрическое</i> , то есть квадратный корень из их произведения: $\sqrt{a \cdot b}$ .
7	У, Ф, Х	Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.
8	Ч, Ш, Щ	Даны катеты прямоугольного треугольника $a$ и $b$ . Найти его гипотенузу $c$ и периметр $P$ : $c = \sqrt{a^2 + b^2}$ , $P = a + b + c$ .
9	Э, Ю, Я	Дана длина $L$ окружности. Найти ее радиус $R$ и площадь $S$ круга, ограниченного этой окружностью, учитывая, что $L = 2 \cdot \pi \cdot R$ , $S = \pi \cdot R^2$ . В качестве значения $\pi$ использовать 3.14.
10	Ё, М	Найти расстояние между двумя точками с заданными координатами $x_1$ и $x_2$ на числовой оси: $ x_2 - x_1 $ .

**Задание 6.** Описать исходные, выходные и промежуточные данные и составить блок-схему решения для задачи по варианту в программе MS Visio.

Вариант	Задача
1	Найти длину окружности и площадь круга
2	Найти площадь поверхности цилиндра
3	Найти площадь поверхности и объем правильного тетраэдра
4	Найти площадь поверхности и объем октаэдра
5	Найти площадь поверхности конуса
6	Найти площадь поверхности и объем шара
7	Найти площадь поверхности и объем куба
8	Найти площадь поверхности шестигранной правильной призмы
9	Найти площадь поверхности и объем прямоугольного параллелепипеда
10	Найти площадь поверхности правильной четырехугольной пирамиды

**Форма отчета:** документ, защита работы.

### Практическое занятие № 2

**Тема:** Составление программ разветвляющейся структуры.

**Цель:** Изучение способов задания алгоритмов, приобретение практических навыков составления блок - схем для программ с разветвляющейся структуры.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

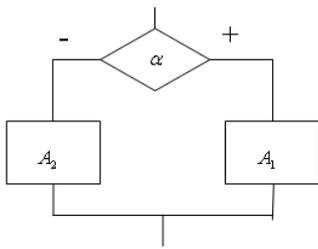
**Ход выполнения:**

**Задание 1.** Описать исходные, выходные и промежуточные данные следующих задач:

1. Даны два неотрицательных числа  $a$  и  $b$ . Найти их среднее арифметическое и среднее геометрическое.
2. Создать калькулятор для сложения и умножения обыкновенных дробей.
3. Даны два неотрицательных числа. Найти сумму, разность, произведение и частное их квадратов.
4. Найти расстояние между двумя точками с заданными неотрицательными координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  на плоскости.
5. Даны неотрицательные переменные  $A, B, C$ . Изменить их значения, переместив содержимое  $A$  в  $B$ ,  $B$  — в  $C$ ,  $C$  — в  $A$ , и вывести новые значения переменных  $A, B, C$ .

**Задание 2.** Составить блок-схему решения для задач 1-5, используя основные алгоритмические конструкции: следование, развилка (в тетради и программе MS Visio).

Структура «развилка» - применяется в тех случаях, когда в зависимости от нескольких условий нужно выполнить одно из двух действий.

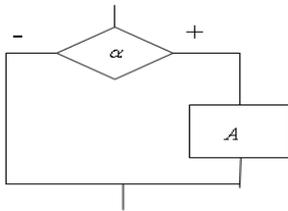


$A_i$  - функциональный блок

$i = \overline{1,2}$

$$\alpha = \begin{cases} 1(\text{true}, +) \\ 0(\text{false}, -) \end{cases}$$

Полная разветвляющаяся (логическая) структура

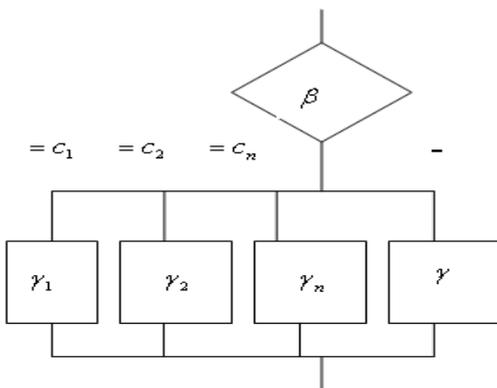


$A$  - функциональный блок

$$\alpha = \begin{cases} 1(\text{true}, +) \\ 0(\text{false}, -) \end{cases}$$

Неполная разветвляющаяся (логическая) структура (другое название неполной логической структуры «обход»)

Замечание: кроме структуры «развилка» так же используется структура выбор, её применяют в тех случаях, когда альтернатива состоит из двух и более вариантов.



$\beta$  - выражение значение, которое анализируется.

$c_i, i = \overline{1, n}$  - возможные значения выражения  $\beta$ .

$\gamma_i, i = \overline{1, n}$  - функциональные блоки, которые выполняются в зависимости от значения  $\beta$ .

**Задание 3.** Описать исходные, выходные и промежуточные данные и составить блок-схему решения для задачи по варианту в программе MS Visio.

Вариант	Задача
1	Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.
2	Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.
3	Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду сотен в записи этого числа..
4	Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду тысяч в записи этого числа.
5	Дано трехзначное число. Вывести вначале его последнюю цифру (единицы), а затем — его среднюю цифру (десятки).

6	Дано трехзначное неотрицательное число. Используя одну операцию деления нацело, вывести первую цифру данного числа (сотни).
7	Дано трехзначное число. Найти сумму и произведение его цифр.
8	Дано трехзначное число. Вывести число, полученное при перестановке цифр десятков и единиц исходного числа.
9	Дано трехзначное число. Вывести число, полученное при перестановке цифр сотен и десятков исходного числа.
10	Дано трехзначное число. Вывести число, полученное при прочтении исходного числа справа налево.

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 3

**Тема:** Составление программ циклической структуры

**Цель:** Изучение способов задания алгоритмов, приобретение практических навыков составления блок - схем для программ с циклической структурой.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Описать исходные, выходные и промежуточные данные следующих задач:

1. Дана последовательность действительных чисел. Найти максимальный элемент в последовательности.

2. Даны натуральные числа  $n$  и  $k$ . Найти сумму  $1k + 2k + 3k + \dots + nk$ . Заблокировать ввод ненатуральных чисел.

3. Составить программу вычисления значения функции

$$y = \frac{(x-2)(x-4)(x-6)\dots(x-64)}{(x-1)(x-3)(x-5)\dots(x-63)}$$

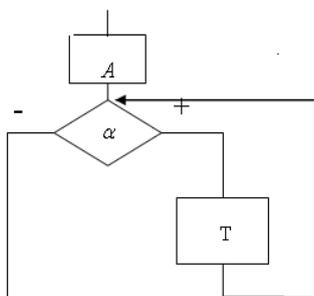
4. Составить программу, которая проверяет, является ли заданное число совершенным. Совершенным называется натуральное число, равное сумме всех своих делителей (исключая само число). Например,  $28 = 1 + 2 + 4 + 7 + 14$ .

5. Спортсмен-лыжник начал тренировки, пробежав в первый день 10 км. Каждый следующий день он увеличивал длину пробега на  $P$  процентов от пробега предыдущего дня ( $P$  — вещественное,  $0 < P < 50$ ). По данному  $P$  определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней  $K$  (целое) и суммарный пробег  $S$  (вещественное число). (WHILE)

**Задание 2.** Составить блок-схему решения для задач 1-5, используя основные алгоритмические конструкции: следование, развилка, цикл (в тетради и программе MS Visio).

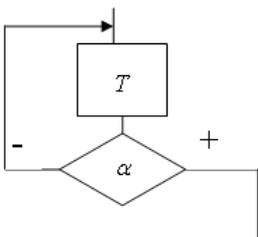
Структура «цикл» - применяется в тех случаях, когда возникает необходимость выполнения некоторой последовательности действий повторно.

а) цикл с предусловием («пока»)



*A* - функциональный блок, начальных присвоений.  
*T* - функциональный блок, тело цикла.  
 $\alpha$  - логическое условие

б) цикл с постусловием («до»)

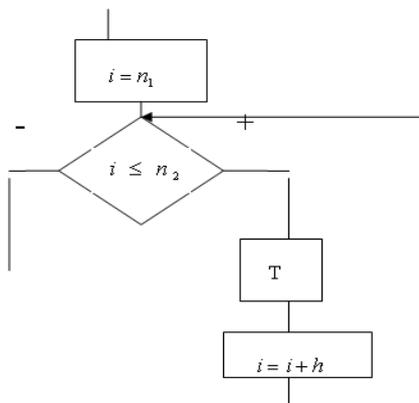


*A* - функциональный блок, начальных присвоений.  
*T* - функциональный блок, тело цикла.  
 $\alpha$  - логическое условие

в) цикл с параметром, счетчик («для»)

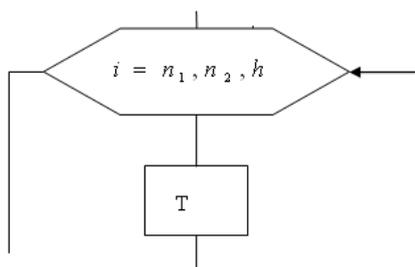
Данный цикл является частным случаем цикла «пока» и применяется в тех случаях, когда известно количество повторений в цикле.

1 форма



*i* - параметр цикла.  
 $n_1, n_2$  - начальное и конечное значение параметра (границы параметра).  
*h* - шаг измерения параметра (если шаг равен 1, то его опускают).  
*T* - функциональный блок.

2 форма



*i* - параметр цикла.  
 $n_1, n_2$  - начальное и конечное значение параметра (границы параметра).  
*h* - шаг измерения параметра (если шаг равен 1, то его опускают).  
*T* - функциональный блок.

**Задание 3.** Описать исходные, выходные и промежуточные данные и составить блок-схему решения для задачи по варианту в программе MS Visio

Вариант	Задача
1	Даны два целых числа <i>A</i> и <i>B</i> ( $A < B$ ).

	Вывести в порядке убывания все целые числа, расположенные между $A$ и $B$ (не включая числа $A$ и $B$ ), а также количество $N$ этих чисел.
2	Дано целое число $N (> 0)$ . Найти квадрат данного числа, используя для его вычисления следующую формулу: $N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1)$ .
3	Дано вещественное число $A$ и целое число $N (> 0)$ . Найти $A$ в степени $N$ : $A^N = A \cdot A \cdot \dots \cdot A$
4	Дано вещественное число $A$ и целое число $N (> 0)$ . Используя один цикл, вывести все целые степени числа $A$ от 1 до $N$ .
5	Дано целое число $N (> 0)$ . Найти произведение $N! = 1 \cdot 2 \cdot \dots \cdot N$
6	Дано целое число $N (> 0)$ . Найти произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ ( $N$ сомножителей).
7	Дано целое число $N (> 0)$ . Найти значение выражения $1.1 - 1.2 + 1.3 - \dots$ ( $N$ слагаемых, знаки чередуются). Условный оператор не использовать.
8	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти сумму всех целых чисел от $A$ до $B$ включительно.
9	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти произведение всех целых чисел от $A$ до $B$ включительно
10	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти сумму квадратов всех целых чисел от $A$ до $B$ включительно.

**Форма отчета:** отчет, защита работы.

#### Практическое занятие № 4

**Тема:** Знакомство со средой программирования.

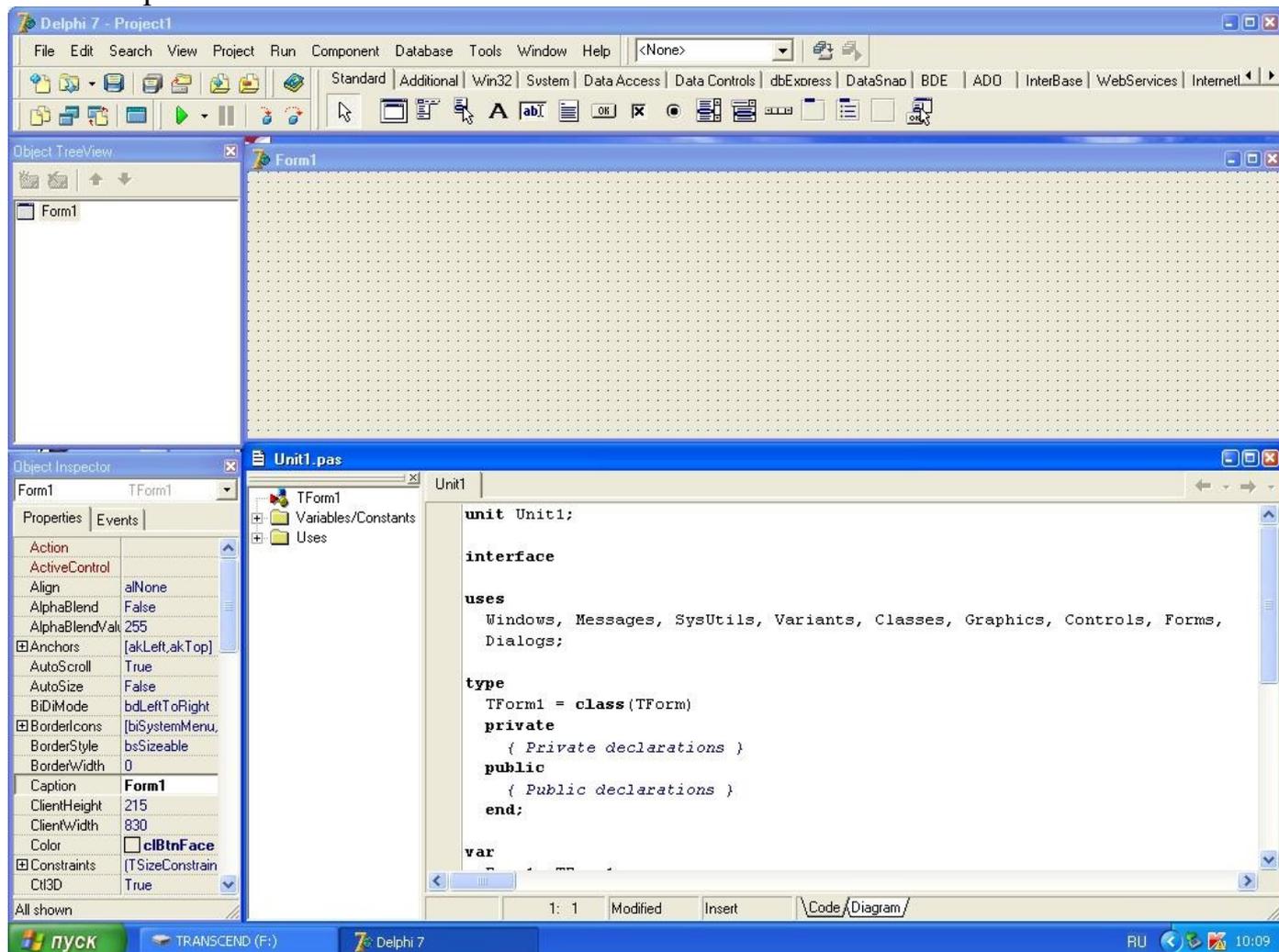
**Цель:** изучить интегрированную среду разработки Delphi 7 для работы с формой и элементами управления.

**Оборудование:** тетрадь, ручка, ПК

## Методическое указания: выполнение заданий

### Ход выполнения:

**Задание 1.** На рисунке нужно отметить следующие объекты: главное окно, окно формы, окно кода программы, окно браузера кода, окно дерева объектов, окно инспектора объектов.



**Задание 2.** Обозначить инструментальные кнопки которые открывают быстрый доступ к наиболее важным командам главного меню.

кнопка	описание	Эквивалентно команде

**Задание 3.** Описать элементы управления в вкладке «Standart»(название, применение и основные свойства).



**Форма отчета:** отчет, защита работы.

### Практическое занятие № 5

**Тема:** Создание проектов с использованием условного оператора if/else.

**Цель:** изучить возможности ИСР Borland Delphi для программирования разветвляющихся алгоритмов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Решить задачи и оформить согласно этапам решения задач на ЭВМ.

1. Даны два неотрицательных числа  $a$  и  $b$ . Найти их среднее арифметическое и среднее геометрическое.

2. Создать калькулятор для сложения и умножения обыкновенных дробей.

3. Даны два неотрицательных числа. Найти сумму, разность, произведение и частное их квадратов.

4. Найти расстояние между двумя точками с заданными неотрицательными координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  на плоскости.

5. Даны неотрицательные переменные  $A, B, C$ . Изменить их значения, переместив содержимое  $A$  в  $B$ ,  $B$  — в  $C$ ,  $C$  — в  $A$ , и вывести новые значения переменных  $A, B, C$ .

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.

2	Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.
3	Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду сотен в записи этого числа..
4	Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду тысяч в записи этого числа.
5	Дано трехзначное число. Вывести вначале его последнюю цифру (единицы), а затем — его среднюю цифру (десятки).
6	Дано трехзначное неотрицательное число. Используя одну операцию деления нацело, вывести первую цифру данного числа (сотни).
7	Дано трехзначное число. Найти сумму и произведение его цифр.
8	Дано трехзначное число. Вывести число, полученное при перестановке цифр десятков и единиц исходного числа.
9	Дано трехзначное число. Вывести число, полученное при перестановке цифр сотен и десятков исходного числа.
10	Дано трехзначное число. Вывести число, полученное при прочтении исходного числа справа налево.

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 6**

**Тема:** Создание проектов с использованием оператора выбора switch и перечислений.

**Цель:** изучить возможности ИСР Borland Delphi для программирования разветвляющихся алгоритмов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Решить задачи и оформить с решить задачи и оформить согласно этапам решения задач на ЭВМ.

1. Дано целое число в диапазоне 1–7. Вывести строку — название дня недели, соответствующее данному числу (1 — «понедельник», 2 — «вторник» и т. д.).
2. Дано целое число К. Вывести строку-описание оценки, соответствующей числу К (1 — «плохо», 2 — «неудовлетворительно», 3 — «удовлетворительно», 4 — «хорошо», 5 — «отлично»). Если К не лежит в диапазоне 1–5, то вывести строку «ошибка».
3. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Вывести название соответствующего времени года («зима», «весна», «лето», «осень»).
4. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.

5. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия  $N$  (целое число в диапазоне 1–4) и вещественные числа  $A$  и  $B$  ( $B$  не равно 0). Выполнить над числами указанное действие и вывести результат.

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины (целое число в диапазоне 1–5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.
2	Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы (целое число в диапазоне 1–5) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах
3	Даны два целых числа: $D$ (день) и $M$ (месяц), определяющие правильную дату невисокосного года. Вывести значения $D$ и $M$ для даты, предшествующей указанной.
4	Даны два целых числа: $D$ (день) и $M$ (месяц), определяющие правильную дату невисокосного года. Вывести значения $D$ и $M$ для даты, следующей за указанной.
5	Робот может перемещаться в четырех направлениях («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, -1 — поворот направо. Дан символ $S$ — исходное направление робота и целое число $N$ — посланная ему команда. Вывести направление робота после выполнения полученной команды.
6	Элементы окружности пронумерованы следующим образом: 1 — радиус $R$ , 2 — диаметр $D = 2 \cdot R$ , 3 — длина $L = 2 \cdot \pi \cdot R$ , 4 — площадь круга $S = \pi \cdot R^2$ . Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке). В

	качестве значения $\pi$ использовать 3.14
7	Элементы равнобедренного прямоугольного треугольника пронумерованы следующим образом: 1 – катет $a$ , 2 – гипотенуза $c = a\sqrt{2}$ , 3 – высота $h$ , опущенная на гипотенузу ( $h = c/2$ ), 4 – площадь $S = c \cdot h/2$ . Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке)..
8	Элементы равностороннего треугольника пронумерованы следующим образом: 1 – сторона $a$ , 2 – радиус $R_1$ вписанной окружности ( $R_1 = a\sqrt{3}/6$ ), 3 – радиус $R_2$ описанной окружности ( $R_2 = 2 \cdot R_1$ ), 4 – площадь $S = a^2\sqrt{3}/4$ . Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).
9	Мастям игральных карт присвоены порядковые номера: 1 – пики, 2 – трефы, 3 – бубны, 4 – червы. Достоинству карт, старших десятки, присвоены номера: 11 – валет, 12 – дама, 13 – король, 14 – туз. Даны два целых числа: $N$ – достоинство ( $6 \leq N \leq 14$ ) и $M$ – масть карты ( $1 \leq M \leq 4$ ). Вывести название соответствующей карты вида «шестерка бубен», «дама червей», «туз треф» и т. п..
10	Дано целое число в диапазоне 20-69, определяющее возраст (в годах). Вывести строку-описание указанного возраста, обеспечив

правильное согласование числа со словом «год», например: 20 – «двадцать лет», 32 – «тридцать два года», 41 – «сорок один год».

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 7**

**Тема:** Создание проектов создание циклической структуры (for).

**Цель:** изучить возможности ИСР Borland Delphi для программирования циклических алгоритмов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Решить задачи и оформить согласно этапам решения задач на ЭВМ:

1. Даны целые числа  $K$  и  $N$  ( $N > 0$ ). Вывести  $N$  раз число  $K$ .
2. Даны два целых числа  $A$  и  $B$  ( $A < B$ ). Вывести в порядке возрастания все целые числа, расположенные между  $A$  и  $B$  (включая сами числа  $A$  и  $B$ ), а также количество  $N$  этих чисел.
3. Даны два целых числа  $A$  и  $B$  ( $A < B$ ). Вывести в порядке убывания все целые числа, расположенные между  $A$  и  $B$  (не включая числа  $A$  и  $B$ ), а также количество  $N$  этих чисел.
4. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1, 2, ..., 10 кг конфет.
5. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 0.1, 0.2, ..., 1 кг конфет.

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти сумму всех целых чисел от $A$ до $B$ включительно.
2	Дано целое число $N$ ( $> 0$ ). Найти сумму $1 + 1/2 + 1/3 + \dots + 1/N$ (вещественное число).
3	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти произведение всех целых чисел от $A$ до $B$ включительно.
4	Дано целое число $N$ ( $> 0$ ). Найти сумму $N^2 + (N+1)^2 + (N+2)^2 + \dots + (2 \cdot N)^2$ (целое число).

5	Даны два целых числа $A$ и $B$ ( $A < B$ ). Найти сумму квадратов всех целых чисел от $A$ до $B$ включительно.
6	Дано целое число $N$ ( $> 0$ ). Найти произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ ( $N$ сомножителей).
7	Дано вещественное число $A$ и целое число $N$ ( $> 0$ ). Найти $A$ в степени $N$ : $A_N = A \cdot A \cdot \dots \cdot A$ (числа $A$ перемножаются $N$ раз).
8	Дано вещественное число – цена 1 кг конфет. Вывести стоимость 1.2, 1.4, ..., 2 кг конфет
9	Дано целое число $N$ ( $> 0$ ). Найти значение выражения $1.1 - 1.2 + 1.3 - \dots$ ( $N$ слагаемых, знаки чередуются). Условный оператор не использовать.
10	Дано целое число $N$ ( $> 0$ ). Найти квадрат данного числа, используя для его вычисления следующую формулу: $N_2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1)$ . После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до $N$ ).

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 8**

**Тема:** Создание проектов создание циклической структуры (while).

**Цель:** изучить возможности ИСР Borland Delphi для программирования циклических алгоритмов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1:** решить задачи и оформить согласно этапам решения задач на ЭВМ:

1. Даны положительные числа  $A$  и  $B$  ( $A > B$ ). На отрезке длины  $A$  размещено максимально возможное количество отрезков длины  $B$  (без наложений). Не используя операции умножения и деления, найти длину незанятой части отрезка  $A$ .
2. Даны положительные числа  $A$  и  $B$  ( $A > B$ ). На отрезке длины  $A$  размещено максимально возможное количество отрезков длины  $B$  (без наложений). Не используя операции умножения и деления, найти количество отрезков  $B$ , размещенных на отрезке  $A$ .
3. Даны целые положительные числа  $N$  и  $K$ . Используя только операции сложения и вычитания, найти частное от деления нацело  $N$  на  $K$ , а также остаток от этого деления.
4. Дано целое число  $N$  ( $> 0$ ). Если оно является степенью числа 3, то вывести True, если не является — вывести False.

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Дано целое число $N$ ( $> 0$ ), являющееся некоторой степенью числа 2: $N = 2^k$ . Найти целое число $K$ — показатель этой степени.
2	Дано целое число $N$ ( $> 0$ ). Найти <i>двойной факториал</i> $N$ : $N!! = N \cdot (N-2) \cdot (N-4) \cdot \dots$ (последний сомножитель равен 2, если $N$ — четное, и 1, если $N$ — нечетное). Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и вывести его как вещественное число.
3	Дано целое число $N$ ( $> 0$ ). Найти наименьшее целое положительное число $K$ , квадрат которого превосходит $N$ : $K^2 > N$ . Функцию извлечения квадратного корня не использовать.
4	Дано целое число $N$ ( $> 0$ ). Найти наибольшее целое число $K$ , квадрат которого не превосходит $N$ : $K^2 \leq N$ . Функцию извлечения квадратного корня не использовать.
5	Дано целое число $N$ ( $> 1$ ). Найти

	наименьшее целое число $K$ , при котором выполняется неравенство $3K > N$ .
6	Дано целое число $N (> 1)$ . Найти наибольшее целое число $K$ , при котором выполняется неравенство $3K < N$ .
7	Дано целое число $N (> 1)$ . Вывести наименьшее из целых чисел $K$ , для которых сумма $1 + 2 + \dots + K$ будет больше или равна $N$ , и саму эту сумму.
8	Дано целое число $N (> 1)$ . Вывести наибольшее из целых чисел $K$ , для которых сумма $1 + 2 + \dots + K$ будет меньше или равна $N$ , и саму эту сумму.
9	Дано число $A (> 1)$ . Вывести наименьшее из целых чисел $K$ , для которых сумма $1 + 1/2 + \dots + 1/K$ будет больше $A$ , и саму эту сумму.
10	Дано число $A (> 1)$ . Вывести наибольшее из целых чисел $K$ , для которых сумма $1 + 1/2 + \dots + 1/K$ будет меньше $A$ , и саму эту сумму.

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 9**

**Тема:** Создание проектов создание циклической структуры (do).

**Цель:** изучить возможности ИСР Borland Delphi для программирования циклических алгоритмов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1:** решить задачи и оформить согласно этапам решения задач на ЭВМ:

1. Даны десять вещественных чисел. Найти их сумму.
2. Даны десять вещественных чисел. Найти их произведение.
3. Даны десять вещественных чисел. Найти их среднее арифметическое.
4. Дано целое число  $N$  и набор из  $N$  вещественных чисел. Вывести сумму и произведение чисел из данного набора.

5. Дано целое число  $N$  и набор из  $N$  положительных вещественных чисел. Вывести в том же порядке целые части всех чисел из данного набора (как вещественные числа с нулевой дробной частью), а также сумму всех целых частей.

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Дано целое число $N$ и набор из $N$ положительных вещественных чисел. Вывести в том же порядке дробные части всех чисел из данного набора (как вещественные числа с нулевой целой частью), а также произведение всех дробных частей.
2	Дано целое число $N$ и набор из $N$ вещественных чисел. Вывести в том же порядке округленные значения всех чисел из данного набора (как целые числа), а также сумму всех округленных значений.
3	Дано целое число $N$ и набор из $N$ целых чисел. Вывести в том же порядке все четные числа из данного набора и количество $K$ таких чисел.
4	Дано целое число $N$ и набор из $N$ целых чисел. Вывести в том же порядке номера всех нечетных чисел из данного набора и количество $K$ таких чисел.
5	Дано целое число $N$ и набор из $N$ целых чисел. Если в наборе имеются положительные числа, то вывести True; в противном случае вывести False.
6	Даны целые числа $K$ , $N$ и набор из $N$ целых чисел. Если в наборе имеются числа, меньшие $K$ , то вывести True; в противном случае вывести False.
7	Дан набор ненулевых целых чисел; признак его завершения – число 0.

	Вывести количество чисел в наборе
8	Дан набор ненулевых целых чисел; признак его завершения – число 0. Вывести сумму всех положительных четных чисел из данного набора. Если требуемые числа в наборе отсутствуют, то вывести 0..
9	Дано целое число $K$ и набор ненулевых целых чисел; признак его завершения – число 0. Вывести количество чисел в наборе, меньших $K$ .
10	Дано целое число $K$ и набор ненулевых целых чисел; признак его завершения – число 0. Вывести номер первого числа в наборе, большего $K$ . Если таких чисел нет, то вывести 0..

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 10**

**Тема:** Обработка одномерных массивов

**Цель:** изучить способы описания массивов, рассмотреть типовые задачи обработки массивов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1:** решить задачи и оформить согласно этапам решения задач на ЭВМ:

1. Дано целое число  $N (> 0)$ . Сформировать и вывести целочисленный массив размера  $N$ , содержащий  $N$  первых положительных нечетных чисел: 1, 3, 5, ... .
2. Дано целое число  $N (> 0)$ . Сформировать и вывести целочисленный массив размера  $N$ , содержащий степени двойки от первой до  $N$ -й: 2, 4, 8, 16, ... .
3. Дано целое число  $N (> 1)$ , а также первый член  $A$  и разность  $D$  арифметической прогрессии. Сформировать и вывести массив размера  $N$ , содержащий  $N$  первых членов данной прогрессии:  $A, A + D, A + 2 \cdot D, A + 3 \cdot D, \dots$  .
4. Дано целое число  $N (> 1)$ , а также первый член  $A$  и знаменатель  $Q$  геометрической прогрессии. Сформировать и вывести массив размера  $N$ , содержащий  $N$  первых членов данной прогрессии:  $A, A \cdot D, A \cdot D^2, A \cdot D^3, \dots$  .

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Дано целое число $N (> 2)$ . Сформировать и вывести целочисленный массив размера $N$ , содержащий $N$ первых элементов последовательности чисел Фибоначчи $F_K: F_1 = 1, F_2 = 1, F_K = F_{K-2} + F_{K-1}, K = 3, 4, \dots$ .
2	Дан массив размера $N$ . Вывести его элементы в обратном порядке.
3	Дан целочисленный массив размера $N$ . Вывести все содержащиеся в данном массиве нечетные числа в порядке возрастания их индексов, а также их количество $K$ .
4	Дан целочисленный массив размера $N$ . Вывести все содержащиеся в данном массиве четные числа в порядке убывания их индексов, а также их количество $K$ .
5	Дан целочисленный массив размера $N$ . Вывести вначале все содержащиеся в данном массиве четные числа в порядке возрастания их индексов, а затем — все нечетные числа в порядке убывания их индексов.
6	Дан массив $A$ размера $N$ и целое число $K (1 \leq K \leq N)$ . Вывести элементы массива с порядковыми номерами, кратными $K: A_K, A_{2 \cdot K}, A_{3 \cdot K}, \dots$ . Условный оператор не использовать.
7	Дан массив $A$ размера $N (N — \text{четное число})$ . Вывести его элементы с четными номерами в порядке возрастания номеров: $A_2, A_4, A_6, \dots, A_N$ . Условный оператор не использовать
8	Дан массив $A$ размера $N (N — \text{нечетное число})$ . Вывести его элементы с нечетными номерами в порядке убывания номеров: $A_N, A_{N-2}, A_{N-4}, \dots, A_1$ . Условный оператор не использовать
9	Дан массив $A$ размера $N$ . Вывести вначале его элементы с четными номерами (в порядке возрастания номеров), а затем — элементы с нечетными номерами (также в порядке возрастания номеров): $A_2, A_4, A_6, \dots, A_1, A_3, A_5, \dots$ . Условный оператор не использовать.
10	Дан массив $A$ размера $N$ . Вывести вначале его элементы с нечетными номерами в порядке возрастания номеров, а затем — элементы с четными номерами в порядке убывания номеров. $A_1, A_3, A_5, \dots, A_6, A_4, A_2$ . Условный оператор не использовать.

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 11**

**Тема:** Обработка двумерных массивов.

**Цель:** изучить способы описания двумерных массивов, рассмотреть типовые задачи обработки массивов.

**Оборудование:** тетрадь, ручка, ПК

**Методическое указания:** выполнение заданий

**Ход выполнения:**

**Задание 1:** решить задачи и оформить согласно этапам решения задач на ЭВМ:

1. Даны массивы  $A$  и  $B$  одинакового размера  $N$ . Поменять местами их содержимое и вывести вначале элементы преобразованного массива  $A$ , а затем — элементы преобразованного массива  $B$ .

2. Дан массив  $A$  размера  $N$ . Сформировать новый массив  $B$  того же размера, элементы которого определяются следующим

$$B_K = \begin{cases} 2 \cdot A_K, & \text{если } A_K < 5, \\ A_K/2 & \text{в противном случае.} \end{cases}$$

образом:

3. Даны два массива  $A$  и  $B$  одинакового размера  $N$ . Сформировать новый массив  $C$  того же размера, каждый элемент которого равен максимальному из элементов массивов  $A$  и  $B$  с тем же индексом.

4. Дан целочисленный массив  $A$  размера  $N$ . Переписать в новый целочисленный массив  $B$  все четные числа из исходного массива (в том же порядке) и вывести размер полученного массива  $B$  и его содержимое.

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

Вариант	Задача
1	Дан целочисленный массив $A$ размера $N$ ( $\leq 15$ ). Переписать в новый целочисленный массив $B$ все элементы с нечетными порядковыми номерами (1, 3, ...) и вывести размер полученного массива $B$ и его содержимое. Условный оператор не использовать
2	Дан целочисленный массив $A$ размера $N$ ( $\leq 15$ ). Переписать в новый целочисленный массив $B$ все элементы с порядковыми номерами, кратными трем (3, 6, ...), и вывести размер полученного массива $B$ и его содержимое. Условный оператор не использовать.
3	Дан целочисленный массив $A$ размера $N$ . Переписать в новый целочисленный массив $B$ того же размера вначале все элементы исходного массива с четными номерами, а затем — с нечетными: $A_2, A_4, A_6, \dots, A_1, A_3, A_5, \dots$ . Условный оператор не использовать.
4	Дан массив $A$ размера $N$ . Сформировать новый массив $B$ того же размера по следующему правилу: элемент $B_K$ равен сумме элементов массива $A$ с номерами от 1 до $K$ .
5	Дан массив $A$ размера $N$ . Сформировать новый массив $B$ того же размера по следующему правилу: элемент $B_K$ равен среднему арифметическому элементов массива $A$ с номерами от 1 до $K$ .
6	Дан массив $A$ размера $N$ . Сформировать новый массив $B$ того же размера по следующему правилу: элемент $B_K$ равен сумме элементов массива $A$ с номерами от $K$ до $N$ .
7	Дан массив $A$ размера $N$ . Сформировать новый массив $B$ того же размера

	по следующему правилу: элемент $B_K$ равен среднему арифметическому элементов массива $A$ с номерами от $K$ до $N$ .
8	Дан массив $A$ размера $N$ . Сформировать два новых массива $B$ и $C$ : в массив $B$ записать все положительные элементы массива $A$ , в массив $C$ — все отрицательные (сохраняя исходный порядок следования элементов). Вывести вначале размер и содержимое массива $B$ , а затем — размер и содержимое массива $C$ .
9	Даны два массива $A$ и $B$ размера 5, элементы которых упорядочены по возрастанию. Объединить эти массивы так, чтобы результирующий массив $C$ (размера 10) остался упорядоченным по возрастанию.
10	Даны три целочисленных массива $A$ , $B$ и $C$ размера $N_A$ , $N_B$ , $N_C$ соответственно, элементы которых упорядочены по убыванию. Объединить эти массивы так, чтобы результирующий целочисленный массив $D$ (размера $N_A + N_B + N_C$ ) остался упорядоченным по убыванию.

**Примечание:**

1. Решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
2. Отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 12**

**Тема:** Работа со строками.

**Цель:** изучить операции над строками, функции и процедуры обработки строк средствами языка программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание.** Решить задачи и оформить согласно этапам решения задач на ЭВМ.

1. Определить, является ли вводимая последовательность символов идентификатором
2. Дана символьная строка. Посчитать, сколько раз в ней встречается слово АВВА.
3. В заданном тексте перевернуть каждое слово.
4. Дана строка символов. Определить, сколько в ней слов, начинающихся и кончающихся на одну и ту же букву.
5. Заданы два текста. Вывести слова первого текста, которые не встречаются во втором.
6. Создать бегущую строку. Остановить бег строки при нажатии клавиши <Esc>.

**Форма отчета:** отчет, защита работы

**Практическое занятие № 13**

**Тема:** Работа с данными типа множество

**Цель:** изучить операции с данными типа множества, функции и процедуры обработки строк средствами языка программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание.** Решить задачи и оформить согласно этапам решения задач на ЭВМ.

1. Вычислить сумму тех элементов матрицы  $A$ , номера строк и столбцов которых принадлежат к заданным множествам целых чисел  $S1$  и  $S2$ .
2. Дан текст из цифр и латинских букв. Определить, каких букв больше: гласных или согласных.
3. Написать программу, формирующую случайным образом множество целых чисел и определяющую мощность этого множества.
4. Написать программу, формирующую случайным образом два числовых множества и определяющую, в каком отношении находятся эти множества.
5. В городе  $N$  имеется 100 кондитерских магазинов. Известно, что в каждом из этих магазинов не более 20 видов сладостей в ассортименте. Какие виды сладостей есть во всех имеющихся магазинах? Существует ли магазин, торгующий уникальной продукцией? Перечислите пять видов сладостей, которые есть в большинстве магазинов города  $N$  (ассортимент кондитерских магазинов рассматривать как данные перечисляемого типа).

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 14

**Тема:** Работа с файлами.

**Цель:** изучить возможности языка программирования для работы с с файлом последовательного доступа.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание:**

- 1) создайте папку для проекта «*Практическое занятие № 14*»;
- 2) в папке «*Практическое занятие № 14*» создайте текстовый файл *quest.txt* с вопросами;
- 3) в папке «*Практическое занятие № 14*» создайте текстовый файл *interp.txt* с интерпретацией результатов теста (текст каждой интерпретации записывается в одну строку и разделяется в файле клавишей <ENTER>)
- 4) в папке *task3* создайте пустой *html*-файл *result.htm*;
- 5) спроектируйте интерфейс программы (интерпретация результатов теста будет выводиться в *html*-файл *result.htm*, для их отображения на экране потребуется компонент *WebBrowser*);  
Объект *WebBrowser*: VCL  вкладка Internet  *WebBrowser*.  
Метод *Navigate(URL)* браузера позволяет загрузить страницу по ее URL (либо по пути к файлу)

**Объект Свойство Значение Событие Программный код**

Form Name frmtest

Caption «Тест»

Button Name btntest OnClick (1)

Caption «Начать тест»

WebBrowser Name wbtest

wbtest

frmtest

btntest

б) Программный код (1)

**procedure Tfrmtest.btntestClick(Sender: TObject);**

**var** fq, fi, fr: textfile; // *fq* - файл с вопросами, *fi* - файл с интерпретацией, *fr* - итоговый

файл *html*

*s*, *rez*: string; // *s* - переменная для считывания из файла, *rez* - переменная, содержащая *html*-код

*k*: 0..10; // номер вопроса

*b*: 0..1; // баллы за ответ пользователя

*but*, *sum*: byte; // *but* - нажатая кнопка, *sum* - сумма набранных баллов

**begin**

*k*:=0; // обнуляем переменную, отвечающую за номер вопроса

*sum*:=0; // обнуляем сумму баллов

*rez*:= '<html><head></head><body><H2>Результаты теста</H2><hr><table border=1><tr><th>№<th>Вопрос<th>Ответ'; // начало *web*-страницы

*assignfile*(fq, 'quest.txt'); // ассоциируем файловую переменную с физическим файлом

*assignfile*(fi, 'interp.txt');

*assignfile*(fr, 'result.htm');

*reset*(fq); // открываем файл с вопросами для чтения

*reset*(fi); // открываем файл с интерпретацией результатов

**while** not EOF(fq) **do** // пока не достигли конца файла с вопросами

**begin**

*readln*(fq, *s*); // считываем вопрос в переменную *s*

*inc*(*k*); // увеличиваем номер вопроса на 1

*but*:=Application.MessageBox(PChar(*s*), PChar('Ответ на вопрос№ '+IntToStr(*k*)), MB\_YESNO+MB\_ICONQUESTION); // окно

*if but*=6 *then* // анализируем нажатую кнопку

**begin**

*inc*(*sum*);

*b*:=1;

**end**

*else b*:=0;

*rez*:=*rez*+'<tr><td>'+inttostr(*k*)+'<td>'+*s*+'<td>'+inttostr(*b*); // прописываем строку к *HTML*-

коду

**end;**

**case sum of** // анализируем сумму баллов

1..3: *readln*(fi, *s*); // считываем соответствующую интерпретацию из файла

4..7: **begin** *readln*(fi); *readln*(fi, *s*); **end;**

8..10: **begin** *readln*(fi); *readln*(fi, *s*); **end;**

```

end;
rez:=rez+'</table>'+<br><b>Итого:</b> '+inttostr(sum) + ' баллов.<p><b>Результат:
</b>' + s + '!' +
'</p></body></html>'; // конец html-кода
closefile(fq); // закрываем файлы с вопросами и интерпретацией
closefile(fi);
rewrite(fr); // открываем html-файл
writeln(fr,rez); // записываем в него html-код
closefile(fr); // закрываем файл
wbtest.Navigate(GetCurrentDir + 'result.htm'); //отображаем web-страницу в браузере
end;

```

### Приложение к задаче 3

#### Вопросы теста

1. Вы нередко действуете необдуманно?
2. Вам быстро все надоедает?
3. Легко расстаетесь со своими намерениями?
4. Вам кажется, что на спор можете сделать многое?
5. Часто опаздываете только из-за своей собственной несобранности?
6. Вы беззаботны и предпочитаете, чтобы все решалось само собой?
7. Чувствуете, что сдержать данные Вами обещания. Вам трудно?
8. Не можете утверждать, что постоянно уверены в себе?
9. Часто не помните, куда что положили?
10. Утром встаете с трудом, долго не можете проснуться?

#### Интерпретация результатов:

1. Вы способны поступать так, как необходимо, стало быть, регулярность в занятиях физкультурой не составит для вас проблемы.
2. У вас есть изъяны в характере, которые мешают вам быть до конца человеком твердым, собранным, т.е., чтобы решить ту или иную задачу, вы должны на ней сосредоточиться, заострить внимание, прилагать волевые усилия. Вам еще с вечера надо подумать о необходимости вставать рано утром, сделать гимнастику и пробежать именно 5 км, хотя, возможно, и не хочется.
3. Вы человек отнюдь не волевой и вам нужно серьезно поработать над своим характером. Начав с простеньких упражнений, с "безболезненного преодоления себя", постепенно приготовьтесь и к решению более серьезных задач. И не только в области физкультуры.

**Форма отчета:** отчет, защита работы.

### Практические занятия № 15

**Тема:** Типизированные и не типизированные файлы

**Цель:** изучить возможности языка программирования для работы с с файлом произвольного доступа.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Изучить теоретический материал

### *Типизированные файлы.*

К типизированным файлам относятся файлы строго определенного типа. Чаще всего это файлы, состоящие из записей. Они применяются для создания различных баз данных.

#### **Описание:**

```
type
fil = record
...      { поля записи }
end;
var
f: file of fil;
```

В типизированных файлах содержимое рассматривается как последовательность записей определенной структуры. Единицей измерения такого набора данных является сама запись. Длина записи определяется как **sizeof(fil)**;

**Seek(f, num)** процедура устанавливает текущую файловую переменную f на запись с номером num.

Типизированные файлы позволяют организовать работу в режиме чтения записи. Информация в типизированных наборах данных представлена в том же виде, как в памяти машины во время выполнения программы, поэтому не надо отслеживать управляющие последовательности типа конец строки или возврат каретки.

#### **Средства для работы с типизированными файлами.**

1. **truncate(f)**; - уничтожает все компоненты файла f, начиная с места текущего положения файлового указателя;

2. **filepos(f): longint**; - функция возвращает для файла f текущую файловую позицию (номер записи, на которую она установлена) в виде значения типа longint;

3. **filesize(f): longint**; - функция нахождения размера файла (количество записей) в виде значения типа longint.

Для того, чтобы очередная запись могла быть записана в конец типизированного файла, необходимо перевести текущую файловую позицию в конец файла. При создании файла это происходит автоматически после формирования каждой новой записи. Если файл уже создан и файловая позиция, установленная по **Seek**, находится в начале файла (**filepos(f) < filesize(f)**), то в конец файла ее позволяет перевести вызов **seek(f, filesize(f))**.

### *Нетипизированные файлы.*

Нетипизированные файлы объявляются как файловые переменные типа file и отличаются тем, что для них не указан тип компонентов. Отсутствие типа делает эти файлы совместимыми с любыми другими файлами и позволяют организовать высокоскоростной обмен данными между диском и памятью.

```
var
f: file;
```

С помощью процедур **reset**, **rewrite** можно задать длину нетипизированного файла: **reset(f, 512)**; **rewrite(f, 65535)**; по умолчанию длина равна 128 байтам.

#### **Средства для работы с нетипизированными файлами.**

1. Используют все процедуры и функции, рассмотренные ранее, кроме **read**, **write**, которые заменяются высокоскоростными.

2. **Blockread(f,buf,c);** - где **buf** – имя переменной, которая участвует в обмене данными с дисками, задает число, считываемых блоков;

3. **Blockwrite(f,buf,c);** - процедура предназначена для быстрой передачи в файл определенного числа записей из переменной **buf**.

Эти процедуры выполняют операции ввода – вывода блоками. Объем блока в байтах определяется по формуле: **объем=c\*reclsize(f);**, где **reclsize** – размер записи файла, заданный при его открытии.

**Задание 2.** Решить задачи и оформить согласно этапам решения задач на ЭВМ.

1. Составить программу, которая создает файл таблицы значений  $\sin(x)$  и  $\operatorname{tg}(x)$  на отрезке  $[0,3]$  с шагом 0.01. Значения  $x$  записывать с одной цифрой в дробной части, значения функций  $\sin(x)$  и  $\operatorname{tg}(x)$  - с пятью.

2. Составьте программу, которая создает файл, состоящий из 100 случайных чисел целого типа в диапазоне от 0 до 300. Исследуйте получившийся файл с целью обнаружения в нем простых чисел. Простые числа вывести на экран.

3. Составьте программу, которая создает файл, состоящий из 30 случайных чисел целого типа в диапазоне от -50 до 100. Исследуйте получившийся файл с целью обнаружения в нем чисел, кратных 5. Если такие числа есть, то вывести их на экран.

4. Имеется текстовый файл. Напечатать первые 10 символов первой строки.

5. Напишите программу, которая создает файл данных, хранящий записи о владельцах автотранспорта: марка автомобиля, номер регистрации в ГАИ, дата постановки на учет, ФИО владельца, домашний адрес, и обеспечивает обслуживание данного файла, запись, изменение данных, удаление, а также поиск данных по регистрационному номеру..

**Форма отчета:** отчет, защита работы.

## Практическое занятие № 16

**Тема:** Организация процедур.

**Цель:** изучить организации и использования процедур в языке программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Написать программу суммирования двух чисел, с использованием одной кнопки и двух текстовых окон.

Процедура Delphi просто выполняет требуемые операции, но никаких результатов своих действий не возвращает. Результат - в тех изменениях, которые произошли в программе в процессе выполнения этой процедуры. В частности, процедура может поменять значения переменных, записать новые значения в ячейки компонентов, сделать запись в файл и т.д.

Для функции необходимо в коде присвоить переменной с именем функции или специальной зарезервированной переменной **Result** (предпочтительно) возвращаемое функцией значение. Примеры:

```
procedure Имя_процедуры((*параметры*));
```

```

begin
  //Код процедуры;
end;
Программный код :
var
  Form1: TForm1;
  A, B, Summa: Integer;
  procedure Sum(A, B: Integer);
implementation
  {$R *.dfm}
  procedure TForm1.Button1Click(Sender: TObject);
begin
  A:=StrToInt(Edit1.Text);
  B:=StrToInt(Edit2.Text);
  Sum(A, B);
  Caption:=IntToStr(Summa);
end;
  procedure Sum(A, B: Integer);
begin
  Summa:=A+B;
end;

```

**Задание 2.** Решить задачу своего варианта и описать создание документации.

Вариант	Задача
1	Описать процедуру $PowerA3(A, B)$ , вычисляющую третью степень числа $A$ и возвращающую ее в переменной $B$ ( $A$ – входной, $B$ – выходной параметр; оба параметра являются вещественными). С помощью этой процедуры найти третьи степени пяти данных чисел.
2	Описать процедуру $PowerA234(A, B, C, D)$ , вычисляющую вторую, третью и четвертую степень числа $A$ и возвращающую эти степени соответственно в переменных $B, C$ и $D$ ( $A$ – входной, $B, C, D$ – выходные параметры; все параметры являются вещественными). С помощью этой процедуры

	найти вторую, третью и четвертую степень пяти данных чисел
3	Описать процедуру $\text{Mean}(X, Y, A\text{Mean}, G\text{Mean})$ , вычисляющую <i>среднее арифметическое</i> $A\text{Mean} = (X + Y) / 2$ и <i>среднее геометрическое</i> $G\text{Mean} = \sqrt{X \cdot Y}$ двух положительных чисел $X$ и $Y$ ( $X$ и $Y$ – входные, $A\text{Mean}$ и $G\text{Mean}$ – выходные параметры вещественного типа). С помощью этой процедуры найти среднее арифметическое и среднее геометрическое для пар $(A, B)$ , $(A, C)$ , $(A, D)$ , если даны $A, B, C, D$ .
4	Описать процедуру $\text{TrianglePS}(a, P, S)$ , вычисляющую по стороне $a$ равностороннего треугольника его периметр $P = 3 \cdot a$ и площадь $S = a^2 \sqrt{3} / 4$ ( $a$ – входной, $P$ и $S$ – выходные параметры; все параметры являются вещественными). С помощью этой процедуры найти периметры и площади трех равносторонних треугольников с данными сторонами
5	Описать процедуру $\text{RectPS}(x_1, y_1, x_2, y_2, P, S)$ , вычисляющую периметр $P$ и площадь $S$ прямоугольника со сторонами, параллельными осям координат, по координатам $(x_1, y_1)$ , $(x_2, y_2)$ его противоположных вершин ( $x_1, y_1, x_2, y_2$ – входные, $P$ и $S$ – выходные параметры вещественного типа). С помощью этой процедуры найти периметры и площади трех прямоугольников с данными противоположными вершинами.
6	Описать процедуру $\text{DigitCountSum}(K, C, S)$ , находящую количество $C$ цифр целого положительного числа $K$ , а также их сумму $S$ ( $K$ – входной,

	<p><math>C</math> и <math>S</math> – выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел</p>
7	<p>Описать процедуру <math>InvertDigits(K)</math>, меняющую порядок следования цифр целого положительного числа <math>K</math> на обратный (<math>K</math> – параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел</p>
8	<p>Описать процедуру <math>AddRightDigit(D, K)</math>, добавляющую к целому положительному числу <math>K</math> справа цифру <math>D</math> (<math>D</math> – входной параметр целого типа, лежащий в диапазоне 0-9, <math>K</math> – параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу <math>K</math> справа данные цифры <math>D_1</math> и <math>D_2</math>, выводя результат каждого добавления</p>
9	<p>Описать процедуру <math>AddLeftDigit(D, K)</math>, добавляющую к целому положительному числу <math>K</math> слева цифру <math>D</math> (<math>D</math> – входной параметр целого типа, лежащий в диапазоне 1-9, <math>K</math> – параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу <math>K</math> слева данные цифры <math>D_1</math> и <math>D_2</math>, выводя результат каждого добавления</p>
10	<p>Описать процедуру <math>Swap(X, Y)</math>, меняющую</p>

	<p>содержимое переменных <math>X</math> и <math>Y</math> (<math>X</math> и <math>Y</math> – вещественные параметры, являющиеся одновременно входными и выходными). С ее помощью для данных переменных <math>A, B, C, D</math> последовательно поменять содержимое следующих пар: <math>A</math> и <math>B, C</math> и <math>D, B</math> и <math>C</math> и вывести новые значения <math>A, B, C, D</math>.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 17

**Тема:** Организация функций.

**Цель:** изучить организации и использования функций в языке программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Написать программу нахождения факториала числа, с использованием компонентов Edit и UpDown.

Функция Delphi также позволяет выполнить всё перечисленное, но дополнительно возвращает результат в присвоенном ей самой значении. То есть вызов функции может присутствовать в выражении справа от оператора присваивания.

function Имя\_функции((\*параметры\*)): тип\_результата;

begin

  //Код функции;

  Result := результат;

end;

Программный код:

**implementation**

{SR \*.dfm}

**function** fak(N: Integer): Int64; //Функция, вычисляющая факториал, принимает число N как параметр

**begin**

**if**((N=0)**or**(N=1)) //Условие прекращения рекурсивных вызовов

**then** Result:=1 //Факториал чисел 0 и 1 равен 1

**else** Result:=N\*fak(N-1); //Если число больше 1, то осуществляется рекурсивный вызов функции самой себя с параметром N-1

**end;**

{используем процедуру **onMouseUp**, так как при использовании простого щелчка (**onClick**) будет вычислен факториал текущего числа, а нужное число появится с опозданием:}

**procedure** TForm1.UpDown1MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

**begin**

Label1.Caption:=IntToStr(fak(UpDown1.Position));

**end;**

**end.**

**Задание 2.** Решить задачу своего варианта и описать создание документации.

Вариант	Задача
1	Описать функцию $\text{Sign}(X)$ целого типа, возвращающую для вещественного числа $X$ следующие значения: $-1$ , если $X < 0$ ; $0$ , если $X = 0$ ; $1$ , если $X > 0$ . С помощью этой функции найти значение выражения $\text{Sign}(A) + \text{Sign}(B)$ для данных вещественных чисел $A$ и $B$ .
2	Описать функцию $\text{RootsCount}(A, B, C)$ целого типа, определяющую количество корней квадратного уравнения $A \cdot x^2 + B \cdot x + C = 0$ ( $A, B, C$ – вещественные параметры, $A \neq 0$ ). С ее помощью найти количество корней для каждого из трех квадратных уравнений с данными коэффициентами. Количество корней определять по значению <i>дискриминанта</i> : $D = B^2 - 4 \cdot A \cdot C$ .
3	Описать функцию $\text{CircleS}(R)$ вещественного типа, находящую площадь круга радиуса $R$ ( $R$ – вещественное). С помощью этой функции найти площади трех кругов с данными радиусами. Площадь круга радиуса $R$ вычисляется по формуле $S = \pi \cdot R^2$ . В качестве значения $\pi$ использовать $3.14$ .
4	Описать функцию $\text{RingS}(R_1, R_2)$ вещественного типа, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами $R_1$ и $R_2$ ( $R_1$ и $R_2$ –

	<p>вещественные, <math>R_1 &gt; R_2</math>). С ее помощью найти площади трех колец, для которых даны внешние и внутренние радиусы.</p> <p>Воспользоваться формулой площади круга радиуса <math>R</math>: <math>S = \pi \cdot R^2</math>. В качестве значения <math>\pi</math> использовать 3.14</p>
5	<p>Описать функцию <math>\text{TriangleP}(a, h)</math>, находящую периметр равнобедренного треугольника по его основанию <math>a</math> и высоте <math>h</math>, проведенной к основанию (<math>a</math> и <math>h</math> – вещественные). С помощью этой функции найти периметры трех треугольников, для которых даны основания и высоты. Для нахождения боковой стороны <math>b</math> треугольника использовать <i>теорему Пифагора</i>: и площади трех прямоугольников с данными противоположными вершинами. <math>b^2 = (a/2)^2 + h^2</math>.</p>
6	<p>Описать функцию <math>\text{Calc}(A, B, Op)</math> вещественного типа, выполняющую над ненулевыми вещественными числами <math>A</math> и <math>B</math> одну из арифметических операций и возвращающую ее результат. Вид операции определяется целым параметром <math>Op</math>: 1 – вычитание, 2 – умножение, 3 – деление, остальные значения – сложение. С помощью <math>\text{Calc}</math> выполнить для данных <math>A</math> и <math>B</math> операции, определяемые данными целыми <math>N_1, N_2, N_3</math>.</p>
7	<p>Описать функцию <math>\text{Quarter}(x, y)</math> целого типа, определяющую номер координатной четверти, в которой находится точка с ненулевыми вещественными координатами <math>(x, y)</math>. С помощью этой функции найти номера</p>

	координатных четвертей для трех точек с данными ненулевыми координатами.
8	Описать функцию $Even(K)$ логического типа, возвращающую True, если целый параметр $K$ является четным, и False в противном случае. С ее помощью найти количество четных чисел в наборе из 10 целых чисел
9	Описать функцию $IsSquare(K)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является квадратом некоторого целого числа, и False в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых положительных чисел
10	Описать функцию $IsPower5(K)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа 5, и False в противном случае. С ее помощью найти количество степеней числа 5 в наборе из 10 целых положительных чисел.

### Практическое занятие № 18

**Тема:** Применение рекурсивных функций.

**Цель:** изучить возможности языка программирования для создания наследованного класса и перегрузки методов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Написать рекурсивную функцию вычисления наибольшего общего делителя по алгоритму Эйлера: Если  $B$  делится на  $A$  нацело, то  $\text{НОД}(A, B) = A$ . В противном случае  $\text{НОД}(A, B) = \text{НОД}(B \bmod A, A)$ .

Установить с помощью Object Inspector следующие свойства формы:

Form1.Height = 347

Form1.Width = 359 Form1.BorderIcons biMaximize = false

Form1.BorderStyle = bsSingle Form1.Position = poScreenCenter

Form1.Caption = 'Нахождение наибольшего общего

Расположить на форме следующие компоненты: три компонента Edit, три компонента Label, один компонент Button и один компонент BitBtn. Установить для них следующие свойства:

Label1.Caption = 'a'

Label2.Caption = 'b'

Label3.Caption = 'НОД'

Edit1.Text = "

Edit2.Text = "

Edit3.Text = "

Button1.Caption = 'Вычислить НОД' BitBtn1.Kind = 'bkClose'

BitBtn1.Caption = '&Закреть'.

**Задание 2.** Написать программу для вычисления выражения  $z(x) = (\text{sign}(x) + \text{sign}(y)) \cdot \text{sign}(x + y)$ .

При решении задачи определите и используйте функцию  $\text{sign}$ (знак числа):  $\text{sign}(x) =$

**Задание 3.** Написать процедуру вычисления определенного интеграла методом прямоугольников: , где  $n$  – количество отрезков разбиения;  $y_0, y_1, \dots, y_n$  – значения функции на концах отрезков. Числа  $a$  и  $b$  – произвольные, вводятся с клавиатуры.

**Задание 4\*.** Напишите рекурсивную функцию вычисления  $i$ -го числа Фибоначчи. Вычислите  $f(k)$ ,  $k = 15, 20, 30, 40$ . Функция  $f(n)$  определена для целых чисел следующим образом:

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 19

**Тема:** Программирование модуля. Создание библиотеки подпрограмм.

**Цель:** изучить возможности Delphi для написания подпрограмм и создания модулей. Составить и отладить программу, использующую внешний модуль Unit с подпрограммой.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание.** Создать модуль.

1. Создание модуля

Создавая модуль, следует обратить внимание на то, что он не должен иметь своей формы. Система DELPHI при начальной загрузке автоматически создает шаблон программы, имеющий в своем составе форму, файл проекта и т. д. Т. к. модуль состоит только из одного файла, то необходимо перед его созданием уничтожить заготовку файла проекта и форму. Для этого в меню File выбрать Close All, файл проекта не сохранять.

Для создания модуля в меню File выбрать File New, и затем в репозитории — пиктограмму r1.tif. В результате будет создан файл с заголовком Unit Unit1. Имя модуля можно сменить на другое, отвечающее внутреннему содержанию модуля, например Unit Matfu;. Затем необходимо сохранить файл с именем, совпадающим с именем заголовка модуля: Matfu. pas. Следует обратить внимание на то, что имя

файла должно совпадать с именем модуля, иначе DELPHI не сможет подключить его к другой программе.

Подключение модуля

Для того чтобы подключить модуль к проекту, необходимо в меню Project выбрать опцию Add to Project... и выбрать файл, содержащий модуль. После этого в разделе Uses добавить имя подключаемого модуля – MatFu. Теперь в проекте можно использовать функции, содержащиеся в модуле.

Тексты модуля и вызывающей программы приведены ниже.

Текст модуля:

```
Unit Matfu;
```

```
Interface
```

```
Function Tg(x:extended) : extended; // Функция для вычисления тангенса
```

```
Function Ch(x:extended) : extended; // Функция для вычисления гиперболического синуса
```

```
Function Sin2(x:extended) : extended; // Функция для вычисления квадрата синуса
```

```
Implementation
```

```
Function Tg;
```

```
begin
```

```
Result:=Sin(x)/Cos(x);
```

```
end;
```

```
Function Ch;
```

```
begin
```

```
Result:=(exp(x)-exp(-x))/2;
```

```
end;
```

```
Function Sin2;
```

```
begin
```

```
Result:=sqr(sin(x));
```

```
end;
```

```
End.
```

Текст вызывающей программы:

```
Unit Unit1;
```

```
Interface
```

```
Uses
```

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

```
Type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
Edit3: TEdit;
```

```
Memo1: TMemo;
```

```

BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
RadioGroup1: TRadioGroup;
procedure FormCreate(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
Type
fun = function(x:extended):extended; // Объявление типа функция
Var
Form1: TForm1;
Implementation
{$R *.DFM}
Procedure TForm1.FormCreate(Sender: TObject);
Begin
Edit1.Text:='0';
Edit2.Text:='3';
Edit3.Text:='0,3';
Memo1.Clear;
RadioGroup1.ItemIndex:=0;
End;
Procedure Tabl(f:fun;xn, xk, h:extended); // Расчет таблицы
var x, y: extended;
begin
x:=xn;
repeat
y:=f(x);
Form1.Memo1.Lines. Add('x='+FloatToStrf(x, ffixed,8,3)+
'y='+FloatToStrf(y, ffixed,8,3));
x:=x+h;
until (x>xk);
End;
Procedure TForm1.BitBtn1Click(Sender: TObject);
Var xn, xk, h : extended;
Begin
xn:=StrToFloat(Edit1.Text); // Начальное значение интервала
xk:=StrToFloat(Edit2.Text); // Конечное значение интервала
h:=StrToFloat(Edit3.Text); // Шаг расчета
case RadioGroup1.ItemIndex of // Выбор функции
0 : Tabl(tg, xn, xk, h);
1 : Tabl(ch, xn, xk, h);

```

```
2 : Tabl(sin2,xn, xk, h);
end;
End;
End.
```

**Форма отчета:** отчет, защита работы

### Практическое занятие № 20

**Тема:** Использование указателей для организации связанных списков.

**Цель:** изучить возможности языка программирования для создания пользовательских подпрограмм с использованием указателей для организации связанных списков.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создать текстовый файл и заполнить его информацией, введённой в редакторе Мемо. Окно работающего приложения:

Создание текстового файла с использованием метода SaveToFile:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
memo1.Lines.SaveToFile('c:\Files\z1.txt');
end;
```

Создание текстового файла без использования метода SaveToFile:

```
procedure TForm1.Button1Click(Sender: TObject);
var f : textfile; n, i : integer;
begin
assignfile(f, 'c:\Files\z1.txt');
rewrite(f);
n:=memo1.Lines.Count;
for i:=1 to n do writeln(f, memo1.lines.strings[i-1]);
closefile(f);
end;
```

Приведённый здесь текст программы содержит имя создаваемого файла (c:\Files\z1.txt) . Поэтому, чтобы создать другой файл, нужно будет внести изменения в текст программы. Добавив на форму компонент SaveDialog, мы получим удобное средство для выбора имени создаваемого файла на этапе работы приложения. Окно приложения после добавления компонента SaveDialog:

Преобразованная процедура Button1Click:

```
procedure TForm1.Button1Click(Sender: TObject);
var f : textfile; n, i :integer;
begin
if not savedialog1.Execute then exit;
assignfile(f, savedialog1.filename);
rewrite(f);
n:=memo1.Lines.Count;
for i:=1 to n do
writeln(f, memo1.lines.strings[i-1]);
```

```
closefile(f);  
end;
```

**Задание 2.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

1. Составить подпрограмму вычисления площади выпуклого четырехугольника, заданного длинами своих сторон и одной из диагоналей. Решить задачу двумя способами: в первом случае составить подпрограмму-процедуру, во втором – подпрограмму-функцию.
2. Составить функцию логического типа, определяющую, является ли слово палиндромом.
3. Составить процедуру обнуления всех положительных элементов в одномерном массиве А. С помощью этой процедуры обнулить все положительные элементы в  $k$ -м столбце произвольной матрицы.
4. Вычислить сумму  $n$  членов последовательности:  $x_0=1, x_1=1, \dots, x_k=0,7x_{k-1}+1, 1x_{k-2}, k = 2, 3 \dots$  При решении задачи использовать рекурсию.
5. Отсортировать одномерный массив по возрастанию элементов, используя рекурсию.

**Форма отчета:** отчет, защита работы

### Практическое занятие № 21

**Тема:** Объявление и использование классов.

**Цель:** изучить возможности языка программирования для создания экземпляра и его объявления.

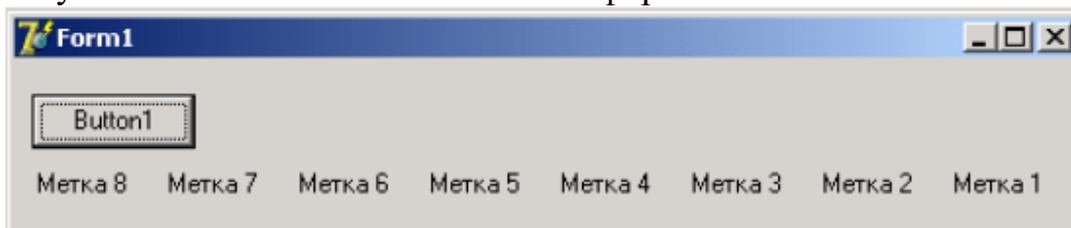
**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

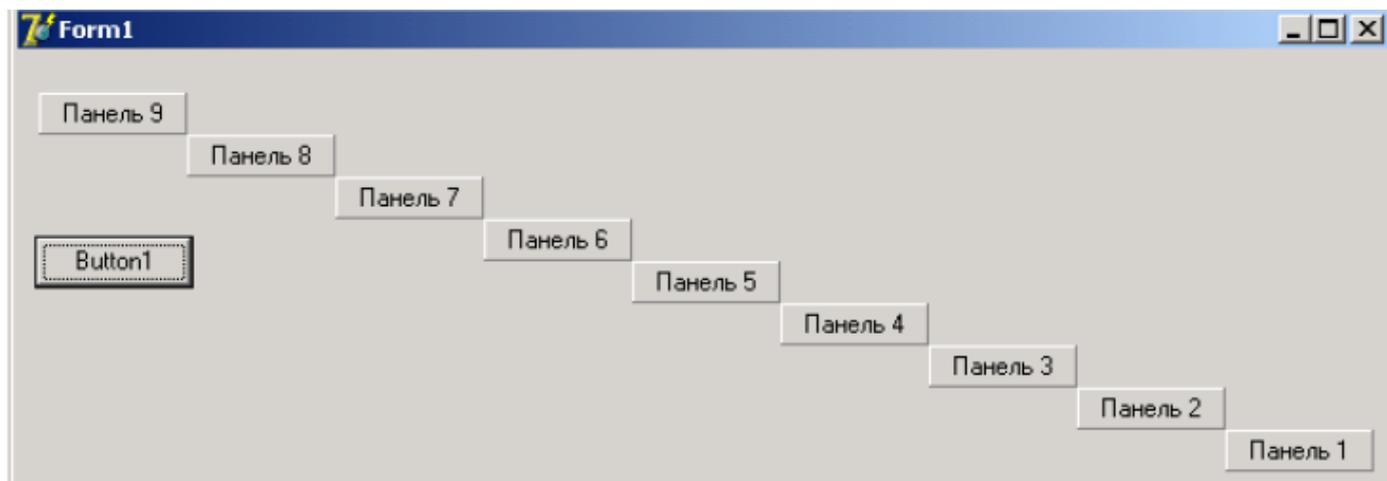
**Задание 1.** Реализовать метод, который будет размещать массив из 8 «новых» меток (Tlabel) на форме в одну строку.

Результат после нажатия на Button 1 формы.



**Задание 2.** Реализовать метод, который будет размещать массив из 9 «новых» меток (Tlabel) на форме по диагонали.

Результат после нажатия на Button 1 формы.



**Форма отчета:** отчет, защита работы.

### Практическое занятие № 22

**Тема:** Объявление и использование классов с использованием свойств

**Цель:** изучить возможности языка программирования для создания наследованного класса и перегрузки методов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Реализовать метод формы, который будет искать зависимость компонентов кнопки TButton и изменять их свойства.

**Задание 2.** Реализовать метод панели, который будет искать среди зависимых компонентов поля ввода TEdit и изменять их свойства Top и Left, таким образом, чтобы поля ввода были расположены по диагонали панели (с верхнего угла в нижний левый).

Программный код:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Menus;
```

```
type
TForm1 = class(TForm)
  Panel1: TPanel;
  Edit1: TEdit;
  Edit2: TEdit;
  Edit3: TEdit;
  Edit4: TEdit;
  Edit5: TEdit;
  Edit6: TEdit;
  Edit7: TEdit;
  procedure panel1click(Sender: TObject);
```

```
private
```

```

    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Panel1Click(Sender: TObject);
var i,x,y:integer;
begin
y:=0;
x:=panel1.Width;
for i:=1 to self.componentcount - 1 do
begin
if self.components[i] is Tedit then
begin
(self.components[i] as Tedit).LEFT:=x-(self.components[i] as Tedit).width;
(self.components[i] as Tedit).top:=y;
x:=x-(self.components[i] as Tedit).width;
y:=y+(self.components[i] as Tedit).height;
end;
end;
end;
end.

```

**Форма отчета:** отчет, защита работы.

### **Практическое занятие №23**

**Тема:** Создание наследованных классов.

**Цель:** изучить класс исключительных ситуации Delphi и создание приложений, генерирующих исключения и обрабатывающие различные формы ввода.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание.** Создать проект, содержащий обработку исключительной ситуации, файл формы и исходный код модуля.

1. В среде программирования Delphi создать новый проект, выбрав пункт меню File/New Application.
2. Сохраните этот проект в папке "C:\ для сохранения документов \....\ Lab10". (Unit1.pas под новым именем Main2.pas, а Project1.dpr под новым именем Lab2.dpr).
3. Разработайте приложение, обрабатывающее исключительную ситуацию, согласно вашему варианту индивидуального задания.

4. Открыть новое приложение.

5. Создать форму с пятью полями редактирования и пятью соответствующими надписями, которые поясняют, какой вид проверки осуществляет соответствующий компонент Edit. Форма также содержит кнопку для проверки содержимого первого поля редактирования.

Событие OnClick кнопки должно проверять целочисленность введенного в первое поле значения, например:

```
var
  Number, Code : Integer ;
begin
  if Edit1.Text <> ' ' then
  begin
    val ( Edit1. Text, Number, Code) ;
    if Code <> 0 then
    begin
      Edit1. SetFocus ;
      MessageDlg ( ' Not a number in the first edit ' , mtError,
        [ mbOK ] , 0 ) ;
    end ;
  end ;
end ;
```

При выходе из компонента Edit2 (событие OnExit) осуществляется аналогичная проверка.

```
var
  Number, Code : Integer ;
begin
  if (Sender as TEdit ). Text <> ' ' then
  begin
    val ((Sender as TEdit ). Text, Number, Code) ;
    if Code <> 0 then
    begin
      (Sender as TEdit ). SetFocus ;
      MessageDlg ( ' The edit field number ' +
        IntToStr ((Sender as TEdit ). Tag) +
        ' does not have a valid number' , mtError, [ mbOK ] , 0 ) ;
    end ;
  end ;
end ;
```

Текст сообщения об ошибке может варьироваться.

Третий компонент Edit выполняет аналогичную проверку при каждом изменении его содержимого (используя событие OnChange).

Записать код для события OnKeyPress компонента Edit4 и проверить, не является ли введенный символ числом или клавишей Back space (которая имеет числовое значение 8).

```
begin
  if not ( key in [ '0' .. '9' , # 8 ] ) then
    begin
      Key := # 0 ;
      MessageBeep ( $ FFFFFFFF ) ;
    end;
  end;
```

Для события OnEnter компонента Edit5 записать код, в котором необходимо преобразовать введенные символы в число с помощью функции StrToInt. Использовать исключение для обработки ошибки EConvertError.

#### 6. Варианты индивидуальных заданий

Вариант	Задача
1	Создать программу, позволяющую пользователю ввести два числа, которые программа разделит. Необходимо поместить на форму три объекта класса TEdit – два для операндов, один – для результата и кнопку (объект класса TButton), нажимая на которую пользователь выполняет деление. Исключить попытку деления на ноль а так же введения символов вместо цифр. Выдать сообщение о типе возникшей ошибке.
2	Создать программу, вычисляющую корни квадратного уравнения ( $ax^2+bx+c=0$ ). Необходимо поместить на форму четыре объекта класса TEdit - три для коэффициентов квадратного уравнения, один – для результата и кнопку (объект класса TButton), нажимая на которую пользователь выполняет нахождение корней. Исключить ввод символов вместо цифр, получение отрицательного дискриминанта и ввод $a = 0$ . Вывести при всех типах ошибок одно и то же сообщение
3	Создать программу с “бесконечным” циклом типа while. В цикле увеличивать переменную I до значения, заданного пользователем. При достижении этого значения выходить из цикла с помощью возбуждения исключения EAbort. Выдать сообщение о выходе из цикла в блоке Except. Необходимо поместить на форму кнопку (объект класса TButton), которая запускает цикл; сообщение можно выдать с помощью функции

	ShowMessage, или поместить на форму метку (объект класса TLabel), в которую помещается сообщение.
4	Создать программу, вычисляющую тангенс угла. Необходимо поместить в форму два компонента Tedit для ввода значения и результата и кнопку Tbutton для вычисления значения тангенса. Исключить ввод символов вместо цифр и получение значения тангенса угла 60 градусов. Предусмотреть возможность ввода значений в радианах
5	Создать программу, вычисляющую логарифм числа. Для этого необходимо поместить в форму два компонента Tedit для ввода значения и результата и кнопку Tbutton для вычисления значения логарифма. Исключить ввод символов вместо цифр и получение значения логарифма 0.
6	Создать программу обработки исключения при обращении к несуществующему элементу массива. В форму поместите поля редактирования для ввода - вывода значений и номеров элементов массива и кнопку для обработки события
7	Создать программу, вычисляющую тангенс угла. Необходимо поместить в форму два компонента Tedit для ввода значения и результата и кнопку Tbutton для вычисления значения тангенса. Исключить ввод символов вместо цифр и получение значения тангенса угла 90 градусов. Предусмотреть возможность ввода значений в радианах
8	Создать программу с “бесконечным” циклом типа while. В цикле увеличивать переменную I до значения, заданного пользователем. При достижении этого значения выходить из цикла с помощью возбуждения исключения EAbort. Выдать сообщение о выходе из цикла в блоке Excerpt. Необходимо поместить на форму кнопку (объект класса TButton), которая запускает цикл; сообщение можно выдать с помощью функции ShowMessage, или поместить на форму метку (объект класса TLabel), в которую помещается сообщение.
9	Создать программу, позволяющую пользователю ввести два числа, которые программа разделит. Необходимо поместить на форму три объекта класса TEdit - два для операндов, один - для результата и кнопку (объект класса TButton), нажимая на которую пользователь выполняет деление. Исключить попытку деления на ноль а также введения символов

вместо цифр. Выдать сообщение о типе возникшей ошибке.

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 24

**Тема:** Изучение интегрированной среды разработчика.

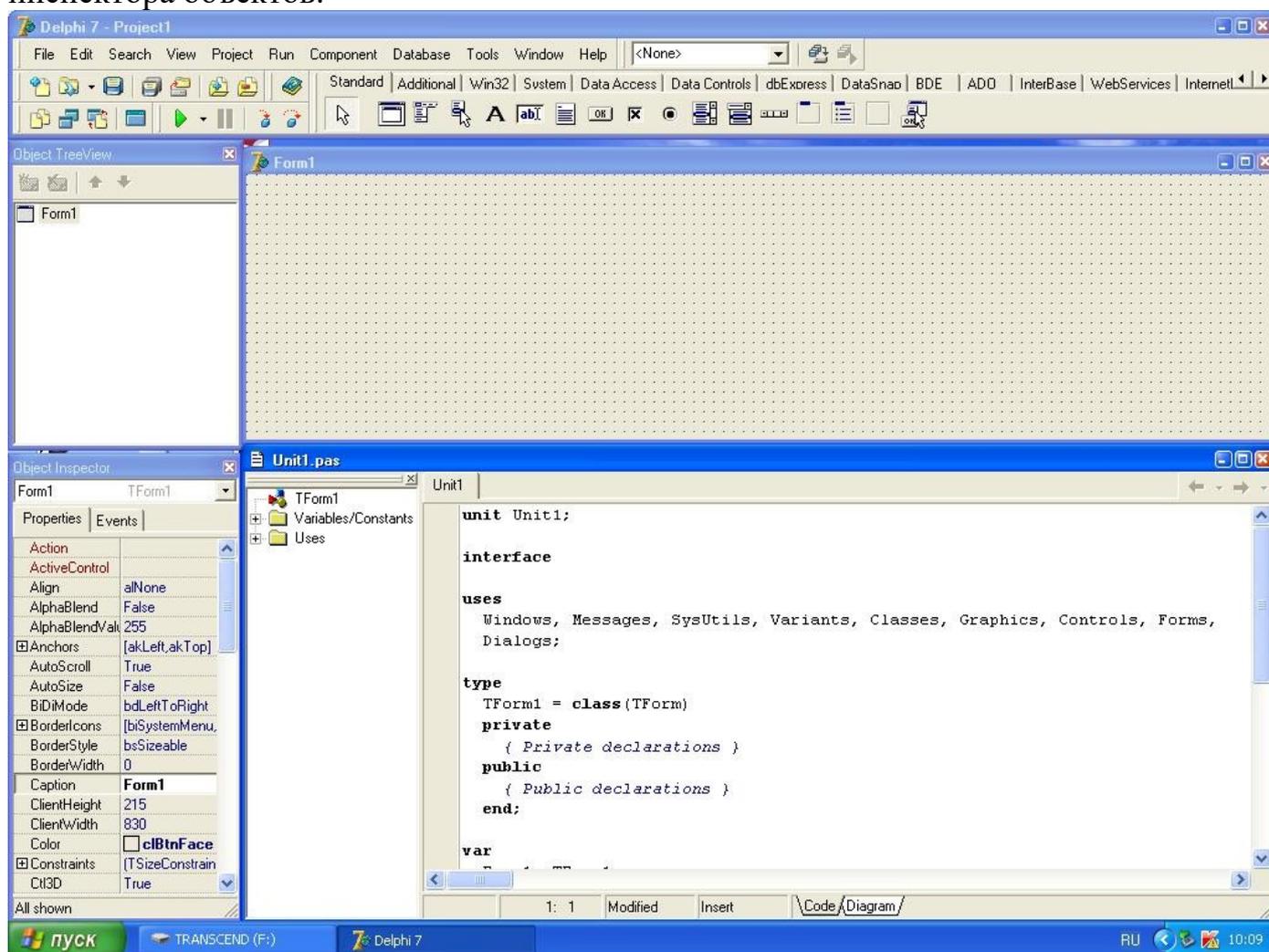
**Цель:** изучить интегрированную среду разработки Delphi 7 для работы с формой и элементами управления.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** На рисунке нужно отметить следующие объекты: главное окно, окно формы, окно кода программы, окно браузера кода, окно дерева объектов, окно инспектора объектов.



**Задание 2.** Обозначить инструментальные кнопки которые открывают быстрый доступ к наиболее важным командам главного меню.

кнопка	описание	Эквивалентно команде

**Задание 3.** Описать элементы управления в вкладке «Standart»(название, применение и основные свойства).



**Форма отчета:** отчет, защита работы

### Практическое занятие № 25

**Тема:** Создание проекта с использованием компонентов для работы с текстом.

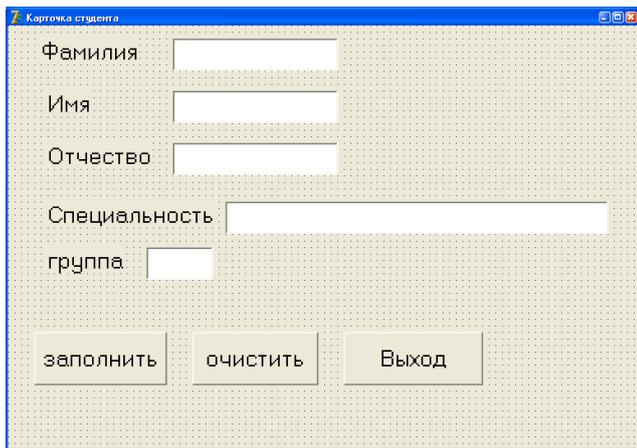
**Цель:** изучить возможности языка программирования для создания проектов с использованием компонентов для работы с текстом.

**Оборудование:** тетрадь, ручка, ПК

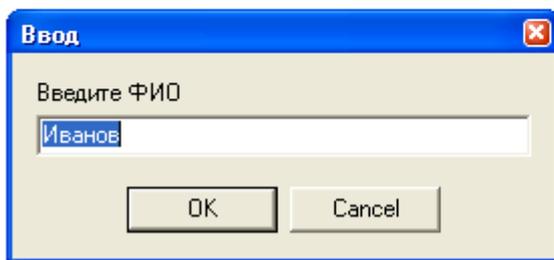
**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создайте диалоговое окно для вывода карточки студента.



**Задание 2.** Создайте диалог для ввода фамилии, имени и отчества клиента.



**Форма отчета:** отчет, защита работы.

### Практическое занятие № 20

**Тема:** Создание проекта с использованием компонентов стандартных диалогов.

**Цель:** изучить возможности языка программирования для создания проектов с использованием компонентов стандартных диалогов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

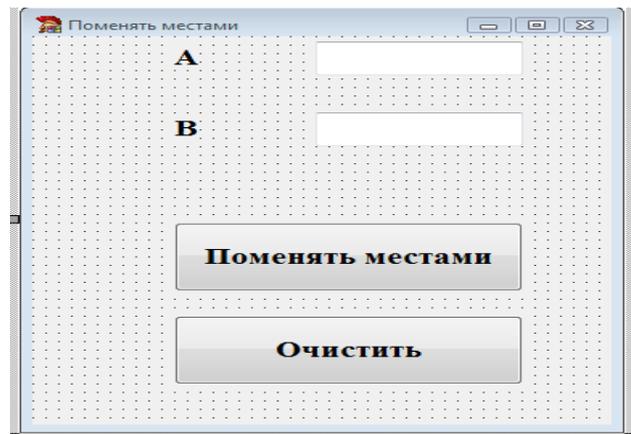
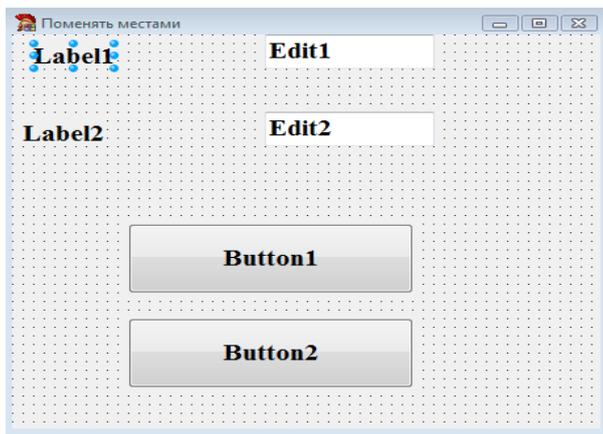
**Ход выполнения:**

**Задание 1.** Описать элементы управления в вкладке «Standart»(название, применение и основные свойства).



**Задание 2.** Создайте приложение, которое при нажатии кнопки меняет местами значение двух переменных (метод пустого стакана).

1. Создайте форму и включите в неё нужные компоненты.



2. Напишите обработчик событий для кнопок «Рассчитать» и «Очистить».

Кнопка «Рассчитать»

```
Procedure TForm1.Button2Click(Sender: TObject);
```

```
Var A,B,C:integer; {объявляются переменные A,B,C целого типа}
```

```
begin
```

```
A:=StrToInt(Edit1.Text); {целой переменной A присваивается значение взятое из компонента Edit1.Text в виде текста и преобразованное в число с помощью функции StrToInt}
```

```
B:=StrToInt(Edit2.Text); {целой переменной B присваивается значение взятое из компонента Edit2.Text в виде текста и преобразованное в число с помощью функции StrToInt}
```

```
C:=A; A:=B; B:=C; {меняем местами значения переменных}
```

```
Edit1.Text:=IntToStr(A); {В компонент Edit1.Text кладется значение целой переменной A преобразованное в текс с помощью функции IntToStr}
```

```
Edit2.Text:=IntToStr(B); {В компонент Edit2.Text кладется значение целой переменной B преобразованное в текс с помощью функции IntToStr}
```

```
end;
```

Кнопка «Очистить»

```
Procedure TForm1.Button2Click(Sender: TObject);
```

```
Begin
```

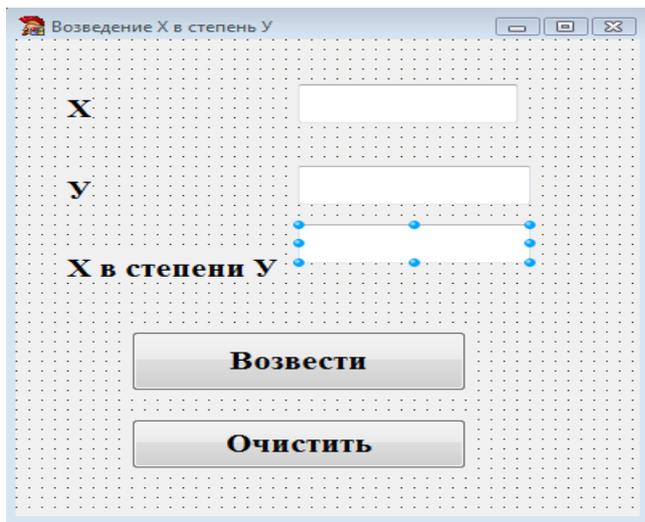
```
Edit1.Text:=' '; {Компонентам Edit1.Text, Edit2.Text присваивается значение
```

```
Edit2.Text:=' '; пустого текста}
```

```
end;
```

**Задание 3.** Создайте приложение, которое при нажатии кнопки возводит X в степень Y.

1. Создайте форму и включите в неё нужные компоненты.



2. Напишите обработчик событий для кнопок «Рассчитать» и «Очистить».

Кнопка «Рассчитать»

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```
Var X,Y,R:integer; {объявляются переменные X,Y,R вещественного типа}
```

```
begin
```

```
X:=StrToFloat(Edit1.Text); {переменной X присваивается значение взятое из компонента Edit1.Text в виде текста и преобразованное в число с помощью функции StrToFloat}
```

```
Y:=StrToFloat(Edit2.Text); { переменной Y присваивается значение взятое из компонента Edit2.Text в виде текста и преобразованное в число с помощью функции StrToFloat}
```

```
R:= Exp(Y*Ln(X)); {В языке Pascal нет стандартной операции или функции для вычисления  $X^Y$ . Для решения такой задачи используют встроенную функцию  $X^Y = E^{LnX^Y} = E^{YLnX}$  Эта формула применяется для нецелых показателей степени Y, а для целых намного эффективней умножать }
```

```
Edit3.Text:=FloatToStr(R); {В компонент Edit3.Text кладется значение переменной R преобразованное в текст с помощью функции FloatToStr}
```

```
end;
```

Кнопка «Очистить»

```
Procedure TForm1.Button2Click(Sender: TObject);
```

```
Begin
```

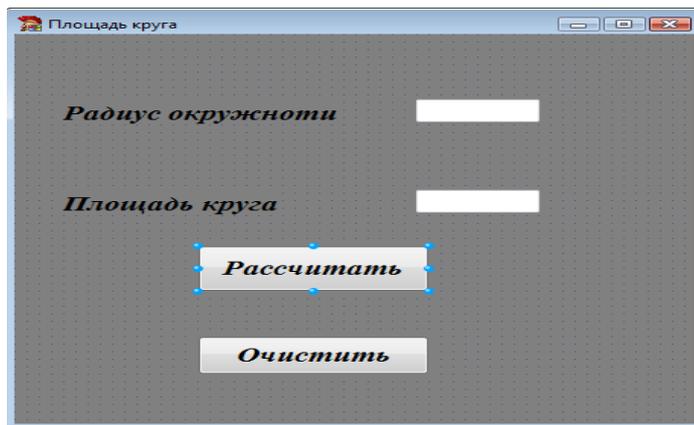
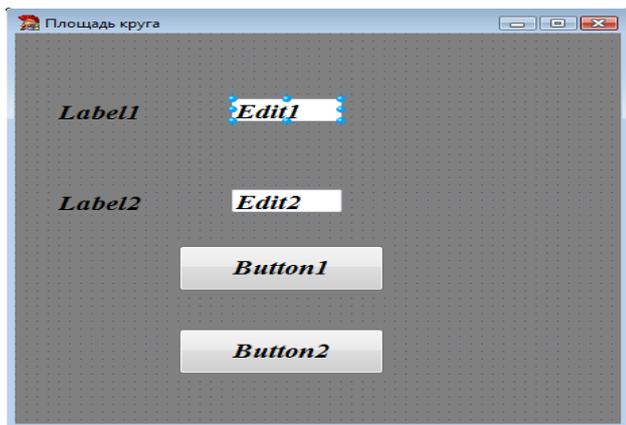
```
Edit1.Text:=' '; {Компонентам Edit1.Text, Edit2.Text присваивается значение
```

```
Edit2.Text:=' '; пустого текста}
```

```
end;
```

**Задание 4.** Создайте приложение, которое при нажатии кнопки вычисляло площадь круга

1. Создайте форму и включите в неё нужные компоненты.



3. Напишите обработчик событий для кнопок «Рассчитать» и «Очистить».

Кнопка «Рассчитать»

```
Procedure TForm1.Button1Click(Sender: TObject);
```

```
var R,Sq:extended; «объявляются переменные R,Sq вещественного типа»
```

```
begin
```

```
R:=StrToFloat(Edit1.Text); «вещественной переменной R присваивается значение  
взятое из компонента Edit1.Text в виде текста и преобразованное в число с помощью  
функции StrToFloat»
```

```
Sq:=Pi*Sqr(R); «вещественной переменной Sq присваивается значение произведения  
Π на  $\sqrt{R}$ »
```

```
Edit2.Text:= FloatToStr(Sq); «В компонент Edit2.Text кладется значение вещественной  
переменной Sq преобразованное в текс с помощью функции FloatToStr»
```

```
end;
```

Кнопка «Очистить»

```
Procedure TForm1.Button2Click(Sender: TObject);
```

```
Begin
```

```
Edit1.Text:=' '; {Компонентам Edit1.Text, Edit2.Text присваивается значение
```

```
Edit2.Text:=' '; пустого текста}
```

```
end;
```

**Задание 5.** Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

варианта	1-ая буква фамилии	задача
1	А, Б, В	Дан диаметр окружности $d$ . Найти ее длину $L = \pi \cdot d$ . В качестве значения $\pi$ использовать 3.14.
2	Г, Д, Е,	Дана длина ребра куба $a$ . Найти объем куба $V = a^3$ и площадь его поверхности $S =$ $6 \cdot a^2$ .
3	Ж, З, И	Даны длины ребер $a, b, c$

		прямоугольного параллелепипеда. Найти его объем $V = a \cdot b \cdot c$ и площадь поверхности $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$ .
4	К, Л	Найти длину окружности $L$ и площадь круга $S$ заданного радиуса $R$ : $L = 2 \cdot \pi \cdot R$ , $S = \pi \cdot R^2$ .
5	Н, О, П	Даны два числа $a$ и $b$ . Найти их <i>среднее арифметическое</i> : $(a + b) / 2$ .
6	Р, С, Т	Даны два неотрицательных числа $a$ и $b$ . Найти их <i>среднее геометрическое</i> , то есть квадратный корень из их произведения: $\sqrt{a \cdot b}$ .
7	У, Ф, Х	Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.
8	Ч, Ш, Щ	Даны катеты прямоугольного треугольника $a$ и $b$ . Найти его гипотенузу $c$ и периметр $P$ : $c = \sqrt{a^2 + b^2}$ , $P = a + b + c$ .
9	Э, Ю, Я	Дана длина $L$ окружности. Найти ее радиус $R$ и площадь $S$ круга, ограниченного этой окружностью, учитывая, что $L = 2 \cdot \pi \cdot R$ , $S = \pi \cdot R^2$ . В качестве значения $\pi$ использовать 3.14.
10	Ё, М	Найти расстояние между двумя точками с заданными координатами $x_1$ и $x_2$ на числовой оси: $ x_2 - x_1 $ .

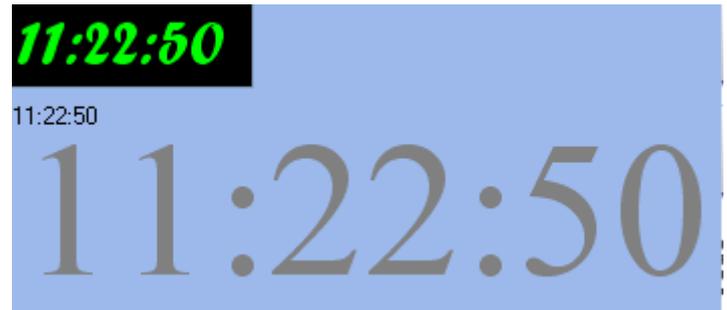
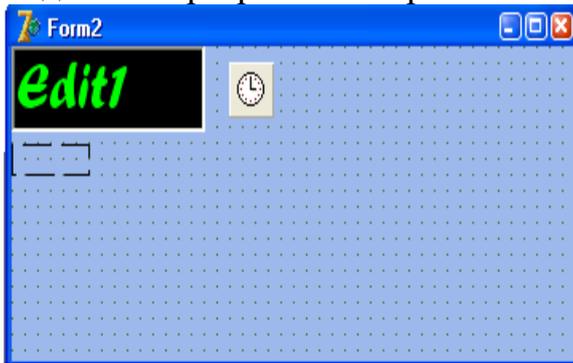
**Форма отчета:** отчет, защита работы.

### Практическое занятие № 26

**Тема:** Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

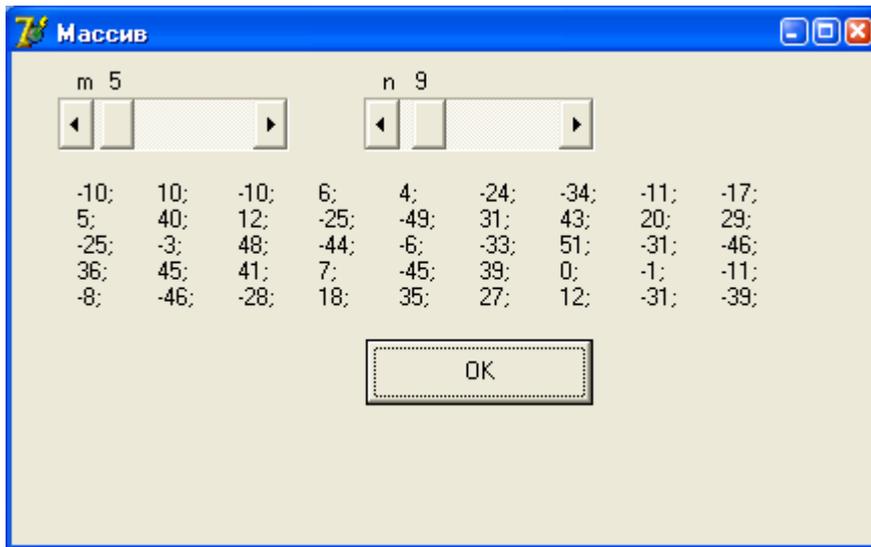
**Цель:** изучить возможности языка программирования для создания проектов с использованием компонентов системы меню.

**Задание 1.** разработать приложение часы.



```
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TForm2 = class(TForm)
    Edit1: TEdit;
    Timer1: TTimer;
    Image1: TImage;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
implementation
{$R *.dfm}
procedure TForm2.Timer1Timer(Sender: TObject);
begin
  edit1.Text := timetostr(time);
  image1.Canvas.TextOut(1, 1, timetostr(time));
  Canvas.TextOut(5, 50, timetostr(time));
end;
end.
```

**Задание 2.** Создать проект «задание матрицы  $m \times n$  со случайно сгенерированными элементами» с использованием компонентов системы меню.



```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
  scm: TScrollBar;
```

```
  scn: TScrollBar;
```

```
  Label1: TLabel;
```

```
  Label2: TLabel;
```

```
  Label3: TLabel;
```

```
  Button1: TButton;
```

```
  Label4: TLabel;
```

```
  Label5: TLabel;
```

```
  procedure Button1Click(Sender: TObject);
```

```
  procedure scmChange(Sender: TObject);
```

```
  procedure scnChange(Sender: TObject);
```

```
private
```

```
  { Private declarations }
```

```
public
```

```
  { Public declarations }
```

```
end;
```

```
var
```

```
  Form1: TForm1;
```

```
implementation
```

```
  {$R *.dfm}
```

```
  procedure TForm1.Button1Click(Sender: TObject);
```

```
  var k:array of array of -50..50;
```

```
  i,j:byte;
```

```
  n,m:1..10;
```

```
begin
```

```

m:=scm.Position;
n:=scn.Position;
setlength(k,m,n);
label3.Caption:="";
randomize;
for i:=0 to m-1 do
begin
for j:=0 to n-1 do
begin
k[i,j]:=round(-50+101*random);
label3.Caption:=label3.Caption+inttostr(k[i,j])+'; '+#9;
end;
label3.Caption:=label3.Caption+#13#10;
end;
end;

```

```

procedure TForm1.scmChange(Sender: TObject);
begin
label4.Caption:=inttostr(scm.Position);
end;

```

```

procedure TForm1.scnChange(Sender: TObject);
begin
label5.Caption:=inttostr(scn.Position);
end;
end.

```

**Задание 3.** Решить задачу своего варианта с использованием компонентов системы меню.

Вариант	Задача
1	Даны два целых числа $A$ и $B$ ( $A < B$ ). Вывести в порядке убывания все целые числа, расположенные между $A$ и $B$ (не включая числа $A$ и $B$ ), а также количество $N$ этих чисел.
2	Дано целое число $N$ ( $> 0$ ). Найти квадрат данного числа, используя для его вычисления следующую формулу: $N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1).$
3	Дано вещественное число $A$ и целое число $N$ ( $> 0$ ). Найти $A$ в степени $N$ : $A^N = A \cdot A \cdot \dots \cdot A$

4	Дано вещественное число $A$ и целое число $N (> 0)$ . Используя один цикл, вывести все целые степени числа $A$ от 1 до $N$ .
5	Дано целое число $N (> 0)$ . Найти произведение $N! = 1 \cdot 2 \cdot \dots \cdot N$
6	Дано целое число $N (> 0)$ . Найти произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ ( $N$ сомножителей).
7	Дано целое число $N (> 0)$ . Найти значение выражения $1.1 - 1.2 + 1.3 - \dots$ ( $N$ слагаемых, знаки чередуются). Условный оператор не использовать.
8	Даны два целых числа $A$ и $B (A < B)$ . Найти сумму всех целых чисел от $A$ до $B$ включительно.
9	Даны два целых числа $A$ и $B (A < B)$ . Найти произведение всех целых чисел от $A$ до $B$ включительно
10	Даны два целых числа $A$ и $B (A < B)$ . Найти сумму квадратов всех целых чисел от $A$ до $B$ включительно.

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 27

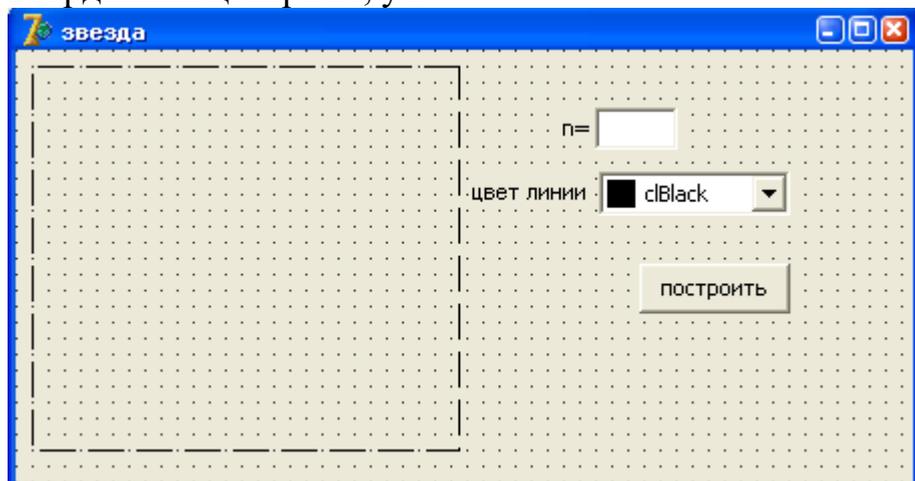
**Тема:** События компонентов (элементов управления), их сущность и назначение.

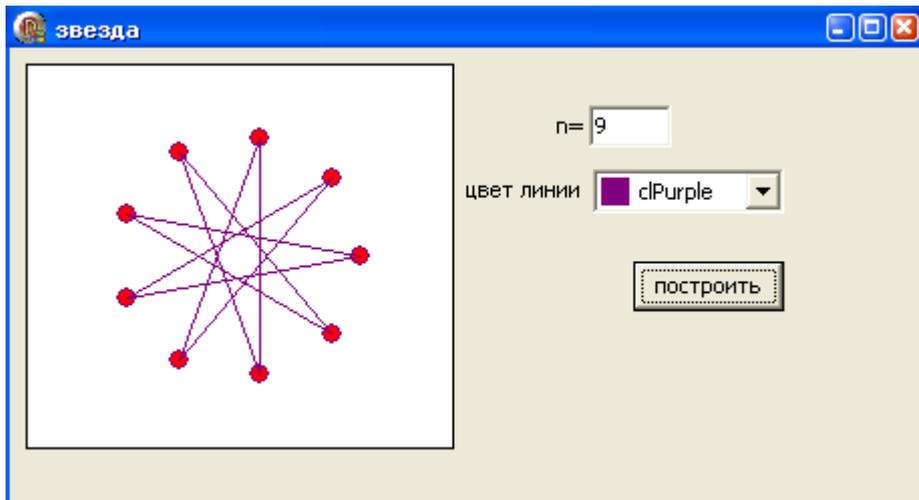
**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Дано натуральное число  $n^3$ . Построить звезду, у которой  $n$  вершин и координаты центра  $x_0, y_0$ .





```

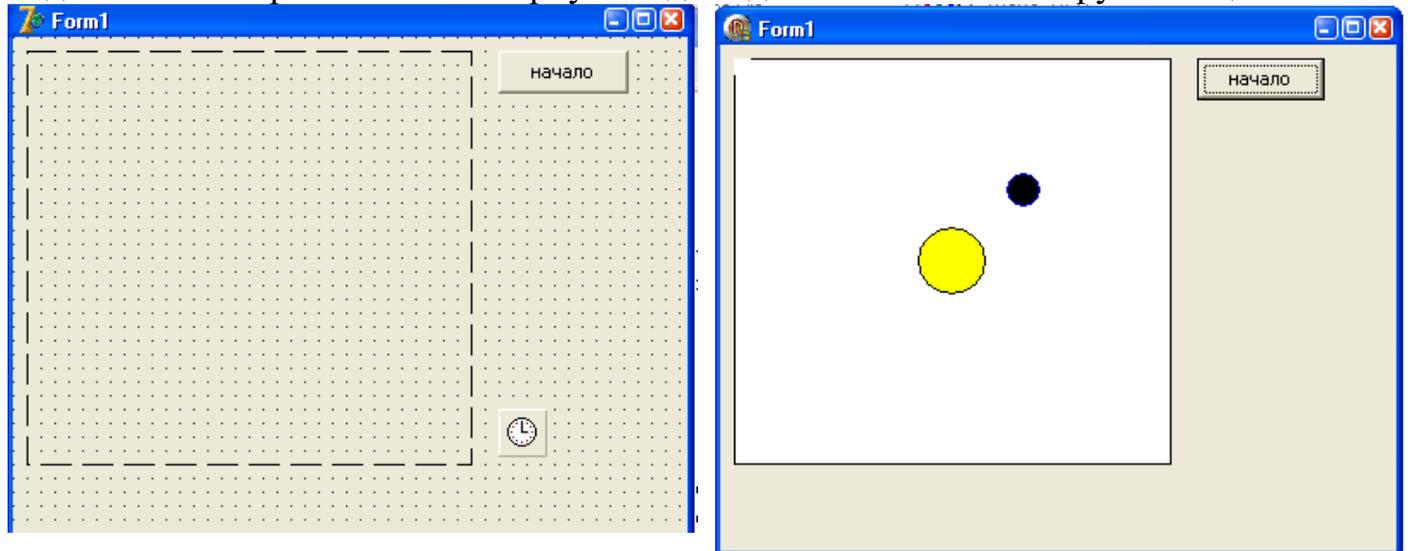
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    CBox: TColorBox;
    Img1: TImage;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
  procedure TForm1.Button1Click(Sender: TObject);
  var x,y: array [1..100] of integer;
  xc,yc,h,r:byte;
  ugol:real;
  n,i:byte;
  begin
    img1.Canvas.Brush.Color:=clwhite;
    img1.Canvas.Rectangle(0,0,img1.Width,img1.height);
    n:=strtoint(edit1.text);
    xc:=round(img1.width/2);
  
```

```

yc:=round(img1.height/2);
r:=60;
ugol:=2*pi/n;
img1.Canvas.Pen.Color:=cbox.selected;
for i:=1 to n do
begin
x[i]:=xc+round(r*cos(-(i-1)*ugol));
y[i]:=yc+round(r*sin(-(i-1)*ugol));
img1.Canvas.Brush.Color:=clred;
img1.Canvas.ellipse(x[i]-5,y[i]-5,x[i]+5,y[i]+5);
end;
h:=(n-1) div 2;
for i:=1 to n do
begin
img1.Canvas.MoveTo(x[i],y[i]);
if (i+h<=n) then img1.Canvas.LineTo(X[i+h],y[i+h])
else img1.Canvas.lineto(x[i+h-n],y[i+h-n]);
end;
end;
end.

```

**Задание 2.** Построить компьютерную модель движения Земли вокруг солнца.



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Img1: TImage;
    Timer: TTimer;

```

```

BtnOK: TButton;
procedure BtnOKClick(Sender: TObject);
procedure TimerTimer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
var xc,yc,r,x,y:byte;
a:real;
{$R *.dfm}
procedure TForm1.BtnOKClick(Sender: TObject);
begin
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(0,0,img1.Width,img1.height);
xc:=round(img1.width/2);
yc:=round(img1.height/2);
a:=0;
r:=60;
img1.Canvas.Brush.Color:=clyellow;
img1.canvas.ellipse(xc-20,yc-20,xc+20,yc+20);
timer.enabled:=true;
end;
procedure TForm1.TimerTimer(Sender: TObject);
begin
img1.Canvas.pen.color:=clwhite;
img1.Canvas.brush.color:=clwhite;
img1.canvas.ellipse(x-10,y-10,x+10,y+10);
a:=a+pi/16;
if a>2*pi then a:=0;
x:=round(xc+r*cos(-a));
y:=round(yc+r*sin(-a));
img1.Canvas.pen.color:=clblue;
img1.Canvas.brush.color:=clblack;
img1.canvas.ellipse(x-10,y-10,x+10,y+10);
end;
end.

```

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 28

**Тема:** Создание процедур на основе событий.

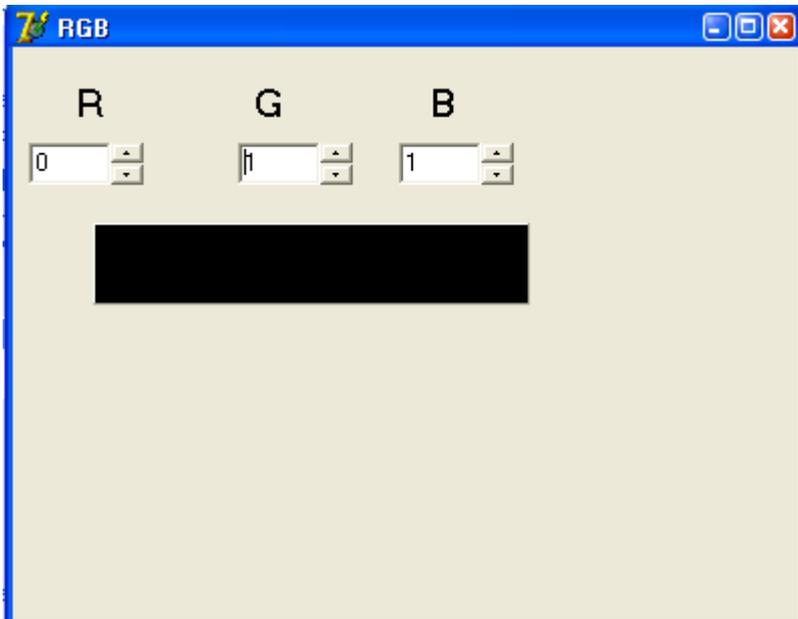
**Цель:** изучить возможности языка программирования для создания проектов с использованием компонентов системы меню.

**Оборудование:** тетрадь, ручка, ПК

**Методическое указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создать проект «RGB- таблица» с использованием компонентов системы меню.



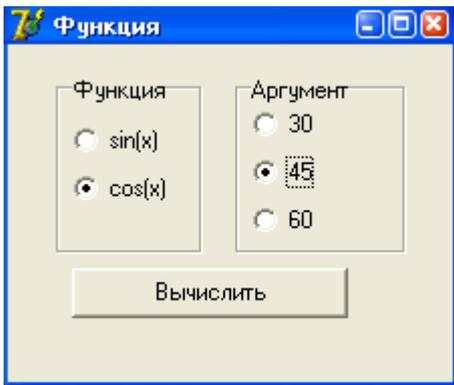
```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    UpDown1: TUpDown;
    UpDown2: TUpDown;
    UpDown3: TUpDown;
    Panel1: TPanel;
    procedure UpDown1Changing(Sender: TObject; var AllowChange: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```

var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.UpDown1Changing(Sender: TObject;
  var AllowChange: Boolean);
begin
  panel1.Color:=rgb(updown1.Position,updown2.Position,updown3.Position);
end;
end.

```

**Задание 2.** Создать проект «вычисление  $\sin 30$ ,  $\sin 45$ ,  $\sin 60$  и  $\cos 30$ ,  $\cos 45$ ,  $\cos 60$ » с использованием компонентов системы меню.



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    rdbS: TRadioButton;
    rdbC: TRadioButton;
    GroupBox2: TGroupBox;
    rdb30: TRadioButton;
    rdb45: TRadioButton;
    rdb60: TRadioButton;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;

```

```

implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var k:1..2;
x:byte; f:real;
begin
if rdbS.Checked<>false or rdbC.Checked<>false then

begin
if rdb30.Checked<>false or rdb45.Checked<>false
or rdb60.Checked<>false then
begin
if rdbS.Checked=true then k:=1 else k:=2;
if rdb30.Checked=true then x:=30 else
begin
if rdb45.Checked=true then x:=45 else x:=60;
end;
if k=1 then f:=sin(pi*x/180) else f:=cos(pi*x/180);
showmessage(floattostrF(f,ffixed,5,4));
end
else
showmessage('Выберите аргумент!');
end
else
showmessage('Выберите функцию!');
end;
end.

```

**Форма отчета:** отчет, защита работы.

### **Практическое занятие № 29**

**Тема:** Создание проекта с использованием кнопочных компонентов.

**Цель:** изучить возможности языка программирования для создания проектов с использованием кнопочных компонентов.

**Оборудование:** тетрадь, ручка, ПК

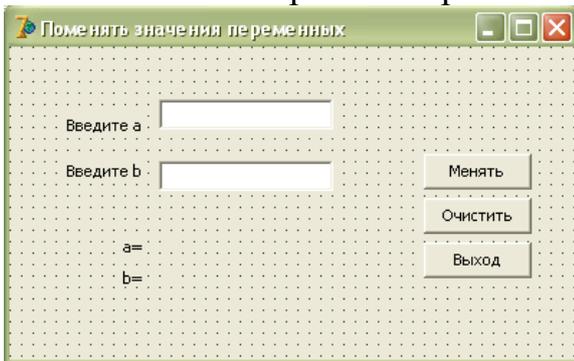
**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1 .** Правительство гарантирует, что инфляция в новом году составит  $p$  % в месяц. Какого роста цен за год можно ожидать.



**Задание 2.** Поменять значения переменных (решить задачу двумя способами: с использованием третьей переменной, без использования третьей переменной)



**Задание 3.** Решить задачу своего варианта

Вариант	Задача
1	Найти длину окружности и площадь круга
2	Найти площадь поверхности цилиндра
3	Найти площадь поверхности и объем правильного тетраэдра
4	Найти площадь поверхности и объем октаэдра
5	Найти площадь поверхности конуса
6	Найти площадь поверхности и объем шара
7	Найти площадь поверхности и объем куба
8	Найти площадь поверхности шестигранной правильной призмы
9	Найти площадь поверхности и объем прямоугольного параллелепипеда
10	Найти площадь поверхности правильной четырехугольной пирамиды

**Примечание:**

- решение каждой задачи оформить согласно этапам решения задач на ЭВМ (Математическая модель, интерфейс программы, алгоритмизация, программирование);
- отчет оформить в печатном виде (с титульным листом).

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 30

**Тема:** Создание проекта с использованием компонентов стандартных диалогов и системы меню.

**Цель:** изучить возможности языка программирования для создания проектов с использованием кнопочных компонентов стандартных диалогов и системы меню.

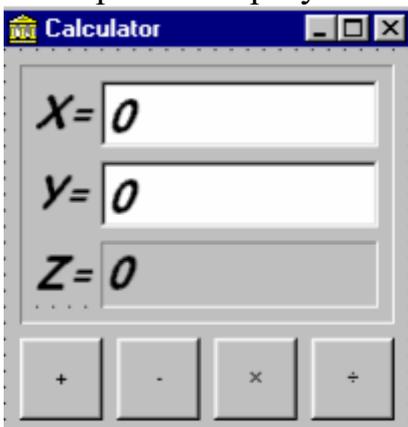
**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

## Ход выполнения:

### Задание 1.

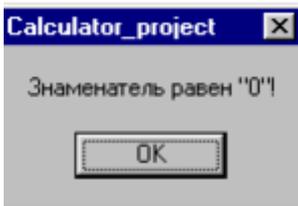
1. Запустить Borland Delphi 7: пуск→все программы→ Borland Delphi → Delphi 7.
2. Совершите экскурс в среду визуального программирования Delphi. При этом результаты своей работы не сохраняйте на диске.
3. Попытайтесь создать различные приложения с помощью Expert-ов Delphi, исследовать в окне редактирования полученный код. Приложения на диске не запоминать.
4. Выберите пункт меню File/New Application, этим вы создадите новый проект приложения. Выберите команду File/Save Project As... → В появившемся диалоге перейдите "рабочий стол"→ Откройте папку ""для сохранения→ откройте свою папку. Если такой нет, то создайте её, щелкнув правой кнопкой мыши на свободном месте, выбрав из появившегося контекстного меню пункт Создать/папку и введя нужное название (после чего не забудьте её открыть). → Создайте (как описано в предыдущем пункте) папку " Lab3". → Сохраните Unit1.pas под новым именем Main.pas, а Project1.dpr под новым именем Lab3.dpr.
5. Выберите из палитры Standard визуальных компонентов и поместите в форму следующие компоненты:
  - Окно редактирования со связанной с ним меткой Operand 1. В этом окне вводится первый операнд.
  - Окно редактирования со связанной с ним меткой Operator. В этом 14 окне вводится операция. В программе следует предусмотреть операции +, -, / и \* .
  - Окно редактирования со связанной с ним меткой Operand 2. В этом окне вводится второй операнд.
  - Окно редактирования со связанной с ним меткой Result. В этом окне отображается результат запрошенной вами операции.



6. В событиях OnClick каждой кнопки опишите соответствующие действия, например, для кнопки код события должен выглядеть следующим образом: 

```
If Edit2.Text='0' Then ShowMessage('Знаменатель равен "0"!') Else Edit3.Text:=FloatToStr(StrToFloat(Edit1.Text)/StrToFloat(Edit2.Text));
```

 Обратите внимание, что в данном действии осуществляется проверка деления на ноль. Оператор ShowMessage выдает сообщение, в случае, когда Y равен "0".



Сообщение об ошибке Процедуры FloatToStr и StrToFloat выполняют преобразования из числовой переменной в строковую и из строковой в числовую соответственно. Остальные клавиши должны быть описаны соответственно.

7. Измените программу, дополнив ее возможностями очистки содержимого полей (например, кнопка Clear) и вычисления значения заданной функции, согласно варианту индивидуального задания. Сохраните форму и проект.

**Задание 2.** Составить программу вычисления значения функции

$$y = \frac{(x-2)(x-4)(x-6)\dots(x-64)}{(x-1)(x-3)(x-5)\dots(x-63)}$$

**Задание 3.** Составить программу, которая проверяет, является ли заданное число совершенным. Совершенным называется натуральное число, равное сумме всех своих делителей (исключая само число). Например,  $28 = 1 + 2 + 4 + 7 + 14$ .

**Задание 4.** Реализовать задачу своего варианта

варианта	1-ая буква фамилии	задача
1	А, Б, В	$Z = \sin(x+5y)$ .
2	Г, Д, Е,	$Z = \tan(3X+2Y)$ .
3	Ж, З, И	$Z = \sin(2x+3y)$ .
4	К, Л, Ё	$Z = \tan(2X+Y)$ .
5	Н, О, П	$Z = \cos(4X-2Y)$ .
6	Р, С, Т	$Z = \log(2X+Y+3)$ .
7	У, Ф, Х	Создайте кнопку для возведения X в целочисленную степень Y
8	Ч, Ш, Щ	$Z = X!$
9	Э, Ю, Я, М	$Z = \cos(6X-3Y+1)$ .

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 31

**Тема:** Создание проектов.

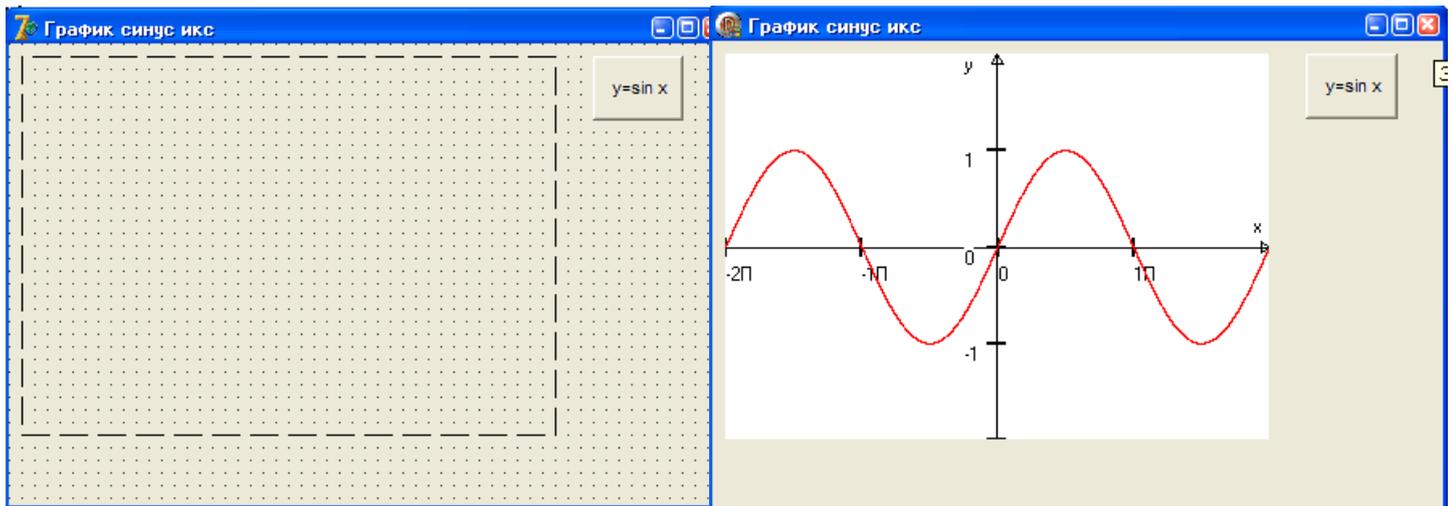
**Цель:** изучить возможности языка программирования для создания проектов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Построить график функции  $y = \sin x$ .



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons;
type
  TForm1 = class(TForm)
    graf: TImage;
    SpeedButton1: TSpeedButton;
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.SpeedButton1Click(Sender: TObject);
var xmin,ymin,xmax,ymax,x,y:real;
    xgmin,ygmin,xgmax,ygmax,xg,yg:integer;
begin
  xmin:=-2*pi;
  xmax:=2*pi;
  ymin:=-2;
  ymax:=2;
  xgmin:=0;
  xgmax:=graf.width;
  ygmin:=graf.Height;
  ygmax:=0;
  graf.Canvas.pen.Color:=clblack;

```

```

xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(-2*pi-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(0-ymin));
graf.Canvas.MoveTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(2*pi-xmin));
graf.Canvas.lineTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(2*pi-(pi/16)-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(0.05-ymin));
graf.Canvas.lineTo(xg,yg);
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(-0.05-ymin));
graf.Canvas.lineTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(2*pi-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(0-ymin));
graf.Canvas.lineTo(xg,yg);
graf.Canvas.TextOut(xg-10,yg-20,'x');
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(0-ymin));
x:=-2*pi;
while x<xmax do
begin
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(x-xmin));
graf.Canvas.rectangle(xg-1,yg-6,xg+1,yg+6);
if round(x/pi)=0 then graf.Canvas.TextOut(xg,yg+10,'0')
else graf.Canvas.TextOut(xg,yg+10,inttostr(round(x/pi))+'\Pi');
x:=x+pi;
end;
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(0-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(-2-ymin));
graf.Canvas.MoveTo(xg,yg);
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(2-ymin));
graf.Canvas.lineTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(-pi/25-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(1.9-ymin));
graf.Canvas.lineTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(pi/25-xmin));
graf.Canvas.lineTo(xg,yg);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(0-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(2-ymin));
graf.Canvas.lineTo(xg,yg);
graf.Canvas.TextOut(xg-20,yg,'y');
y:=-2;
while y<ymax do
begin
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(y-ymin));
graf.Canvas.rectangle(xg-6,yg-1,xg+6,yg+1);
if round(x/pi)<>0 then graf.Canvas.TextOut(xg-20,yg,inttostr(round(y)));

```

```

y:=y+1;
end;
x:=xmin;
while x<=xmax do
begin
y:=sin(x);
xg:=xgmin+round((xgmax-xgmin)/(xmax-xmin)*(x-xmin));
yg:=ygmin+round((ygmax-ygmin)/(ymax-ymin)*(y-ymin));
x:=x+(xmax-xmin)/10000;
graf.Canvas.Pixels[xg,yg]:=clred;
end;
end;
end.

```

**Задание 2.** Построить график функции точечными кусочно-линейным способами, используя канву. Построить график этой же функции с помощью объекта Chart.

**Задание 3.** Построить график функции своего варианта.

Вариант	функция
1	$y=\cos x$
2	$y=\sin(2x)$
3	$y=\cos(2x)$
4	$y=\sin(3x/2)$
5	$y=\cos(5x/7)$
6	$y=\sin x/\cos 5x$
7	$y=5/(7x+1)$ .
8	$y=x^2+3x+7$
9	$y=x^4+2x^3+7$
10	$y=x^3+7x+10$

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 32

**Тема:** Сбор элементов данных во время работы с проектом.

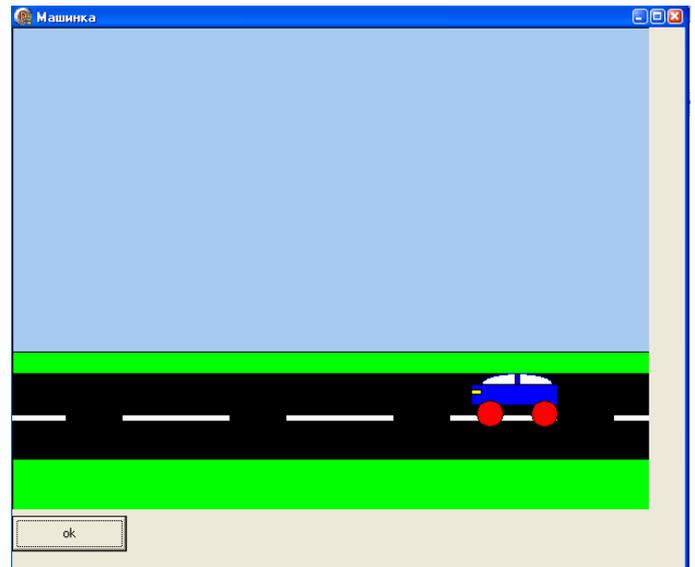
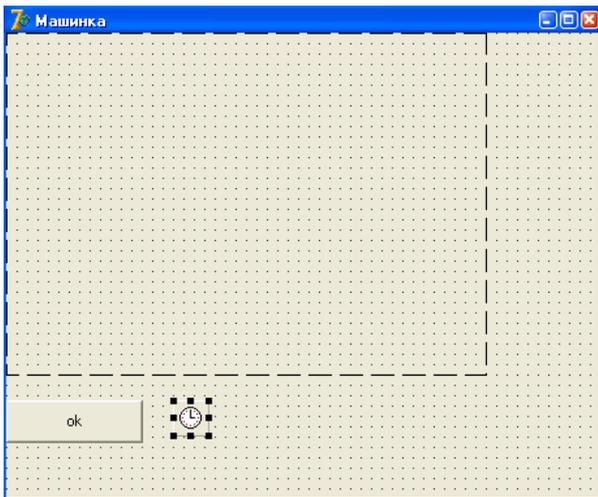
**Цель:** изучить возможности сбора элементов данных во время работы с проектом.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создать анимацию движения машины с прорисовыванием каждого элемента.



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TForm4 = class(TForm)
    Img1: TImage;
    Btnok: TButton;
    Timer1: TTimer;
    procedure BtnokClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
  {$R *.dfm}
  var a:real;
  r,xc,yc,x,y,up,xu,z,d:integer; //upiter
  procedure TForm4.BtnokClick(Sender: TObject);
  begin
    d:=0;
    x:=0;
    y:=0;
    z:=0;
    xc:=round(img1.Width/2);
    yc:=round(img1.Height/2);
  
```

```

    {if (x>150) and (y>150) then
begin
x:=x+10;
y:=y+10;
end ; }
timer1.Enabled:=true;
end;
procedure TForm4.Timer1Timer(Sender: TObject);
begin
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(0,0,img1.Width,img1.Height);
d:=d-5;
img1.Canvas.Brush.Color:=clskyblue;
img1.Canvas.Rectangle(0,x+400,x+800,x) ;
img1.Canvas.Brush.Color:=cllime;
img1.Canvas.Rectangle(0,x+500,x+3000,x+300) ;
img1.Canvas.Brush.Color:=clblack;
img1.Canvas.Rectangle(0,x+400,x+600,x+320) ;
//polosi dvigenia
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(50,x+365,x+-100,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(100,x+365,x+200,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(250,x+365,x+350,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(400,x+365,x+500,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(550,x+365,x+650,x+358) ;
//кабина
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.pen.Color:=clblue;
img1.Canvas.Ellipse(d+460,y+340,d+525,y+320);
//перегородка кабины
img1.Canvas.Brush.Color:=clblue;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Rectangle(d+490,y+340,d+495,y+320);
//корпус
img1.Canvas.Brush.Color:=clblue;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Rectangle(d+450,y+330,d+530,y+350);
//koleso1
img1.Canvas.Brush.Color:=clred;
img1.Canvas.pen.Color:=clblack;

```

```

img1.Canvas.Ellipse(d+455,y+345,d+480,y+370);
//koleso2
img1.Canvas.Brush.Color:=clred;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Ellipse(d+505,y+345,d+530,y+370);
//fara
img1.Canvas.Brush.Color:=clyellow;
img1.Canvas.Rectangle(d+450,y+335,d+460,y+340);
end;
end.

```

**Задание 2.** Создать индивидуальный проект «анимацию» по самостоятельно выбранной теме.

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 33

**Тема:** Разработка интерфейса приложения, разработка функциональной схемы работы приложения.

**Цель:** изучить возможности сбора элементов данных во время работы с проектом

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

Задание:

1 Создайте карту навигации для выбранной системы. На карте в зависимости от специфики системы выделите разделы, доступные различным пользователям в зависимости от роли, опишите условия перехода из различных разделов (при необходимости).

2 Используя графический редактор на выбор, создайте макеты графического интерфейса пользователя (от каждого члена бригады – не менее 3 макетов).

Предлагаемые системы:

-Microsoft Visio 2010

-Axure

-Adobe Photoshop

-Balsamiq

-Cacoo

3 Для разработанных макетов подготовьте их текстовое описание в следующем виде:

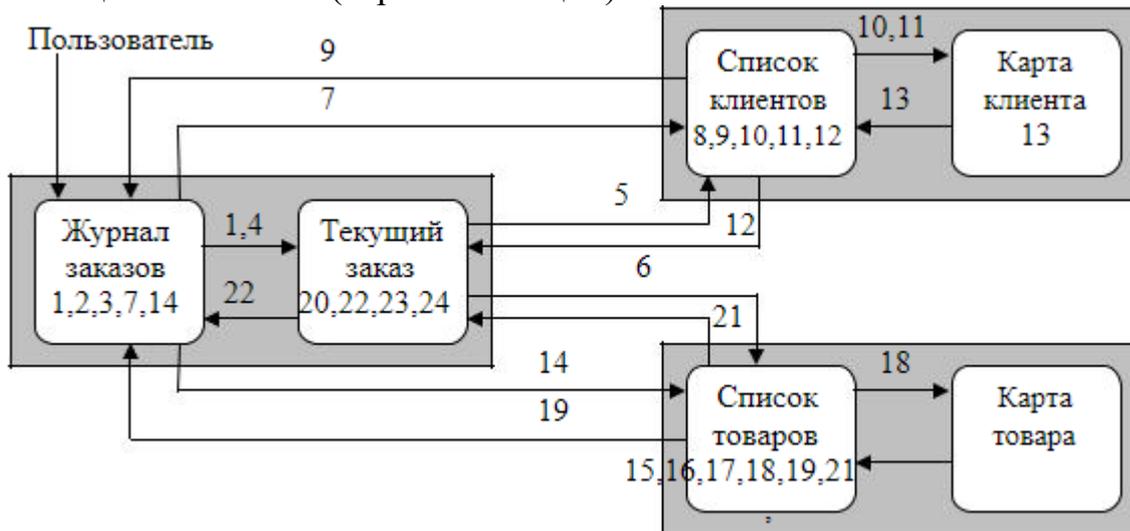
Название поля	Тип	Условия видимости	Условие доступности	Описание
				Формат, допустимые значения, макс. и мин. длина, поведение

Содержание отчета

1 Титульный лист

## 2 Цель работы

## 3 Навигационная схема (карта навигации)



## 4 Макеты графического интерфейса пользователя

Определим операции, которые должен выполнять пользователь в рамках возможностей, предоставляемых ему приложением (функций приложения):

- 1) создать новый заказ;
- 2) задать атрибуты поиска заказа;
- 3) найти заказ по текущим атрибутам поиска;
- 4) открыть текущий заказ на редактирование;
- 5) открыть список клиентов для добавления в текущий заказ;
- 6) открыть список товаров для добавления в текущий заказ;
- 7) просмотреть список клиентов;
- 8) выбрать клиента из списка клиентов;
- 9) добавить атрибуты текущего клиента к поиску заказа;
- 10) ввести данные нового клиента в текущий заказ;
- 11) редактировать данные текущего клиента в списке клиентов;
- 12) добавить текущего клиента в текущий заказ;
- 13) сохранить данные о текущем клиенте;
- 14) просмотреть список товаров;
- 15) задать атрибуты поиска товаров;
- 16) найти товар по текущим атрибутам;
- 17) выбрать товар из списка товаров;
- 18) просмотреть подробные данные текущего товара;
- 19) добавить атрибуты текущего товара к поиску заказа;
- 20) редактировать данные по текущему товару в текущем заказе;
- 21) добавить данные текущего товара в текущий заказ;
- 22) сохранить текущий заказ;
- 23) распечатать информацию по текущему заказу;
- 24) сформировать счет по текущему заказу

Соответствие приведенных операций функциональным блокам, экранным формам и навигационным переходам указано на рисунке.

Далее, необходимо сгруппировать операции таким образом, чтобы их группы соответствовали пунктам главного меню. В рассматриваемом примере предлагается сформировать следующие группы.

1. Действия над объектами. В качестве объектов выступают заказ, клиент, товар

Действия	Объект	Примечания
Создать	Заказ	1
	Клиент	10
Открыть	Заказ	4
	Клиент	11
	Товар	18
Сохранить	Заказ	22
	Клиент	13
Выбрать (отобрать для добавления)	Клиент	12 (в заказ)
	Товар	21 (в заказ)
	Атрибуты клиента	9 (к поиску)
	Атрибуты товара	19 (к поиску)
Печать	Заказ	23
Счет	Заказ	24

2. Поиск. Специфическое действие, выделено отдельно; объекты – заказ (3), товар (16).

3. Работа со списками. Объекты – клиент, заказ

Списки	Операции	Примечания
Клиенты	Просмотреть	7
	Открыть для выбора (добавления) в заказ	5

5. Описание элементов управления по таблице 1

Таблица 1

Пользователи	Менеджер по направлению товара	Представители обслуживающего персонала
Социальные характеристики	Мужчины, женщины	Женщины
	Взрослые	Взрослые
	Русскоязычные	Русскоязычные
	Средний уровень владения компьютером	Низкий уровень владения компьютером
Мотивационно целевая среда	Прямая производственная необходимость, удобство	Производственная необходимость, Престиж
	Мотивация к обучению высокая	Мотивация к обучению низкая
Навыки и умения	Должны иметь значительный тренинг работы с программой	Прошли предварительный тренинг работы с программой
Требования к ПО ИС	Возможность использования ПО ИС в локальной сети	Возможность использования программы одновременно с телефонным общением с клиентом
	Отсутствие жестких ограничений по времени	Время реакции ПО ИС, допустимое для ожидания клиента
	Обеспечение текущей	Обеспечение текущей

	информацией по содержанию заказов Обеспечение текущей информацией по товарам Возможность проводить обобщение информации по заказам	информацией по содержанию заказов Обеспечение текущей информацией по товарам Возможность формирования новых заказов
Задачи пользователя	Просмотр/фильтрация информации по заказам/клиентам/товарам Сортировка информации по заказам/клиентам/товарам	Просмотр данных по товарам Создание/поиск/модификация заказа
	Агрегирование информации по заказам/клиентам/товарам	Сохранение/печать заказа Формирование счета по заказу
Рабочая среда	Стандартизированные ПК, локальная сеть	Стандартизированные ПК, специализированное телефонное обслуживание

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 34

**Тема:** Разработка игрового приложения.

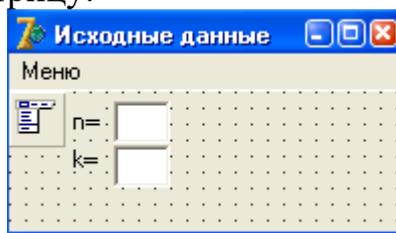
**Цель:** изучить возможности языка программирования для создания проектов многооконного приложения.

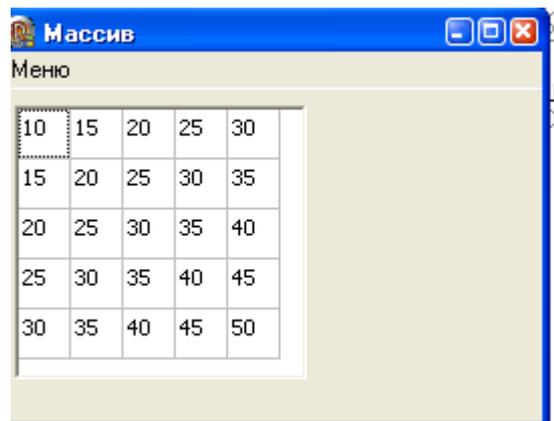
**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Дана квадратная матрица. Заменить все элементы, расположенные ниже побочной диагонали средним арифметическим элементов главной диагонали. Транспонируйте итоговую матрицу.





Unit1

```

procedure Tfrm1.exitClick(Sender: TObject);
var q:byte;
begin
q:=application.MessageBox('Точно
выйти?','Подтверждение',MB_YesNo+MB_iconquestion);
if q=6 then application.Terminate;
end;

```

```

procedure Tfrm1.FormClose(Sender: TObject; var Action: TCloseAction);
var q:byte;
begin
q:=application.MessageBox('Точно
выйти?','Подтверждение',MB_YesNo+MB_iconquestion);
if q=7 then action:=caNone;
end;

```

```

procedure Tfrm1.OKClick(Sender: TObject);
var k:integer;
i,j:byte;
begin
if (edtn.Text<>") and (edtk.Text<>") then
begin
n:=strtoint(edtn.text);
k:=strtoint(edtk.text);
setlength(a,n,n);
with frm1mass.gr1 do
begin
rowcount:=n;
colcount:=n;
for i:=1 to n do
for j:=1 to n do
begin
a[i-1,j-1]:=k*(i+j);
cells[j-1,i-1]:=inttostr(a[i-1,j-1]);

```

```

end;
end;
with frmMass do
begin
show;
left:=self.Left+self.Width+10;
top:=self.Top;
end;
end;
end;
end.

```

Unit 2.

```

object frmMass: TfrmMass
  Left = 0
  Top = 0
  Caption = #1052#1072#1089#1089#1080#1074
  ClientHeight = 166
  ClientWidth = 270
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  Menu = MainMenu1
  OldCreateOrder = False
  OnClose = FormClose
  PixelsPerInch = 96
  TextHeight = 13
  object Gr1: TStringGrid
    Left = 8
    Top = 8
    Width = 145
    Height = 137
    DefaultColWidth = 25
    FixedCols = 0
    FixedRows = 0
    TabOrder = 0
  end
  object MainMenu1: TMainMenu
    object mainm: TMenuItem
      Caption = #1052#1077#1085#1102
      object Sum: TMenuItem

```

```

Caption = #1057#1091#1084#1084#1072
OnClick = SumClick
end
object Clear: TMenuItem
Caption = 'OK'
OnClick = ClearClick
end
object Excit: TMenuItem
Caption = #1042#1099#1093#1086#1076
OnClick = ExcitClick
end
end
end
end
end

```

**Задание 2.** Разработать многооконное приложение из предложенных вариантов:

1. Игра-перевертыш.
2. Игра-пазл.
3. Автоматическое написание характеристик.
4. Алгоритмы сравнений двух изображений.
5. Автоматическое написание характеристик.
6. Построение графиков функций (первая задана с помощью ряда Тейлора, корень второго уравнения необходимо найти при помощи метода Ньютона);
7. Графическое интегрирование методом Симпсона;
8. Графическое интегрирование методом Гаусса;
9. Графическое интегрирование методом прямоугольников;
10. Графическое интегрирование методом трапеций;
11. Решение систем линейных уравнений методом Гаусса;
12. Решение систем линейных уравнений методом Крамера;
13. Решение систем линейных уравнений методом Зейделя;
14. Решение систем уравнений методом Ньютона;
15. Решение уравнений методом половинного деления, секущих, Ньютона, хорд;
16. Сортировка методом простых вставок;
17. Сортировка методом бинарных вставок;
18. Сортировка методом слияния;
19. Сортировка методом выбора;
20. Сортировка методом пузырька;
21. Моделирование кипящей жидкости;
22. Моделирование броуновского движения;
23. Моделирование кругов на воде;
24. Изображение электронных часов;
25. Моделирование движения футбольного мяча после удара;
26. Модель атома.

**Форма отчета:** отчет, защита работы.

**Практическое занятие № 35**

**Тема:** Разработка оконного приложения с несколькими формами.

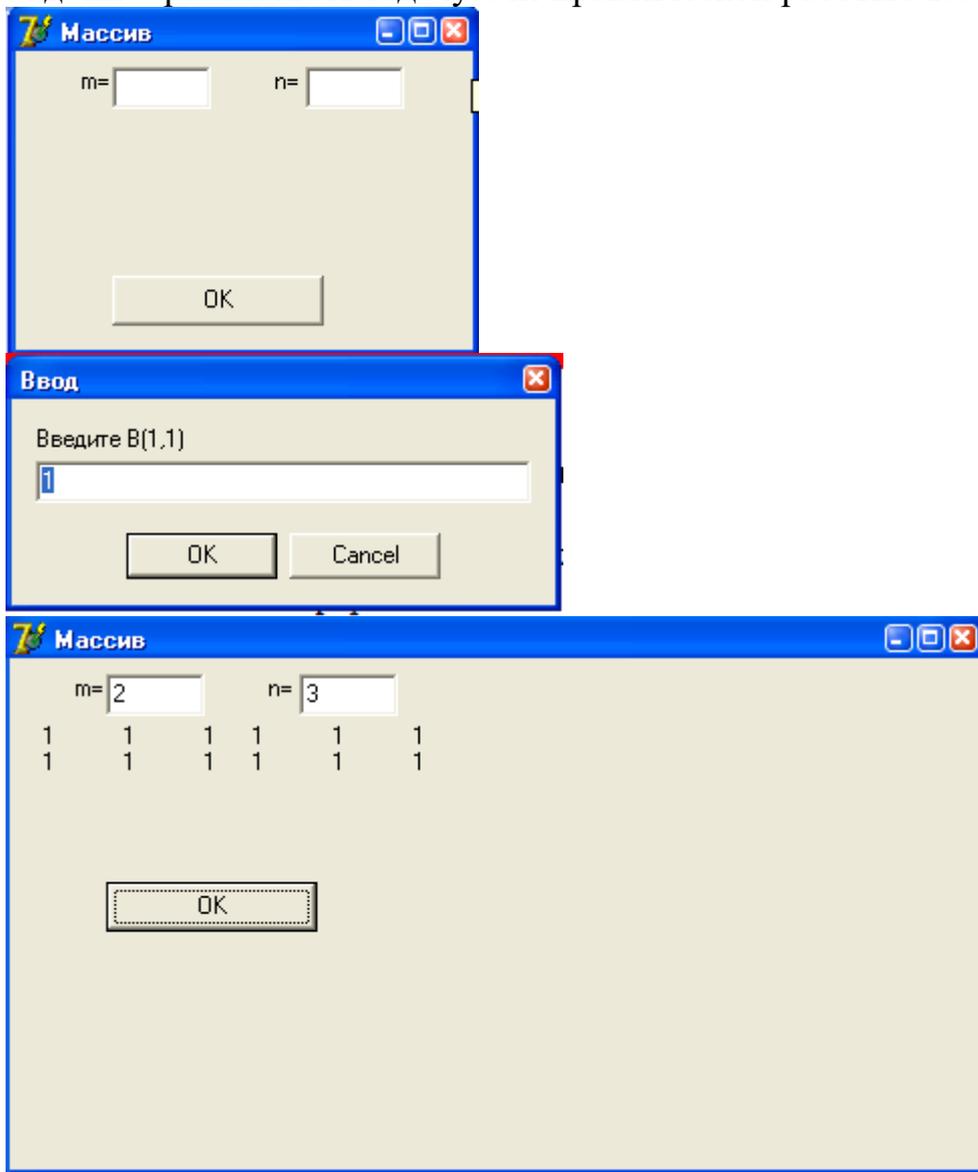
**Цель:** изучить возможности языка программирования для создания проектов многооконного приложения с несколькими формами.

**Оборудование:** тетрадь, ручка, ПК

**Методическое указание:** выполнение заданий

**Ход выполнения:**

**Задание:** реализовать задачу 4 из практической работы №8.



```
Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, CheckLst;
```

```
type
```

```
TForm1 = class(TForm)
```

```
  BtnOK: TButton;
```

```
  Btnf: TButton;
```

```
  Lblp: TLabel;
```

```

cl: TCheckListBox;
Lblm: TLabel;
Button1: TButton;
c: TCheckListBox;
procedure BtnOKClick(Sender: TObject);
procedure BtnfClick(Sender: TObject);

procedure clClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
procedure TForm1.BtnOKClick(Sender: TObject);
var i,k,m,p:byte;
begin
k:=cl.Items.Count;
if (k>=5) then
begin
p:=6*(ord(cl.Checked[0])+ord(cl.Checked[1]))+9*(ord(cl.Checked[2])+ord(cl.Checked[3]))
+12*(ord(cl.Checked[4]));
for i:=5 to (k-1) do
p:=p+2*(i-1)*ord(cl.Checked[i]);
case p of
0..14:m:=2;
15..24:m:=3;
25..34:m:=4
else m:=5;
end;
lblp.Caption:=Format('Сумма баллов %d',[p]);
lblm.Caption:=Format('Оценка %d',[m]);
end
else showMessage ('Недостаточно данных!');
end;
procedure TForm1.BtnfClick(Sender: TObject);
begin
cl.Items.LoadFromFile('1.txt');
end;

```

```

procedure TForm1.clClick(Sender: TObject);
var a,k,n,i,l:integer;
begin
k:=cl.Items.Count;
n:=cl.ItemIndex;
//showmessage(inttostr(n));
case n of
0: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
1: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
2: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
3: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
4: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
5: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
6: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
7: if cl.Checked[n]=true then c.Items.Add(cl.Items.Strings[n]);
end;
//l:=c.Items.IndexOf(cl.Items.Strings[n]);
//showmessage(inttostr(l));
//c.Header[1];
case n of
0: if cl.Checked[n]=false then
//begin
//c.Header[c.Items.IndexOf(cl.Items.Strings[n])]:=true;
c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
//end;
1: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
2: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
3: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
4: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
5: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
6: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
7: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
end;
//l:=c.Items.Count;
//for i := 0 to l - 1 do
c.Header[c.Items.IndexOf(cl.Items.Strings[n])]:=true;
//for i:=0 to k-1 do
//begin
//if cl.Checked[i]=true then a:=cl.Items.IndexOf(cl.Items.Strings[i]);
//end;

end;
procedure TForm1.Button1Click(Sender: TObject);
begin

```

```
c.Items.SaveToFile('3.txt');
end;
end.
```

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 36

**Тема:** Тестирование и отладка приложений.

**Цель:** изучить этапы создания документации для реализации программы на языке программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Изучить этапы документации проекта на основании примера.

**Задача.** По заданным значениям  $x, y, z$  вычислить значение  $u$ :  $u = \max(x+y+z, xyz)$

1. Постановка задачи: корректна

2. Математическая модель:

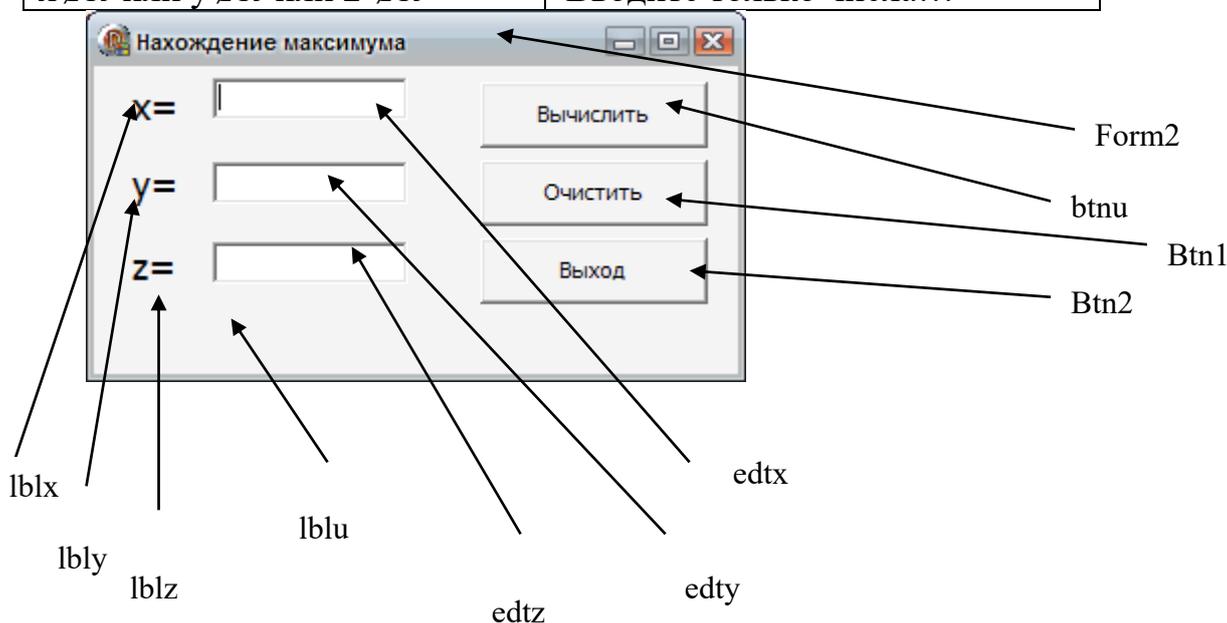
И.Д.:  $x, y, z \in R$

В.Д.:  $u \in R$

Связь:  $u = \max(x+y+z, xyz)$

3. Техническое задание:

Условия на И. Д.	Действие алгоритма
$x, y, z \in R$	$u$
$x \notin R$ или $y \notin R$ или $z \notin R$	'Вводите только числа!!!'



Объект	Свойство	Значение	Событие	Программный код
Form	Name Caption	Form2 'Нахождение максимума'		
label	Name Caption	Lblx 'x='		
label	Name	Lbly		

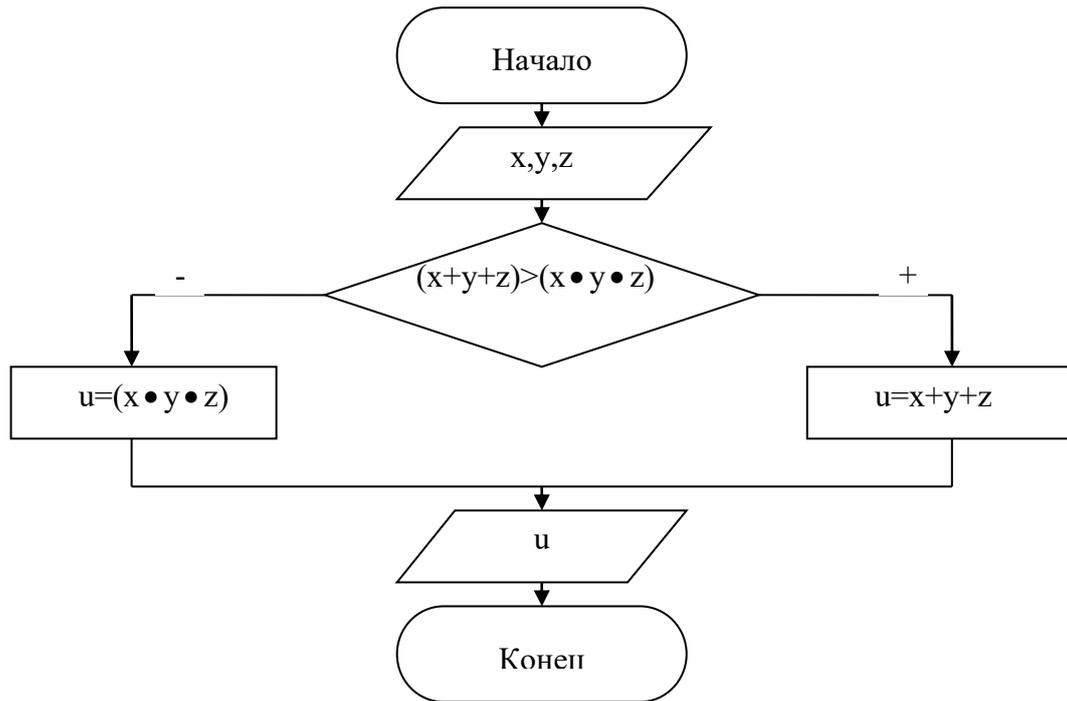
	Caption	'y='		
label	Name Caption	Lblz 'z='		
label	Name Caption	Lblu “		
edit	Name Text	Edtx “		
edit	Name Text	Edty “		
edit	Name Text	Edtz “		
Button	Name Caption	btnu 'ВЫЧИСЛИТЬ'	onClick	1
Button	Name Caption	Btn1 'ОЧИСТИТЬ'	onClick	2
Button	Name Caption	Btn2 'ВЫХОД'	onClick	3

#### 4. Выбор ПО: Turbo Delphi

#### 5. Тестирование:

№ теста	Исходные данные			Выходные данные	
	x	y	z	u	Примечание
1	2	3	5	30	
2	-9	5	1.25	-2.75	
3	-0.32	5	-8	12.8	
4	4	0	65.9	69.9	
5	*	4	58		'Вводите только числа!!!'
6	52	*	*		'Вводите только числа!!!'
7	0	0	0		'значения равны'

## 6. Алгоритмизация



## 7. Программный код 1:

```
procedure TForm2.btnuClick(Sender: TObject);
Var x,y,z,max: real;
begin
try
x:=StrToFloat(edtx.Text);
y:=StrToFloat(edty.Text);
z:=StrToFloat(edtz.Text);
If (x+y+z)>(x*y*z) Then
begin
ShowMessage('Сумма больше произведения');
max:=x+y+z;
end
else
begin
if (x+y+z)=(x*y*z) Then ShowMessage('значения равны')
else
begin
ShowMessage('Произведение больше суммы');
max:=x*y*z;
end;
end;
lblu.Caption:='Максимум равен '+FloatToStr(max);
except
showmessage ('Вводите только числа!!!')
```

end;

end;

Программный код 2:

```
procedure TForm2.btn1Click(Sender: TObject);
```

```
begin
```

```
edtx.Text:='';
```

```
edty.Text:='';
```

```
edtz.Text:='';
```

```
lblu.Caption:='';
```

```
end;
```

Программный код 3:

```
procedure TForm2.btn2Click(Sender: TObject);
```

```
begin
```

```
Application.Terminate;
```

```
end;
```

**Задание 2.** Решить задачу своего варианта и описать создание документации на примере из задания 1.

Вариант	Задача
1	Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.
2	Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число
3	Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число
4	Дано целое число. Если оно является положительным, то прибавить к нему 10; если отрицательным, то разделить его на 2; если нулевым, то заменить его на 10. Вывести полученное число
5	Дано целое число. Если оно является положительным, то увеличить его в 2 раза; если отрицательным, то уменьшить его в 2 раза; если

	нулевым, то заменить его на 100. Вывести полученное число
6	Дано целое число. Если оно является положительным, то прибавить к нему 101; если отрицательным, то вычесть из него 10; если нулевым, то заменить его на 5. Вывести полученное число
7	Дано целое число. Если оно является положительным, то уменьшить его в два раза; если отрицательным, то увеличить его в 2 раза; если нулевым, то заменить его на 555. Вывести полученное число
8	Дано целое число. Если оно является положительным, то прибавить к нему 50; если отрицательным, то вычесть из него 50; если нулевым, то заменить его на 50. Вывести полученное число
9	Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 1; если нулевым, то заменить его на 100. Вывести полученное число
10	Дано целое число. Если оно является положительным, то увеличить его в 2 раза; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 2. Вывести полученное число

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 37

**Тема:** Перегрузка методов.

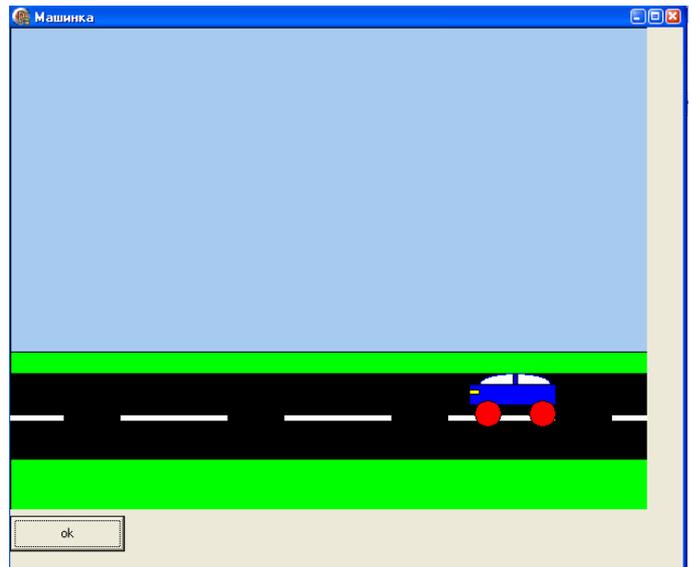
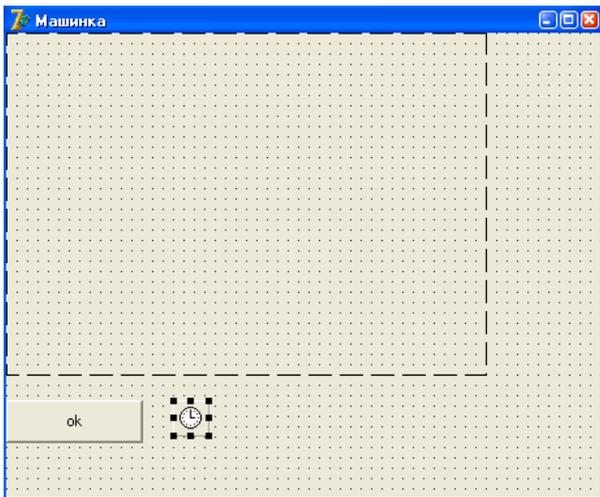
**Цель:** изучить этапы создания документации для реализации программы на языке программирования.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создать анимацию движения машины с прорисовыванием каждого элемента.



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TForm4 = class(TForm)
    Img1: TImage;
    Btnok: TButton;
    Timer1: TTimer;
    procedure BtnokClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
  {$R *.dfm}
  var a:real;
  r,xc,yc,x,y,up,xu,z,d:integer; //upiter
  procedure TForm4.BtnokClick(Sender: TObject);
  begin
    d:=0;
    x:=0;
    y:=0;
    z:=0;
    xc:=round(img1.Width/2);
    yc:=round(img1.Height/2);
  
```

```

    {if (x>150) and (y>150) then
begin
x:=x+10;
y:=y+10;
end ; }
timer1.Enabled:=true;
end;
procedure TForm4.Timer1Timer(Sender: TObject);
begin
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(0,0,img1.Width,img1.Height);
d:=d-5;
img1.Canvas.Brush.Color:=clskyblue;
img1.Canvas.Rectangle(0,x+400,x+800,x) ;
img1.Canvas.Brush.Color:=cllime;
img1.Canvas.Rectangle(0,x+500,x+3000,x+300) ;
img1.Canvas.Brush.Color:=clblack;
img1.Canvas.Rectangle(0,x+400,x+600,x+320) ;
//polosi dvigenia
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(50,x+365,x+-100,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(100,x+365,x+200,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(250,x+365,x+350,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(400,x+365,x+500,x+358) ;
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.Rectangle(550,x+365,x+650,x+358) ;
//кабина
img1.Canvas.Brush.Color:=clwhite;
img1.Canvas.pen.Color:=clblue;
img1.Canvas.Ellipse(d+460,y+340,d+525,y+320);
//перегородка кабины
img1.Canvas.Brush.Color:=clblue;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Rectangle(d+490,y+340,d+495,y+320);
//корпус
img1.Canvas.Brush.Color:=clblue;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Rectangle(d+450,y+330,d+530,y+350);
//koleso1
img1.Canvas.Brush.Color:=clred;
img1.Canvas.pen.Color:=clblack;

```

```

img1.Canvas.Ellipse(d+455,y+345,d+480,y+370);
//koleso2
img1.Canvas.Brush.Color:=clred;
img1.Canvas.pen.Color:=clblack;
img1.Canvas.Ellipse(d+505,y+345,d+530,y+370);
//fara
img1.Canvas.Brush.Color:=clyellow;
img1.Canvas.Rectangle(d+450,y+335,d+460,y+340);
end;
end.

```

**Задание 2.** Создать индивидуальный проект «анимацию» по самостоятельно выбранной теме.

**Форма отчета:** отчет, защита работы.

### Практическое занятие № 38

**Тема:** Перегрузка методов.

**Цель:** изучить перезагрузку методов.

**Оборудование:** тетрадь, ручка, ПК

**Методические указания:** выполнение заданий

**Ход выполнения:**

**Задание 1.** Создание классов. создать класс TVector и написать для него методы (процедуры и функции), реализующие стандартные операции для векторов. В Delphi есть зарезервированное слово class, которое позволяет описывать класс. При создании в Delphi нового проекта в модуле unit1 появляется объявление класса основной формы. Ниже приведен пример стандартного объявления класса:

```

type
TForm1 = class(TForm)
private
{ Private declarations }
public
{ Public declarations }
end;

```

По негласному соглашению программисты Delphi начинают имя типа класса с большой буквы T, а имя поля (что будет использовано далее) – с большой буквы F. После служебного слова class в круглых скобках указывается базовый класс, от которого порождается данный.

Раздел public (открытый) предназначен для объявлений, которые доступны внешнему миру, а в разделе private можно объявлять переменные, процедуры и функции, используемые только внутри данного класса. Подобным образом можно описать класс TVector, необходимый для выполнения первой лабораторной работы. Для примера приведем следующий фрагмент объявления.

```

type
TVector = class
private
FValues: array of double;

```

```

//Далее идут 4 accessing метода, которые, в соответствии
// с принципом инкапсуляции, должны быть объявлены в разделе private
function GetLength: integer;
procedure SetLength(const Value: integer);
function GetValues(Index: integer):double;
procedure SetValues(Index: integer; const Value: double);
public
//Receive the vector as string
function AsString: string;
//Length of the vector
property Length: integer read GetLength write SetLength;
//Receive the product of the vector to scalar
procedure Product( Factor: double); overload;
//Receive the product of two vectors
procedure Product( const SecondVector: TVector); overload;
end;

```

Разумное использование комментариев, а также мнемонических имен делает текст достаточно понятным. В приведенном объявлении указаны некоторые составляющие объявления класса: поля и методы. Поле FValues объявлено как динамический массив вещественных чисел, т.е. его длину можно произвольно менять по ходу исполнения программы.

В разделе private объявлены методы доступа, начинающиеся со слов Get и Set. Способ написания имен тоже принят программистским сообществом. В разделе public объявлены имена некоторых необходимых методов в виде процедур и функций, а также свойство (служебное слово property) класса Length (число элементов в массиве) и действия, связанные с чтением и записью значения этого атрибута. Можно было бы назвать это свойство другим именем (например, size), но мы намеренно взяли имя Length, для того чтобы показать, как выходить из положения, когда используемое имя совпадает с системным (см. далее). Обращаем также внимание на то, что в разделе public объявлены два метода с одинаковым именем Product, и каждое из них заканчивается служебным словом overload (перегрузка). Эта новая возможность позаимствована, начиная с Delphi 4, из C++. Она разрешает иметь несколько функций и процедур с одинаковым именем. Однако у них должны быть разные наборы параметров или их типы. Тип возвращаемого результата не может служить различием. В нашем случае перегружается метод Product. В одном случае вычисляется произведение вектора на некоторое значение, а в другом – поэлементное произведение двух векторов. Приведенное объявление помещается в разделе interface модуля Delphi, а текст объявленных методов (процедур и функций) – в разделе implementation. Для рассматриваемого примера код соответствующего модуля будет выглядеть так:

```

unit Vectors;
interface
uses
SysUtils, Classes;

```

```

Type
//Vector
TVector = class
private
FValues: array of double;
function GetLength: integer;
procedure SetLength(const Value: integer);
function GetValues(Index: integer):double;
procedure SetValues(Index: integer; const Value: double);
procedure CheckIndex(Index: integer);
public
constructor Create;
//Get vector values as string
function AsString: string;
//Item values of the vector
property Values[Index: integer]: double
read GetValues Write SetValues; default;
//Length of the vector
property Length: integer read GetLength write SetLength;
//Receive the product of the vector to scalar
procedure Product( Factor: double); overload;
//Receive the product of two vectors
procedure Product( const SecondVector: TVector); overload;
end;
implementation
{TVector}
constructor TVector.Create;
begin
System.SetLength(Fvalues, 1);
end;
function TVector.AsString: string;
var i: integer;
begin
Result := '(';
for i:=0 to Length-1 do
begin
if i>0 then Result:= Result+' , ';
Result:=Result+Format('%0.3n',[Values[i]]);
end;
Result:=Result+')';
end;
function TVector.GetLength: integer;
begin
Result:=System.Length(FValues);

```

```

end;
function TVector.GetValues(Index: integer): double;
begin
CheckIndex(Index);
Result:=FValues[Index];
end;
procedure TVector.SetLength(const Value: integer);
begin
if Value<1 then
raise Exception.Create('Invalid vector length');
System.SetLength(FValues, Value);
end;
procedure TVector.SetValues(Index: integer; const Value: double);
begin
CheckIndex(Index);
FValues[Index]:=Value;
end;
procedure TVector.CheckIndex(Index: integer);
begin
if (Index>=0) and (Index>=Length) then
raise Exception.Create('Index out of the bounds');
end.
Procedure TVector.Product(const SecondVector: TVector);
var i: integer;
Begin
For i:=0 to SecondVector.Length-1 do
Fvalues[I]:= Self[i] * SecondVector[i];
end;
end.

```

## **Задание 2.:**

1. Изучить назначение метода construtor.
2. Написать методы:
  - для сложения векторов (TVector.Add(const SecondVector: TVector)),
  - умножения вектора на значение (TVector.Product(Factor: double)),
  - получения скалярного произведения векторов.
3. Создать форму для отображения результатов работы проекта.

Ниже приведен возможный вариант реализации п.3

```

unit UnitMain;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)

```

```

ButtonInput: TButton;
Memo1: TMemo;
ButtonExit: TButton;
procedure ButtonInputClick(Sender: TObject);
procedure ButtonExitClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
uses Vectors;
{$R *.dfm}
procedure TForm1.ButtonInputClick(Sender: TObject);
var V, SecondV: TVector;
i, N: integer; R: double;
begin
//Создание экземпляров класса
V:=TVector.Create; SecondV:=TVector.Create;
try
V.Length:=4; SecondV.Length:= 4;
for i:=0 to V.Length-1 do begin
Randomize;
V[i]:=Random(100);
end;
for i:=0 to SecondV.Length-1 do
SecondV[i]:= 10;
Memo1.Lines.add(V.AsString);
V.Product(10);
Memo1.Lines.add(V.AsString);
V.Add(SecondV);
Memo1.Lines.Add(V.AsString);
R:=V.ScalarProduct(SecondV);
Memo1.Lines.Add(FloatToStr(R));
finally
V.Free; SecondV.Free;
end;
end;
procedure TForm1.ButtonExitClick(Sender: TObject);
begin
Close;
end;

```

end.

**Форма отчета:** отчет, защита работы.

## **4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ**

### **4.1 Основные печатные и (или) электронные издания:**

О-1. Семакин, И.Г. Основы алгоритмизации и программирования: учебник / И.Г. Семакин, А.П. Шестаков. – 6-е изд., стер. – М.: Образовательно-издательский центр «Академия», 2024. – 304 с. – URL: <https://academia-moscow.ru/catalogue/5546/768351/>. – Режим доступа: Электронная библиотека «Academia-moscow». – Текст: электронный.

О-2. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — 4-е изд. — Москва: Издательство Юрайт, 2024. — 119 с. — (Профессиональное образование). — ISBN 978-5-534-17498-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539994> (дата обращения: 02.05.2025).

### **4.2 Дополнительные печатные и (или) электронные издания (электронные ресурсы):**

Д-1. Голицына, О.Л. Попов И.И. Основы алгоритмизации и программирования: учебник. – М.: ИД "ФОРУМ"-ИНФРА-М, 2006. – 432 с.

Д-2. Голицына, О.Л. Попов И.И. Основы алгоритмизации и программирования: учебник. – М.: ИД "ФОРУМ"-ИНФРА-М, 2004. – 432 с.

Д-3. Колдаев, В.Д. Основы алгоритмизации и программирования: учебник. – М.: ИД "ФОРУМ"-ИНФРА-М, 2009. – 416 с.

Д-4. Канцедал, С.А. Алгоритмизации и программирования: учебник. – М.: ИД "ФОРУМ"-ИНФРА-М, 2008. – 352 с.

Д-5. Голицына, О.Л., Партыка Т.Л., Попов И.И. Программное обеспечение: Учебное пособие – М.: ИД "ФОРУМ"-ИНФРА-М, 2006. – 432 с.

Д-6. Голицына, О.Л., Партыка Т.Л., Попов И.И. Языки программирования: Учебное пособие – М.: ИД "ФОРУМ", 2008. – 400 с.

Д-7. Голицына, О.Л., Попов И.И., Попов И.И. Программирование на языках высокого уровня: Учебное пособие – М.: ИД "ФОРУМ", 2008. – 496 с.

Д-8. Голицына О.Л., Партыка Т.Л., Попов И.И. Программное обеспечение: Учебное пособие – М.: ИД "ФОРУМ"-ИНФРА-М, 2008. – 432 с.

Д-9. Семакин, И.Г. Основы программирования: Учебное пособие. – М.: Академия, 2003. – 432 с.

Д-10. Мишенин А.И. Сборник задач по программированию: учебное пособие – М.: Инфра-М, 2009. – 224 с.

Д-11. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие – М: ИД "ФОРУМ"-ИНФРА-М, 2009. – 400 с.

Д-12. Шамис, В.А. С++ Builder 4. Техника визуального программирования – М: Нолис, 2000. – 656 с.

- Д–13. Вальпа О.Д. С++Builder. Экспресс-курс. - М: БХВ -Петербург, 2006. – 224 с.
- Д–14. Культи, Н. С++Builder в задачах и примерах– М: БХВ -Петербург, 2007. – 336 с.
- Д–15. Пахомов, Б. С/С++ и Borland С++Builder . Для начинающих– М: БХВ -Петербург, 2007. – 640 с.
- Д–16. Архангельский, А.Я. Приемы программирования С++Builder 6 и 2006 – М: Бинوم-Пресс, 2006. – 992 с.
- Д–17. Лаптев, В.В. С++.Объектно-ориентированное программирование: учебное пособие – М: Питер, 2008. – 464 с.
- Д–18. Павловская, Т.А., Щупак Ю.А. С++.Объектно-ориентированное программирование. Практикум: Практикум – М: Питер, 2008. – 265 с.
- Д–19. Карпов, Б., Баранова Т. С++.: Справочник – М: Питер, 2005. – 381 с.
- Д–20. Пол Айра Объектно-ориентированное программирование. С++ – М: Бинум, 1999. – 462 с.
- Д–21. Иванова, Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: учебник – М: МГТУ им Баумана, 2003. – 368 с.
- Д–22. Павловская, Т.А., Щупак Ю.А. ПАСКАЛЬ. Программирование на языке высокого уровня: учебник – М: Питер, 2004. – 368 с.
- Д-23. Начало программирования [Электронный ресурс]. – Режим доступа: [www.url: https://pas1.ru/](https://pas1.ru/). – 02.05.2025.

## ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В МЕТОДИЧЕСКИЕ УКАЗАНИЯ

<b>№ изменения, дата внесения, № страницы с изменением</b>	
<b>Было</b>	<b>Стало</b>
<b>Основание:</b>	
<b>Подпись лица, внесшего изменения</b>	