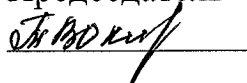



**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ
«ЧЕРЕМХОВСКИЙ ГОРНОТЕХНИЧЕСКИЙ КОЛЛЕДЖ
ИМ. М.И. ЩАДОВА»**

Рассмотрено на
заседании ЦК
«01» 06 2020 г.
Протокол № 10
Председатель
 Т.В.Окладникова

УТВЕРЖДАЮ
Зам. директора по УР
 Н.А. Шаманова
«23» 06 2020 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для выполнения
практических работ студентов
по учебной дисциплине

ОП. 06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
программы подготовки специалистов среднего звена
09.02.04 информационные системы (по отраслям)

Разработал преподаватель: Коровина Н.С.

2020 г.

СОДЕРЖАНИЕ

	СТР.
1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ РАБОТ	5
3. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ РАБОТ	7
4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРАКТИЧЕСКИХ РАБОТ	66
5. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЁННЫХ В МЕТОДИЧЕСКИЕ УКАЗАНИЯ	68

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по выполнению практических (лабораторных) работ по учебной дисциплине **«Основы алгоритмизации и программирования»** предназначены для студентов специальности **09.02.04 информационные системы (по отраслям)**, составлены в соответствии с рабочей программой дисциплины **«Основы алгоритмизации и программирования»** и направлены на достижение следующих целей:

- формирование у обучающихся умений осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;
- формирование у обучающихся умений применять, анализировать, преобразовывать информационные модели реальных объектов и процессов, используя при этом ИКТ, в том числе при изучении других дисциплин;
- развитие у обучающихся познавательных интересов, интеллектуальных и творческих способностей путем освоения и использования методов информатики и средств ИКТ при изучении различных учебных предметов;
- приобретение обучающимися опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности;
- приобретение обучающимися знаний этических аспектов информационной деятельности и информационных коммуникаций в глобальных сетях; осознание ответственности людей, вовлечённых в создание и использование информационных систем, распространение и использование информации;
- владение информационной культурой, способностью анализировать и оценивать информацию с использованием информационно-коммуникационных технологий, средств образовательных и социальных коммуникаций

Методические указания являются частью учебно-методического комплекса по дисциплине **основы алгоритмизации и программирования** и содержат задания, указания для выполнения практических (лабораторных) работ, теоретический минимум и т.п. Перед выполнением практической работы каждый студент обязан показать свою готовность к выполнению работы:

- пройти инструктаж по технике безопасности;
- ответить на теоретические вопросы преподавателя.

По окончании работы студент оформляет отчет в тетради и защищает свою работу.

В результате выполнения полного объема практических работ студент должен **уметь:**

- составлять алгоритмы с использованием основных алгоритмических структур;
- строить логически правильные и эффективные программы;
- использовать языки программирования;
- строить логически правильные и эффективные программы.

При проведении практических работ применяются следующие технологии и методы обучения:

1. проблемно-поисковых технологий
2. тестовые технологии
3. метод проектов

Правила выполнения практических работ:

1. Внимательно прослушайте инструктаж по технике безопасности, правила поведения в кабинете информатики.
2. Запомните порядок проведения практических работ, правила их оформления.
3. Изучите теоретические аспекты практической работы
4. Выполните задания практической работы.
5. Оформите отчет в тетради.

Требования к рабочему месту:

1. Количество ученических ПЭВМ, необходимых для оснащения кабинета ИВТ должно быть из расчета одной машины на одного обучающегося с учетом деления класса на две группы.
2. В состав кабинета ИВТ должна быть включена одна машина для учителя с соответствующим периферийным оборудованием.
3. Кабинет ИВТ должен быть оснащен диапроектором и экраном.

Критерии оценки:

Оценки «5» (отлично) заслуживает студент, обнаруживший при выполнении заданий всестороннее, систематическое и глубокое знание учебно - программного материала, учения свободно выполнять профессиональные задачи с всесторонним творческим подходом, обнаруживший познания с использованием основной и дополнительной литературы, рекомендованной программой, усвоивший взаимосвязь изучаемых и изученных дисциплин в их значении для приобретаемой специальности, проявивший творческие способности в понимании, изложении и использовании учебно-программного материала, проявивший высокий профессионализм, индивидуальность в решении поставленной перед собой задачи, проявивший неординарность при выполнении практических заданий.

Оценки «4» (хорошо) заслуживает студент, обнаруживший при выполнении заданий полное знание учебно- программного материала, успешно выполняющий профессиональную задачу или проблемную ситуацию, усвоивший основную литературу, рекомендованную в программе, показавший систематический характер знаний, умений и навыков при выполнении теоретических и практических заданий по дисциплине «Информатика».

Оценки «3» (удовлетворительно) заслуживает студент, обнаруживший при выполнении практических и теоретических заданий знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебной и

профессиональной деятельности, справляющийся с выполнением заданий, предусмотренных программой, допустивший погрешности в ответе при защите и выполнении теоретических и практических заданий, но обладающий необходимыми знаниями для их устранения под руководством преподавателя, проявивший какую-то долю творчества и индивидуальность в решении поставленных задач.

Оценки «2» (неудовлетворительно) заслуживает студент, обнаруживший при выполнении практических и теоретических заданий проблемы в знаниях основного учебного материала, допустивший основные принципиальные ошибки в выполнении задания или ситуативной задачи, которую он желал бы решить или предложить варианты решения, который не проявил творческого подхода, индивидуальности.

В соответствии с учебным планом программы подготовки специалистов среднего звена по специальности **09.02.04 Информационные системы (по отраслям)** и рабочей программой на практические работы по дисциплине **«основы алгоритмизации и программирования»** отводится 76 часа.

2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ РАБОТ

№ п/п	Название практической работы (указать раздел программы, если это необходимо)	Количество часов
1.	Составление блок-схем алгоритмов.	4
2.	Составление программ линейной структуры.	2
3.	Составление программ разветвляющейся структуры	4
4.	Составление программ циклической структуры.	4
5.	Обработка одномерных и двумерных массивов.	4
6.	Обработка динамических массивов.	2
7.	Работа со строковыми переменными. Работа с данными типа множество	4
8.	Организация и использование процедур.	2
9.	Организация и использование функций.	2
10.	Работа с файлом последовательного доступа	2
11.	Работа с файлом произвольного доступа.	2
12.	Программирование модуля.	2
13.	Создание библиотеки подпрограмм. Использование библиотеки подпрограмм.	2
14.	Создание простого проекта.	2
15.	Проектирование и формирование формы	2
16.	Создание документации. Тестирование, отладка приложения	2
17.	Создание экземпляров класса. Объявление класса.	2
18.	Создание наследованного класса. Перегрузка методов.	2
19.	Создание проекта с использованием компонентов для работы с текстом.	2
20.	Создание проекта с использованием компонентов	4

	стандартных диалогов.	
21.	Создание проекта с использованием компонентов системы меню.	4
22.	Обработка исключительных ситуации	4
23.	Графические возможности системы программирования.	2
24.	Создание анимации.	2
25.	Графические примитивы контурной графики.	2
26.	Средства прорисовки контурного изображения.	4
27.	Разработка оконного приложения. Разработка оконного приложения с несколькими формами.	2
28.	Разработка многооконного приложения.	4
	Итого	76

3. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ РАБОТ

Практическая работа № 1

Составление блок-схем алгоритмов.

Цель: Изучение способов задания алгоритмов, приобретение практических навыков составления блок - схем решения задач на ЭВМ.

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.

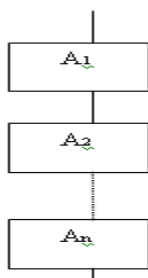
$$h = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a}$$

Задача 4. Дано два числа. Найти наименьшее.

Задача 5. 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 решения для задач 1-5, используя основные алгоритмические конструкции: следование, развилка и цикл.

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

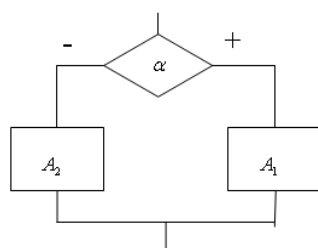


A_i - функциональный блок

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

Функциональный блок – это любая базовая структура или их комбинация

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

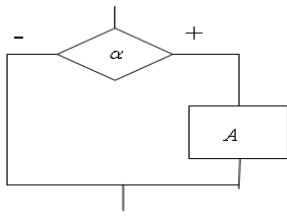


A_i - функциональный блок

$$i = \overline{1..2}$$

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

Полная разветвляющаяся (логическая) структура



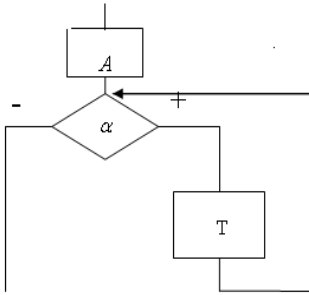
A - функциональный блок

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

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

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

а) цикл с предусловием («пока»)

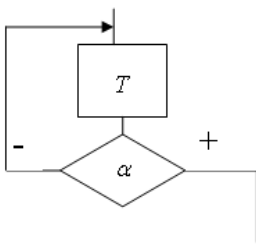


A - функциональный блок, начальных присвоений.

T - функциональный блок, тело цикла.

α - логическое условие

б) цикл с постусловием («до»)



A - функциональный блок, начальных присвоений.

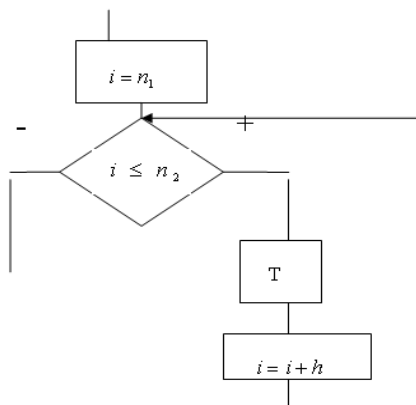
T - функциональный блок, тело цикла.

α - 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



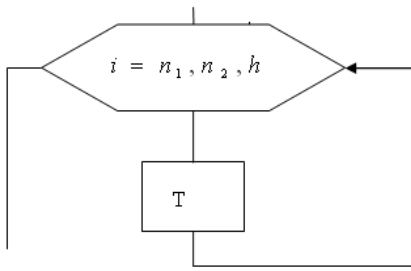
i-параметр цикла.

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



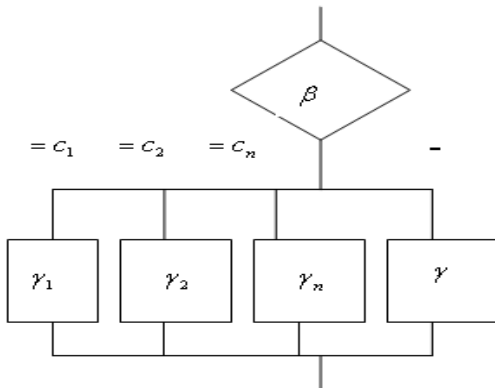
i -параметр цикла.

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.

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



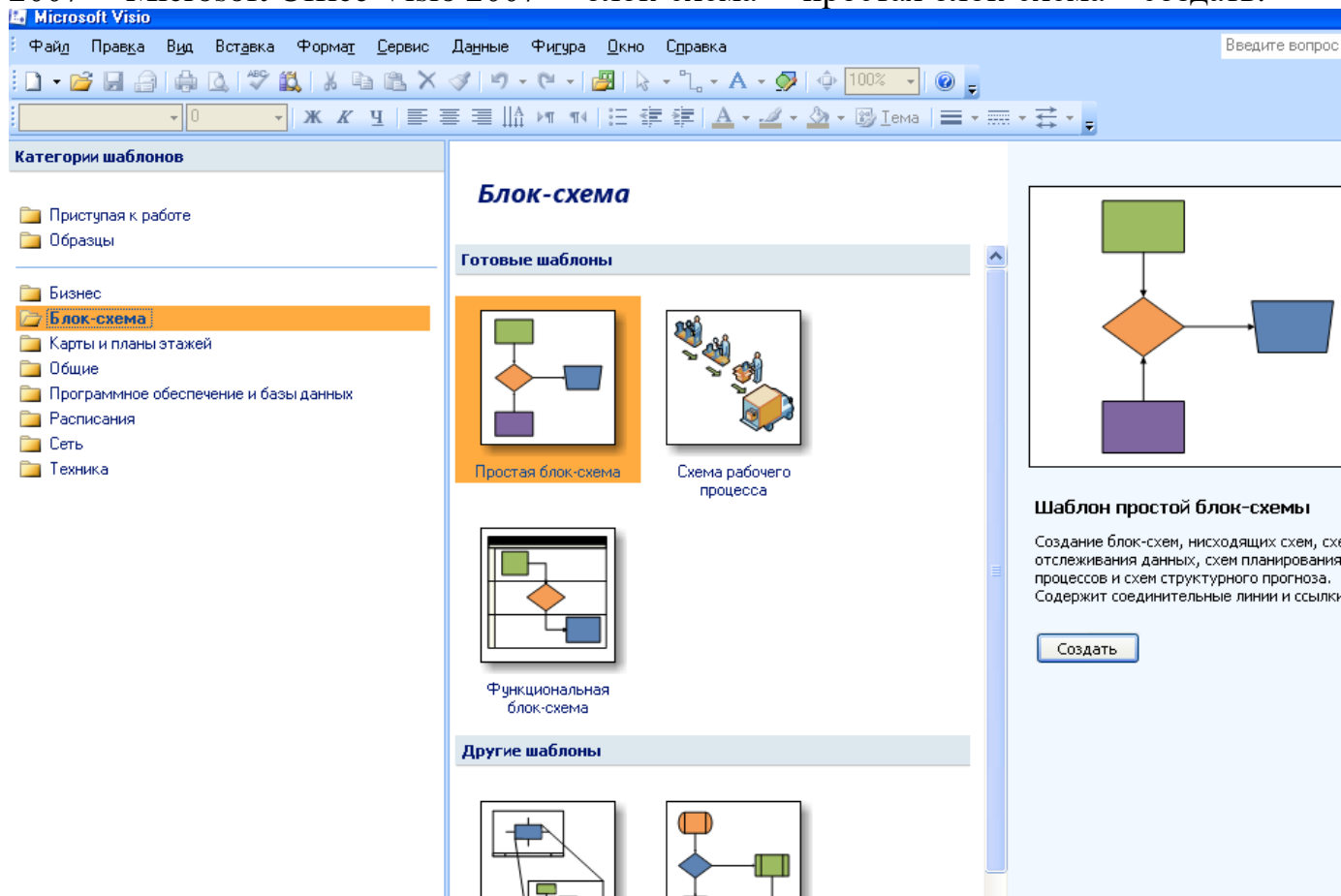
β - выражение значение, которое анализируется.

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

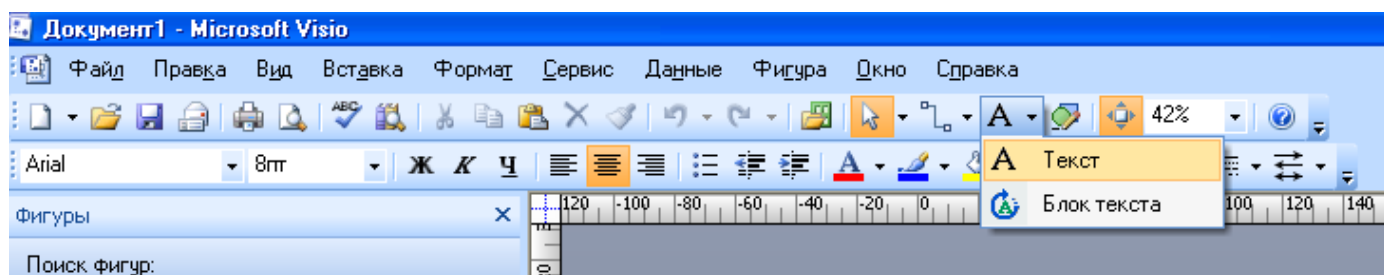
$\gamma_i, i = \overline{1, n}$ - функциональные блоки, которые выполняются в зависимости от значения β .

элемент	наименование	содержание
	терминатор	Начало или конец алгоритма
	данные	Общее 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. Описать исходные, выходные и промежуточные данные задач 1-5 в программе MS Visio 2007 с помощью кнопки «Текст» на панели быстрого доступа.



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

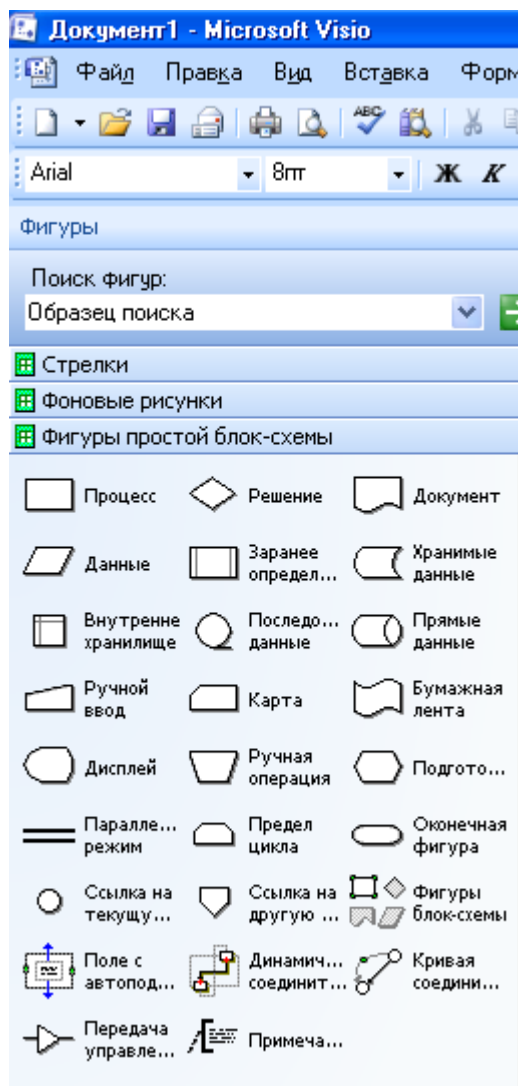
Задача 2. Дана сторона квадрата a . Найти его площадь $S = a^2$.

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. Составить блок-схему решения для задач 1-5, используя основные алгоритмические конструкции в программе MS Visio 2007.



Задание 6. Описать исходные, выходные и промежуточные данные и составить блок-схему решения для задачи по варианту.

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

		<i>геометрическое</i> , то есть квадратный корень из их произведения: $a \cdot b$.
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 : $c = a^2 + b^2$, $P = a + b + c$.
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 π 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 $.

Итог работы: документ, защита работы.

Практическая работа № 2

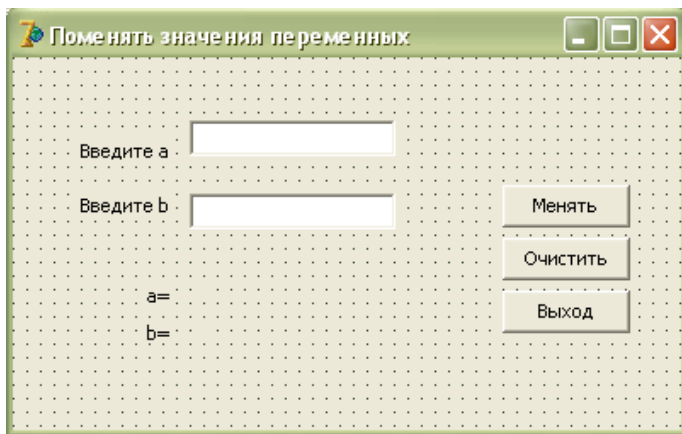
Составление программ линейной структуры.

Цель: изучить возможности ИСР Borland Delphi для разработки линейных алгоритмов.

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).

Итог работы: отчет, защита работы.

Практическая работа № 3 Составление программ разветвляющейся структуры

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

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

- Даны два неотрицательных числа a и b . Найти их среднее арифметическое и среднее геометрическое.
- Создать калькулятор для сложения и умножения обыкновенных дробей.
- Даны два неотрицательных числа. Найти сумму, разность, произведение и частное их квадратов.
- 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.

Итог работы: отчет, защита работы.

Практическая работа № 4 **Составление программ циклической структуры.**

Цель: изучить операторы, реализующие циклические алгоритмы.

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

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

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.

Итог работы: отчет, защита работы.

Практическая работа № 5 Обработка одномерных и двумерных массивов.

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

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

1. Дан вектор $A(n)$. В данном векторе определить минимальный элемент, максимальный элемент, среднее арифметическое положительных элементов, произведение отрицательных элементов, количество нулевых элементов с четными номерами (решение задачи реализовать двумя способами: с применением статических массивов, с применением динамических массивов).
2. В произвольно заданном одномерном массиве определить два элемента с наибольшими значениями и обнулить все элементы, расположенные между найденными значениями (решение задачи реализовать двумя способами: с применением статических массивов, с применением динамических массивов).
3. В произвольно заданном одномерном массиве целых чисел определить элементы, сумма цифр в записи которых максимальна и минимальна. Поместить найденные элементы в начало и в конец соответственно (решение задачи реализовать двумя способами: с применением статических массивов, с применением динамических массивов).
4. Дан вектор $K(n)$. Если в данном векторе ни один элемент не расположен после нечетного, то вывести все отрицательные компоненты вектора, иначе – все положительные. Порядок следования чисел в обоих случаях заменить обратным (решение задачи реализовать двумя способами: с применением статических массивов, с применением динамических массивов).
5. Дана квадратная матрица. Заменить все элементы, расположенные ниже побочной диагонали средним арифметическим элементов главной диагонали. Транспонировать итоговую матрицу.
6. Дана матрица $A_{m \times n}$. Сформировать вектор K , каждый элемент которого k_j , $j=1,2,\dots,n$ представляет собой сумму максимального и минимального элементов j -го столбца матрицы. Упорядочить элементы каждой строки матрицы по убыванию (решение задачи реализовать двумя способами: с применением статических массивов, с применением динамических массивов).

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

1. Пусть даны вещественные числа $x_1, x_2, x_3, \dots, x_n, y_1, y_2, y_3, \dots, y_n, r_1, r_2, r_3, \dots, r_n$. Выяснить, есть ли на плоскости точка, принадлежащая всем кругам $w_1, w_2, w_3, \dots, w_n$, где w_i имеет центр с координатами (x_i, y_i) и радиус r_i .
2. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы вывести индексы всех ее седловых точек. Седловые точки выделить.
3. Матрица $L_{m \times n}$ состоит из нулей и единиц. Удалить из нее совпадающие строки, а оставшиеся упорядочить по возрастанию двоичных чисел, образуемых строками.
4. Переставить строки и столбцы квадратной матрицы так, чтобы элементы главной диагонали образовали неубывающую последовательность.

Итог работы: отчет, защита работы.

Практическая работа № 7

Работа со строковыми переменными. Работа с данными типа множество.

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

Задание 1. 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>.

Задание 2. 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).

Итог работы: отчет, защита работы.

Практическая работа № 8

Организация и использование процедур.

Цель: изучить организации и использования процедур в языке программирования.

Задание: 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;
```

Итог работы: отчет, защита работы.

Практическая работа № 9 Организация и использование функций.

Цель: изучить организации и использования функций в языке программирования.

Задание: 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

{\$R *.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.

Итог работы: отчет, защита работы.

Практическая работа № 10

Работа с файлом последовательного доступа

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

Задание:

- 1) создайте папку для проекта «Практическая работа № 14»;
 - 2) в папке «Практическая работа № 14» создайте текстовый файл *quest.txt* с вопросами;
 - 3) в папке «Практическая работа № 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:='<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 интер-
претацию из файла
4..7: begin readln(fi); readln(fi,s); end;
8..10: begin readln(fi); 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 встать 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.

Итог работы: отчет, защита работы.

Практическая работа № 11 Работа с файлом произвольного доступа.

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

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

1. В первой строке текстового файла задано число n – количество первых элементов вектора, сумму которых нужно найти. Сам вектор находится во второй строке. Найти сумму n первых элементов вектора. Результат вывести в текстовый файл (см. лекцию).
2. В текстовом файле хранится массив $A=[a_{ij}]_{m \times n}$, $m, n \in \mathbb{N}$, $m \leq 10$, $n \leq 10$. Получить массив $B=[b_{ij}]_{m \times n}$, $m, n \in \mathbb{N}$, $m \leq 10$, $n \leq 10$, для которого $b_{ij}=2a_{ij}$, $i=1,2,\dots,m$, $j=1,2,\dots,n$. Итоговый массив записать в текстовый файл (см. лекцию «Работа с файлом произвольного доступа»).
3. Написать программу-тест «Вы тверды? Непреклонны?» (приложение 1, мини-тест, предложенный редакцией газеты «Черемховские новости»). Вопросы хранятся в текстовом файле. Ответы пользователя («да», «нет») записываются в текстовый файл (для ввода ответа пользователя использовать метод MessageBox). Требуется вывести результат тестирования.

Итог работы: отчет, защита работы.

Практическая работа № 12 Программирование модуля.

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

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.

```

Итог работы: отчет, защита работы

Практическая работа № 13

Создание библиотеки подпрограмм. Использование библиотеки подпрограмм.

Цель: изучить возможности языка программирования для создания пользовательских подпрограмм.

Задание. 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.

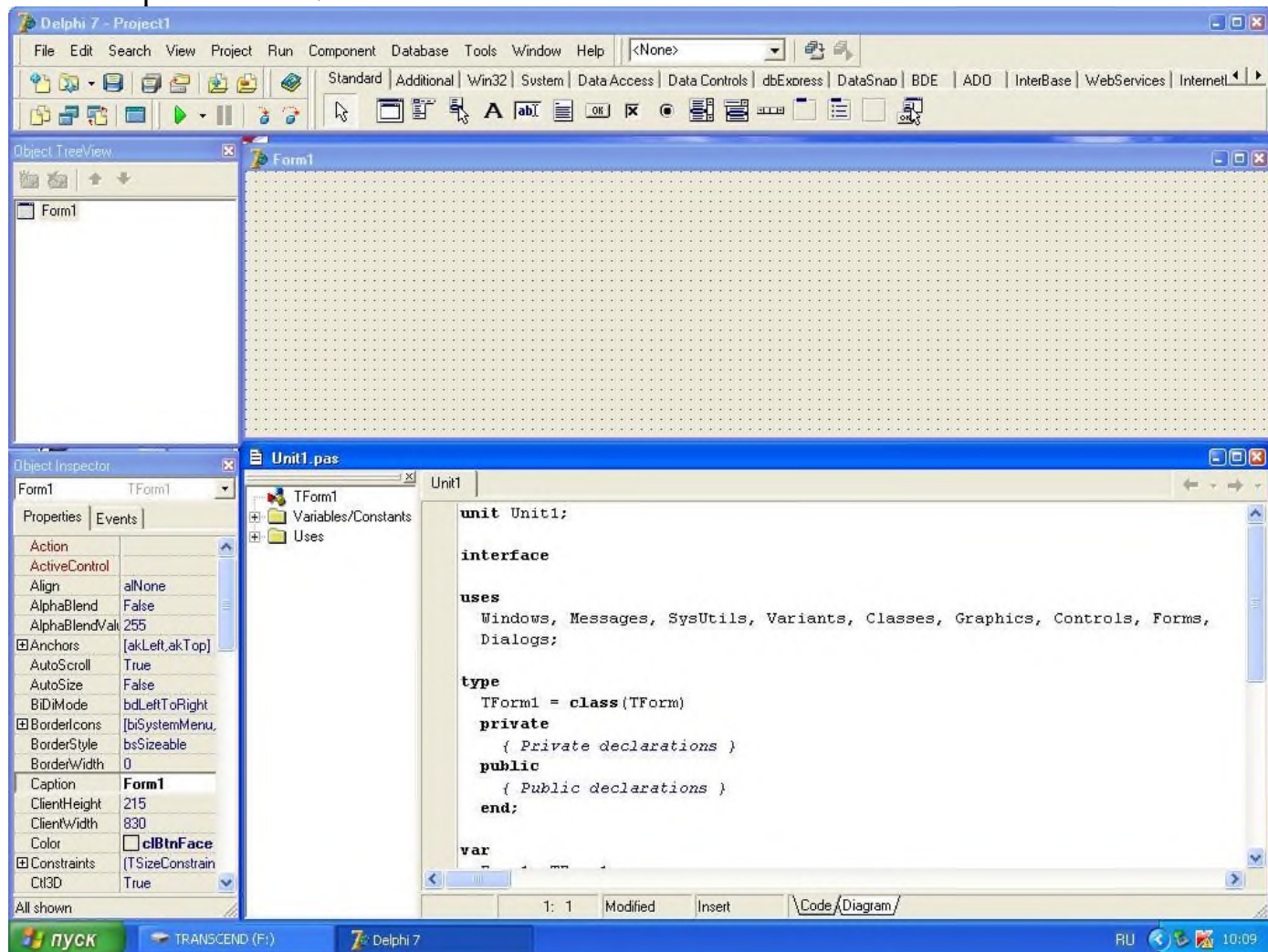
Итог работы: отчет, защита работы.

Практическая работа № 14

Создание простого проекта. Отладка проекта.

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

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).



Итог работы: отчет, защита работы

Практические занятия № 15 Проектирование и формирование формы

Цель: изучить основные свойства, методы и события формы.

Проектирование форм – ядро визуальной разработки в среде Delphi. Каждый помещаемый в форму компонент или любое задаваемое свойство сохраняется в файле, описывающем форму (DFM-файл), а также оказывает некоторое влияние на исходный текст, связанный с формой (PAS-файл).

Задание 1.

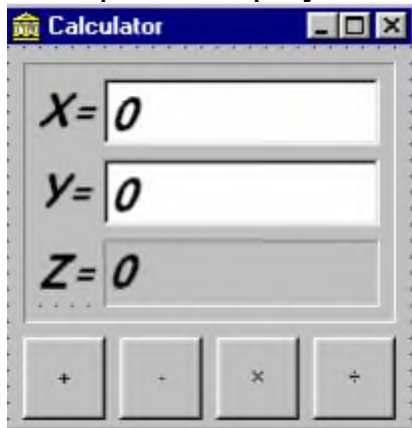
1. Запустить Borland Delphi 7: пуск→все программы→ Borland Delphi → Delphi 7.
2. Совершите экскурс в среду визуального программирования Delphi. При этом результаты своей работы не сохраняйте на диске.

3. Попробуйте создать различные приложения с помощью Expert-ов Delphi, исследовать в окне редактирования полученный код. Приложения на диске не запоминать.

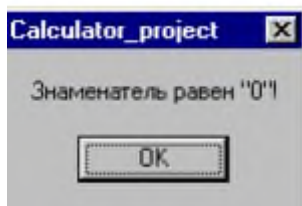
4. Выберите пункт меню File/New Application, этим вы создадите новый проект приложения. Выберите команду File/Save Project As... → В появившемся диалоге перейдите "рабочий стол" → Откройте папку "" для сохранения → откройте свою папку. Если такой нет, то создайте её, щелкнув правой кнопкой мыши на свободном 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. 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	$Z = \cos(6X-3Y+1)$.

Итог работы: отчет, защита работы.

Практическая работа № 16

Создание документации. Тестирование, отладка приложения.

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

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:

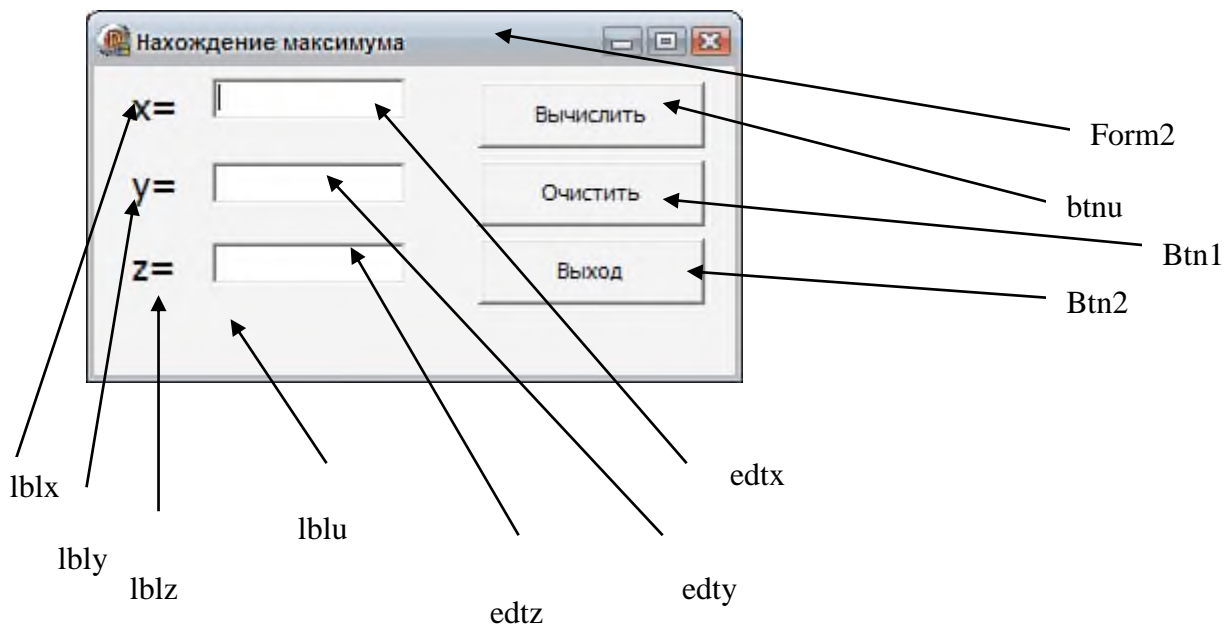
И.Д.: $x, y, z \in R$

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!!!'



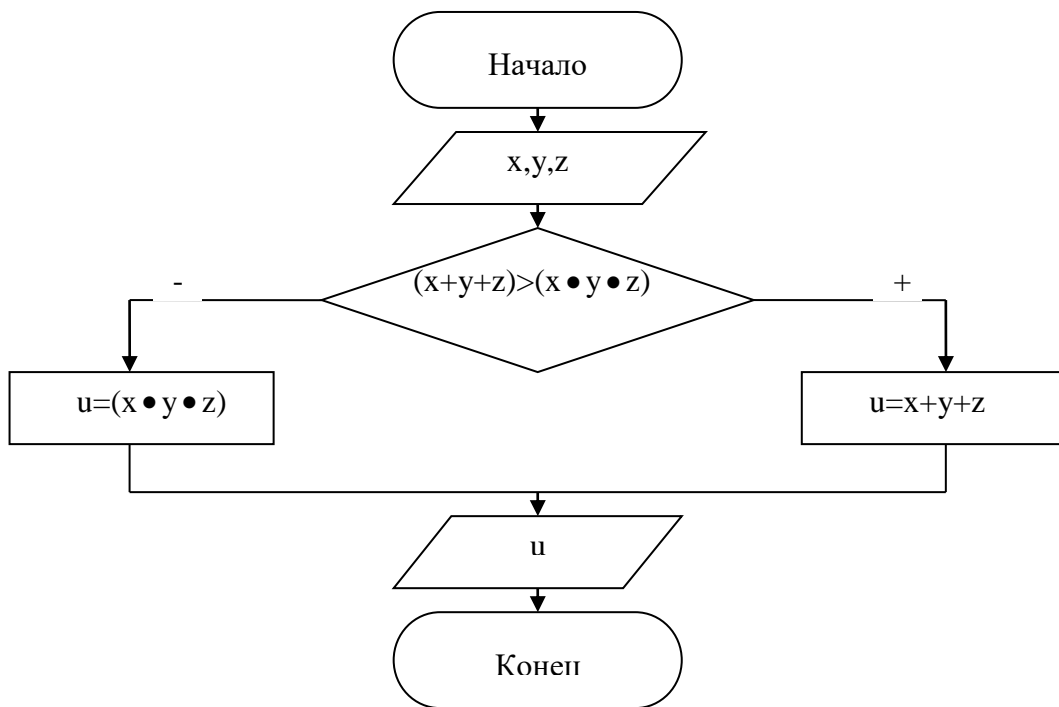
Объект	Свойство	Значение	Событие	Программный код
Form	Name Caption	Form2 'Нахождение максимума'		
label	Name Caption	Lblx 'x='		
label	Name Caption	Lbly 'y='		
label	Name Caption	Lblz 'z='		
label	Name Caption	Lblu "		
edit	Name Text	Edtx "		
edit	Name Text	Edty "		
edit	Name Text	Edtz "		
Button	Name 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
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;
lbl.Caption:='Максимум равен '+FloatToStr(max);
except
showmessage ('Вводите только числа!!!')
end;
end;

```

Программный код 2:

```

procedure TForm2.btn1Click(Sender: TObject);
begin
edtX.Text:="";
edy.Text:="";
edtz.Text:="";
lbl.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

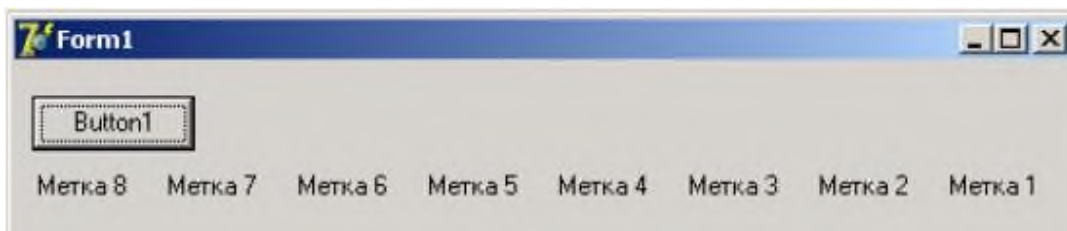
Итог работы: отчет, защита работы.

Практическая работа № 17 Создание экземпляров класса. Объявление класса.

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

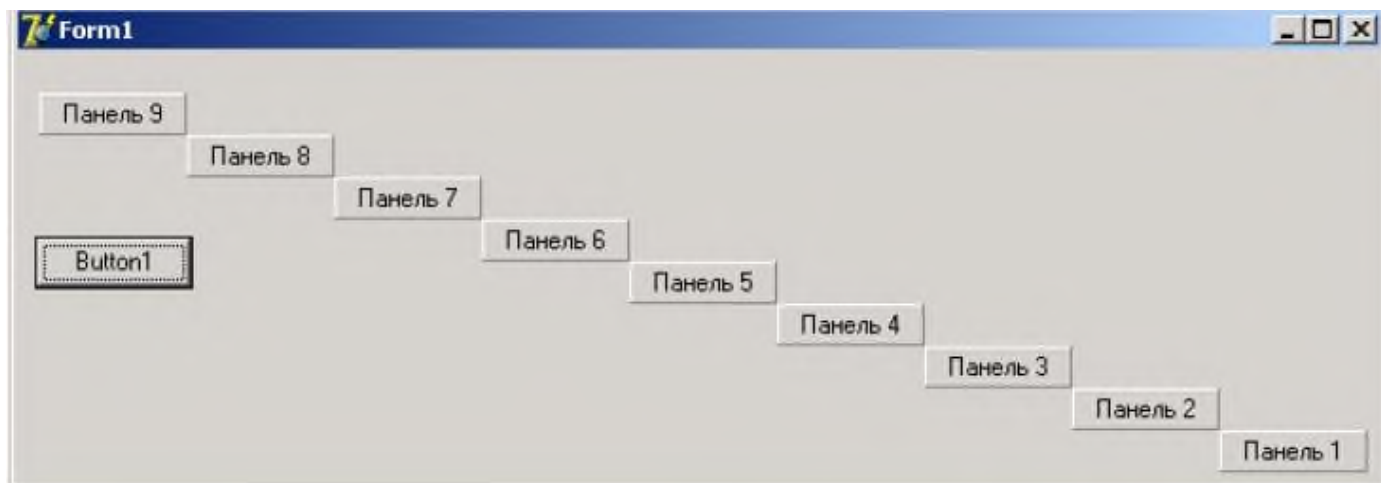
Задание 1. Реализовать метод, который будет размещать массив из 8 «новых» меток (TLabel) на форме в одну строку.

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.



Практическая работа № 18

Создание наследованного класса. Перегрузка методов.

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

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.

```

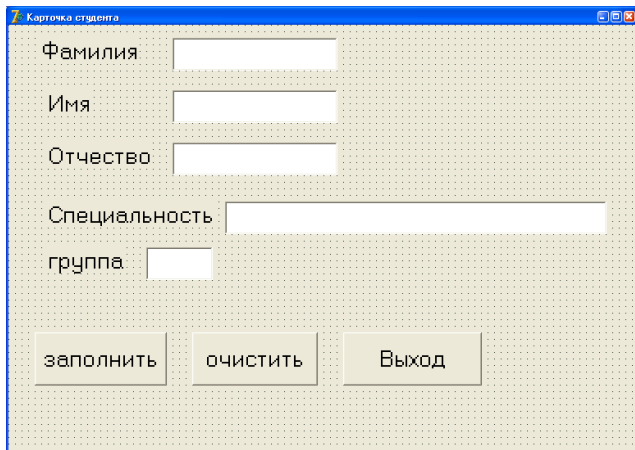
Итог работы: отчет, защита работы.

Практическая работа № 19

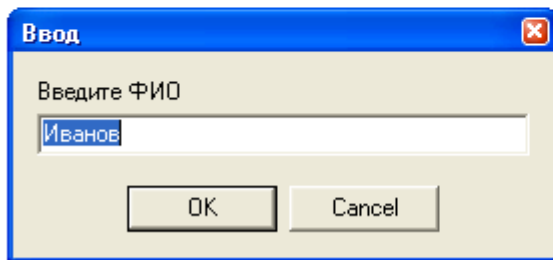
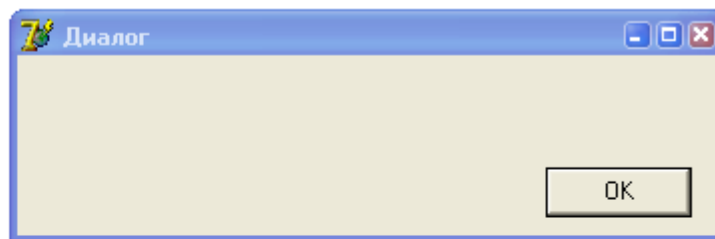
Создание проекта с использованием компонентов для работы с текстом.

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

Задание 1. Создайте диалоговое окно для вывода карточки студента.



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



Итог работы: отчет, защита работы.

Практическая работа № 20

Создание проекта с использованием компонентов стандартных диалогов.

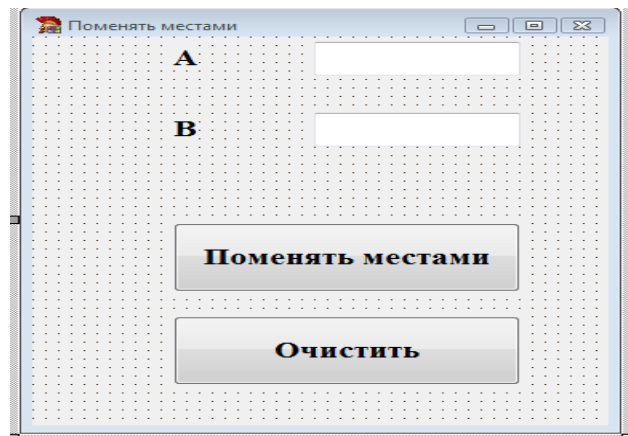
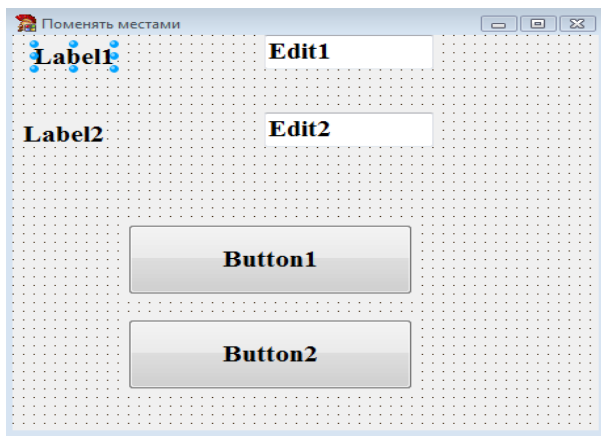
Цель: изучить возможности языка программирования для создания проектов с использованием компонентов стандартных диалогов.

Задание 1. Описать элементы управления в вкладке «Standard» (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
```

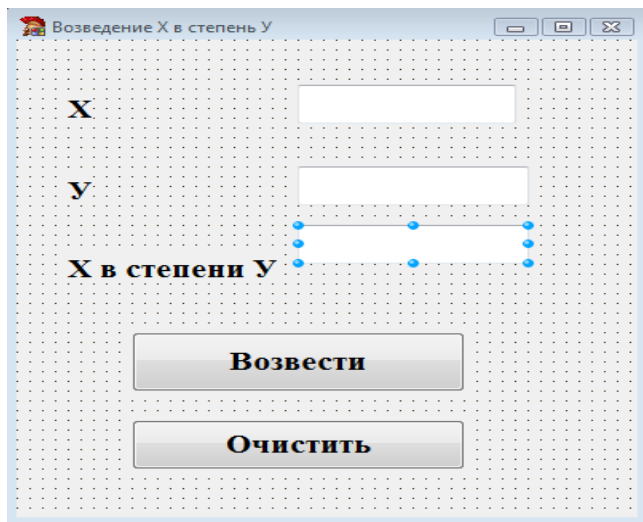
```
Edit1.Text:=' '; {Компонентам Edit1.Text, Edit2.Text присваивается значение
```

```
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
```

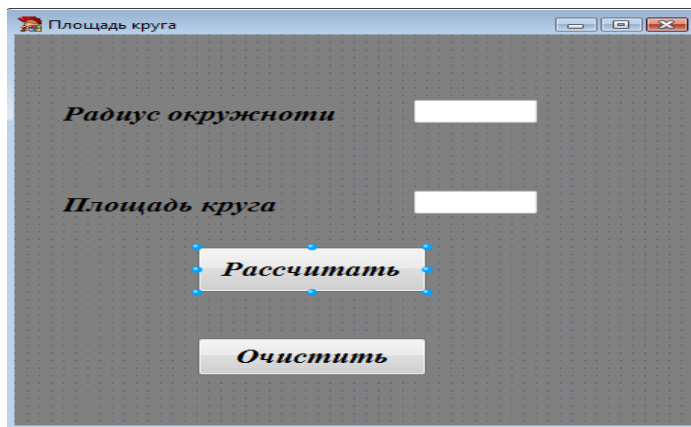
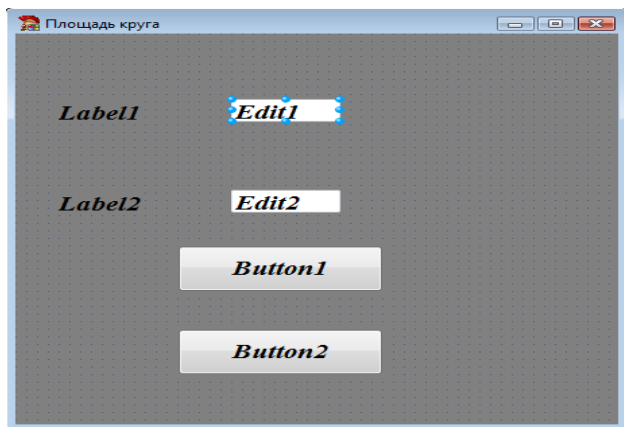
```
Edit1.Text:=' '; {Компонентам Edit1.Text, Edit2.Text присваивается значение
```

```
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 π 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: $a \cdot b$.
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 π 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 $.

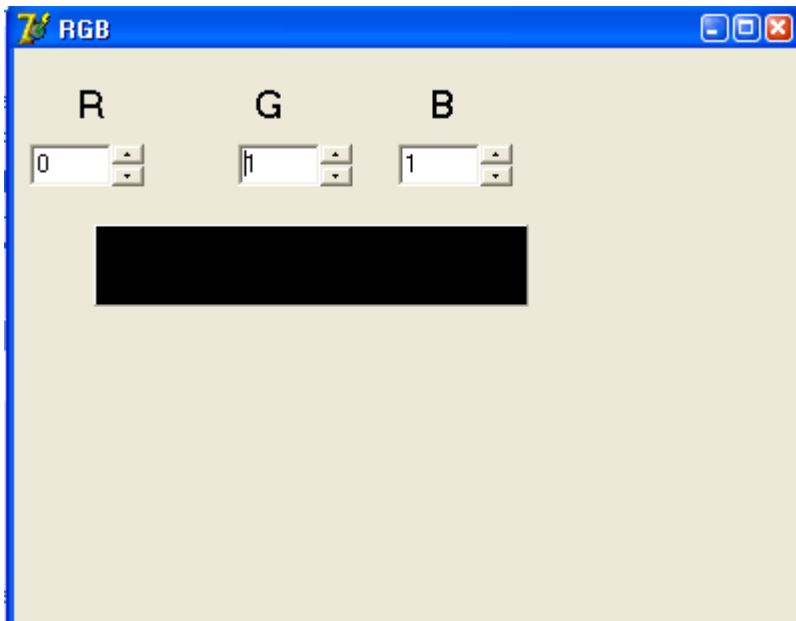
Итог работы: отчет, защита работы.

Практическая работа № 21

Создание проекта с использованием компонентов системы меню.

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

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

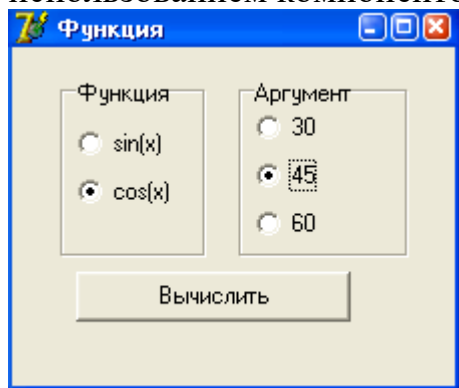
```

```

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

```

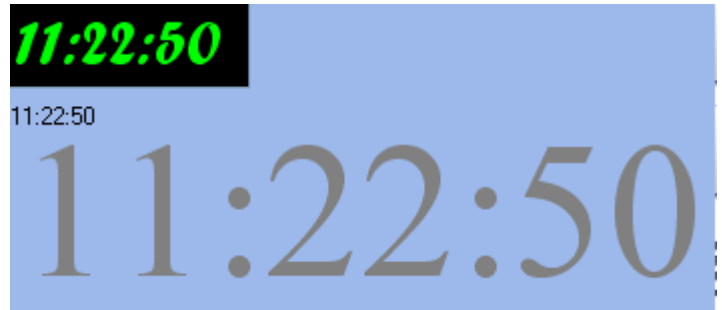
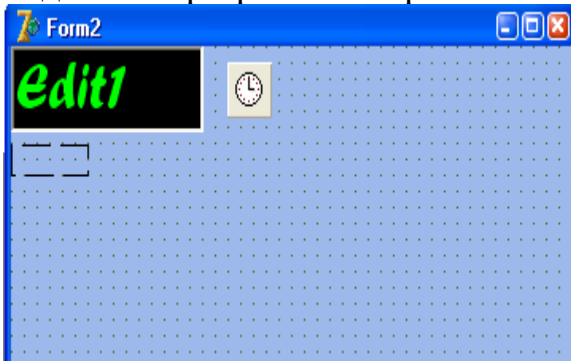
```

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.

```

Задание 3. 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

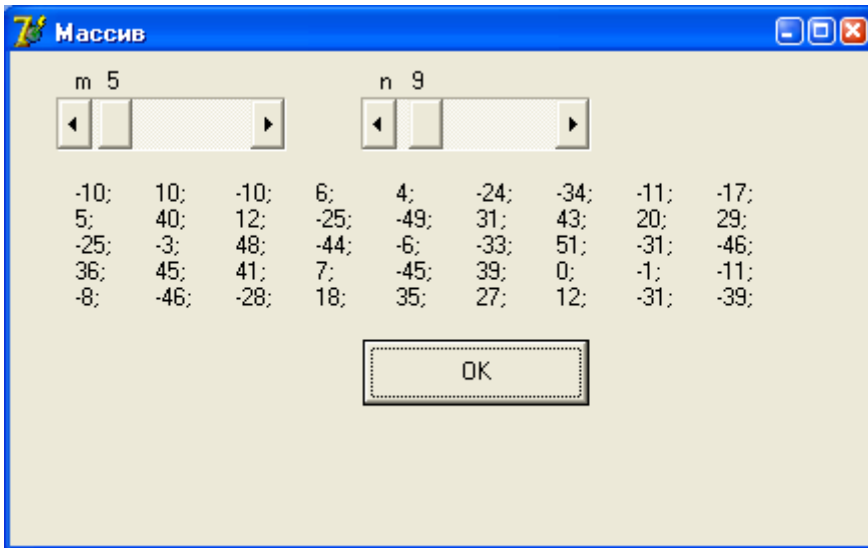
```

```

    { 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 4. 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;
    end;

```

```

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 5. 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.

Итог работы: отчет, защита работы.

Практическая работа № 22 Обработка исключительных ситуаций

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

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 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 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.
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 так же введения символов вместо цифр. Выдать сообщение о типе возникшей ошибке.

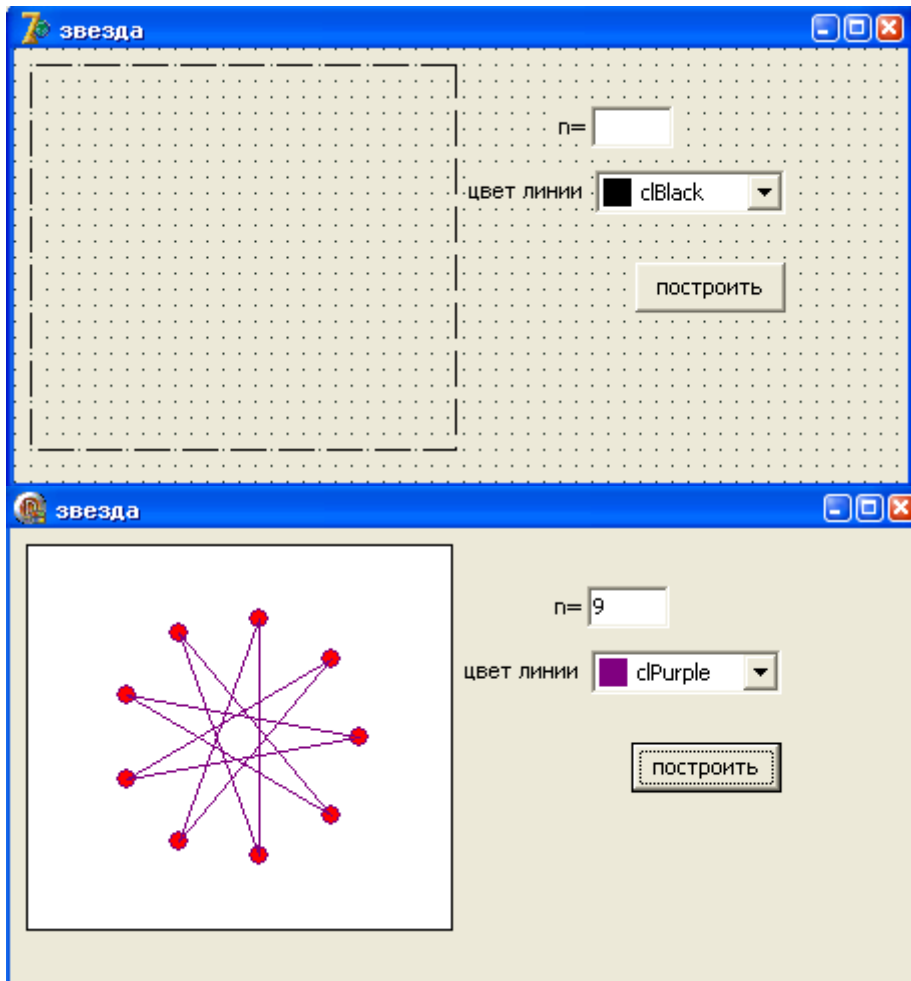
Итог работы: отчет, защита работы.

Практическая работа №23

Графические возможности системы программирования.

Цель: изучить графические примитивы контурной графики языка программирования.

Задание 1. Дано натуральное число n^3 . Построить звезду, у которой n вершин и координаты центра x_s, y_s .



```

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

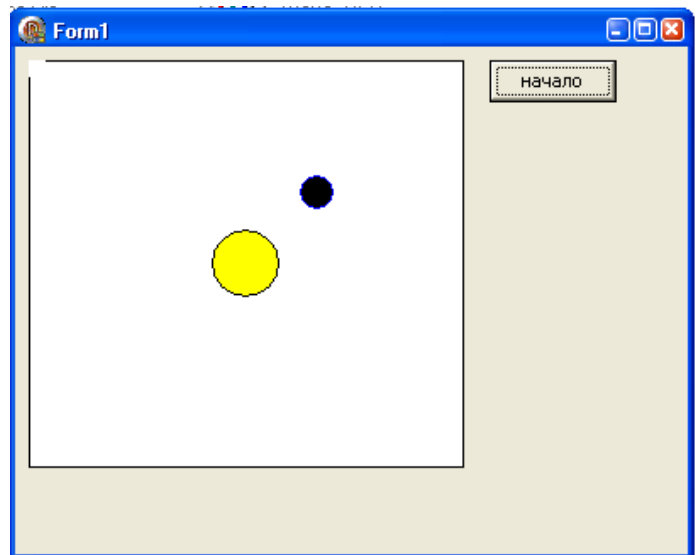
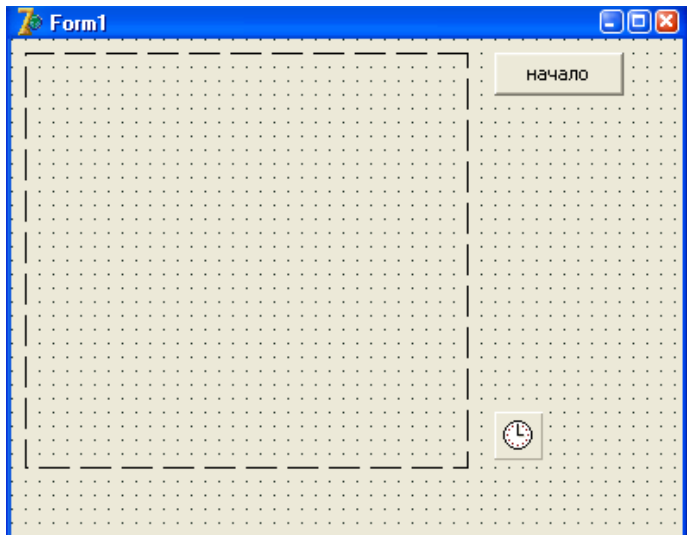
```

```

{$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.

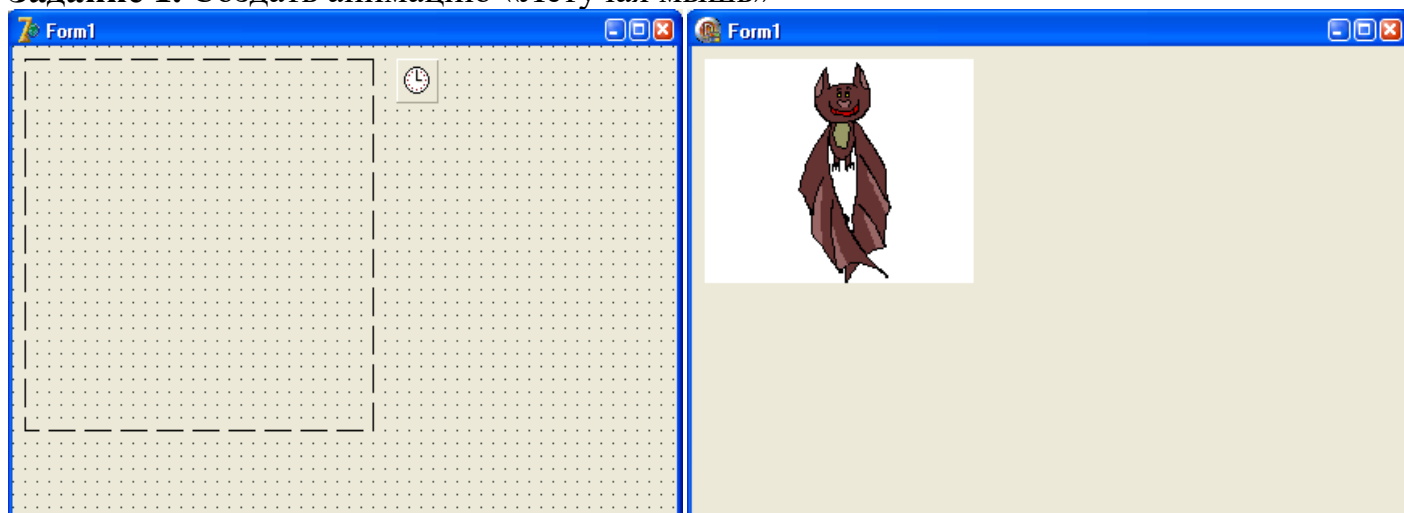
```

Итог работы: отчет, защита работы.

Практическая работа № 24 Создание анимации.

Цель: изучить создание анимации.

Задание 1. Создать анимацию «Летучая мышь»



```

unit Unit1;
interface
uses

```

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

type

```
TForm1 = class(TForm)
  Img1: TImage;
  Timer1: TTimer;
  procedure Timer1Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
```

implementation

```
var i:byte=1;
{$R *.dfm}
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if i>6 then i:=1;
  img1.picture.loadfromfile(getcurrentdir+'\pics\'+inttostr(i)+'.bmp');
  inc(i);
end;
end.
```

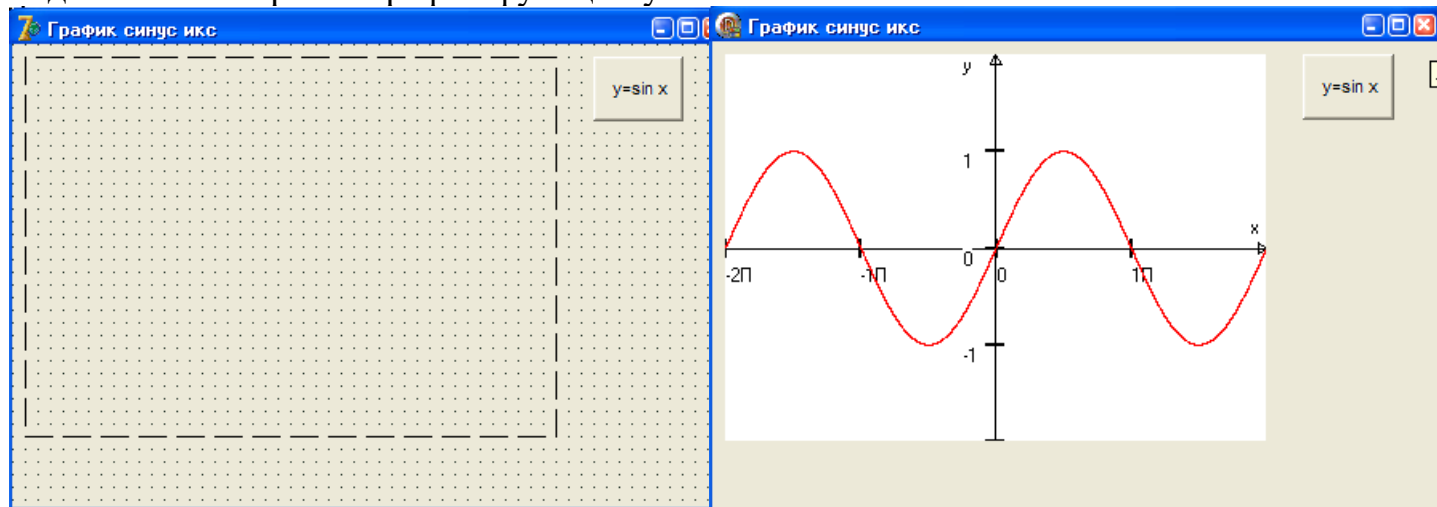
Задание 2. Создать анимацию «Летучая мышь»

Практическая работа № 25

Графические примитивы контурной графики.

Цель: изучить графические примитивы контурной графики языка программирования.

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))+'Π');
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	$y=x^3+7x+10$

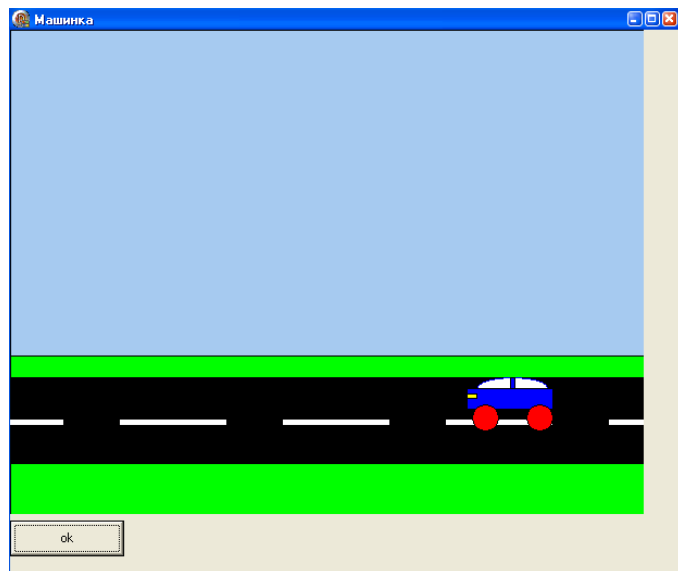
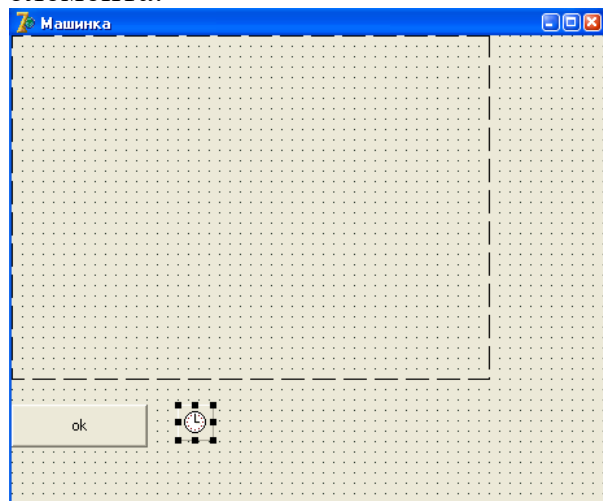
Итог работы: отчет, защита работы.

Практическая работа № 26

Средства прорисовки контурного изображения.

Цель: изучить средства прорисовки контурного изображения.

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.

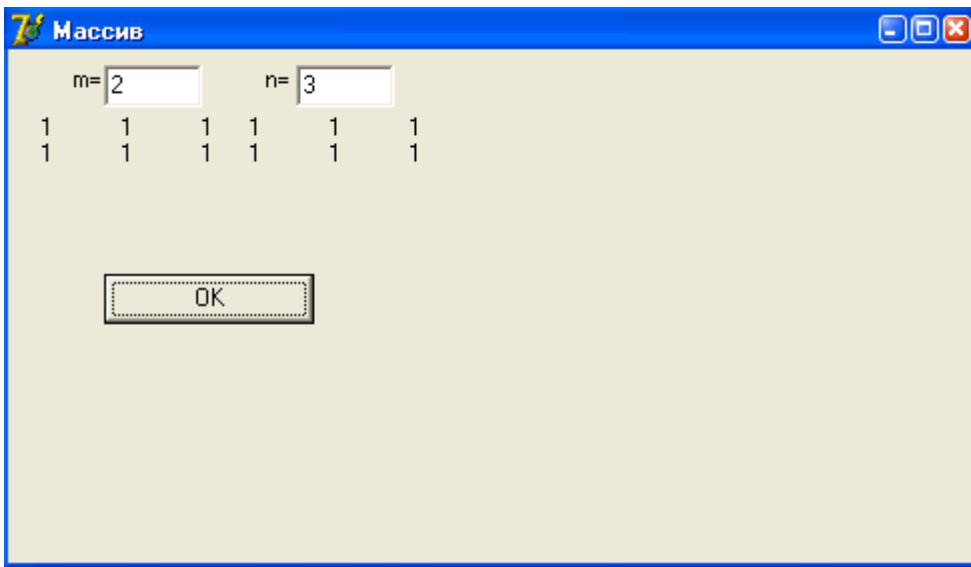
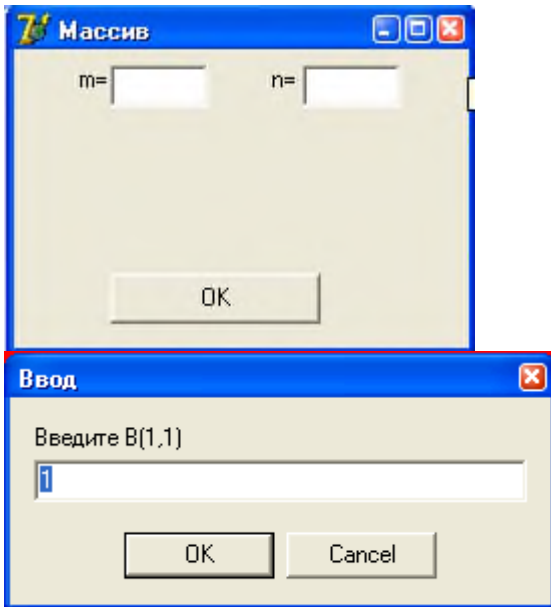
Итог работы: отчет, защита работы.

Практическая работа № 27

Разработка оконного приложения. Разработка оконного приложения с несколькими формами.

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

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;
Lb1p: 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;
lbp.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[l];
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.

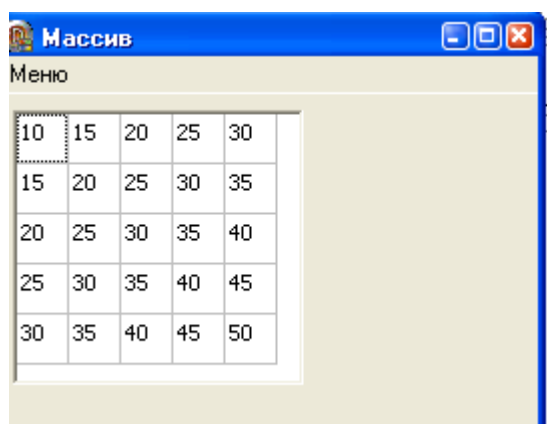
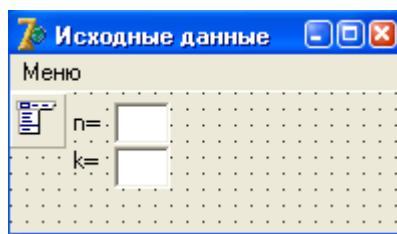
```

Итог работы: отчет, защита работы.

Практическая работа № 28 Разработка многооконного приложения.

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

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 frm1mass do
begin
show;
left:=self.Left+self.Width+10;
top:=self.Top;
end;
end;
end;
end.

```

Unit 2.

```

object frm1mass: Tfrm1mass
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
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.

Итог работы: отчет, защита работы.

4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРАКТИЧЕСКИХ РАБОТ

Основные:

О–1. Семакин, Основы алгоритмизации и программирования: учебник- ИЦ Академия, 2019 г.

О-2 Семакин И.Г., Шестаков А.П., Основы алгоритмизации и программирования. Практикум, учебник - ИЦ Академия, 2018 г.

Дополнительные:

Д–1. Голицына О.Л., Партыка Т.Л., Попов И.И. Программное обеспечение: Учебное пособие – М.: ИД "ФОРУМ"-ИНФРА-М, 2008.

Д–2. Семакин И.Г. Основы программирования: Учебное пособие – М.: Академия, 2003.

4.2 Электронные издания (электронные ресурсы)

1. Семакин, Основы алгоритмизации и программирования: учебник- ИЦ Академия, 2019 г., 15 подключений.

**5. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЕННЫХ В
МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

№ изменения, дата внесения, № страницы с изменением	
Было	Стало
Основание:	
Подпись лица, внесшего изменения	