

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КРАСНОЯРСКОГО КРАЯ

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«КРАСНОЯРСКИЙ КОЛЛЕДЖ ОТРАСЛЕВЫХ ТЕХНОЛОГИЙ
И ПРЕДПРИНИМАТЕЛЬСТВА»**

РАССМОТРЕНО

методической комиссией
протокол № 06 от «24» июня 2021 г.

УТВЕРЖДЕНО

Директор КГБПОУ «Красноярский колледж
отраслевых технологий и предпринимательства»

_____/Н. В. Журова/
Приказ № 01-91-1п от « 30 » июня 2021 г.

**АДАПТИРОВАННАЯ ПРОГРАММА ПОДГОТОВКИ
СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

09.02.07 Информационные системы и программирование

на базе среднего общего образования

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ**

ОП.04 Основы алгоритмизации и программирования

и.о. зам. директора по УР _____ / Е.В. Миля /
подпись

Красноярск 2021

Организация-разработчик: КГБПОУ «Красноярский колледж отраслевых технологий и предпринимательства»

Разработчики: Лавренков Семен Сергеевич, преподаватель

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
2. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ И
ОФОРМЛЕНИЮ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ
3. МЕТОДИКА ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И
ЛАБОРАТОРНЫХ РАБОТ
4. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ

1.ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания к проведению практических занятий и лабораторных работ по учебной дисциплине ОП.04 Основы алгоритмизации и программирования, предназначены для обучающихся СПО по специальности 09.02.07 Информационные системы и программирование.

Уровень профессиональной подготовки по специальности 09.02.07 Информационные системы и программирование, определяемый ФГОС СПО, предусматривает владение практическими навыками выбора материалов для профессиональной деятельности.

Для успешного выполнения и защиты работы обучающиеся должны иметь:

умения	– 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
знания	– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. – эволюцию языков программирования, их классификацию, понятие системы программирования. – основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. – подпрограммы, составление библиотек подпрограмм. объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.

Выполнение лабораторных работ способствует формированию и развитию общих и профессиональных компетенций:

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 09	Использовать информационные технологии в профессиональной деятельности
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке

Код	Наименование профессиональных компетенций
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

2. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы выполняются обучающимися по графику, составленному в соответствии с рабочей программой учебной дисциплины ОП.04 Основы алгоритмизации и программирования.

Учебная дисциплина ОП.04 Основы алгоритмизации и программирования зависит от содержания лабораторных работ, которые соответствуют более глубокому освоению дисциплины, закреплению теоретических знаний и прививают обучающимся практические навыки самостоятельной работы. Задача лабораторных работ – закрепить теоретические знания обучающихся. Согласно учебного плана по специальности и программы учебной дисциплины на лабораторные (практические) занятия обучающихся выделено 76 академических часов, из них:

№ и наименование разделов (тем)	Количество часов
ЛР №1: Составление сложных алгоритмов	2
ЛР №2: Знакомство с системой программирования	2
ЛР №3: Программирование линейных алгоритмов	2
ЛР №4: Программирование разветвляющихся алгоритмов	2
ЛР №5: Операторы цикла	2
ЛР №6: Использование процедур в программировании	2
ЛР №7: Работа с массивами	2
ЛР №8: Строковые данные в программировании	2
ЛР №9: Программирование линейных алгоритмов на языке C++	3
ЛР №10: Программирование разветвляющихся алгоритмов на языке C++	2
ЛР №11: Программирование циклических алгоритмов на языке C++	2
ЛР №12: Одномерные массивы	1
ЛР №13: Двумерные массивы	2
ЛР №14: Процедуры и функции	2
ЛР №15: Указатели	2
ЛР №16: Сортировка массивов	2
ЛР №17: Динамические массивы	2
ЛР №18: Строки типа char	2
ЛР №19: Обработка текстовых файлов	2
ЛР №20: Структуры	2
ЛР №21: Классы	2
ЛР №22: Конструкторы и деструкторы	2
ЛР №23: Перегрузка операций	2
ЛР №24: Основные принципы ООП	2
ЛР №25: Наследование с использованием абстрактного базового класса	2
ЛР №26: Множественное наследование	2
ЛР №27: Шаблоны классов	4
ЛР №28: Стандартная библиотека шаблонов	4
ЛР №29: Создание простейшей программы на языке Visual C++	2
ЛР №30: Условные операторы. Вычисление значений функции, заданной условно	2
ЛР №31: Циклический алгоритм. Табулирование функции и поиск экстремумов	2
ЛР №32: Построение графика функции на промежутке с определенным шагом	2
ЛР №33: Понятие одномерного массива. Селективная обработка элементов массива	2
ЛР №34: Понятие матрицы. Селективная обработка элементов строк, столбцов и диагоналей матрицы	2
ЛР №35: Разработка приложения для обработки базы данных	4
ИТОГО:	76

3. МЕТОДИКА ПРОВЕДЕНИЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Лабораторное или практическое занятие состоит из следующих элементов: вводная часть, основная и заключительная.

Вводная часть обеспечивает подготовку студентов к выполнению заданий работы. В ее состав входят:

- формулировка темы, цели и задач занятия, обоснование его значимости в профессиональной подготовке студентов;
- изложение теоретических основ работы;
- характеристика состава и особенностей заданий работы и объяснение методов (способов, приемов) их выполнения;
- характеристика требований к результату работы;
- инструктаж по технике безопасности при эксплуатации технических средств;
- проверка готовности студентов выполнять задания работы;
- указания по самоконтролю результатов выполнения заданий студентами.

Основная часть включает процесс выполнения лабораторной работы, оформление отчета и его защиту. Она может сопровождаться дополнительными разъяснениями по ходу работы, устранением трудностей при ее выполнении, текущим контролем и оценкой результатов отдельных студентов, ответами на вопросы студентов.

Заключительная часть содержит:

- подведение общих итогов занятия;
- оценку результатов работы отдельных студентов;
- ответы на вопросы студентов;
- выдачу рекомендаций по устранению пробелов в системе знаний и умений студентов, по улучшению результатов работы;
- сбор отчетов студентов для проверки, изложение сведений, касающихся подготовки к выполнению следующей работы.

Вводная и заключительная части занятия проводятся фронтально. Основная часть может выполняться индивидуально или коллективно (в зависимости от формы организации занятия).

Оценивание лабораторно-практических работ проводится дифференцированно (по пятибалльной системе) и при определении оценок за семестр рассматривается как один из основных показателей текущего учета знаний.

Критериями оценки результативности лабораторного занятия являются:

- степень реализации цели и задач работы;
- степень выполнения заданий;
- степень соответствия результатов работы заданным требованиям;
- степень сформированности у студентов необходимых умений и навыков

Критерии лабораторно-практического занятия:

Оценка «5» (отлично): выполнены все задания лабораторно-практического занятия, студент четко и без ошибок ответил на все контрольные вопросы.

Оценка «4» (хорошо): выполнены все задания лабораторно-практического занятия; студент ответил на все контрольные вопросы с замечаниями.

Оценка «3» (удовлетворительно): выполнены все задания лабораторно-практического занятия с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

Оценка «2» (не зачтено): студент не выполнил или выполнил неправильно задания лабораторно-практического занятия; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

4. СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ

ЛР №1: Составление сложных алгоритмов + ЛР №2: Знакомство с системой программирования (2+2 часа)

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.

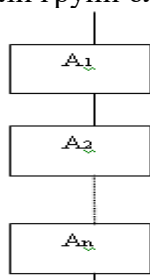
$$h = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a}$$

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.



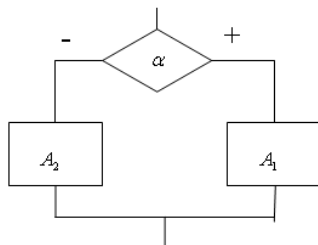
A_i - функциональный блок

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

Функциональный блок – это любая базовая структура или их комбинация

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

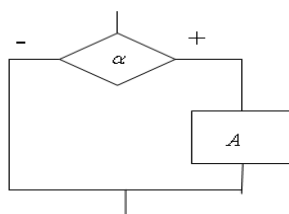
A_i - функциональный блок



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

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

Полная разветвляющаяся (логическая) структура



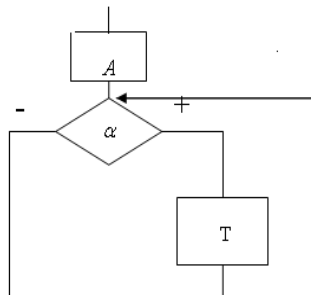
A - функциональный блок

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

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»)

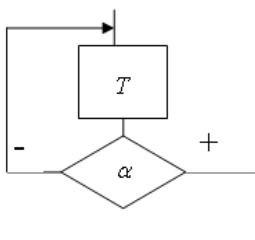


A - функциональный блок, начальных присвоений.

T - функциональный блок, тело цикла.

α - логическое условие

б) цикл с постусловием («до»)



A - функциональный блок, начальных присвоений.

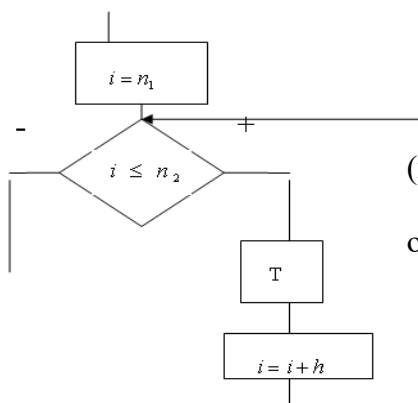
T - функциональный блок, тело цикла.

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



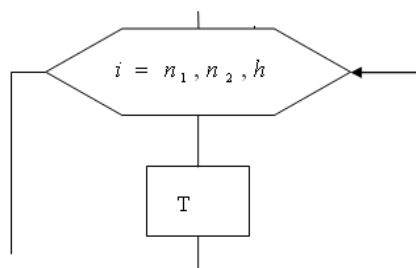
i-параметр цикла.

n_1, n_2 - начальное и конечное значение параметра (границы параметра).

h- шаг измерения параметра (если шаг равен 1, то его опускают).

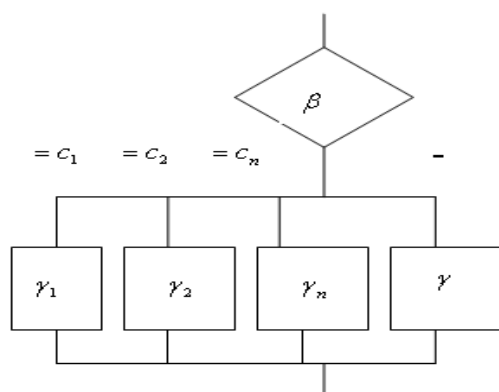
T - функциональный блок.

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
НТ		
	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

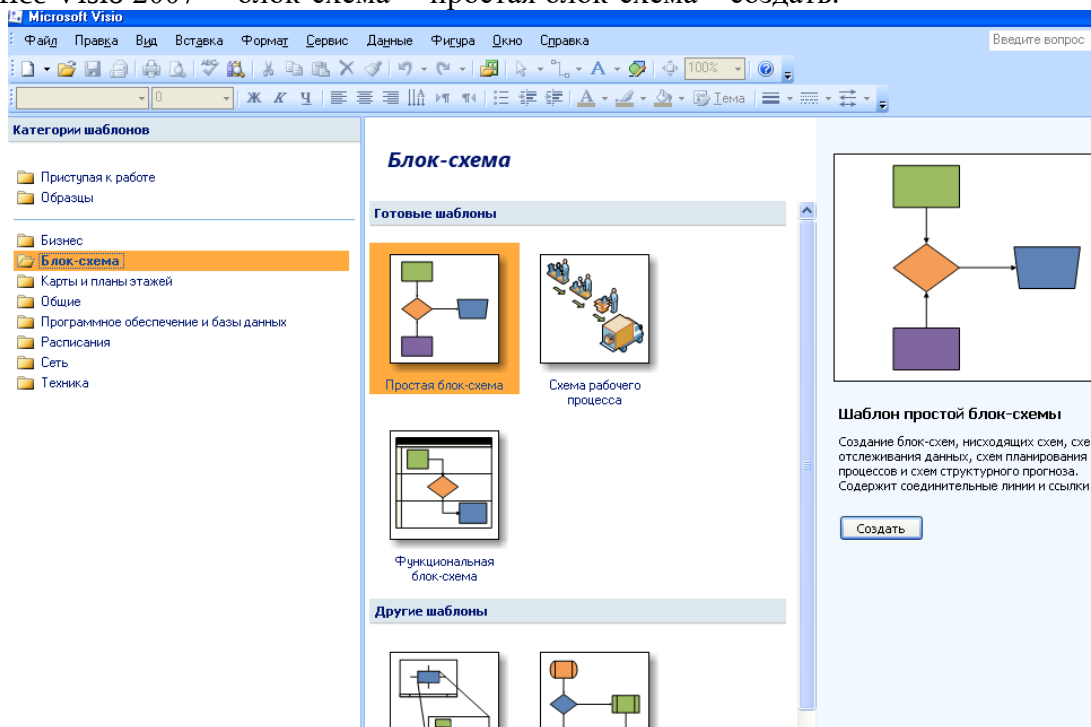
----()	комментарии	для внесения пометок
---------	-------------	----------------------

Итог работы: тетрадь, защита работы.

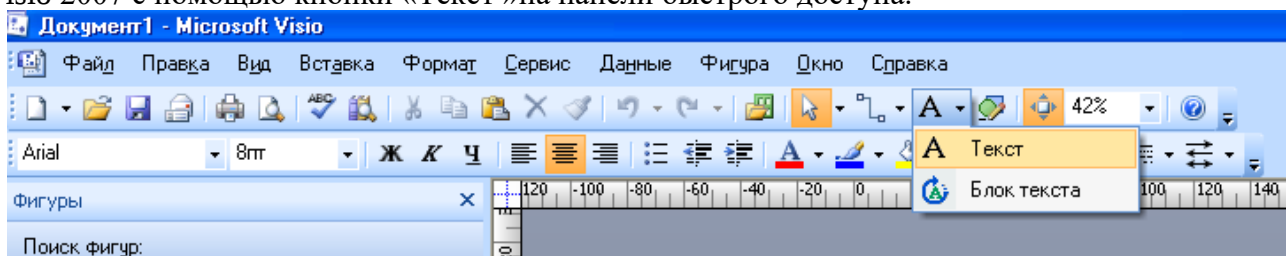
ЛР №3: Программирование линейных алгоритмов+ ЛР №4: Программирование разветвляющихся алгоритмов (2+2 часа)

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

Задание 1. Запустить MS Visio 2007: пуск→все программы→ Microsoft Office 2007→ Microsoft Office Visio 2007→ блок-схема →простая блок-схема→создать.



Задание 2. 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 $P = 4 \cdot a$.

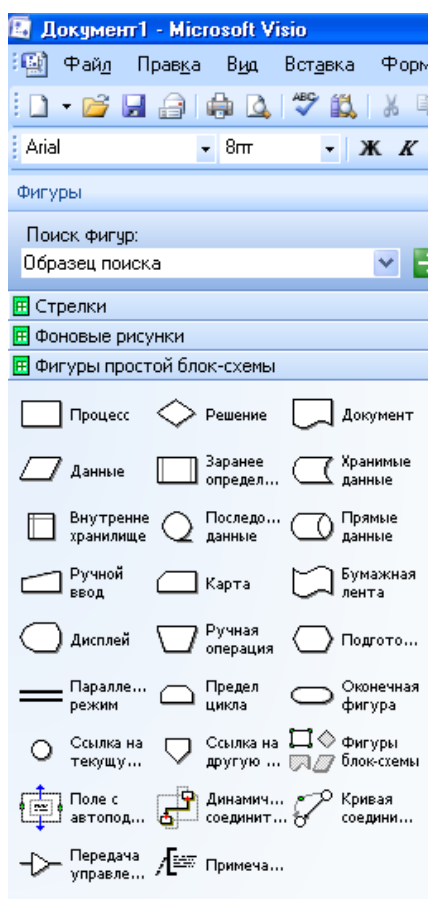
Задача 2. Дана сторона квадрата a . Найти его площадь $S = a^2$.

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



Задание 4. 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 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: документ, защита работы.

ЛР №5: Операторы цикла (2 часа)

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:

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).

Итог работы: отчет, защита работы.

ЛР №6: Использование процедур в программировании (2 часа)

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

Задание. 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 .

Итог работы: отчет, защита работы.

ЛР №7: Работа с массивами (2 часа)

Задание. 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.

ЛР №8: Строковые данные в программировании (2 часа)

Цель: изучить операторы, реализующие циклические алгоритмы.

Задание. 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)

Итог работы: отчет, защита работы.

ЛР №9: Программирование линейных алгоритмов на языке C++ (3 часа)

Цель: изучить операторы, реализующие циклические алгоритмы.

Задание. 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.

ЛР №10: Программирование разветвляющихся алгоритмов на языке C++ (2 часа)

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

Задание. 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).

Итог работы: отчет, защита работы.

ЛР №11: Программирование циклических алгоритмов на языке C++ (2 часа)

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

Задание. 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.

ЛР №12: Одномерные массивы (1 час)

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

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>.

Итог работы: отчет, защита работы.

ЛР №13: Двумерные массивы (2 часа)

Цель: изучить возможности применения множеств в языке программирования, операции над множествами.

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).

Итог работы: отчет, защита работы.

ЛР №14: Процедуры и функции + ЛР №15: Указатели (2+2 часа)

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 менять 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;
A, B: Integer);
begin
  Summa:=A+B;
end;
```

Итог работы: отчет, защита работы.

ЛР №16: Сортировка массивов (2 часа)

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.

ЛР №17: Динамические массивы + ЛР №18: Строки типа char (2+2 часа)

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

Задание:

- 1) создайте папку для проекта «Практическая работа № 14»;
 - 2) в папке «Практическая работа № 17-18» создайте текстовый файл *quest.txt* с вопросами;
 - 3) в папке «Практическая работа № 17-18» 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

6) 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 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 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.

Итог работы: отчет, защита работы.

ЛР №19: Обработка текстовых файлов (2 часа)

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.

ЛР №20: Структуры + ЛР №21: Классы (2+2 часа)

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

ЛР №22: Конструкторы и деструкторы (2 часа)

Цель: изучить возможности языка программирования для создания пользовательских подпрограмм.

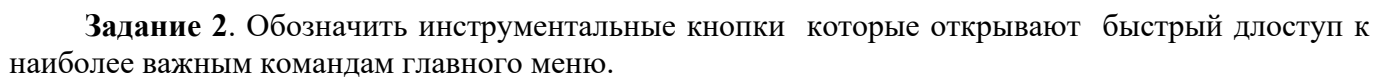
Задание. Решить задачу своего варианта и оформить согласно этапам решения задач на ЭВМ.

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.

ЛР №23: Перегрузка операций (2 часа)

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

23

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



Итог работы: отчет, защита работы

ЛР №24: Основные принципы ООП + ЛР №25: Наследование с использованием абстрактного базового класса (2+2 часа)

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:

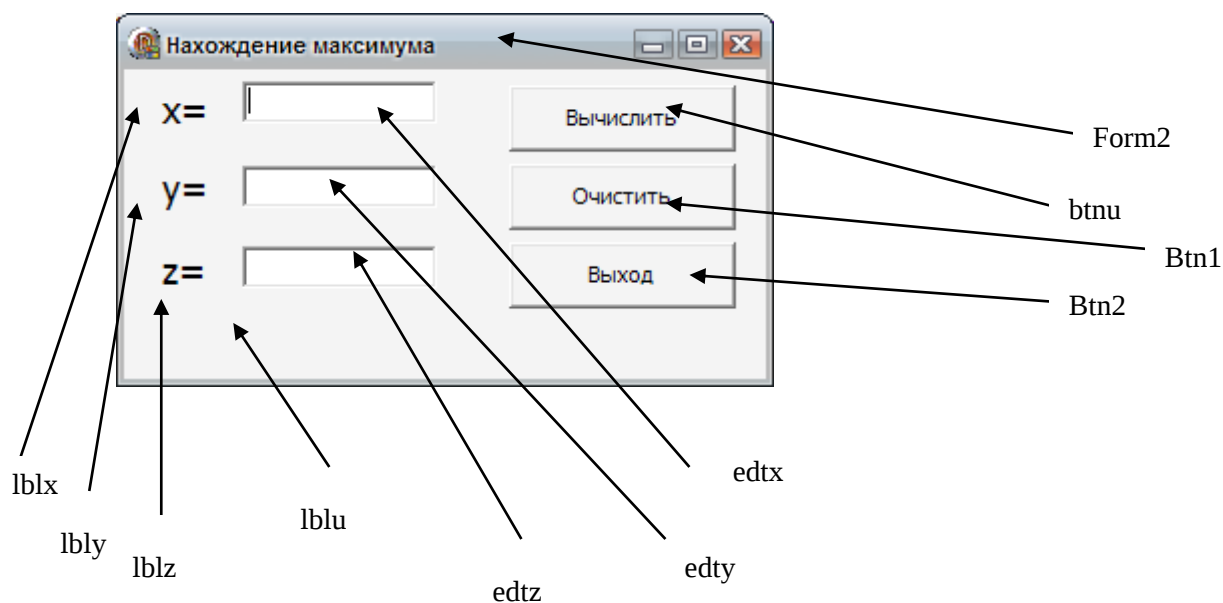
И.Д.: $x, y, z \in R$

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!!!'



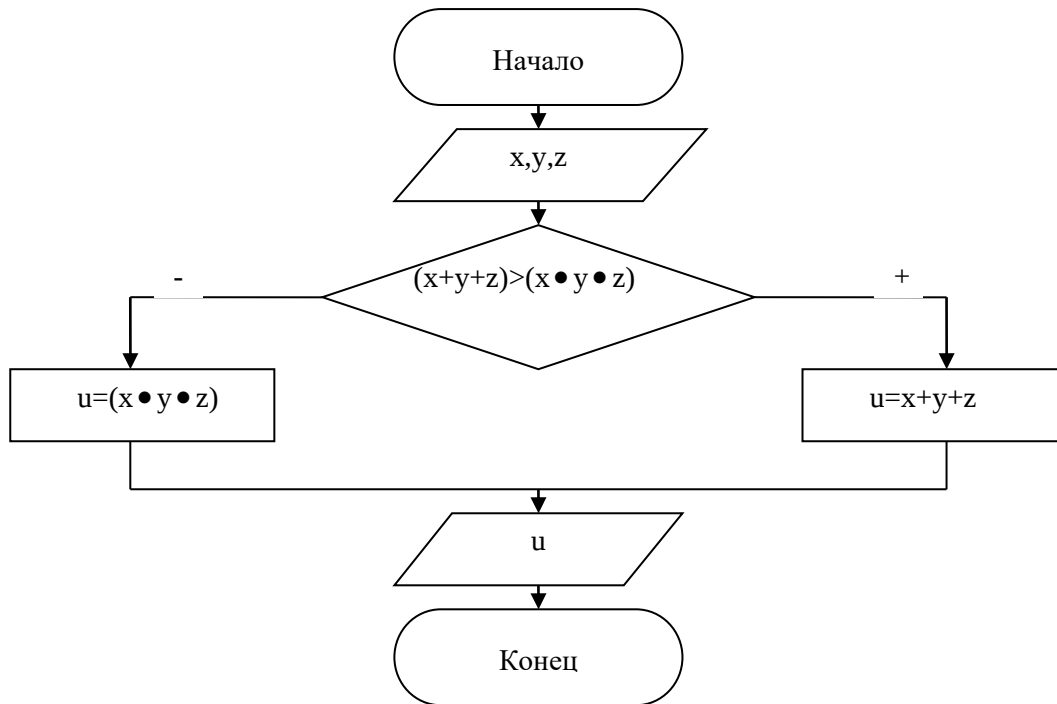
Объект	Свойство	Значение	Событие	Программный код
Form	Name Caption	Form2 'Нахождение максимума'		
label	Name Caption	Lblx 'x='		
label	Name Caption	Lbly 'y='		
label	Name Caption	Lblz 'z='		
label	Name Caption	Lblu ' '		
edit	Name Text	Edtx ' '		
edit	Name Text	Edty ' '		
edit	Name Text	Edtz ' '		
Button	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
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

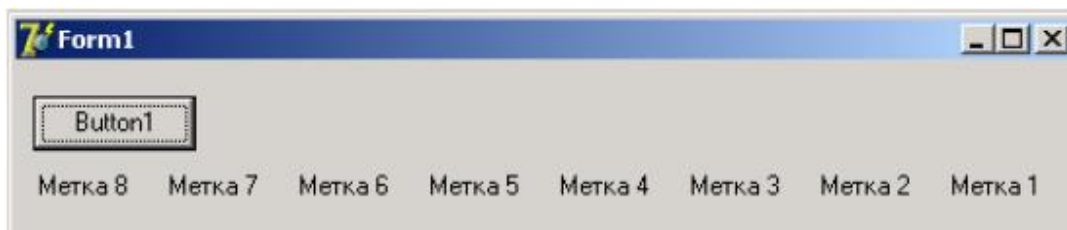
Итог работы: отчет, защита работы.

ЛР №26: Множественное наследование (2 часа)

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

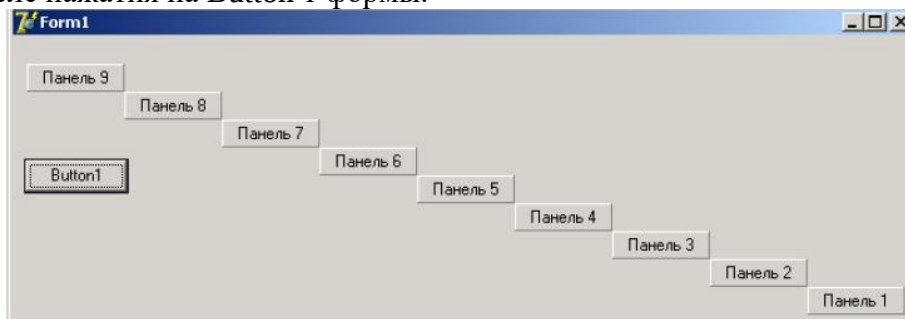
Задание 1. Реализовать метод, который будет размещать массив из 8 «новых» меток (TLabel) на форме в одну строку.

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.



ЛР №27: Шаблоны классов (4 часа)

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.

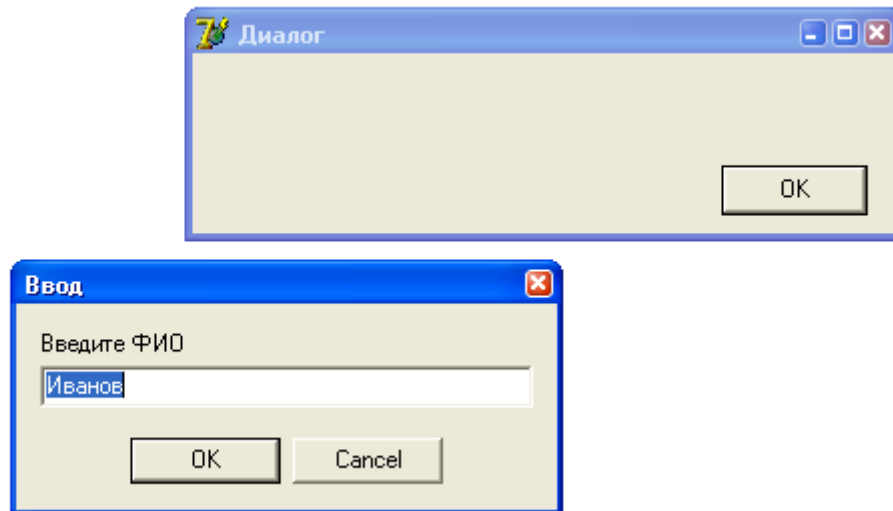
ЛР №28: Стандартная библиотека шаблонов (4 часа)

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

Задание 1. Создайте диалоговое окно для вывода карточки студента.

The screenshot shows a standard Windows-style dialog box with a blue title bar and a light gray background. The title bar text is 'Карточка студента'. Inside the window, there are five text input fields arranged vertically, each preceded by a label: 'Фамилия', 'Имя', 'Отчество', 'Специальность', and 'группа'. The 'Специальность' field is wider than the others. At the bottom of the window, there are three buttons: 'заполнить', 'очистить', and 'Выход'.

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

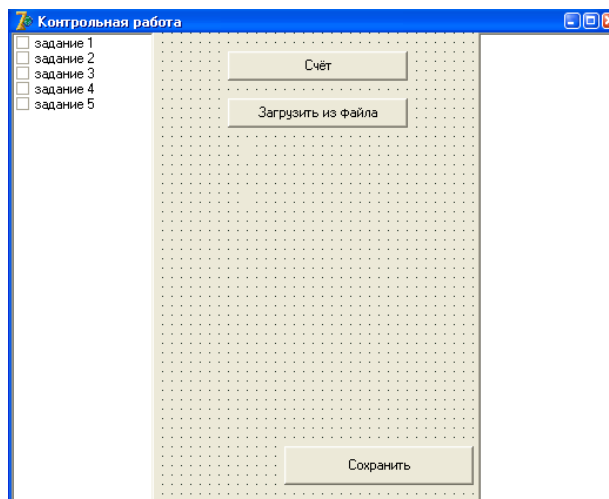


Итог работы: отчет, защита работы.

ЛР №29: Создание простейшей программы на языке Visual C++ + ЛР №30: Условные операторы. Вычисление значений функции, заданной условно (2+2 часа)

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

Задание: Реализовать задачу «Контрольная работа» найти ошибку в программном коде и пояснить, на что она влияла.



```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, CheckLst;  
type  
  TForm1 = class(TForm)  
    BtnOK: TButton;  
    Btnf: TButton;  
    Lblp: TLabel;  
    Lblm: TLabel;  
    cl: TCheckBoxList;  
  end;  
end;
```

```

    c: TCheckListBox;
    Button1: TButton;
    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 i,k,m,n,l,p:integer;
    begin
        n:=cl.ItemIndex;
        //showmessage(inttostr(n));
        case n of
            0: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
            1: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
            2: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
            3: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;

```

```

4: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
5: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
6: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
7: if cl.Checked[n]=true then c.Header[c.Items.Add(cl.Items.Strings[n])]:=true;
end;
case n of
0: if cl.Checked[n]=false then c.Items.Delete(c.Items.IndexOf(cl.Items.Strings[n]));
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;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
c.Items.SaveToFile('результат.txt');
end;
end.

```

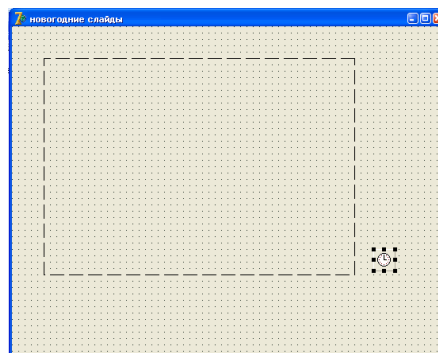
Итог работы: отчет, защита работы.

ЛР №31: Циклический алгоритм. Табулирование функции и поиск экстремумов (2 часа)

Создание проекта с использованием компонентов системы меню.

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

Задание 1. Создать проект «анимация» с использованием компонентов системы меню: Timer1 и Image1.



```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, jpeg;
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>14 then i:=1;
  img1.picture.loadfromfile(getcurrentdir+'\pics\'+inttostr(i)+'.jpg');
  inc(i);
end;
end.

```

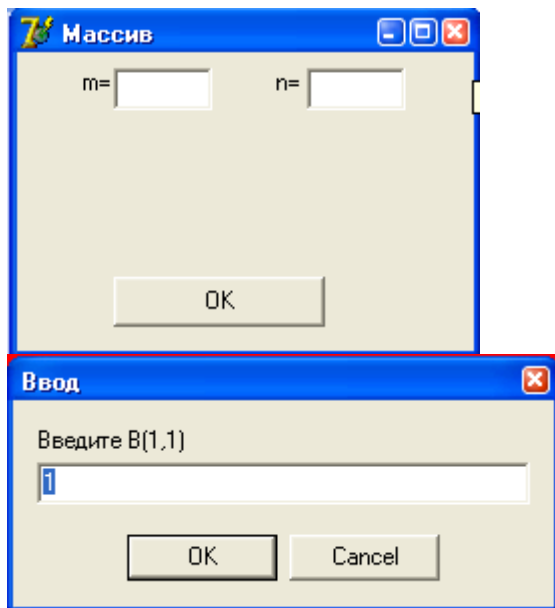
Задание 2. Создать проект «анимация» по различной тематике с использованием компонентов системы меню: Timer1 и Image1.

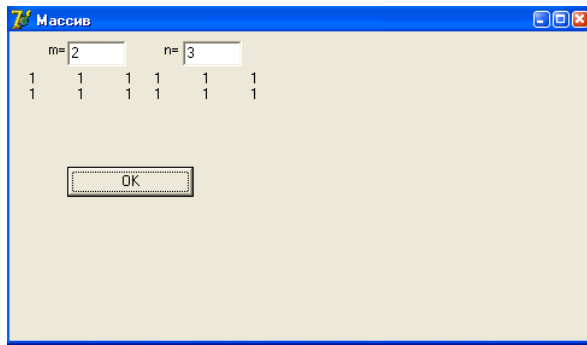
Итог работы: отчет, защита работы.

ЛР №32: Построение графика функции на промежутке с определенным шагом + ЛР №33: Понятие одномерного массива. Селективная обработка элементов массива (2+2 часа)

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;
    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
d(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[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.

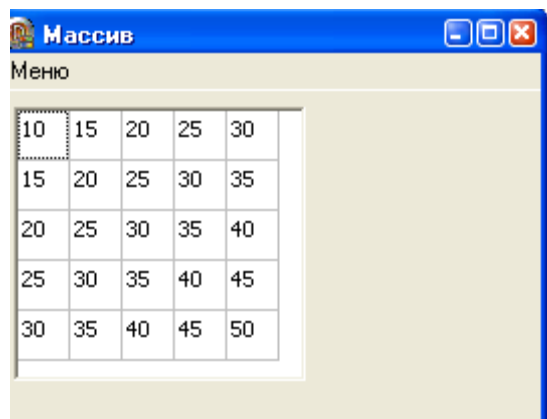
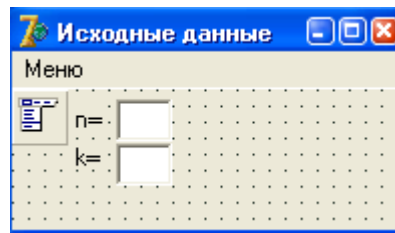
```

Итог работы: отчет, защита работы.

ЛР №34: Понятие матрицы. Селективная обработка элементов строк, столбцов и диагоналей матрицы (2 часа)

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

Задание: Реализовать задачу 5 из практической работы №10.



```

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 frmmass.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

```

Итог работы: отчет, защита работы.

ЛР №35: Разработка приложения для обработки базы данных (4 часа)

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

Задание: Разработать многооконое приложение из предложенных вариантов:

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

Итог работы: отчет, защита работы.

Информационные источники

Печатные издания:

1. Иванченко А.Н. Теоретические основы разработки и реализации языков программирования. (Бакалавриат). Учебное пособие Кнорус, 2020.

Электронные издания (электронные ресурсы):

1. Российское образование. Федеральный портал. [Электронный ресурс]. Режим доступа: <http://www.edu.ru>

2. Единая коллекция цифровых образовательных ресурсов. [Электронный ресурс]. Режим доступа: <http://school-collection.edu.ru>

3. Социальная сеть работников образования. [Электронный ресурс]. Режим доступа: <http://nsportal.ru>

4. Электронная информационная образовательная среда. [Электронный ресурс]. Режим доступа: <http://edu.dvgups.ru>

5. Открытый урок. Первое сентября. [Электронный ресурс]. Режим доступа: <http://festival.1september.ru>

6. Педагогическое сообщество «урок.рф». [Электронный ресурс]. Режим доступа: <http://урок.рф>

7. Инфоурок. Ведущий образовательный портал России. [Электронный ресурс]. Режим доступа: <https://infourok.ru>

8. Профобразование. [Электронный ресурс]. Режим доступа: <http://проф-обр.рф>

9. Учебно-методический кабинет. [Электронный ресурс]. Режим доступа: <http://ped-kopilka.ru>

10. Единое окно доступа к образовательным ресурсам. [Электронный ресурс]. Режим доступа: <http://window.edu.ru>

11. Электронное обучение, компьютерная филология. Информационные технологии. [Электронный ресурс]. Режим доступа: <http://it.lang-study.com/>

Дополнительные источники:

1. ЭБС «Юрайт»: Советов, Б. Я. Информационные технологии: учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский. — 7-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 327 с. — (Профессиональное образование). — ISBN 978-5-534-06399-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450686>

2. Гохберг Г.С., Зафиевский А.В., Короткин А.А. Информационные технологии: Издание: учебник для среднего профессионального образования / Гохберг Г.С., Зафиевский А.В., Короткин А.А — 3-е изд. стер. М.: Изд.центр «Академия», 2020 — 240 с.

3. Михеева Е.В. Информационные технологии в профессиональной деятельности. Технические специальности: учебник для студ. Учреждений сред. проф. образования / Е.В. Михеева, О.И. Титова. — 2-е изд., стер. — М.: Издательский центр «Академия», 2015. — 416 с. Режим доступа: <http://www.academia-moscow.ru/reader/?id=168074&demo=Y>.

4. Михеева Е. В. Практикум по информационным технологиям в профессиональной деятельности. — М.: Изд.центр «Академия», 2014. — 256 с. Режим доступа: <http://www.academia-moscow.ru/reader/?id=106719>