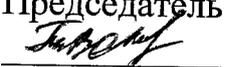
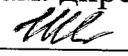


**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ  
«ЧЕРЕМХОВСКИЙ ГОРНОТЕХНИЧЕСКИЙ КОЛЛЕДЖ  
ИМ. М.И. ЩАДОВА»**

Рассмотрено на  
заседании ЦК  
«02» 06 2020 г.  
Протокол № 10  
Председатель  
 Т.В.Окладникова

УТВЕРЖДАЮ  
Зам. директора по УР  
 Н.А. Шаманова  
«03» 06 2020 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
для выполнения  
практических (лабораторных) работ студентов 2 курса  
по  
**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

	<b>СТР.</b>
1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ РАБОТ	5
3. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ РАБОТ	6
4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРАКТИЧЕСКИХ РАБОТ	18
5. ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ, ВНЕСЁННЫХ В МЕТОДИЧЕСКИЕ УКАЗАНИЯ	20

## 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»** с учетом (помещение кабинета информатики должны удовлетворять требованиям санитарно-эпидемиологических правил и нормативов (СанПиН 2.4.2 № 178-02), и оснащено типовым оборудованием, указанным в настоящих требованиях, в том числе специализированной учебной мебелью и средствами обучения, достаточными для выполнения требований к уровню подготовки обучающихся<sup>1</sup>) 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 техники безопасности;
- ответить на теоретические вопросы преподавателя.

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

---

<sup>1</sup> См. Письмо Минобрнауки РФ от 24 ноября 2011 г. N МД-1552/03 «Об оснащении общеобразовательных учреждений учебным и учебно-лабораторным оборудованием»

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 (лабораторные) работы по дисциплине **«основы алгоритмизации и программирования»** отводится 10 часа.

## 2. ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ РАБОТ

№ п/п	Название практической работы (указать раздел программы, если это необходимо)	Количество часов
1.	Составление блок-схем алгоритмов.	1
2.	Составление программ линейной структуры.	1
3.	Составление программ разветвляющейся структуры	1
4.	Составление программ циклической структуры.	1
5.	Обработка одномерных и двумерных массивов.	1
6.	Работа со строковыми переменными. Работа с данными типа множество.	1
7.	Организация и использование процедур и функций	2
8.	Программирование модуля. Создание библиотеки подпрограмм	1
9.	Создание экземпляров класса. Объявление класса.	1
<b>Итого</b>		<b>10</b>

### 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.

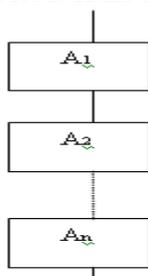
$$h = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a}$$

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.



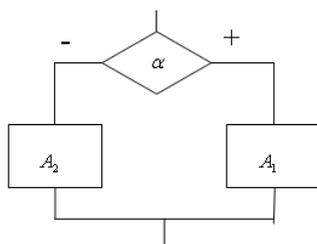
$A_i$  - функциональный блок

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

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.

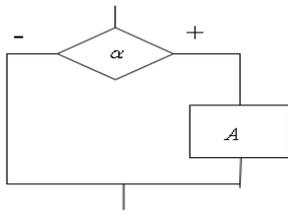
$A_i$  - функциональный блок



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

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

Полная разветвляющаяся (логическая) структура



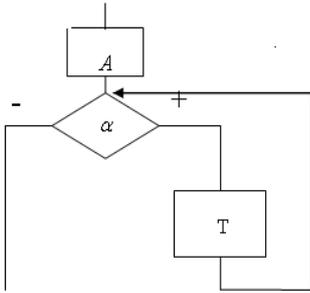
A - функциональный блок

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

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»)

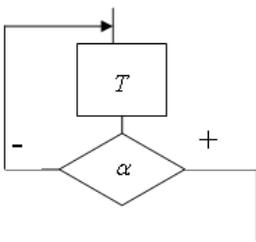


A - функциональный блок, начальных присвоений.

T - функциональный блок, тело цикла.

alpha - логическое условие

б) цикл с постусловием («до»)



A - функциональный блок, начальных присвоений.

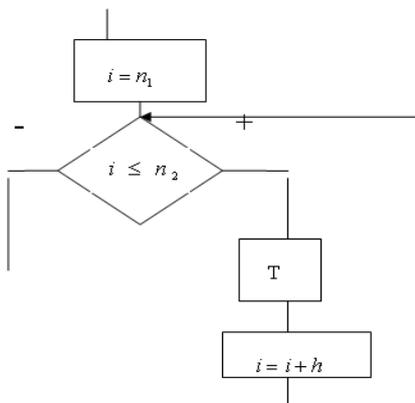
T - функциональный блок, тело цикла.

alpha - логическое условие

в) цикл с параметром, счетчик («для»)

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

1 форма



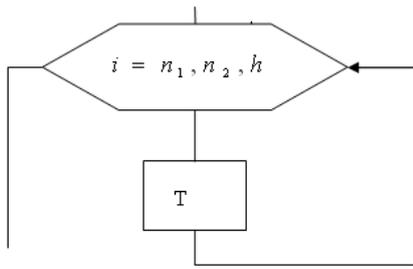
i - параметр цикла.

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

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

T - функциональный блок.

2 форма



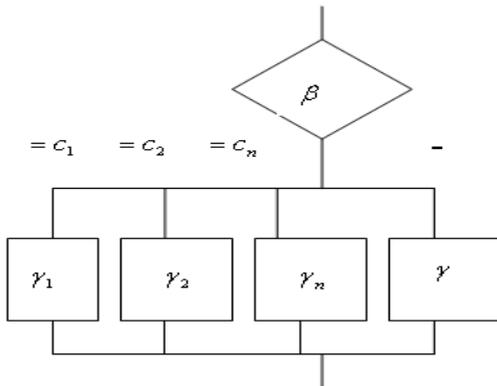
$i$ -параметр цикла.

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

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

$T$  - функциональный блок.

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



$\beta$  - выражение значение, которое анализируется.

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

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

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

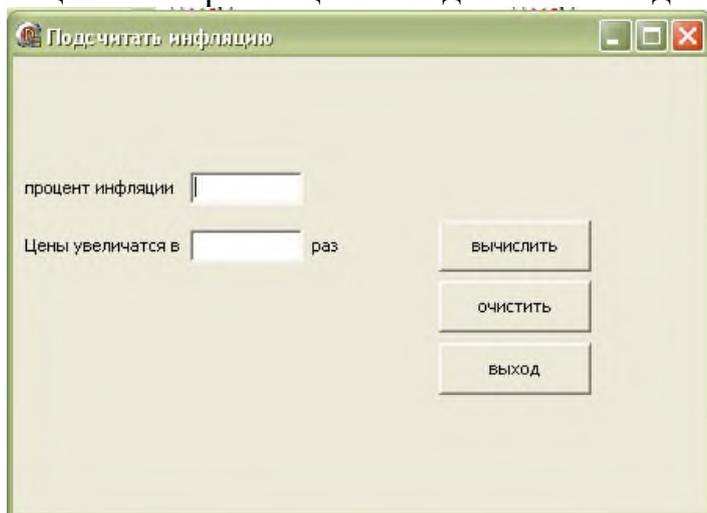
**Итог работы:** тетрадь, защита работы.

## Практическая работа № 2

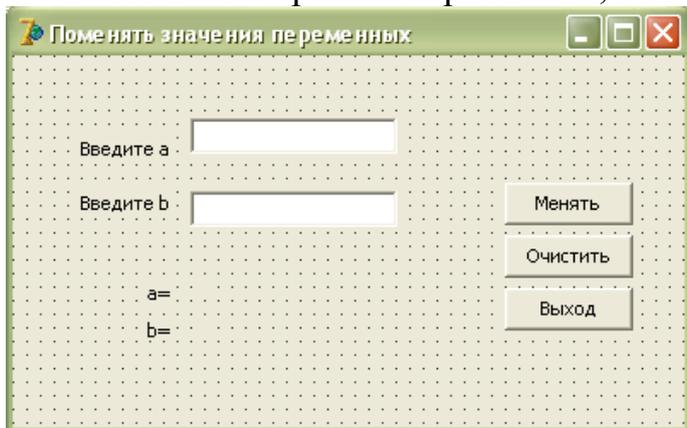
### Составление программ линейной структуры.

**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:**

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.

### Практическая работа № 3 Составление программ разветвляющейся структуры

**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.

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

**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.

### **Практическая работа № 5** **Обработка одномерных и двумерных массивов.**

**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).

**Итог работы:** отчет, защита работы.

### **Практическая работа № 6** **Работа со строковыми переменными. Работа с данными типа множество.**

**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.

## Практическая работа № 7

### Организация и использование процедур и функций.

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

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

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

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

```
procedure Имя_процедуры((*параметры*));  
begin  
  //Код процедуры;  
end;
```

Программный код :

```
var  
  Form1: TForm1;  
  A, B, Summa: Integer;  
  procedure Sum(A, B: Integer);  
implementation  
  {$R *.dfm}  
  procedure TForm1.Button1Click(Sender: TObject);  
  begin  
    A:=StrToInt(Edit1.Text);  
    B:=StrToInt(Edit2.Text);
```

```

Sum(A, B);
Caption:=IntToStr(Summa);
end;
procedure Sum(A, B: Integer);
begin
  Summa:=A+B;
end;

```

**Задание 2.** 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 № 16

### Программирование модуля. Создание библиотеки подпрограмм.

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

**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.

```

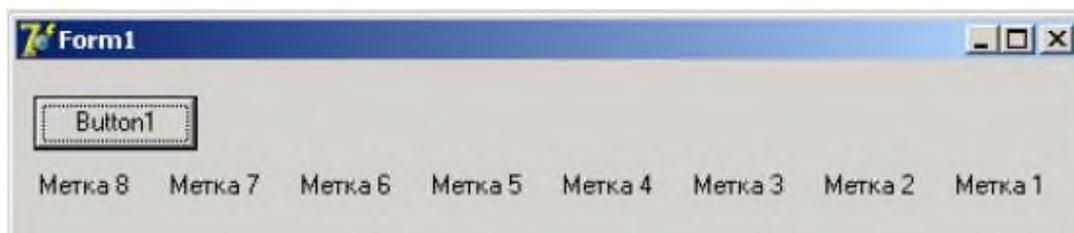
**Итог работы:** отчет, защита работы

## Практическая работа № 9 Создание экземпляров класса. Объявление класса.

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

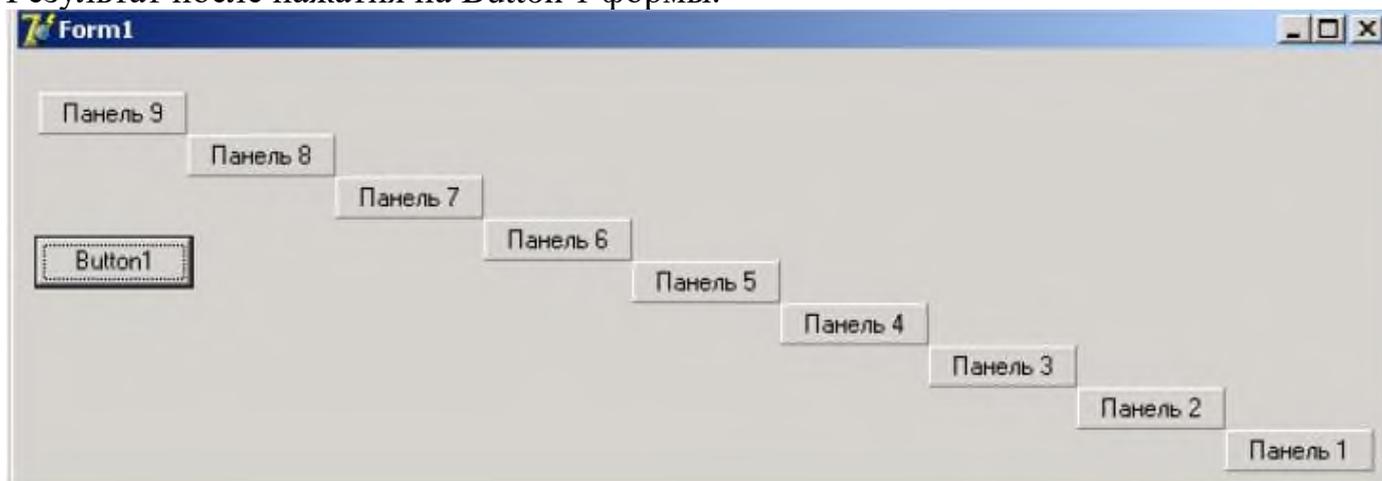
**Задание 1.** Реализовать метод, который будет размещать массив из 8 «новых» меток (TLabel) на форме в одну строку.

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.



## **4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРАКТИЧЕСКИХ РАБОТ**

### **4.1 Печатные изделия:**

#### **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.

#### **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**

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