

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
АГРАРНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ
НА АЛГОРИТМИЧЕСКОМ ЯЗЫКЕ
Object Pascal**

*Лабораторный практикум по дисциплине
«Информатика»*

Минск
БГАТУ
2010

УДК 004.43 (07)
ББК 32.97-018я7
Р47

*Рекомендовано научно-методическим советом агроэнергетического
факультета БГАТУ.
Протокол № 6 от 9 февраля 2010 г.*

Авторы: кандидат технических наук, доцент *Н. В. Исаеня*,
кандидат технических наук, доцент *Ю. Н. Силкович*,
кандидат технических наук, доцент *А. И. Шакирин*,
старший преподаватель *О. М. Львова*

Рецензенты:

доктор физико-математических наук, профессор кафедры
«Вычислительные методы и программирование» УО «БГУИР»
А. В. Аксенчик;

кандидат технических наук, доцент кафедры «Информационные
технологии автоматизированных систем» УО «БГУИР» *А. М. Севернев*

**Решение прикладных задач на алгоритмическом языке
Object Pascal** : лабораторный практикум / Н. В. Исаеня
[и др.] – Минск: БГАТУ, 2010. – 176 с.
ISBN 978-985-519-238-2.

УДК 004.43 (07)
ББК 32.97-018я7

ISBN 978-985-519-238-2

© БГАТУ, 2010

ВВЕДЕНИЕ

Лабораторный практикум предназначен для приобретения и закрепления навыков разработки алгоритмов и решения прикладных задач при выполнении лабораторных работ по дисциплине «Информатика» студентами агроинженерного, агроэнергетического факультетов и факультета «Технический сервис в АПК» БГАТУ.

Практикум содержит методический материал для выполнения 8 лабораторных работ по следующим темам:

- 1) решение прикладных задач, основанных на использовании линейных алгоритмов;
- 2) решение прикладных задач, основанных на использовании разветвляющихся алгоритмов;
- 3) решение прикладных задач, основанных на использовании циклических алгоритмов с известным количеством повторений цикла;
- 4) решение прикладных задач, основанных на использовании циклических алгоритмов с неизвестным количеством повторений цикла;
- 5) решение прикладных задач, основанных на использовании алгоритмов с одномерными массивами;
- 6) решение прикладных задач, основанных на использовании алгоритмов с двумерными массивами;
- 7) решение прикладных задач, основанных на использовании алгоритмов численных методов вычисления интегралов;
- 8) решение прикладных задач, основанных на использовании алгоритмов численных методов решения нелинейных уравнений.

Студентам предлагаются задания разного уровня сложности, что позволяет преподавателю дифференцированно оценить степень подготовки каждого студента.

По каждой лабораторной работе студент обязан оформить и защитить отчет.

Содержание отчета:

- 1) номер и название лабораторной работы;
- 2) цель работы;
- 3) полная формулировка индивидуального задания;
- 4) схемы алгоритмов, программы и полученные результаты.

ЛАБОРАТОРНАЯ РАБОТА №1

Решение прикладных задач, основанных на использовании линейных алгоритмов

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании линейных алгоритмов, в визуальной среде *Delphi*.

Порядок выполнения лабораторной работы

1. Изучить раздел «Визуальная среда *Delphi*».
2. Выполнить пример создания приложения. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

1.1. Визуальная среда *Delphi*

Для запуска *Delphi* используется последовательность команд: **Пуск** → **Программы (Все программы)** → **Borland Delphi 7** → **Delphi 7**.

После запуска *Delphi* на экране появляются несколько окон, показанных на рисунке 1:

- главное окно *Delphi* с именем проекта **Project1**;
- окно формы — **Form1**;
- окно редактора свойств объектов — **Object Inspector**;
- окно просмотра списка объектов — **Object TreeView**;
- окно редактора кода — **Unit1**.

Главное окно (1) всегда присутствует на экране и предназначено для управления процессом создания приложения.

Главное меню (2) содержит все необходимые средства для управления проектом.

Пиктограммы главного меню (3) облегчают доступ к наиболее часто применяемым командам.

Палитра Компонентов (6) обеспечивает доступ к набору библиотечных процедур среды *Delphi*, которые описывают некоторый элемент (компонент), размещенный Вами в окне Формы. Каждый компонент имеет определенный набор свойств, которые Вы можете выбирать и изменять по своему усмотрению, например, заголовок окна, надпись на кнопке, размер, цвет и тип шрифта и др.

Окно Инспектора Объектов (5) (Object Inspector) – предназначено для изменения свойств выбранных компонентов и состоит из

двух страниц: **Properties** (Свойства) и **Events** (События). Страница *Properties* предназначена для изменения необходимых свойств компонента. Страница *Events* (События) – для определения реакции компонента на то или иное событие (например, щелчок кнопки мыши).

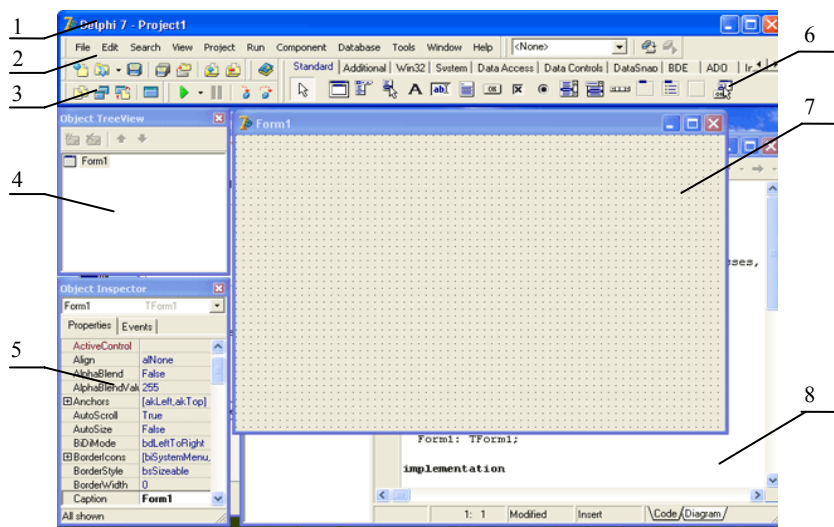


Рисунок 1. Вид экрана после запуска *Delphi*:

- 1 – главное окно; 2 – главное меню; 3 – пиктограммы главного меню;
4 – окно просмотра дерева объектов; 5 – окно Инспектора Объектов;
6 – Палитра Компонентов; 7 – окно пустой Формы;
8 – окно Редактора Кода

Окно Формы (7) (Form1) представляет собой интерфейс проектируемого Windows-приложения. В это окно на этапе проектирования приложения помещаются необходимые компоненты, которые разработчик берет из Палитры Компонентов. Каждой Форме проекта соответствует модуль (**Unit**), текст которого на языке *Object Pascal* размещается в окне Редактора Кода.

Окно Редактора Кода (8) (Unit1) предназначено для просмотра, создания и редактирования текстов модулей проекта. При первоначальной загрузке в окне Редактора Кода находится текст модуля, содержащий минимальный набор операторов для нормального функционирования пустой Формы в качестве Windows-приложения. При размещении некоторого компонента в окне Формы текст модуля автоматически дополняется необходимыми операторами.

Обо всех происходящих в системе событиях (создание Формы, нажатие кнопки мыши или клавиатуры и т.д.), операционная система Windows информирует окна *Delphi* путем послышки соответствующих сообщений. Среда *Delphi* принимает и обрабатывает сообщения с помощью обработчиков событий (например, щелчок кнопки мыши – событие **OnClick**, создание Формы – **OnCreate**).

При разработке приложения пользователь с помощью *Delphi* создает интерфейс в соответствии с разработанным дизайном, пишет обработчики событий на языке *Object Pascal (OP)*. Язык программирования *OP* – это основа *Delphi*.

1.2. Основы программирования на языке *Object Pascal*

Алгоритмический язык *OP* для персональных компьютеров (ПК) разработан фирмой *Borland* и широко используется как для обучения основам программирования, так и для создания программных продуктов профессионального назначения.

ПК выполняет действия с данными: читает данные из памяти, производит с ними нужные действия, результаты вновь заносит в память. Какие действия и с какими данными выполнять, куда записывать результат – указывается в программе.

Данные могут представлять собой числа, буквы (символы), текст – в зависимости от решаемой задачи.

Одно и тоже число на *OP* можно записать различными способами (таблица 1).

Таблица 1 – Запись чисел на языке *OP*

Математическая запись	Запись на <i>OP</i>		
	Целое	Действительное (вещественное)	
		В форме с фиксированной точкой	В форме с плавающей точкой
25	25	25.0	2.5E01
-892	-892	-892.0	8.92E02
0	0	0.0	0.0E00
-4,3	-	-4.3	-4.3E00
$-3,56 \cdot 10^8$	-	-356000000.0	-3.56E08
$27,84 \cdot 10^{-6}$	-	0.00002784	27.84E-06

Все данные в *OP* относятся к определенному типу. Тип данных показывает, какие значения принимают данные, какие операции

можно с ними выполнять и как данные представляются в памяти ПК. Типы данных указываются в специальном разделе программы, который называется **раздел объявлений (описаний)**.

В программе обычно используется много данных. Чтобы указать, с какими данными требуется выполнить действия, данные обозначаются **именами**. Имя может содержать буквы и цифры, не должно разделяться промежутками и должно начинаться с буквы, например: *Delta, Sum5, MinAk*.

Иногда в программах на ОР используются данные, которые получают значение в начале программы и не изменяют своего значения в процессе ее выполнения. Такие данные называются **константами**. Константы указываются в разделе объявлений после слова **Const**, например:

Const

```
MaxNumber = 50;           {целая константа}
Delta = 1.7E-8;          {действительная константа}
Stroka = 'Pascal';       {строковая константа}
```

Константу π указывать в разделе объявлений не нужно, в программе в качестве ее используется имя Pi.

Данные, значения которых могут изменяться при выполнении программы, называются **переменными**. Типы **всех переменных, используемых в программе**, указываются в разделе объявлений после слова **Var**, например:

Var

```
Index:integer;           {переменная целого типа}
Sum,Min:extended;       {переменные действительного типа}
St1,St2:string;         {переменные строкового типа}
```

Эти типы данных чаще всего используются в ОР.

Все данные, с которыми работает программа, хранятся в соответствующих полях памяти ПК. Эти поля выделяются при трансляции программы в соответствии с типом данных и закрепляются за данными в течение всего времени выполнения программы. Так происходит при статическом распределении памяти, динамическое выделение памяти здесь не рассматривается. Доступ к содержимому поля памяти осуществляется с помощью имени соответствующего данного.

Для изменения содержимого переменной **используется оператор присваивания**. Оператор содержит операцию присваивания, которая

обозначается двумя символами **:=** (двоеточие и равенство). Слева от операции присваивания указывается имя той переменной, которой нужно присвоить новое значение, справа – значение, которое следует присвоить указанной переменной. Один оператор отделяется от другого с помощью символа **;** (точка с запятой), например:

```
Index := 5;
Min := Max;
```

Действия, которые должен выполнить ПК, указываются в выражениях. **Выражение** задает порядок выполнения действий над данными и состоит из операндов, круглых скобок и знаков операций. **Операнды** – это константы, переменные и функции языка ОР. Круглые скобки используются для изменения последовательности выполнения операций в выражении. **Операции** – это действия, выполняемые над операндами. В арифметических выражениях используются арифметические операции. В таблице 2 представлены основные арифметические операции ОР.

Таблица 2 – Арифметические операции

Операция	Действие	Тип операндов	Тип результата
+	Сложение	Целый, действительный	Целый, действительный
-	Вычитание	Целый, действительный	Целый, действительный
*	Умножение	Целый, действительный	Целый, действительный
/	Деление	Целый, действительный	Целый, действительный
Div	Целочисленное деление	Целый	Целый
Mod	Остаток от деления	Целый	Целый

Результат вычисления выражения заносится в переменную с помощью оператора присваивания, например:

```
z := (Summa+5) *Adr-RezInter / (Znam-2.6);
```

Рекомендуется следить за тем, чтобы тип переменной совпадал с типом выражения. При несовпадении типов ПК попытается преобразовать тип результата вычисления арифметического выражения к типу переменной. Если переменная действительного типа

(extended), а выражение целого типа (integer) – оператор присваивания выполнится успешно. Если переменная целого типа (integer), а выражение действительного типа (extended) – компьютер выведет сообщение об ошибке преобразования типа.

Пример 1. Записать оператор для вычисления выражения

$$Y = a^2 + b^2.$$

В языке ОР отсутствует операция возведения в степень, поэтому вместо нее чаще всего используют многократное умножение операндов на свое собственное значение:

```
y:=a*a+b*b*b;
```

Пример 2. В переменных *m* и *n* находятся численные значения. Перенести содержимое переменной *m* в переменную *n*, а содержимое переменной *n* наоборот – в переменную *m*.

1-й вариант

```
DopPer:= m;
m:=n;
n:=DopPer;
```

2-й вариант

```
DopPer:=n;
n:=m;
m:=DopPer;
```

Дополнительная переменная **DopPer** потребовалась для того, чтобы сохранить значение одной из переменных перед тем, как оно будет уничтожено значением второй переменной.

Пример 3. В переменной **DvaZnaka** находится двухзначное целое число. Написать последовательность операторов, которые вычисляют сумму цифр этого числа.

С использованием арифметических операций над целочисленными операндами `div` и `mod` алгоритм решения может быть таким. Первая (старшая) цифра двухзначного числа – это количество десятков. Ее можно определить, разделив заданное число на 10:

```
Cifra1 := DvaZnaka div 10;
```

Остаток от деления заданного числа на 10 – это количество единиц, т.е. вторая (младшая) цифра двухзначного числа:

```
Cifra2 := DvaZnaka mod 10;
```

Таким образом, вычисление суммы цифр числа **DvaZnaka** обеспечит последовательность операторов:

```
Cifra1 := DvaZnaka div 10;
Cifra2 := DvaZnaka mod 10;
SummaCifr := Cifra1 + Cifra2;
```

В выражениях могут использоваться функции ОР. Основные функции языка ОР приведены в таблице 3.

Таблица 3 – Стандартные функции языка Object Pascal

Математическая запись	Действие	Запись на ОР	Пример
$ x $	Вычисление абсолютного значения x	Abs(x)	Abs(-4) = 4
x^2	Вычисление квадрата x	Sqr(x)	Sqr(5) = 25
\sqrt{x}	Вычисление квадратного корня из x	Sqrt(x)	Sqrt(25) = 5
e^x	Вычисление экспоненты x	Exp(x)	Exp(1) = 2,7183...
Ln x	Вычисление натурального логарифма	Ln(x)	Ln(Exp(1)) = 1
Sinx	Вычисление синуса x	Sin(x)	Sin(Pi/6) = 0,5
Cos x	Вычисление косинуса x	Cos(x)	Cos(Pi/3) = 0,5
Arctg x	Вычисление арктангенса x	Arctan(x)	Arctan(1) = pi/4 = 0,7854...
	Округление x до целого	Round(x)	Round(2.5) = 3
	Вычисление целой части x , результат – <i>действительный</i>	Int(x)	Int(2.5) = 2,0
	Вычисление целой части x , результат <i>целый</i>	Trunc(x)	Trunc(2.5) = 2
	Вычисление дробной части x	Frac(x)	Frac(2.5) = 0,5
	<i>True</i> , если x – нечетный; <i>False</i> , если x – четный	Odd(x)	Odd(3) = true

Если операнды выражений содержат другие функции, необходимо использовать известные математические соотношения, например:

- для вычисления логарифма с произвольным основанием *a* используется соотношение:

$$\log_a(x) = \frac{\ln(x)}{\ln(a)};$$

- для возведения положительного числа x в действительную степень a используется соотношение:

$$x^a = e^{a \ln x};$$

- для вычисления других тригонометрических функций используются известные соотношения из справочной литературы.

2. Пример создания приложения

Разработать *Windows*-приложение для вычисления выражения

$$u = \operatorname{tg}^5(\sqrt{x} - y^3) + e^{y/z} \cdot \sin(z^2 + \pi/3) \quad (1)$$

Проверить правильность вычисления выражения (1) при следующих значениях исходных данных: $x = 0,4$; $y = 8,75 \times 10^{-2}$; $z = 19,63$. Результат: $u = 0,242$.

Этап 1. Создание интерфейса

Для удобства работы с приложением создается интерфейс — панель с элементами (надписями, полями ввода/вывода, кнопками), с помощью которых пользователь вводит исходные данные, запускает приложение на выполнение. Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке 2.

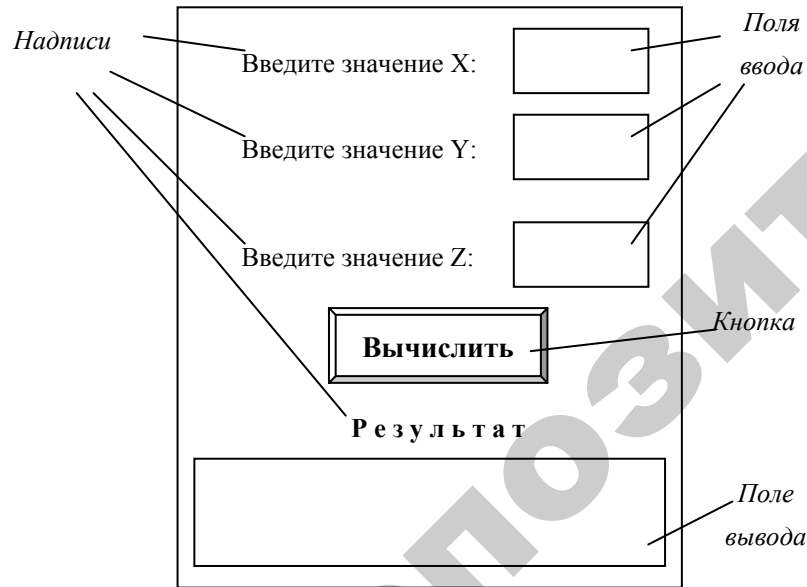


Рисунок 2. Панель интерфейса приложения

Этап 2. Разработка сценария работы приложения

Последовательность действий пользователя с созданным интерфейсом называется *сценарием* работы приложения. Для разработкиваемого приложения сценарий может быть таким:

- 1) щелчок мышью в поле ввода X , после чего ввод с клавиатуры значения исходного данного X ;
- 2) щелчок мышью в поле ввода Y , после чего ввод с клавиатуры значения исходного данного Y ;
- 3) щелчок мышью в поле ввода Z , после чего ввод с клавиатуры значения исходного данного Z ;
- 4) щелчок на кнопке **Вычислить**.

Этап 3. Разработка алгоритма

Для уменьшения количества возможных ошибок при записи сложных формул желательно использовать промежуточные вычисления. Обозначим отдельные части формулы (1) следующим образом:

$$a = \sqrt{x} - y^3; \quad (2)$$

$$b = \frac{\sin(x)}{\cos(x)}; \quad (3)$$

$$c = b^5; \quad (4)$$

$$d = e^{y/z}; \quad (5)$$

$$f = \sin(z^2 + \pi/3). \quad (6)$$

Тогда с учетом принятых обозначений в формулах (2) – (6) формула (1) запишется так:

$$u = c + d \cdot f. \quad (7)$$

Изобразим последовательность этих действий в виде алгоритма, блок-схема которого приведена на рисунке 3.

Согласно разработанному алгоритму при нажатии кнопки **Вычислить** на панели интерфейса (рисунок 2) приложение должно ввести значения исходных данных X , Y , Z , выполнить вычисления по формулам (2) – (7) и вывести полученный результат U .

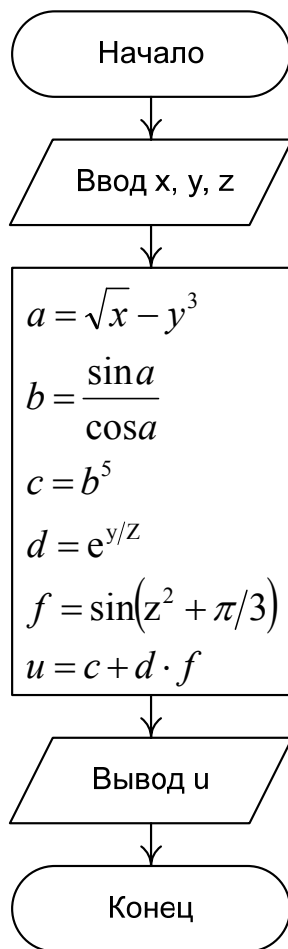


Рисунок 3. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Такие алгоритмы, в которых действия выполняются последовательно, одно за другим, в единственно возможном порядке следования называются **линейными**.

Методические указания по созданию приложения

В процессе проектирования *Delphi* создает несколько файлов – **проект**. Каждый проект следует хранить в отдельной, заранее созданной папке. Поэтому в своей рабочей папке создайте новую папку и назовите ее **LAB1**. Далее следуйте рекомендациям, описанным ниже.

Сохранение проекта

Запустите *Delphi* с помощью последовательности команд: **Пуск → Программы (Все программы) → Borland Delphi 7 → Delphi 7**.

Сохраните новый проект: **File → Save Project As... (Файл → Сохранить Проект как ...)**.

Сначала *Delphi* откроет панель диалога **Save Unit1 As (Сохранить Unit1 как)** для сохранения модуля проекта (рисунок 4). В этой панели найдите созданную папку **LAB1** и сохраните в ней модуль под именем **UnLab1 (Unit-модуль, Lab1 – лабораторная работа № 1)**. Обратите внимание на то, что *Delphi* по умолчанию присвоит этому файлу тип Delphi unit с расширением *.pas.

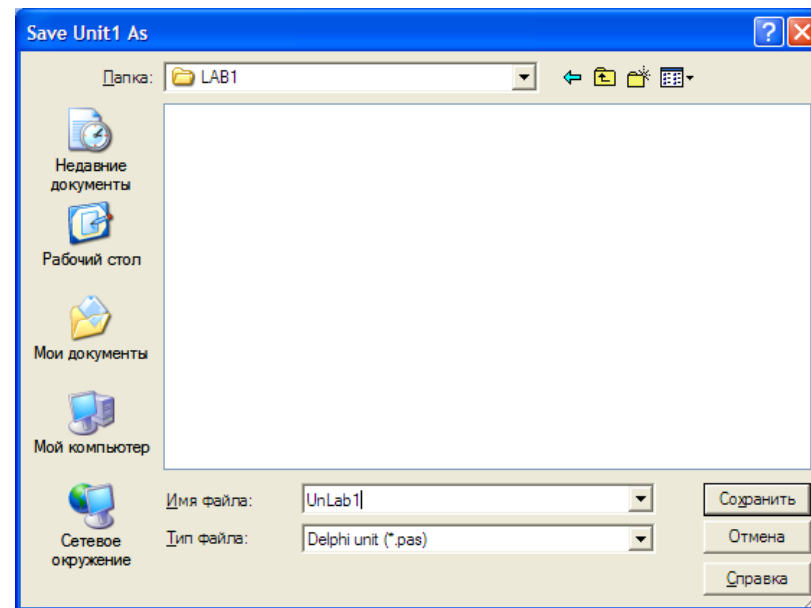


Рисунок 4. Окно **Save Unit1 As**

Затем откроется панель диалога **Save Project1 As (Сохранить Project1 как)**. Назовем проект **PrLab1** (Project-проект, Lab1-лабораторная работа №1) и сохраним его в этой же папке (рисунок 5). Здесь *Delphi* даст файлу тип Delphi project и расширение *.dpr. Убедитесь в том, что главное окно *Delphi* теперь называется **PrLab1**, окно главного файла проекта — **PrLab1.dpr**, а окно модуля проекта — **UnLab1.pas**.

Следует давать файлам осмысленные имена, например, **UnLab1** и **PrLab1** вместо Unit1 и Project1, предлагаемых *Delphi*.

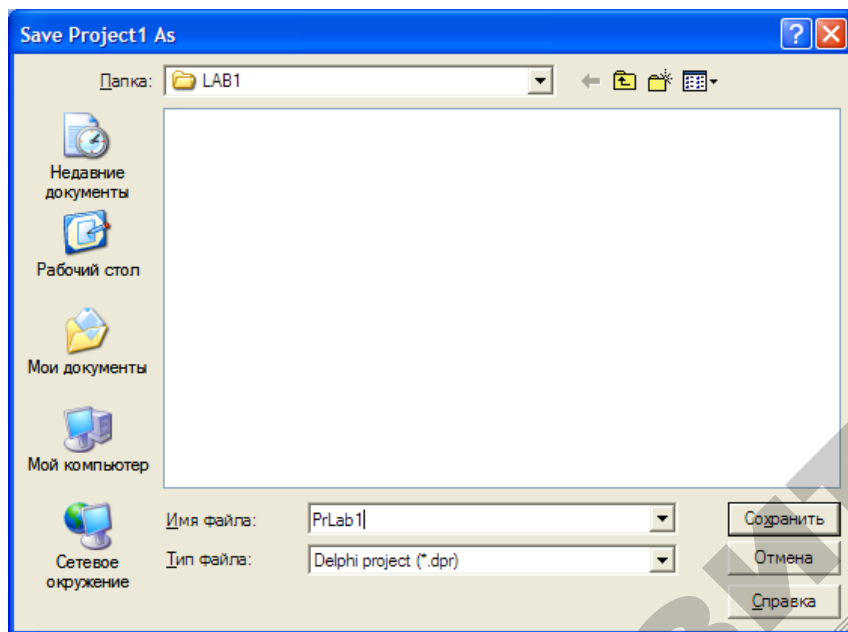



Рисунок 5. Окно Save Project1 As...

Чтобы избежать потери файлов проекта в аварийных ситуациях связанных, например, с выключением питания, зависании системы и т.д., рекомендуется периодически сохранять проект, используя пиктограмму  главного меню или опцию **Save All (Сохранить Все)** в меню **File (Файл)**.

Изменение заголовка Формы

Новая Форма имеет одинаковые имя (*Name*) и заголовок (*Caption*) — **FORM1**. Имя Формы менять не рекомендуется, т. к. оно используется в тексте модуля.

Для изменения заголовка Формы перейдите в окно **Object Inspector (Инспектор Объектов)** и на странице **Properties (Свойства)** найдите свойство **Caption (Заголовок)**. По умолчанию это свойство имеет значение Form1. Замените заголовок **Form1** на **Лаб1** (рисунок 6). Убедитесь, что одновременно изменился заголовок окна Формы.

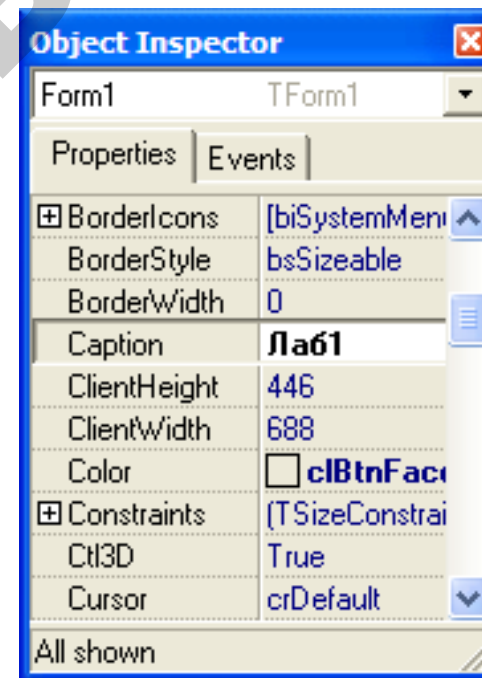


Рисунок 6. Изменение заголовка Формы в Инспекторе объектов.

Размещение компонентов на Форме

Форма предназначена для создания внешнего вида программы, ее интерфейса. Разместите на Форме необходимые компоненты, чтобы панель интерфейса выглядела так, как показано на рисунке 7.

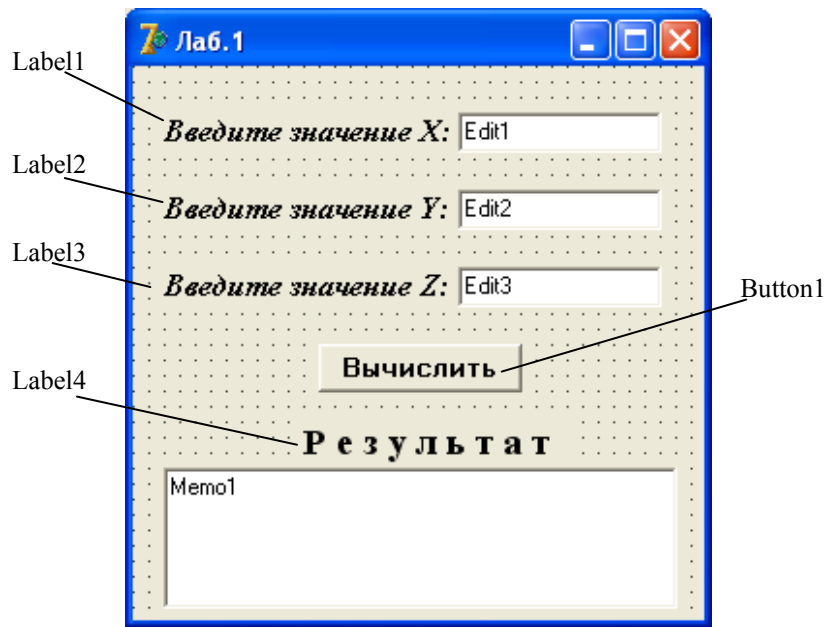


Рисунок 7. Размещение компонентов Label, Edit и Memo на Форме

Все компоненты выбираются со страницы **Standart** Палитры Компонентов.

Для нанесения надписей на Форму используется компонент **Label**. Выберите в Палитре Компонентов на странице **Standart** пиктограмму **A** компонента **Label** и щелкните на ней мышью. Затем в нужном месте Формы щелкните мышью — появится компонент **Label1**. В свойстве **Caption** Инспектора Объектов замените надпись **Label1** на **Введите значение X:**. В свойстве **Font** установите параметры шрифта. Аналогично нанесите на Форму остальные надписи. Щелкнув мышью на любом из размещенных компонентов, отрегулируйте его местоположение на Форме и размер.

Для создания полей ввода численных значений переменных **x**, **y** и **z** используем компонент **Edit**. Выберите в Палитре Компонентов на странице **Standart** пиктограмму **ab1** и разместите компоненты **Edit** в нужных местах Формы так же, как Вы это делали с компонентом **Label**.

Для вывода результатов используем компонент **Memo**. Выберите в Палитре Компонентов на странице **Standart** компонент **Memo** и поместите его на Форму. Откорректируйте его местоположение и размеры.

Создание процедуры обработки события нажатия кнопки **Вычислить**

Поместите на Форму кнопку, нажатие которой приведет к выполнению алгоритма, показанного на рисунке 2. Для этого выберите в Палитре Компонентов на странице **Standart** пиктограмму компонента **Button** и поместите его на Форму. В свойстве **Caption** Инспектора Объектов замените надпись на кнопке **Button1** надписью **Вычислить**. В свойстве **Font** подберите шрифт. Отрегулируйте положение и размер кнопки.

Для создания процедуры-обработчика поместите курсор на изображение кнопки и **дважды щелкните** левой кнопкой мыши. На экране появится текст модуля **UnLab1**, в котором Delphi автоматически создаст пустую процедуру, в которой будет всего три оператора:

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

end;
```

Замечание.** При создании этой процедуры необходимо очень аккуратно наводить курсор мыши на изображение кнопки. Небрежное действие приведет к созданию совершенно другой процедуры. Убедитесь в том, что пустая процедура, которую создала Delphi, называется именно **procedure Form1.Button1Click(Sender:TObject)!


Между операторами **begin** и **end** этой процедуры вставьте операторы, которые выполняют необходимые действия. Кроме этого, после заголовка процедуры необходимо добавить раздел объявления **var** в котором следует перечислить переменные, участвующие в вычислениях, и указать их тип:

```
procedure TForm1.Button1Click(Sender: TObject);
var
x, y, z, a, b, c, d, f, u: extended; // объявление локальных переменных
```

```

begin
x:=StrToFloat(Edit1.Text); // X присваивается содержимое
Edit1
y:=StrToFloat(Edit2.Text); // Y присваивается содержимое
Edit2
z:=StrToFloat(Edit3.Text); // Z присваивается содержимое
Edit3
a:=sqrt(x)-y*y*y; // Вычисляется выражение (2)
b:=sin(a)/cos(a); // Вычисляется выражение (3)
c:=b*b*b*b*b; // Вычисляется выражение (4)
d:=exp(y/z); // Вычисляется выражение (5)
f:=sin(z*z+pi/3); // Вычисляется выражение (6)
u:=c+d*f; // Вычисляется выражение (7)
// Вывод результата в Memo1
Memo1.Lines.Add(' U = '+FloatToStrF(u,ffixed,8,3));
end;

```

Переключение между окном Формы и окном Редактора Кода осуществляется клавишей **F12** или кнопкой главного меню с пиктограммой .

Пояснения к тексту процедуры Form1.Button1Click

Данные, с которыми работает компонент **Edit**, имеют строковый тип **String**. Поэтому в процедуре TForm1.Button1Click при присваивании содержимого полей **Edit1**, **Edit2**, **Edit3** переменным **x**, **y**, **z** с помощью функции **StrToFloat** осуществляется преобразование данных типа **String** в действительные значения с плавающей точкой типа **Extended**. Если необходимо работать с данными целого типа, используется функция преобразования **StrToInt**.

При выводе данных в **Memo1** используется метод **Add** свойства **Lines**, причем для преобразования данных из действительного значения в строковое и управления формой представления выводимого результата используется функция **FloatToStrF**.

Перечень основных функций преобразования данных, используемых при вводе и выводе данных, приведен в таблице 4.

Таблица 4 – Функции преобразования данных в ОР


Функция	Действие
Function StrToFloat (St: String):Extended;	Преобразует символы строки St в вещественное число
Function StrToInt (St: String):Integer;	Преобразует символы строки St в целое число
Function IntToStr (Value:Integer):String;	Преобразует целое значение Value в строку символов
Function FloatToStr (Value:Extended):String;	Преобразует вещественное значение Value в строку символов
Function FloatToStrF (Value:Extended; Format: TFloatFormat; Precision, Digits:Integer):String;	Преобразует вещественное значение Value в строку символов с учетом параметров <i>Precision</i> и <i>Digits</i> (см. таблицу 5)


Функция **FloatToStrF** позволяет управлять формой выводимых результатов. В зависимости от значения параметра **Format** результаты будут выводиться в форме с плавающей или фиксированной точкой. Правила использования всех параметров функции **FloatToStrF** приведены в таблице 5.


Таблица 5 – Параметры функции FloatToStrF

Значение Format	Описание
ffExponent	Формат с плавающей десятичной точкой и множителем E+(-)XX (умножить на 10 в степени +(-)XX). Precision задает общее количество десятичных цифр мантииссы. Digits – количество цифр в десятичном порядке XX. Число округляется с учетом первой отбрасываемой цифры: 3.1416E+00
ffFixed	Формат с фиксированной десятичной точкой. Precision задает общее количество десятичных цифр в представлении числа. Digits – количество цифр в дробной части. Число округляется с учетом первой отбрасываемой цифры: 3.14
ffGeneral	Универсальный формат, использующий наиболее удобную для чтения форму представления вещественного числа. Соответствует формату ffFixed, если количество цифр в целой части меньше или равно Precision, а само число – больше или равно 0.00001, в противном случае соответствует формату ffExponent: 3.1416

Работа с приложением

Перед запуском приложения обязательно сохраните проект (пиктограмма  главного меню) или **File→Save All (Файл→Сохранить Все)**.

Для запуска созданного приложения нажмите пиктограмму  главного меню или клавишу **F9**. При этом происходит компиляция модулей и, если нет ошибок, компоновка проекта и создание выполняемого файла **PrLab1.exe**. На экране появляется панель интерфейса приложения. Занесите в поля Edit1, Edit2, Edit3 значения переменных **x, y, z**, как показано на рисунке 8.

Щелкните мышью на кнопке **Вычислить** и в поле Memo1 появится результат. Измените исходные значения **x, y, z** и снова нажмите кнопку **Вычислить**. Убедитесь, что в поле Memo1 отображаются новые результаты. Завершить работу приложения можно нажатием кнопки  в правом верхнем углу панели интерфейса.

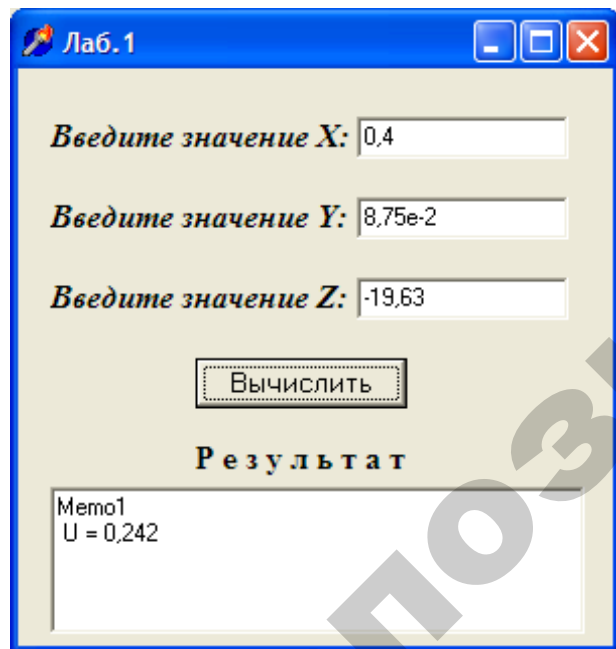


Рисунок 8. Панель интерфейса приложения с результатами

В случае нештатного функционирования приложения восстановить первоначальный режим работы с проектом можно выбором в меню **Run (Запуск)** опции **ProgramReset (Сброс программы)** или нажатием клавиши **Ctrl+F2**.

Вопросы для самоконтроля

1. Какие основные этапы необходимо выполнить при создании приложения?
2. Для чего предназначены основные окна среды Delphi?
3. Какие формы записи чисел используются в языке ОР?
4. Каким образом в разделе объявлений указываются типы данных?
5. Как изменить заголовок Формы?
6. Как можно создать заготовку процедуры-обработчика события нажатия кнопки?
7. Как закончить работу приложения?

Варианты индивидуальных заданий разного уровня

В каждом варианте имеются два задания. Задание 1 по уровню сложности соответствует примеру разработанного приложения. Задание 2 – повышенного уровня сложности и потребует дополнительных усилий.

1–2 уровни – выполните задание 1;

3 уровень – выполните задания 1, 2.

В задании 1 каждого варианта разработать *Windows*-приложение для вычисления выражения. Проверить правильность вычислений при заданных численных значениях исходных данных.

Вариант 1

$$1. t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0,5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right).$$

При $x = 14,26$; $y = -1,22$; $z = 3,5 \times 10^{-2}$ результат $t = 0,56485$.

2. Найти сумму цифр заданного четырехзначного числа.

Вариант 2

$$1. u = \frac{\sqrt[3]{8+|x-y|^2+1}}{x^2+y^2+2} - e^{|x-y|} (\operatorname{tg}^2 z + 1)^x.$$

При $x = -4,5$; $y = 0,75 \times 10^{-4}$; $z = 0,845 \times 10^2$ результат $u = -55,6848$.

2. Определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного числа.

Вариант 3

$$1. v = \frac{1 + \sin^2(x+y)}{\left| \frac{2y}{x - \frac{1}{1+x^2 y^2}} \right|} x^{|y|} + \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right).$$

При $x = 3,74 \times 10^{-2}$; $y = -0,825$; $z = 0,16 \times 10^2$ результат $v = 1,0553$.

2. Определить сумму двух первых цифр заданного четырехзначного числа и произведение двух его последних цифр.

Вариант 4

$$1. w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$$

При $x = 0,4 \times 10^4$; $y = -0,875$; $z = -0,475 \times 10^{-3}$ результат $w = 1,9873$.

2. Определить куб суммы цифр заданного трехзначного числа.

Вариант 5

$$1. \alpha = \ln \left(y^{-\sqrt{|x|}} \right) \left(x - \frac{y}{2} \right) + \sin^2 \operatorname{arctg}(z).$$

При $x = -15,246$; $y = 4,642 \times 10^{-2}$; $z = 0,001 \times 10^2$ результат $\alpha = -182,036$.

2. Определить произведение целой и дробной части заданного положительного вещественного числа.

Вариант 6

$$1. \beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\sin^2 z - |x-y|).$$

При $x = 16,55 \times 10^{-3}$; $y = -2,75$; $z = 0,15$ результат $\beta = -40,636$.

2. Присвоить целой переменной k третью от конца цифру в записи положительного целого числа n (например, если $n = 130985$, то $k = 9$).

Вариант 7

$$1. \gamma = 5 \operatorname{arctg}(x) - \frac{1}{4} \cos(x) \frac{x+3|x-y|+x^2}{|x-y|z+x^2}.$$

При $x = -0,1722$; $y = 6,33$; $z = 3,25 \times 10^{-4}$ результат $\gamma = -150,99$.

2. Присвоить целой переменной k первую цифру из дробной части положительного вещественного числа x (так, если $x = 32,597$, то $k = 5$).

Вариант 8

$$1. \varphi = \frac{e^{|x-y|} |x-y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}.$$

При $x = -2,235 \times 10^{-2}$; $y = 2,23$; $z = 15,221$ результат $\varphi = 39,374$.

2. Целой переменной S присвоить сумму цифр трехзначного целого числа k .

Вариант 9

$$1. \psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y-x) \frac{\cos y - z/(y-x)}{1+(y-x)^2}.$$

При $x = 1,825 \times 10^2$; $y = 18,225$; $z = -3,298 \times 10^{-2}$ результат $\psi = 1,2131$.

2. Идет k -я секунда суток. Определить, сколько целых часов h и целых минут m прошло к этому моменту (например, $h = 3$ и $m = 40$, если $k = 13257 = 3 \times 3600 + 40 \times 60 + 57$).

Вариант 10

$$1. a = 2^{-x} \sqrt{x + 4\sqrt{|y|}} \sqrt[3]{e^{x-1/\sin z}}$$

При $x = 3,981 \times 10^2$; $y = -1,625 \times 10^3$; $z = 0,512$ результат $a = 1,26185$.

2. Определить f – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h часов, m минут и s секунд.

Вариант 11

$$1. b = y^{\sqrt[3]{|x|}} + \cos^3(y) \frac{|x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{|x-y|} + \frac{x}{2}}$$

При $x = 6,251$; $y = 0,827$; $z = 25,001$ результат $b = 0,7121$.

2. Определить h – полное количество часов и m – полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 < f < 360$, f – вещественное число).

Вариант 12

$$1. c = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctg} z - \frac{\pi}{6} \right)}{|x| + \frac{1}{y^2 + 1}}$$

При $x = 3,251$; $y = 0,325$; $z = 0,466 \times 10^{-4}$ результат $c = 4,2514$.

2. Поменять местами значения целых переменных x и y , не используя дополнительные переменные.

Вариант 13

$$1. f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \operatorname{tg} z)}$$

При $x = 17,421$; $y = 10,365 \times 10^{-3}$; $z = 0,828 \times 10^5$ результат $f = 0,33056$.

2. Дано трехзначное число. Вывести число, полученное при перестановке первой и второй цифр заданного числа.

Вариант 14

$$1. g = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x+y/2}{2|x+y|} (x+1)^{-1/\sin z}$$

При $x = 12,3 \times 10^{-1}$; $y = 15,4$; $z = 0,252 \times 10^3$ результат $g = 82,8257$.

2. Дано трехзначное число. В нем зачеркнули первую цифру и записали ее в конец. Вывести полученное число на экран.

Вариант 15

$$1. h = \frac{x^{y+1} + e^{y-1}}{1+x|y-\operatorname{tg} z|} (1+|y-x|) + \frac{|y-x|^2}{2} - \frac{|y-x|^3}{3}$$

При $x = 2,444$; $y = 0,869 \times 10^{-2}$; $z = -0,13 \times 10^3$ результат $h = -0,49871$.

2. Из трехзначного числа x вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа x , то получилось число z . Найти число x .

Вариант 16

$$1. t = \left(1 + \frac{z^2}{3 - \frac{z^2}{5}} \right) \frac{2 \cos \left(x - \frac{\pi}{6} \right)}{0,5 + \sin^2 y}$$

При $x = 14,26$; $y = -1,22$; $z = 3,5 \times 10^{-2}$ результат $t = 0,565$.

2. Найти сумму цифр заданного трехзначного числа.

Вариант 17

$$1. u = e^{|x-y|} (\operatorname{tg}^2 z + 1)^x - \frac{\sqrt[3]{8 + |x-y|^2 + 1}}{x^2 + y^2 + 2}$$

При $x = -4,5$; $y = 0,75 \times 10^{-4}$; $z = 0,845 \times 10^2$ результат $u = 55,685$.

2. Определить число, полученное выписыванием в обратном порядке цифр заданного четырехзначного числа.

Вариант 18

$$1. v = \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right) + \frac{1 + \sin^2(x+y)}{\left| x - \frac{2y}{1+x^2y^2} \right|} x^{|y|}.$$

При $x = 3,74 \times 10^{-2}$; $y = -0,825$; $z = 0,16 \times 10^2$ результат $v = 1,0553$.

2. Определить квадрат суммы цифр заданного трехзначного числа.

Вариант 19

$$1. w = \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right) |\cos x - \cos y|^{(1+2\sin^2 y)}.$$

При $x = 0,4 \times 10^4$; $y = -0,875$; $z = -0,475 \times 10^{-3}$ результат $w = 1,9873$.

2. Определить произведение целой и дробной части заданного положительного вещественного числа.

Вариант 20

$$1. a = \sin^2 \left[\operatorname{arctg}(z) \right] + \left(x - \frac{y}{2} \right) \ln \left(\frac{1}{y^{\sqrt{|x|}}} \right).$$

При $x = -15,246$; $y = 4,642 \times 10^{-2}$; $z = 20,001 \times 10^2$ результат $a = -182,04$.

2. Присвоить целой переменной k вторую от конца цифру в записи положительного целого числа n .

Вариант 21

$$1. b = \left(\sin^2 z - |x-y| \right) \sqrt{10 \left(\sqrt[3]{x} + x^{y+2} \right)}.$$

При $x = 16,55 \times 10^{-3}$; $y = -2,75$; $z = 0,15$ результат $b = -40,6357$.

2. Присвоить целой переменной k первую цифру из дробной части положительного вещественного числа.

Вариант 22

$$1. g = \frac{1}{4} \cos(x) \frac{x+3|x-y|+x^2}{|x-y|z+x^2} - 5 \operatorname{arctg}(x).$$

При $x = -0,1722$; $y = 6,33$; $z = 3,25 \times 10^{-4}$ результат $g = 150,99$.

2. Из трехзначного числа x вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа x , то получилось число t . Найти число x .

Вариант 23

$$1. f = \sqrt[3]{x^6 + \ln^2 y} + \frac{e^{|x-y|} |x-y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)}.$$

При $x = -2,235 \times 10^{-2}$; $y = 2,23$; $z = 15,221$ результат $f = 39,374$.

2. Идет k -я секунда суток. Определить, сколько целых часов h и целых минут m прошло к этому моменту.

Вариант 24

$$1. w = \frac{\cos y - \frac{z}{y-x}}{1 + (y-x)^2} (y-x) + \left| x^{\frac{|y|}{|x|}} - \sqrt[3]{\frac{y}{x}} \right|.$$

При $x = 1,825 \times 10^2$; $y = 18,225$; $z = -3,298 \times 10^{-2}$ результат $w = 1,2131$.

2. Определить f – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h часов, m минут и s секунд ($0 \leq h \leq 11$, $0 \leq m \leq 59$, $0 \leq s \leq 59$).

Вариант 25

$$1. a = 2^{-x} \sqrt[3]{e^{\frac{x-1}{\sin z}} \sqrt{x + 4\sqrt{|y|}}}.$$

При $x = 3,981 \times 10^{-2}$; $y = -1,625 \times 10^3$; $z = 0,512$ результат $a = 1,262$.

2. Определить h – полное количество часов и m – полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 \leq f < 360$, f – вещественное число).

Вариант 26

$$1. b = \frac{|x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{|x-y|} + \frac{x}{2}} \cos^3 y + y^{\sqrt{|x|}}.$$

При $x = 6,251$; $y = 0,827$; $z = 25,001$ результат $b = 0,7121$.

2. Поменять местами значения целых переменных x и y , не используя дополнительные переменные.

Вариант 27

$$1. c = \frac{y \left(\arctg z - \frac{\pi}{6} \right)}{|x| + \frac{1}{y^2 + 1}} - (3^x)^y - 2^{(y^x)}.$$

При $x = 3,251$; $y = 0,325$; $z = 0,466 \times 10^{-4}$ результат $c = -4,2514$.

2. Дано трехзначное число. Вывести число, полученное при перестановке первой и второй цифр заданного числа.

Вариант 28

$$1. f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{(\sin^2 z + \operatorname{tg} z) |x-y|}.$$

При $x = 17,421$; $y = 10,365 \times 10^{-3}$; $z = 0,828 \times 10^5$ результат $f = 0,3306$.

2. Дано трехзначное число. В нем зачеркнули первую цифру и записали ее в конец. Вывести полученное число на экран.

Вариант 29

$$1. g = \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{-1/\sin z} + \frac{y^{x+1}}{3 + \sqrt[3]{|y-2|}}.$$

При $x = 1,23$; $y = 15,4$; $z = 2,52 \times 10^3$ результат $g = 82,826$.

2. Целой переменной S присвоить сумму цифр трехзначного целого числа k .

Вариант 30

$$1. h = \frac{|y-x|^3}{3} - (1+|y-x|) \frac{x^{y+1} + e^{y-1}}{1+x|y-\operatorname{tg} z|} - \frac{|y-x|^2}{2}.$$

При $x = 2,444$; $y = 0,869 \times 10^{-2}$; $z = -0,13 \times 10^3$ результат $h = 0,4987$.

2. Определить произведение двух первых цифр заданного четырехзначного числа и сумму двух его последних цифр.

ЛАБОРАТОРНАЯ РАБОТА №2

Решение прикладных задач, основанных на использовании разветвляющихся алгоритмов

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании разветвляющихся алгоритмов, в визуальной среде Delphi.

Порядок выполнения лабораторной работы

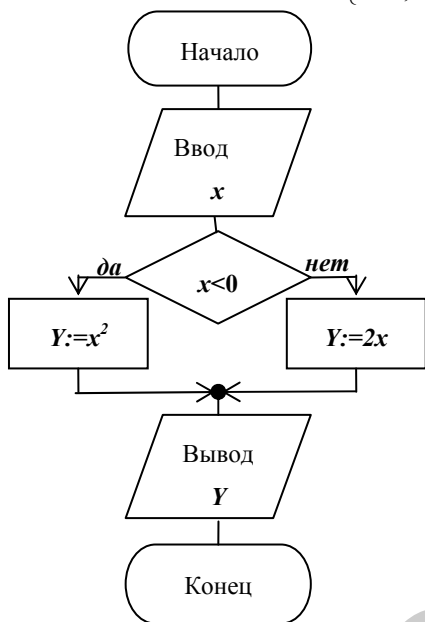
1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания приложения.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

1.1. Программирование разветвляющихся алгоритмов

Пусть необходимо вычислить значение выражения:

$$Y = \begin{cases} x^2, & \text{если } x < 0; \\ 2x, & \text{если } x \geq 0. \end{cases} \quad (1)$$



Блок-схема алгоритма, который выполняет необходимые действия, приведена на рисунке 1.

В этом алгоритме осуществляется ввод произвольного значения x , а затем, в зависимости от значения x , выполняются различные действия. Такие алгоритмы, в которых в зависимости от некоторых условий выполняется одна или другая последовательность действий (ветвь), называются **разветвляющимися**.

Разветвляющийся алгоритм может содержать любое количество ветвей.

Рисунок 1. Блок-схема алгоритма вычисления выражения (1)

1.2. Условные операторы перехода

Для программирования разветвляющихся алгоритмов в языке *OP* используются условные и безусловные операторы перехода, которые позволяют изменять последовательность выполнения операторов программы. Операторы перехода, обеспечивающие выполнение той или иной ветви алгоритма в зависимости от выполнения некоторого условия, называются **условными**. Эти операторы содержат **условие перехода** на соответствующую ветвь алгоритма.

Оператор if

Используется для программирования алгоритма, который разветвляется на две ветви, причем для разных схем алгоритмов применяются разные формы записи оператора.

Если алгоритм имеет две ветви, в каждой из которых выполняются действия (рисунок 2), то такой алгоритм программируется с помощью оператора **if** следующего вида:

if логическое выражение **then**

```
begin
  оператор1;
  оператор2;
  ...
  операторn;
end
```

else

```
begin
  оператор1;
  оператор2;
  ...
  операторk;
end;
```

Оператор выполняется следующим образом. Вначале вычисляется **логическое выражение**. Если результат его вычисления равен **True** (истина), тогда вычисляются операторы, стоящие после слова **then**, а затем операторы, которые следуют за оператором **if**.

Если же результат вычисления логического выражения оказался равен **False** (ложь), то выполняются операторы, расположенные после слова **else**.

Обращаем внимание на то, что перед словом **else** символ ; (точка с запятой) **не ставится!**

В логических выражениях используются **операции отношения**. Результатом вычисления логического выражения могут быть только два значения: истина (True) или ложь (False). Операции отношения представлены в таблице 1.

Таблица 1 – Операции отношения

Операция	Действие	Выражение	Результат
=	Равно	A=B	True, если A=B
<>	Не равно	A<>B	True, если A<B или A>B
<	Меньше	A<B	True, если A	Больше	A>B	True, если A>B
<=	Меньше или равно	A<=B	True, если A<B или A=B
>=	Больше или равно	A>=B	True, если A>B или A=B

Блок-схема алгоритма, иллюстрирующая работу оператора **if**, который содержит две ветви **then** и **else**, показана на рисунке 2.

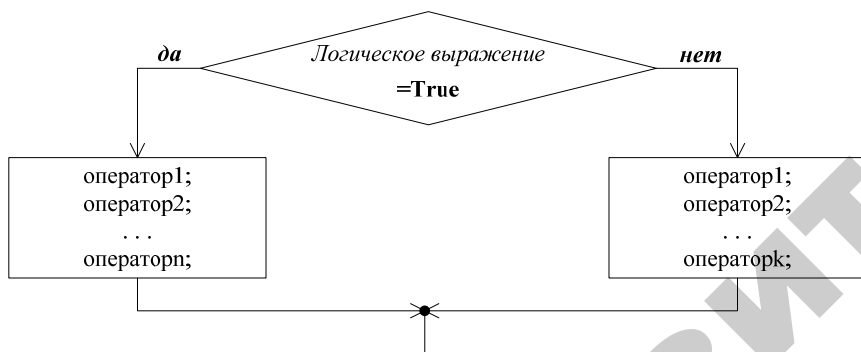


Рисунок 2. Блок-схема алгоритма работы оператора **if**, который содержит две ветви

Пример 1. Даны значения x и y . Переменной Z присвоить наибольшее из них

$$Z = \max(x, y). \quad (2)$$

Блок-схема алгоритма, который выполняет необходимые действия, приведена на рисунке 3.

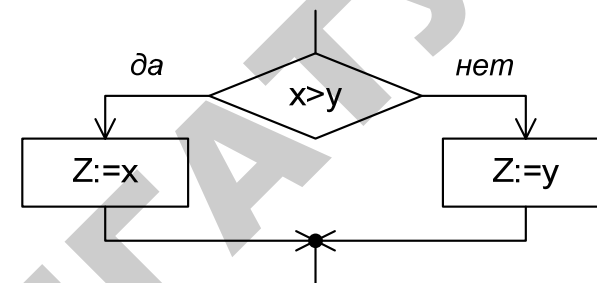


Рисунок 3. Блок-схема алгоритма вычисления выражения (2)

Оператор **if**, организующий разветвление алгоритма в этом примере, запишется так:

```

if x>y then
    z :=x
    else
    z :=y;
    
```

Замечание. Если в ветвях **then** и **else** используется один оператор, то **begin** и **end** можно не писать!

При решении некоторых задач необходимо использовать алгоритм с двумя ветвями, в одной из которых выполняются операторы, а в другой операторы отсутствуют. Блок-схема такого алгоритма изображена на рисунке 4.

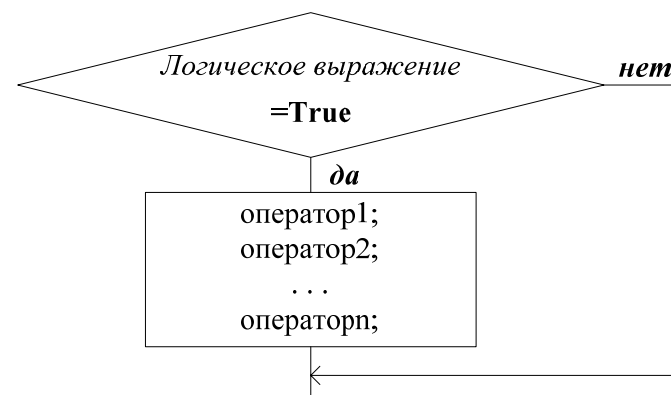


Рисунок 4. Блок-схема разветвляющегося алгоритма с операторами в одной ветви

Такой алгоритм программируется с помощью оператора **if** следующего вида:

```

if логическое выражение
then
    begin
        оператор1;
        оператор2;
        . . .
        операторn;
    end;

```

Оператор выполняется следующим образом. Если результат вычисления логического выражения равен **True** (истина), тогда выполняются операторы, стоящие после слова **then**, а затем операторы, которые следуют за оператором **if**. Если же результат вычисления логического выражения оказался равен **False** (ложь), то выполняются операторы, которые следуют за оператором **if**.

Пример 2. Даны значения x и y . Если $x < y$, то содержимое x занести в y , а содержимое y занести в x .

На рисунке 5 приведены два возможных варианта блок-схемы алгоритма, который выполняет необходимые действия.

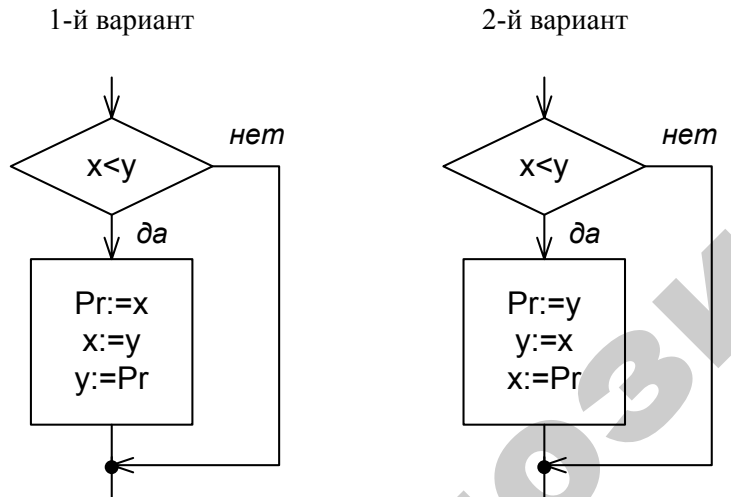


Рисунок 5. Варианты блок-схемы алгоритма

Здесь для промежуточного хранения значений переменных x и y использована переменная Pr .

Оператор **if**, организующий разветвление алгоритма в этом примере, запишется так:

```

1-й вариант          2-й вариант
if  $x < y$  then      if  $x < y$  then
    begin              begin
         $Pr := x;$        $Pr := y;$ 
         $x := y;$        $y := x;$ 
         $y := Pr;$      $x := Pr;$ 
    end;              end;

```

В логических выражениях для записи сложных условий, содержащих, например, несколько сравнений, кроме операций отношения могут использоваться **логические операции** (таблица 2).

Таблица 2 – Логические операции

Операция	Действие	Выражение	A	B	Результат
Not	Отрицание	Not A	True		False
And	И	A and B	True	True	True
			True	False	False
			False	True	False
Or	ИЛИ	A or B	True	True	True
			True	False	True
			False	True	True
Xor	Исключающее ИЛИ	A xor B	True	True	False
			True	False	True
			False	True	True
			False	False	False

Пример 3. Вычислить значение выражения:

$$y = \begin{cases} \cos^2(x), & \text{если } 0 \leq x \leq 2; \\ \sin^2(x), & \text{иначе.} \end{cases} \quad (3)$$

Оператор **if**, с помощью которого можно вычислить выражение (3), запишется следующим образом:

```
if (0<=x) and (x<=2) then
    y:=sqr(cos(x))
else
    y:=sqr(sin(x));
```

Замечание. Операции отношения обязательно заключаются в круглые скобки, так как они имеют более низкий приоритет, чем логические операции.

Операторы, стоящие в ветвях **then** или **else**, тоже могут быть условными. В этом случае говорят, что операторы **if** *вложены* друг в друга.

Пример 4. Вычислить значение выражения:

$$c = \begin{cases} \max(a,b), & \text{если } ab < 0; \\ \min(a,b), & \text{если } ab \geq 0. \end{cases} \quad (4)$$

На рисунке 6 приведена блок-схема алгоритма, который выполняет необходимые действия:

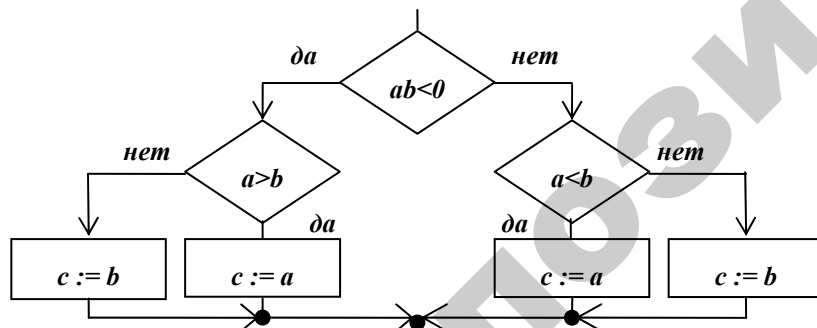


Рисунок 6. Блок-схема алгоритма вычисления выражения (4)

Оператор **if**, организующий разветвление алгоритма в этом примере, запишется так:

```
if a*b<0 then
    if a>b then
        c:=a
    else
        c:=b
    else
        if a<b then
            c:=a
        else
            c:=b;
```

Замечание. Для упрощения проверки работы оператора и уменьшения количества ошибок рекомендуется записывать вложенные операторы **if** так, чтобы логические выражения, ключевые слова (выделены жирным шрифтом) и операторы, стоящие в ветвях **then** и **else**, были расположены в отдельных строках друг против друга, как это показано в примере.

При реализации алгоритма, имеющего более двух ветвей, иногда вместо нескольких вложенных друг в друга операторов **if** удобно использовать оператор **case**.

Оператор case

Используется вместо нескольких вложенных операторов **if** для программирования алгоритма, который разветвляется на любое количество ветвей. В общем случае оператор **case** записывается так:

```
case селектор of
    список1:оператор1;
    список2:оператор2;
    . . .
    списокn:операторn
else
    оператор;
end;
```

Здесь использованы следующие обозначения:

- **селектор** – переменная или выражение одного из следующих типов: целого, символьного, логического, интервального или перечисляемого;
- **список1...списокn** – значения, которые принимает **селектор**. Значения в списках отделяются друг от друга запятыми.

Оператор **case** выполняется следующим образом. Вначале вычисляется значение **селектор**. Если результат его вычисления равен одному из значений **список1**, тогда выполняется оператор1. Если результат вычисления **селектор** равен одному из значений **список2**, выполняется оператор2 и т.д. Если же результат вычисления

селектора не равен ни одному из значений *список1 . . списокn*, то выполняется оператор, расположенный после слова **else**. Обращаем внимание на то, что перед словом **else** символ ; (точка с запятой) **не ставится!**

При использовании оператора **case** кроме целого, применяются данные символьного, логического, интервального или перечисляемого типов.

Символьный тип позволяет работать с данными, значения которых представляют собой любой символ кодовой таблицы персонального компьютера.

Логический тип используется в тех случаях, когда необходимо работать с данными, которые принимают только два значения **TRUE** (истина) или **FALSE** (ложь).

Интервальный тип позволяет задавать две константы, определяющие границы интервала значений для данных этого типа.

Перечисляемый тип задается непосредственно перечислением всех значений, которые могут принимать данные этого типа.

В программе переменные этих типов могут быть объявлены таким образом:

```

var
    Simvol:char;           { символьный тип }
    Priznak, Selektor1:boolean; { логический тип }
    DniNedeli:1..7;       { интервальный тип }
    VremenaGoda:(Zima,Vesna,Leto,Osen); { перечисляемый тип }

```

Пример 5. Вычислить значение выражения (5), если x – целое:

$$f = \begin{cases} \alpha, & \text{если } 1 < x \leq 5; \\ 2\alpha, & \text{если } 5 < x \leq 10; \\ 3\alpha, & \text{если } 10 < x \leq 15; \\ 4\alpha, & \text{иначе.} \end{cases} \quad (5)$$

На рисунке 7 приведена блок-схема алгоритма, который выполняет необходимые действия.

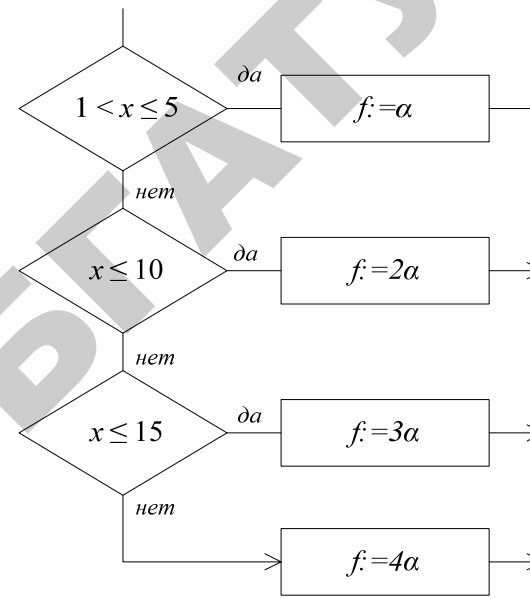


Рисунок 7. Блок-схема алгоритма вычисления выражения (5)

Ниже для сравнения приведены варианты реализации вычисления выражения (5) в соответствии с рисунком 7 при помощи операторов **if** и **case**:

<pre> if (1<x) and (x<=5) then f:=alpha else if x<=10 then f:=2*alpha else if x<=15 then f:=3*alpha else f:=4*alpha; </pre>	<pre> case x of 1..5:f:=alpha; 6..10:f:=2*alpha; 11..15:f:=3*alpha else f:=4*alpha; end; </pre>
--	---

В операторе **case**, так же как и в операторе **if**, ветвь **else** может отсутствовать.

1.3. Безусловный оператор перехода

В безусловном операторе перехода условие перехода на соответствующую ветвь алгоритма *отсутствует*. В общем виде оператор записывается так:

```
goto метка;
```

Меткой может быть любой идентификатор, но на практике в качестве меток рекомендуется использовать положительные целые числа. Метки должны быть указаны в разделе объявлений после ключевого слова **Label** например:

```
Label 3, 27;
```

Безусловный оператор перехода чаще всего используется в ветвях **then** или **else** оператора **if** для перехода на операторы, стоящие перед оператором **if**, или после него, например:

```
...
Label 5,17;
Begin
...
5:a:=cos(y+z);
...
if c<d then
    goto 5
else
    goto 17;
...
17:b:=sin(y-z);
```

Метка отделяется от оператора, на который осуществляется переход, символом : (двоеточие).

Здесь, в операторе **if**, при выполнении условия $c < d$ с помощью оператора **goto 5** осуществляется переход на оператор, помеченный меткой 5, т.е. на $a := \cos(y + z)$.

Если же в операторе **if** условие $c < d$ не выполняется, то с помощью оператора **goto 17** происходит переход на оператор, помеченный меткой 17, т.е. на $b := \sin(y - z)$.

2. Пример создания приложения

Разработать *Windows*-приложение для вычисления выражения

$$y = \begin{cases} \cos^2 x, & \text{если } 0 < x \leq 2; \\ 1 - \sin x^2, & \text{иначе.} \end{cases} \quad (6)$$

В панели интерфейса предусмотреть поля для ввода исходных данных с клавиатуры и вывода результатов расчета. Для удобства пользования панелью интерфейса снабдить поля соответствующими надписями.

Создание интерфейса и разработка алгоритма

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке 8.

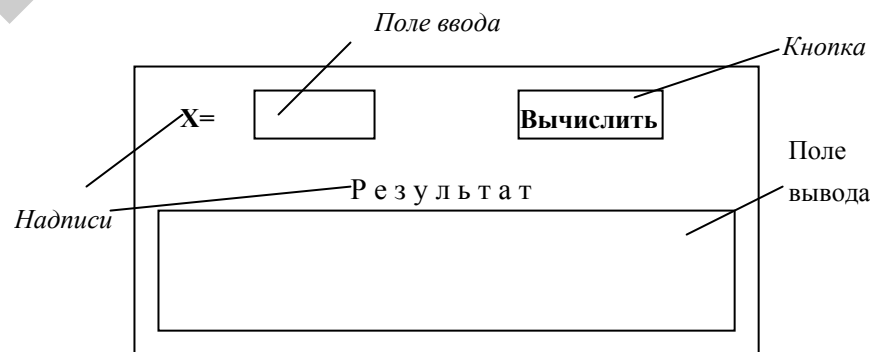


Рисунок 8 — Панель интерфейса приложения

Сценарий работы приложения:

- 1) щелчок мышью в поле ввода x , после ввод с клавиатуры значения исходного данного x ;
- 2) щелчок на кнопке **Вычислить**.

Блок-схема алгоритма процедуры обработки события нажатия кнопки **Вычислить** может иметь следующий вид, представленный на рисунке 9.

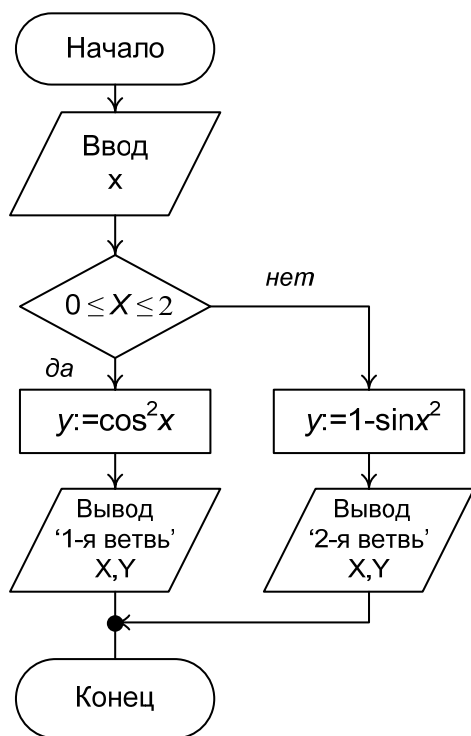


Рисунок 9. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Методические указания по созданию приложения

Сохранение проекта

1. В своей рабочей папке создайте новую папку и назовите ее **LAB2**.
2. Сохраните проект.
 - 1) **File**→ **Save Project As...** (**Файл**→**Сохранить Проект как ...**).
 - 2) В окне диалога **Save Unit1 As (Сохранить Unit1 как)** найдите созданную папку **LAB2** и сохраните в ней модуль под именем **UnLab2 (Unit-модуль Lab2-лабораторная работа №2)**.
 - 3) Затем в окне диалога **Save Project1 As (Сохранить Project1 как)** сохраните файл проекта в папке **LAB2** под именем **PrLab2**.
 - 4) Для сохранения проекта используйте опцию **Save All (Сохранить Все)** в меню **File (Файл)** или пиктограмму

Изменение заголовка Формы

Для изменения заголовка активизируйте окно Инспектора Объектов и на странице **Properties** в свойстве **Caption** замените заголовок **Form1** на **Лаб2**. Убедитесь, что одновременно изменился заголовок окна Формы.

Размещение компонентов на Форме

Разместите компоненты на Форме в соответствии с рисунком 10.

Для нанесения надписей на Форму используется компонент **Label**. Выберите в Палитре Компонентов на странице **Standard** пиктограмму компонента **Label** и щелкните на ней мышью. Затем в нужном месте Формы щелкните мышью – появится надпись **Label1**. В свойстве **Caption** Инспектора Объектов замените надпись **Label1** на **X=**. В свойстве **Font** подберите шрифт. Аналогично нанесите на Форму надпись **Label2** – **Результат**.

Для поля ввода численных значений переменной **X** используем компонент **Edit**. Выберите в Палитре Компонентов на странице **Standard** пиктограмму и поместите компонент **Edit** в нужном месте Формы так, как показано на рисунке 10.

Для поля вывода используем компонент **Memo**.

Отрегулируйте положение на Форме и размеры размещенных компонентов.

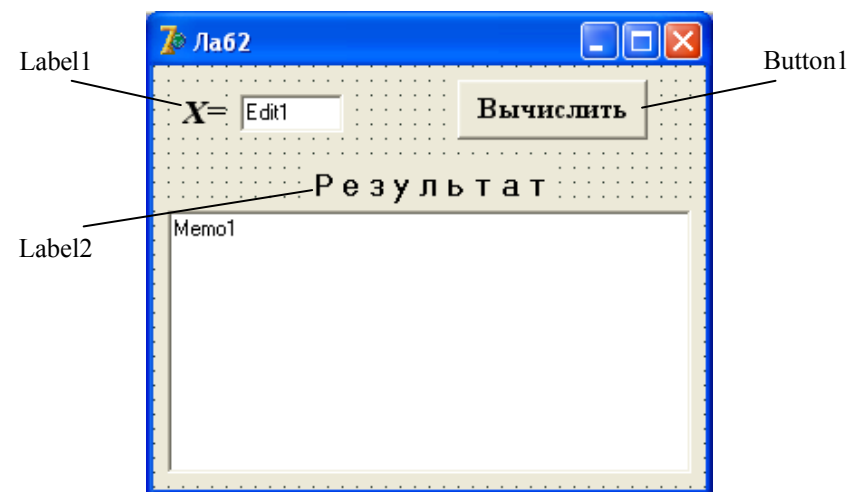



Рисунок 10. Форма с компонентами

Создание процедуры `Button1Click` обработки события нажатия кнопки **Вычислить**

Поместим на Форму кнопку, нажатие которой приведет к вычислению выражения. Выберите в Палитре Компонентов на странице **Standart** пиктограмму  компонента **Button**. Затем в нужном месте Формы щелкните мышью – появится кнопка с надписью **Button1**. В свойстве **Caption** Инспектора Объектов замените надпись на кнопке **Button1** надписью **Вычислить**. В свойстве **Font** подберите шрифт. Отрегулируйте положение и размер кнопки.

Для создания процедуры-обработчика поместите курсор на изображение кнопки и дважды щелкните левой кнопкой мыши. На экране появится текст модуля `UnLab2`, в котором *Delphi* автоматически создает заготовку процедуры-обработчика события нажатия кнопки `Button1`:

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


Замечание. При создании этой процедуры необходимо очень аккуратно наводить курсор мыши на изображение кнопки. Небрежное действие приведет к созданию совершенно другой процедуры. Убедитесь в том, что пустая процедура, которую создала *Delphi*, называется именно `procedure Form1.Button1Click(Sender:TObject)!`

Между операторами `begin` и `end` этой процедуры вставьте операторы, которые выполняют необходимые действия. Кроме этого, после заголовка процедуры необходимо добавить раздел объявлений `var`, в котором следует перечислить переменные, участвующие в вычислениях, и указать их тип:

```
procedure TForm1.Button1Click(Sender: TObject);
var
x, y: extended;           // объявление локальных переменных
begin
x:=StrToFloat(Edit1.Text); // x присваивается содержимое Edit1
Memo1.Clear;              // очищается поле компонента Memo1
if (0<=x) and (x<2) then
begin
y:=cos(x)*cos(x);
Memo1.Lines.Add('1-я ветвь:');
```

```
Memo1.Lines.Add('X='+FloatToStrF(x, ffFixed, 8, 1)+
' Y='+FloatToStrF(y, ffFixed, 8, 3))
end
else
begin
y:=1-sin(x*x);
Memo1.Lines.Add('2-я ветвь:');
Memo1.Lines.Add('X='+FloatToStrF(x, ffFixed, 8, 1)+
' Y='+FloatToStrF(y, ffFixed, 8, 3))
end;
end;
```

Работа с приложением

Для запуска созданного приложения нажмите пиктограмму  главного меню или клавишу **F9**. При этом происходит компиляция модулей и, если нет ошибок, компоновка проекта и создание выполняемого файла **PrLab2.exe**. На экране появляется панель интерфейса приложения, показанная на рисунке 11.

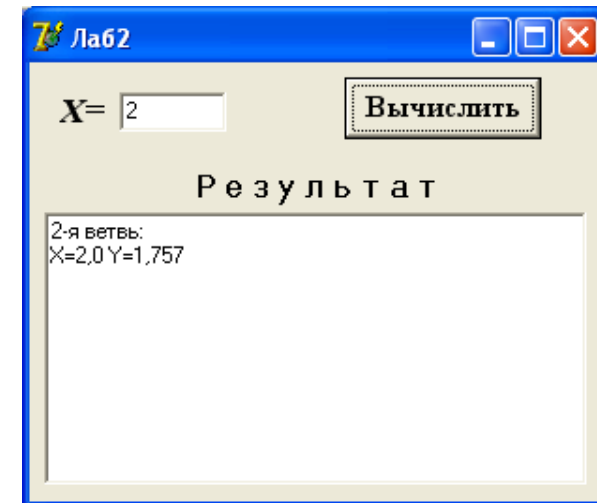


Рисунок 11. Панель интерфейса приложения

Используя рисунок 11, выполните действия в соответствии со сценарием работы приложения. Чтобы проверить правильность работы всех

ветвей алгоритма, введите другие значения X и снова запустите приложение. Убедитесь, что приложение дает верные результаты при всех значениях исходных данных, указанных в задании.

В случае нештатного функционирования приложения восстановить первоначальный режим работы с проектом можно выбором в меню **Run** опции **ProgramReset** или нажатием клавиши **Ctrl+F2**.

Вопросы для самоконтроля

1. Каким оператором перехода программируется алгоритм, который содержит две ветви, в каждой из которых выполняются действия?
2. Чему равен результат вычисления логического выражения?
3. Как в условных операторах перехода записываются сложные логические выражения, содержащие несколько операций отношения?
4. Почему в логических выражениях, содержащих логические операции, операции отношения необходимо заключать в круглые скобки?
5. Что такое вложенные условные операторы и как они записываются?

Варианты индивидуальных заданий разного уровня:

- 1 уровень** – выполните задание 1;
2 уровень – выполните задания 1, 2;
3 уровень – выполните задания 1, 2, 3.

Вариант 1

1. Вычислить значение функции:

$$y = \begin{cases} 1, & t < 1; \\ at^2 \cdot \ln t, & 1 \leq t \leq 2; \\ e^{at} \cos bt, & t > 2. \end{cases}$$

2. Определить, является ли число B наименьшим из трех, неравных между собой чисел A, B, C .
3. По номеру n ($n > 0$) некоторого года определить c – номер его столетия (учесть, что, к примеру, началом XX столетия был 1901, а не 1900 год!).

Вариант 2

1. Вычислить значение функции:

$$y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 1,3; \\ ax^3 + 7\sqrt{x}, & x = 1,3; \\ \lg(x + 7\sqrt{x}), & x > 1,3. \end{cases}$$

2. Определить, могут ли три произвольных числа a, b, c служить длинами сторон треугольника. Если могут, то рассчитать площадь этого треугольника. В противном случае вывести соответствующее сообщение.
3. Для целого числа k от 1 до 99 вывести фразу «мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года».

Вариант 3

1. Вычислить значение функции:

$$w = \begin{cases} ax^2 + bx + c, & x < 1,2; \\ \frac{a}{x} + \sqrt{x^2 + 1}, & x = 1,2; \\ \frac{a + bx}{\sqrt{x^2 + 1}}, & x > 1,2. \end{cases}$$

2. Вычислить $Q = \min(x, y^2, z^2) + 1$.
3. Дано целое число в диапазоне от 1 до 5. Вывести строку – словесное описание соответствующей оценки (1 – «плохо», 2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отлично»).

Вариант 4

1. Вычислить значение функции:

$$Q = \begin{cases} \pi x^2 - 7/x^2, & x < 1,4; \\ ax^3 + 7\sqrt{x}, & x = 1,4; \\ \ln(x + 7\sqrt{|x+a|}), & x > 1,4. \end{cases}$$

2. Даны три произвольных числа a , b , c . Определить среднее арифметическое положительных чисел среди них. Если таких чисел нет – вывести сообщение об этом.

3. В старояпонском календаре был принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цвета: **green** (зеленый), **red** (красный), **yellow** (желтый), **white** (белый) и **black** (черный). Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. (1984 год – год зеленой крысы – был началом очередного цикла). Разработать программу, которая вводит номер некоторого года нашей эры и выводит его название по старояпонскому календарю.

Вариант 5

1. Вычислить значение функции:

$$y = \begin{cases} 1,5 \cos^2 x, & x < 1; \\ 1,8ax, & x = 1; \\ (x-2)^2 + 6, & 1 < x < 2; \\ 3 \operatorname{tg} x, & x \geq 2. \end{cases}$$

2. Даны три неравных между собой числа x , y , z . Найти сумму чисел, кратных 3.

3. Составьте программу, определяющую название ноты по ее номеру: 1 – До, 2 – Ре и т.д. При вводе числа вне диапазона [1...7] программа должна выдавать сообщение об ошибке.

Вариант 6

1. Вычислить значение функции:

$$w = \begin{cases} x^3 \sqrt{x-a}, & x > a; \\ x \sin ax, & x = a; \\ e^{-ax} \cos ax, & x < a. \end{cases}$$

2. Имеются четыре результата t_1, t_2, t_3, t_4 (с) в беге на 100 метров. Вывести результат и номер победителя.

3. Дано целое число k в диапазоне от 1 до 9. Вывести значение переменной k римскими цифрами.

Вариант 7

1. Вычислить значение функции:

$$R = \begin{cases} bx - \ln|bx|, & bx < 1; \\ 1, & bx = 1; \\ bx - \ln bx, & bx > 1. \end{cases}$$

2. Заданы три точки с координатами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$ и круг с центром в начале координат и радиусом R . Определить, сколько точек принадлежит кругу.

3. Дано натуральное число n , определяющее стоимость товара в копейках. Вывести стоимость в рублях и копейках в виде «3 рубля 21 копейка»; «15 рублей 5 копеек» или «1 рубль 2 копейки».

Вариант 8

1. Вычислить значение функции:

$$y = \begin{cases} \sin x \cdot \lg|x|, & x < 3,5; \\ \cos^2 x, & x \geq 3,5. \end{cases}$$

2. Даны действительные числа a , b , c . Удвоить эти числа, если $a < b < c$, и заменить их абсолютными значениями, если это не так.

3. Единицы длины пронумерованы следующим образом: 1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах.

Вариант 9

1. Вычислить значение функции:

$$z = \begin{cases} \frac{\ln^3 x + x^3}{\sqrt{x+t}}, & x < 0,5; \\ \sqrt{x+t} + \frac{1}{x}, & x = 0,5; \\ \cos x + t \sin^2 t, & x > 0,5. \end{cases}$$

2. Бригада из четырех рабочих собирала картофель, каждый собрал соответственно a, b, c, d (кг). Определить, превышает ли средняя выработка в бригаде заданную величину S (кг) и вывести на экран номера рабочих, которые не достигли средней выработки по бригаде.

3. Составить программу, которая в зависимости от порядкового номера дня недели (1, 2, ..., 7) выводит на экран его название (понедельник, вторник, ..., воскресенье).

Вариант 10

1. Вычислить значение функции:

$$f = \begin{cases} \ln(x+1), & x > 1; \\ \sin^2 \sqrt{|ax|}, & x \leq 1. \end{cases}$$

2. Определить, какая из трех точек с координатами $A(x_1, y_1)$, $B(x_2, y_2)$ или $C(x_3, y_3)$ наиболее удалена от начала координат, и вывести на печать соответствующее сообщение.

3. Составить программу, которая в зависимости от порядкового номера месяца (1, 2, ..., 12) выводит на экран количество дней в этом месяце. Учесть, является ли год високосным или нет.

Вариант 11

1. Вычислить значение функции:

$$s = \begin{cases} \frac{a+b}{e^x + \cos x}, & x < 2,8; \\ \frac{a+b}{x+1}, & 2,8 \leq x < 6; \\ e^x + \sin x, & x \geq 6. \end{cases}$$

2. Четыре студента сдали экзамен по физике и получили соответственно оценки x, y, z, d . Вывести номера студентов, оценка которых превышает средний балл по предмету.

3. С начала 2000 года по некоторый день прошло n месяцев. Определить название месяца (январь, февраль и т.п.) этого дня.

Вариант 12

1. Вычислить значение функции:

$$y = \begin{cases} a \lg x + \sqrt{x}, & x > 1; \\ 2a \cos x + 3x^3, & x \leq 1. \end{cases}$$

2. Даны три точки с координатами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Определить количество точек, расположенных в I четверти координатной плоскости.

3. Для натурального числа k выведите фразу «мы нашли k грибов», согласовав окончание слова «гриб» с числом k .

Вариант 13

1. Вычислить значение функции:

$$w = \begin{cases} \frac{a}{i} + bi^2 + c, & i < 4; \\ i, & 4 \leq i \leq 6; \\ ai + bi^3, & i > 6. \end{cases}$$

2. Заданы радиус круга R и сторона квадрата A . Определить, можно ли круг вписать в квадрат.

3. Записать программу, которая на ввод времени суток выводит соответствующее пожелание доброго утра, доброго дня, доброго вечера и спокойной ночи.

Вариант 14

1. Вычислить значение функции:

$$y = \begin{cases} \frac{1}{\sin x + 2} & x \leq 0; \\ \lg x + e^x, & 0 < x \leq 2; \\ 2x^2, & x > 2. \end{cases}$$

2. Даны три целых числа. Найти произведение тех чисел, которые кратны 3.

3. Арифметические действия над числами пронумерованы следующим образом: 1 – сложение, 2 – вычитание, 3 – умножение, 4 – деление. Дан номер действия и два числа A и B (B не равно нулю). Выполнить над числами указанное действие и вывести результат.

Вариант 15

1. Вычислить значение функции:

$$g = \begin{cases} e^x + \frac{1}{x+1}, & 0 \leq x < 3; \\ \sin x + \sqrt{x}, & x = 3; \\ \cos x + |b|, & x > 3. \end{cases}$$

2. Даны три неравных между собой числа x , y , z . Наибольшее из них разделить на сумму двух других.

3. По дате (день и месяц) рождения определите знак Зодиака.

Вариант 16

1. Вычислить значение функции:

$$C = \begin{cases} ax^2 - 4, & x < -2; \\ x^4 - 1, & -2 \leq x < 0; \\ x^3 + 8x, & 0 \leq x \leq 2; \\ 4 \sin x, & x > 2. \end{cases}$$

2. Даны три целых числа. Вывести среднее арифметическое четных чисел.

3. В чемпионате по футболу команде за выигрыш дается 3 очка, за проигрыш – 0, за ничью – 1. Известно количество очков, полученных командой за игру. Вывести результат игры словами: **выигрыш**, **проигрыш** или **ничья**.

Вариант 17

1. Вычислить значение функции:

$$Z = \begin{cases} \sqrt{3x^2 + 7 \sin^2 x + 5}, & x < 0,1; \\ ax + b\sqrt{\sin x + 1}, & x = 0,1; \\ \sqrt{x^2 + 7 \cos^2 x + x}, & x > 0,1. \end{cases}$$

2. Четыре студента сдали экзамен по информатике и получили соответственно оценки a , b , c , d . Определить и вывести на экран номера студентов, которые получили по экзамену оценку 9.

3. Составить программу, которая по возрасту человека (вводится с клавиатуры как целое число) определяет его принадлежность к возрастной группе: от 0 до 13 – мальчик; от 14 до 20 – юноша; от 21 до 70 – мужчина; более 70 – старец.

Вариант 18

1. Вычислить значение функции:

$$D = \begin{cases} |x| + a^2 + 4, & a > 2; \\ \frac{x^2 + 2ax}{3x}, & a = 2; \\ 8a + \operatorname{tg} x, & a < 2 \end{cases}$$

2. Составить программу нахождения произведения наименьшего и наибольшего из трех различных чисел x, y, z .

3. Геометрические фигуры пронумерованы следующим образом: 1 – треугольник, 2 – прямоугольник, 3 – квадрат, 4 – ромб, 5 – трапеция. По номеру фигуры вывести ее название и формулу, по которой вычисляется площадь этой фигуры.

Вариант 19

1. Вычислить значение функции:

$$Z = \begin{cases} \frac{a+b}{4 + \cos^2 x}, & x < 2,3; \\ \frac{a+b}{x+1}, & 2,3 \leq x \leq 5; \\ e^x + \sin x, & x > 5 \end{cases}$$

2. Определить количество неотрицательных чисел среди трех заданных a, b, c . Если таких чисел нет – вывести соответствующее сообщение.

3. Дано целое число d в диапазоне от 0 до 9. Вывести на экран название этой цифры (0 – «ноль», 1 – «один», 2 – «два» и т.д.).

Вариант 20

1. Вычислить значение функции:

$$Y = \begin{cases} e^{-bx} \sin bx, & x < 0; \\ \cos ax, & 0 \leq x \leq 1; \\ e^{-ax} \sin x, & x > 1. \end{cases}$$

2. На плоскости даны две точки: $P(x_1, y_1)$ и $Q(x_2, y_2)$. Составьте программу, определяющую, находится ли каждая из этих точек на прямой $y = ax + b$.

3. Дано целое число в диапазоне от 100 до 999. Вывести словесное описание данного числа, например: 256 – «двести пятьдесят шесть», 814 – «восемьсот четырнадцать».

Вариант 21

1. Вычислить значение функции:

$$y = \begin{cases} \pi x^2 - \frac{7}{x^2}, & x < 3,4; \\ ax^3 + 7\sqrt{x}, & x = 3,4; \\ \ln(x + 7\sqrt{x}), & x > 3,4. \end{cases}$$

2. Вывести на экран номер четверти координатной плоскости (в виде **I**, **II**, **III**, **IV**), которой принадлежит точка с координатами (x, y) , при условии, что $x \neq 0$ и $y \neq 0$.

3. Составить программу, которая по номеру семестра печатает курс, к которому относится введенный семестр (1 и 2 семестр – 1 курс, 3 и 4 семестр – 2 курс и т.д.).

Вариант 22

1. Вычислить значение функции:

$$B = \begin{cases} \sqrt{6-x}, & x < c; \\ \sin \frac{x}{2}, & x \geq c. \end{cases}$$

2. Три студента сдали три экзамена и получили соответственно оценки: 1-й студент – x_1, x_2, x_3 ; 2-й студент – y_1, y_2, y_3 ; 3-й студент – z_1, z_2, z_3 . Вывести номера тех студентов, которые будут получать стипендию, если для этого необходимо иметь средний балл, равный 7,5.

3. С начала 2009 года по некоторый день прошло n дней. Вывести название месяца (январь, февраль и т.п.)

Вариант 23

1. Вычислить значение функции:

$$Y = Z - 5, \quad \text{где} \quad Z = \begin{cases} x^2 - 2x - 2, & x > 2; \\ 2, & -2 \leq x \leq 2; \\ -x^2 - x - 4, & x < -2. \end{cases}$$

2. Определить, могут ли три произвольных вещественных числа a, b, c служить сторонами треугольника. Если могут, определить, является ли треугольник прямоугольным.

3. Дано натуральное число n , определяющее возраст человека в месяцах. Вывести возраст в годах и месяцах числами и словами, например: «21 год 10 месяцев», «52 года 1 месяц», «46 лет ровно».

Вариант 24

1. Вычислить значение функции:

$$c = \begin{cases} e^{2x} + y^2 + \sin(y), & x + y > 0; \\ (e^x - y^3)^2 + \cos(y), & x + y = 0; \\ (y - e^x)^2 + \operatorname{tg}(y), & x + y < 0. \end{cases}$$

2. Заданы координаты 3-х точек: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Вывести координаты точек, лежащих во 2-ой четверти.

3. Единицы массы пронумерованы следующим образом: 1 – килограмм, 2 – грамм, 4 – тонна, 5 – центнер. Дан номер единицы массы и масса тела M в этих единицах (вещественное число). Вывести массу в килограммах.

Вариант 25

1. Вычислить значение функции:

$$S = \begin{cases} \frac{a}{2c} + bc^2 + \cos b^2, & c < 3; \\ \cos^3 a - c, & 3 \leq c \leq 7; \\ ac + bc^3 + 2, & c > 7. \end{cases}$$

2. Имеется 10 луковиц диаметром d_1 , 6 луковиц диаметром d_2 и 15 луковиц диаметром d_3 . Радиус отверстия в сепарирующем решете p . Определить, сколько луковиц пройдет сквозь отверстие решета.

3. Составить программу, которая по номеру дня в месяце печатает день недели. Считаем, что 1-е число месяца – понедельник.

Вариант 26

1. Вычислить значение функции:

$$a = \begin{cases} (x^2 + y)^2 - \sqrt{y}, & x \leq 0; \\ (x^2 + y)^3 + \sqrt{|y|}, & 0 < x \leq 1; \\ (x^2 + y)^4 + 1, & x > 1. \end{cases}$$

2. Имеются четыре результата z_1, z_2, z_3, z_4 (м) по прыжкам в высоту. Вывести номера спортсменов, которые выполнили заданный норматив L (м).

3. По порядковому номеру дня недели (1, 2, ..., 7) определить, является ли этот день рабочим или выходным.

Вариант 27

1. Вычислить значение функции:

$$g = \begin{cases} e^{x-|b|}, & x < 0,1; \\ \sqrt{|x^2 + b|}, & 0,1 \leq x < 0,5; \\ 2 \cos^2 x, & x \geq 0,5. \end{cases}$$

2. Заданы четыре переменные x, y, z, d . Вывести имена переменных, значения которых попали в интервал от -5 до 5 . Если таких значений нет, то вывести соответствующее сообщение.

3. Дано целое число n , соответствующее количеству углов геометрической фигуры. Составить программу, которая по введенному числу n выводит на экран название фигуры (например, при $n = 3$ программа печатает «треугольник», при $n > 8$ – «многоугольник»). Если вводится число меньше 2, выводится сообщение об ошибке.

Вариант 28

1. Вычислить значение функции:

$$d = \begin{cases} y + \operatorname{arctg}(x^2), & x < y; \\ (y - x^2)^2 + \operatorname{arctg}(x), & x = y; \\ (y + x^2)^3 + 0,5, & \text{иначе.} \end{cases}$$

2. Заданы три положительных числа a, b, c . Определить являются ли они последовательно стоящими элементами арифметической прогрессии. Если являются, то определить разность прогрессии.

3. Составьте программу для определения числа дней в месяце, если даны: номер месяца N – целое число от 1 до 12, целое число A , равное 1 для високосного года и 0 в противном случае.

Вариант 29

1. Вычислить значение функции:

$$S = \begin{cases} e^{2x}, & x < 1; \\ \sqrt{|x^2 - 4b|}, & 1 \leq x < 5,1; \\ b \sin x^2, & x \geq 5,1. \end{cases}$$

2. Известно, что из четырех чисел a_1, a_2, a_3, a_4 одно отлично от трех других, равных между собой; присвоить номер этого числа переменной n .

3. Составить программу, которая печатает номера дней в месяце, если вводится день недели. Считаем, что 1-е число месяца – понедельник, в месяце – 31 день. Вывести на экран словесное описание дня недели и соответствующие числа месяца (например, вводится число 2, на экране появляется: «Вторник – 2, 9, 16, 23, 30»).

Вариант 30

1. Вычислить значение функции:

$$j = \begin{cases} \sin(5x + 3m|x - m|), & x < -1,1; \\ \cos(3x^2 + 5m|x|), & -1,1 \leq x < 4; \\ (x^2 + m)^2, & x \geq 4. \end{cases}$$

2. Определить, какие из трех точек с координатами $A(x_1, y_1)$, $B(x_2, y_2)$, или $C(x_3, y_3)$ лежат в 3-ей четверти, вывести их имена.

3. Составить программу, определяющую название слоя атмосферы, соответствующего определенной высоте (тропосфера – 0-16 км, стратосфера – 16-50 км, мезосфера – 50-80 км, термосфера – 80-800 км, экзосфера – свыше 800 км).

ЛАБОРАТОРНАЯ РАБОТА № 3

Решение прикладных задач, основанных на использовании циклических алгоритмов с известным количеством повторений цикла

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании циклических алгоритмов с известным количеством повторений цикла, в визуальной среде *Delphi*.

Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания приложения. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

Пример 1. Вычислить значение выражения:

$$S = 1 + 2 + 3 + \dots + 10 = \sum_{i=1}^{10} i. \quad (1)$$

Блок-схема алгоритма, который выполняет необходимые действия, приведена на рисунке 1.

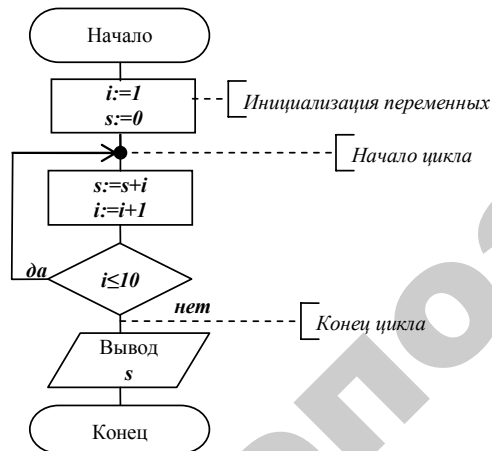


Рисунок 1. Блок-схема алгоритма вычисления выражения (1)

Анализ блок-схемы показывает, что в ней имеется участок, который выполняется 10 раз, образуя цикл. Такие алгоритмы, в которых действия многократно повторяются, называются **циклическими**.

Любой циклический алгоритм можно запрограммировать с помощью условных и безусловных операторов перехода, но для эффективной организации циклов в языке *OP* используются специальные **операторы цикла**. При программировании циклических алгоритмов, в которых количество повторений цикла **известно**, используется оператор **for**.

Существует два варианта оператора **for**.

1-й вариант	2-й вариант
<p>После каждого повторения параметр цикла <i>parametr</i> увеличивается на 1:</p> <pre> for parametr := n to k do begin операторы цикла end; </pre> <p>Блок-схема выполнения оператора:</p>	<p>После каждого повторения параметр цикла <i>parametr</i> уменьшается на 1:</p> <pre> for parametr := n downto k do begin операторы цикла end; </pre> <p>Блок-схема выполнения оператора:</p>

Здесь использованы следующие обозначения:

parametr – переменная или выражение одного из следующих типов: целого, логического, символьного, интервального или перечисляемого;

n – начальное значение переменной $parametr$;
 k – конечное значение переменной $parametr$.

Если в цикле требуется выполнить только один оператор, то слова **begin** и **end** можно опустить.

Схемы циклических алгоритмов с известным количеством повторений цикла изображают с помощью символа ГОСТ «модификация», внутри которого указываются все параметры цикла:

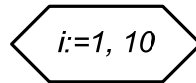
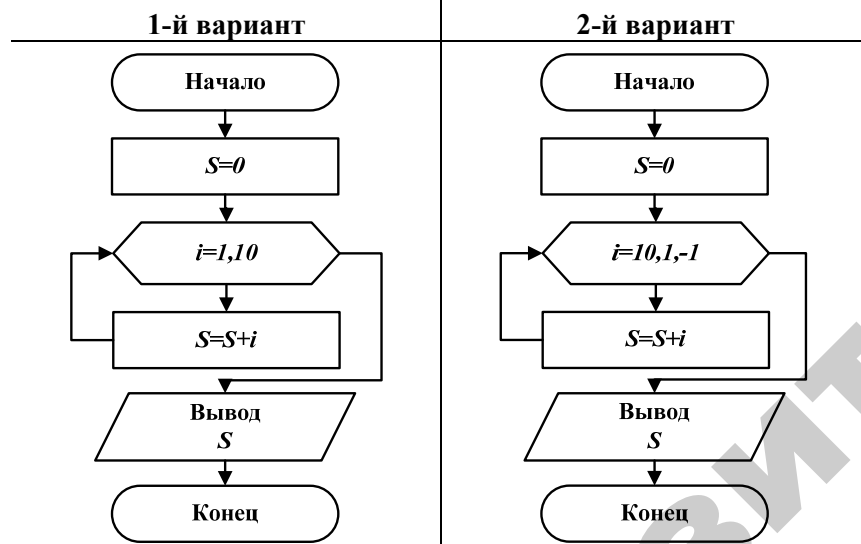


Рисунок 2. Изображение символа «модификация»

Приведем два варианта изображения блок-схемы алгоритма, показанного на рисунке 1, с использованием символа «модификация»:



Операторы, реализующие этот алгоритм:

```
s:=0;
for i:=1 to 10 do
s:=s+i;
...
```

Операторы, реализующие этот алгоритм:

```
s:=0;
for i:=10 downto 1 do
s:=s+i;
...
```

Рассмотрим несколько примеров, поясняющих использование оператора **for**.

Пример 2. Вычислить сумму ряда:

$$S = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots + \frac{1}{n^2} = \sum_{i=1}^n \frac{1}{i^2}. \quad (2)$$

Для вычисления суммы будем использовать алгоритм накопления, который заключается в том, что для вычисления суммы n слагаемых необходимо выполнить n раз выполнить оператор:

$S := S + i$ – *ое слагаемое*;

где $i = 1, 2, 3 \dots n$;

i -ое слагаемое обозначим через t , значение переменной t определяется по формуле $t = 1/i^2$.

Начальное значение переменной S равно 0.

Блок-схема алгоритма может иметь следующий вид.

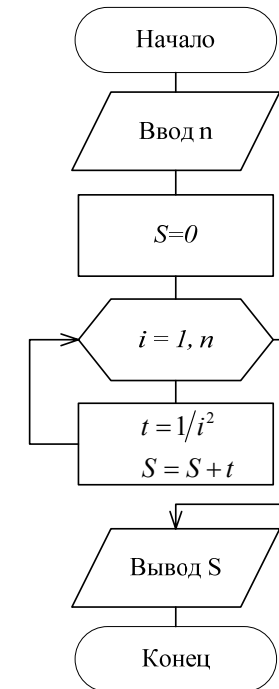


Рисунок 3. Блок-схема алгоритма вычисления выражения (2)

Для реализации этой задачи можно предложить следующий интерфейс в среде Delphi.

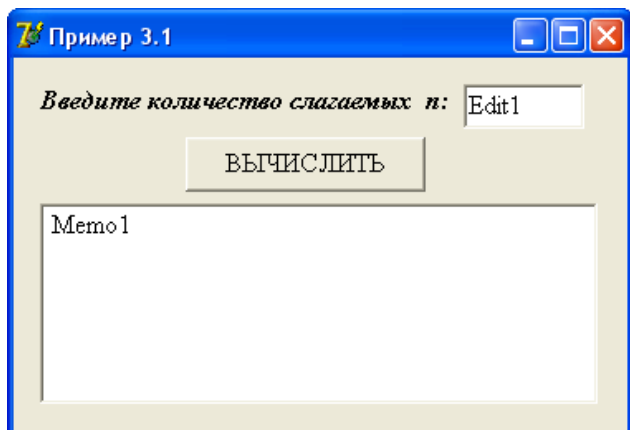


Рисунок 4. Интерфейс приложения для вычисления выражения (2)

Процедура, реализующая алгоритм на рисунке 3.

```

procedure TForm1.Button1Click(Sender: TObject);
Var i, n: integer;
      s, t: extended;
begin
  n:=StrToInt(Edit1.Text);
  S:=0;
  for i:=1 to n do
    begin
      t:=1/sqr(i);
      S:=S+t;
    end;
  Memo1.Lines.Add('S='+FloatToStrF(S, ffFixed, 8, 3));
end;

```

Пример 3. Вычислить факториал заданного числа N .

Факториал вычисляется по формуле: $N! = 1 * 2 * 3 * \dots * N$ и является частным случаем произведения. Поэтому можно записать:

$$F = N! = 1 * 2 * 3 * \dots * N = \prod_{k=1}^N k. \quad (3)$$

Блок-схема алгоритма представлена на рисунке 5.

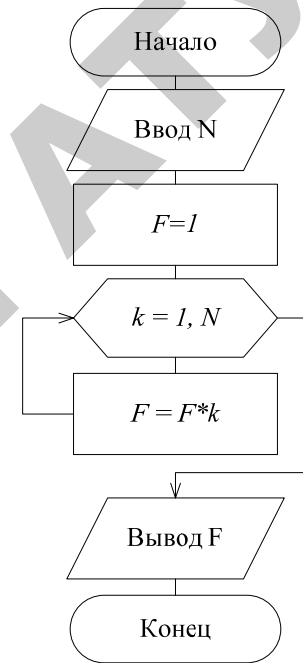


Рисунок 5. Блок-схема алгоритма вычисления выражения (3)

Для ввода числа N используем компонент Edit1, для вывода результата – компонент Memo1. Тогда процедура для вычисления $N!$ будет иметь вид:

```

procedure TForm1.Button1Click(Sender: TObject);
Var k, N, F: integer;
begin
  N:=StrToInt(Edit1.Text);
  F:=1;
  for k:=1 to N do
    F:=F*k;
  Memo1.Lines.Add('F='+IntToStr(F));
end;

```

Примечание. По определению факториал является целым числом, но при больших значениях факториала происходит переполнение переменной F , что приводит к некорректной работе программы. Поэтому для переменной F лучше использовать `тип extended`, который имеет более широкий диапазон данных.

2. Пример создания приложения

Разработать *Windows*-приложение для вычисления суммы ряда:

$$\sum_{i=1}^n (-1)^i \frac{x}{i} = -x + \frac{x}{2} - \frac{x}{3} + \dots + (-1)^n \frac{x}{n}.$$

Разработка алгоритма

Данный ряд называется знакоперевающимся. Знаки слагаемых образуют последовательность $-, +, -, +, \dots$. Введем обозначения: $znak$ – коэффициент $(-1)^i$, a – текущее (i -ое) слагаемое, S – искомая сумма. Тогда для вычисления суммы можно воспользоваться оператором:

```
S := S + a;  
где  
a := znak * x / i;  
i = 1, 2, 3, ..., n
```

Если известно значение переменной $znak$ на i -ом шаге, то на $i + 1$ шаге ее значение будет равно: $znak := -znak$;

Начальное значение переменной $znak$ равно -1 , начальное значение переменной S равно 0 . Блок-схема алгоритма данной задачи представлена на рисунке 6.

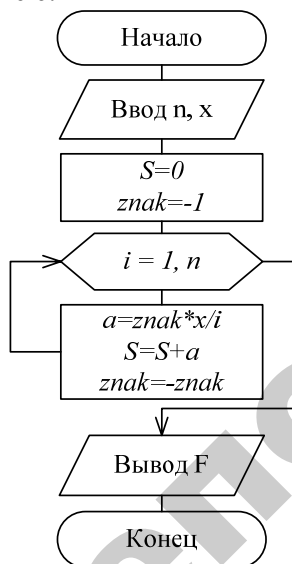


Рисунок 6. Блок-схема алгоритма решения задачи

Методические указания по созданию приложения

Сохранение проекта


1. Для нового проекта в своей рабочей папке создайте новую папку и назовите ее **LAB3**.

2. Сохраните проект.

1) **File** → **Save Project As...** (**Файл** → **Сохранить Проект как ...**).

В окне диалога **Save Unit1 As (Сохранить Unit1 как)** найти папку **LAB3** и сохраните в ней модуль под именем **UnLab3**. *Delphi* по умолчанию присвоит этому файлу тип *Delphi unit* с расширением *.pas.

2) Затем в окне диалога **Save Project1 As... (Сохранить Project1 как...)** сохраните файл проекта в папке **LAB3** под именем **PrLab3**. Здесь *Delphi* даст файлу тип *Delphi project* и расширение *.dpr.

В дальнейшем для сохранения проекта используйте пиктограмму  или опцию **Save All (Сохранить Все)** в меню **File (Файл)**.

Изменение заголовка Формы

Для изменения заголовка активизируйте окно Инспектора Объектов и на странице **Properties** в свойстве **Caption** замените заголовок **Form1** на **Лаб3**.

Размещение компонентов на Форме

Разместите компоненты на Форме согласно рисунку 7.

Для нанесения надписей на Форму используется компонент **Label**. В свойстве **Font** подберите шрифт. Для ввода исходных данных используются компоненты **Edit**. Для вывода результатов вычислений на экран будем использовать компонент **Memo**.

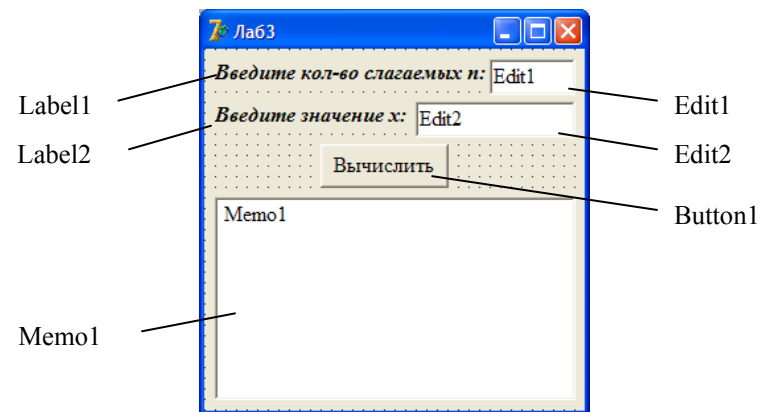


Рисунок 7. Форма с компонентами

Написание процедуры Button1Click обработки события нажатия кнопки Button1


Для создания процедуры-обработчика поместите курсор на изображение кнопки и дважды щелкните левой кнопкой мыши. На экране появится текст модуля UnLab3, в котором *Delphi* автоматически создает заготовку процедуры-обработчика события нажатия кнопки Button1:


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

Между операторами **begin** и **end** этой процедуры вставьте операторы, которые выполняют необходимые действия. После заголовка добавьте раздел объявлений **var**:

```
procedure TForm1.Button1Click(Sender: TObject);
Var i, n, znak: integer;
    s, a, x: extended;
begin
n:=StrToInt(Edit1.Text);
x:=StrToFloat(Edit2.Text);
S:=0;
znak:=-1;
for i:=1 to n do
begin
a:=znak*x/i; //Вычисление очередного члена последовательности
S:=S+a; //Вычисление суммы последовательности
znak:=-znak;
end;
Memo1.Lines.Add('при n='+IntToStr(n));
//Вывод результирующей суммы
Memo1.Lines.Add('S='+FloatToStrF(S, ffFixed, 8, 3));
end;
```

Работа с приложением

Перед запуском приложения сохраните проект: пиктограмма  главного меню или **File → Save All** (**Файл → Сохранить Все**).

Для запуска созданного приложения нажмите пиктограмму  или клавишу **F9**. При этом происходит компоновка проекта и создание выполняемого файла **PrLab3.exe**. На экране появляется панель интерфейса приложения, показанная на рисунке 8.

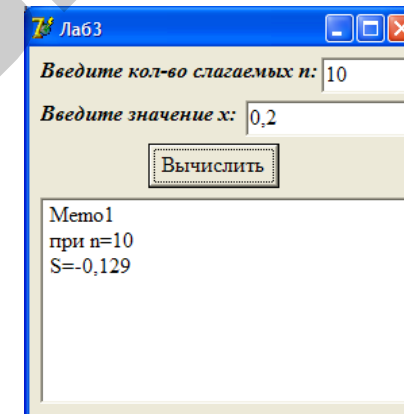


Рисунок 8. Панель интерфейса приложения

Внесите в поля соответствующие значения переменных. Убедитесь, что в *Memo* отображается результат **S**, совпадающий со значением, показанным на рисунке 8.

В случае нештатного функционирования приложения восстановить первоначальный режим работы с проектом можно выбором в меню **Run (Запуск)** опции **ProgramReset (Сброс программы)** или нажатием клавиши **Ctrl+F2**.

Вопросы для самоконтроля

1. Какие вычислительные процессы называются *циклическими*?
2. Какие формы оператора цикла **for** вы знаете, чем они отличаются между собой?
3. Какой тип должен иметь параметр цикла в цикле **for**?
4. Какой тип должны иметь начальное и конечное значения в цикле **for**?

Варианты индивидуальных заданий разного уровня:

1 уровень – выполните задание 1;

2 уровень – выполните задания 1, 2;

3 уровень – выполните задания 1, 2, 3.

Вариант 1

1. Вычислить сумму первых n элементов ряда:

$$S = -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} - \frac{1}{\sqrt{4}} + \dots + (-1)^n \frac{1}{\sqrt{n+1}}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{N^2} \prod_{k=1}^N \ln(x^2 + k).$$

Число N и значение аргумента x ввести с клавиатуры.

3. Напечатать таблицу температур по шкале Цельсия t_C от -5 до 10 градусов и их эквивалентов по шкале Фаренгейта t_F с шагом 1°C . Для перевода используйте формулу: $t_F = 1,8t_C + 32$.

Вариант 2

1. Вычислить сумму элементов ряда

$$S = \frac{1}{\sqrt{1+x}} + \frac{1}{2\sqrt{1+2x}} + \frac{1}{3\sqrt{1+3x}} + \dots + \frac{1}{n\sqrt{1+nx}}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \left(x - \frac{kx}{2} \right).$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти сумму кубов всех целых от 20 до n :
 $20^3 + 21^3 + 22^3 + \dots + n^3$.

Значение n вводится с клавиатуры.

Вариант 3

1. Вычислить сумму первых n элементов ряда

$$S = e^{-x^2} + \frac{e^{-4x^2}}{2} + \frac{e^{-9x^2}}{3} + \dots + \frac{e^{-n^2x^2}}{n}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \frac{x+k}{k^2}$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти произведение всех целых чисел от c до 20 . Значение c вводится с клавиатуры: $c \leq 20$.

Вариант 4

1. Вычислить сумму первых n элементов ряда

$$S = \frac{1}{11} - \frac{2}{17} + \frac{3}{23} - \frac{4}{29} + \dots + (-1)^{n+1} \frac{n}{6n+5}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N (x+k)^2.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вычислить приближенно площадь одной арки синусоиды.
Примечание. Разделите отрезок от 0 до π на 10 частей. Затем найдите сумму площадей полученных прямоугольников с основанием $\pi/10$ и высотой, равной значению функции $y = \sin x$ на правой границе каждого интервала.

Вариант 5

1. Вычислить сумму первых n элементов ряда

$$S = \cos(x+1) + \cos(x+2) + \cos(x+3) + \dots + \cos(x+n).$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \sqrt{N} \prod_{k=1}^N \frac{x^2}{k-1}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Определить сумму первых t членов геометрической прогрессии, первый член которой равен a_1 , знаменатель q .

Вариант 6

1. Вычислить сумму первых n элементов ряда
- $$S = -\frac{\cos(x)}{5} + \frac{\cos(2x)}{9} - \frac{\cos(3x)}{37} + \dots + (-1)^n \frac{\cos(nx)}{1+4n^2}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \frac{x+k}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Напечатать таблицу умножения на число p . Значение p вводится с клавиатуры: $1 \leq p \leq 9$.

Вариант 7

1. Вычислить сумму первых n элементов ряда

$$S = |x+1| + |x+4| + |x+9| + \dots + |x+n^2|.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = 2N \prod_{k=1}^N \frac{x^2+k}{x}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти произведение всех целых чисел от a до b . Значения a и b вводятся с клавиатуры.

Вариант 8

1. Вычислить сумму первых n элементов ряда

$$S = -\frac{1}{1-\ln 1} + \frac{1}{2-\ln 2} - \frac{1}{3-\ln 3} + \dots + (-1)^n \frac{1}{n-\ln n}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{N} \prod_{k=1}^N \frac{k^2}{k+1} x.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Каждая бактерия делится на две в течение одной минуты. В начальный момент времени имеется одна бактерия. Составить программу для расчета количества бактерий через заданное целое количество минут.

Вариант 9

1. Вычислить сумму первых n элементов ряда

$$S = \frac{2 \sin^2 nx}{\sqrt{2}} + \frac{3 \sin^2 2x}{2\sqrt{3}} + \dots + \frac{(n+1) \sin^2 nx}{n\sqrt{n+1}}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{6}{N} \prod_{k=1}^N \sqrt{x+k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вывести на экран все целые числа от a до b , кратные 3. Значения a и b вводятся с клавиатуры.

Вариант 10

1. Вычислить сумму первых n элементов ряда

$$S = 1 - \frac{1}{2\sqrt[3]{2}} + \frac{1}{3\sqrt[3]{3}} - \frac{1}{4\sqrt[3]{4}} + \dots + (-1)^{n-1} \frac{1}{n\sqrt[3]{n}}.$$

Число n с клавиатуры.

2. Вычислить произведение

$$P = N^2 \prod_{k=1}^N e^{-kx}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти среднее арифметическое всех целых чисел от 20 до c . Значение c вводится с клавиатуры: $c \geq 20$.

Вариант 11

1. Вычислить сумму первых n элементов ряда

$$S = \frac{1}{x^2+1} + \frac{\sqrt{2}}{x^2+4} + \frac{\sqrt{3}}{x^2+9} + \dots + \frac{\sqrt{n}}{x^2+n^2}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \frac{\sqrt{|x-k|}}{k^2}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Составить программу, вычисляющую и печатающую n ($n \geq 3$) первых чисел Фибоначчи. Числа Фибоначчи образуются следующим образом: первый и второй члены последовательности равны 1, каждый следующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13, ...).

Вариант 12

1. Вычислить сумму первых n элементов ряда

$$S = \frac{1}{\sqrt{x+e^2}} + \frac{1}{\sqrt{2x+e^4}} + \dots + \frac{1}{\sqrt{nx+e^{2n}}}.$$

Число n и значение аргумента x ($x \geq 0$) ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{n} \prod_{k=1}^n (x-k)^2$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вывести на экран все двузначные числа, обладающие следующим свойством: если к сумме цифр числа прибавить квадрат этой суммы, то получится снова искомое число.

Вариант 13

1. Вычислить сумму первых n элементов ряда

$$S = \sin(\pi x) - \frac{\sin(2\pi x)}{2} + \frac{\sin(3\pi x)}{3} \dots + (-1)^{n+1} \frac{\sin(n\pi x)}{n}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \sqrt{2N} \prod_{k=1}^N \frac{e^{-kx}}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Напечатать таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, ..., 10 фунтов (1 фунт = 453 г).

Вариант 14

1. Вычислить сумму первых n элементов ряда

$$S = (1 + \sqrt{|x|}) + (2 + \sqrt{|x|}) + \dots + (n + \sqrt{|x|}).$$

Число N и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{\sqrt{N}} \prod_{k=1}^N \sin(kx).$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вычислить значение выражения

$$\left(\left(\dots (20^2 - 19^2)^2 - 18^2 \right)^2 - \dots - 1^2 \right)^2.$$

Вариант 15

1. Вычислить сумму первых n элементов ряда

$$S = -\sin \frac{\pi x}{3} + \frac{1}{2} \sin \frac{2\pi x}{3} - \frac{1}{3} \sin \pi x + \dots + \frac{(-1)^n}{n} \sin \frac{n\pi x}{3}.$$

Число N и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{N} \prod_{k=1}^N \frac{(k+1)^2}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вывести на экран все двузначные числа, сумма квадратов цифр которых делится на 13.

Вариант 16

1. Вычислить сумму первых n элементов ряда

$$S = -e^x + \frac{e^{4x}}{2} - \frac{e^{9x}}{3} + \dots + (-1)^n \frac{e^{n^2x}}{n}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{\sqrt{\pi N}} \prod_{k=1}^N (2kx + \sin(kx)).$$

Число N и значение аргумента x ввести с клавиатуры.

3. Плотность воздуха убывает с высотой по закону $\rho = \rho_0 e^{-hz}$, где ρ – плотность на высоте h метров; $\rho_0 = 1,25 \text{ кг/м}^3$, $z = 1,25 \cdot 10^{-4}$. Напечатать таблицу зависимости плотности от высоты для значений от 0 до 1000 м через каждые 100 м.

Вариант 17

1. Вычислить сумму первых n элементов ряда

$$S = \frac{\cos(x)}{5} + \frac{\cos(2x)}{9} + \frac{\cos(3x)}{37} + \dots + \frac{\cos(nx)}{1 + 4n^2}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{2}{N} \prod_{k=1}^N \left(x + \frac{k}{2}\right)^2.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти произведение целых чисел из промежутка от c до 30, кратных четырем. Значение c вводится с клавиатуры.

Вариант 18

1. Вычислить сумму первых n элементов ряда

$$S = 2 \sin^2 nx - \frac{3 \sin^2 2x}{2} + \dots + (-1)^{n+1} \frac{(n+1) \sin^2 nx}{n}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{6}{N} \prod_{k=1}^N \sqrt{x+k}.$$

3. Определить сумму первых t членов арифметической прогрессии, первый член которой равен a_1 , разность – d . Значения t , a_1 , d задать самостоятельно и ввести с клавиатуры.

Вариант 19

1. Вычислить сумму первых n элементов ряда

$$S = -\frac{1}{\sqrt{x+e^2}} + \frac{1}{\sqrt{2x+e^4}} - \frac{1}{\sqrt{3x+e^4}} + \dots + (-1)^n \frac{1}{\sqrt{nx+e^{2n}}}.$$

Число n и значение аргумента x ($x \geq 0$) ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N (2k+1)x^2.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Составить программу поиска трехзначных чисел, которые при делении на 47 дают в остатке заданное число k .

Вариант 20

1. Вычислить сумму первых n элементов ряда

$$S = -\cos(x+1) + \cos(x+2) - \cos(x+3) + \dots + (-1)^n \cos(x+n).$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \sin(0.05+x) \sqrt{1+k}.$$

3. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько клеток будет через k часов (k – натуральное число), если первоначально была одна амеба.

Вариант 21

1. Вычислить сумму первых n элементов ряда

$$S = 1 + \frac{1}{2\sqrt[3]{2}} + \frac{1}{3\sqrt[3]{3}} + \frac{1}{4\sqrt[3]{4}} + \dots + \frac{1}{n\sqrt[3]{n}}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{3}{2} + \prod_{k=1}^N k^2 \sin(kx).$$

3. Составить программу вычисления при заданных x и a значения функции y вида:

$$y = (((((x-a)x-a)x-a)x-a)x-a).$$

Вариант 22

1. Вычислить сумму первых n элементов ряда:

$$S = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{4}} + \dots + \frac{1}{\sqrt{n+1}}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{N} \prod_{k=1}^N x \frac{(k+1)^2}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вычислить приближенно площадь фигуры, образованной кривой $y = 0,3(x-1)^2 + 4$, осью абсцисс и двумя прямыми $y = 1$ и $y = 10$.

Примечание. Разделите фигуру на 10 частей и найдите сумму площадей полученных прямоугольников.

Вариант 23

1. Вычислить сумму первых n элементов ряда

$$S = |x+1| - |x+4| + |x+9| - |x+16| + \dots + (-1)^{n+1} |x+n^2|.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = 2N \prod_{k=1}^N \frac{x^2}{k+1}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Вывести на экран все натуральные числа из интервала от 100 до 500, сумма цифр которых равна k (k ввести с клавиатуры).

Вариант 24

1. Вычислить сумму первых n элементов ряда

$$S = (1 + \sqrt{|x|}) - (4 + \sqrt{|x|}) + \dots + (-1)^{n+1} (n^2 + \sqrt{|x|}).$$

Число N и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \sqrt{N} \prod_{k=1}^N \sin(k\pi x).$$

Число N и значение аргумента x ввести с клавиатуры.

3. Даны натуральные числа x и y . Вычислить произведение $x \cdot y$, используя лишь операцию сложения.

Вариант 25

1. Вычислить сумму первых n элементов ряда

$$S = \sin \frac{\pi x}{3} + \frac{1}{2} \sin \frac{2\pi x}{3} + \frac{1}{3} \sin \pi x + \dots + \frac{1}{n} \sin \frac{n\pi x}{3}.$$

Число N и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = N \prod_{k=1}^N \frac{\sqrt{x+1}}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Определить суммарный объем в литрах двенадцати вложенных друг в друга шаров со стенками толщиной 5 мм. Внутренний диаметр внутреннего шара равен 10 см. Принять, что шары вкладываются друг в друга без зазоров.

Вариант 26

1. Вычислить сумму первых n элементов ряда

$$S = \frac{1}{11} + \frac{2}{17} - \frac{3}{23} + \dots + \frac{n}{6n+5}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{2\sqrt{N}} \prod_{k=1}^N \sin(x+k)^2.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти среднее арифметическое всех целых чисел от a до 200. Значение x вводится с клавиатуры: $a \leq 200$.

Вариант 27

1. Вычислить сумму элементов ряда

$$S = -\frac{1}{\sqrt{1+x}} + \frac{1}{2\sqrt{1+2x}} - \frac{1}{3\sqrt{1+3x}} + \dots + (-1)^n \frac{1}{n\sqrt{1+nx}}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = N \prod_{k=1}^N \left(\sin x - \frac{kx}{2} \right).$$

3. Число N и значение аргумента x ввести с клавиатуры.
Напечатать таблицу перевода расстояний в дюймах в сантиметры для значений 10, 11, ..., 22 дюйма (1 дюйм = 25,4 мм).

Вариант 28

1. Вычислить сумму первых n элементов ряда

$$S = \sin(\pi x) + \frac{\sin(2\pi x)}{2} + \frac{\sin(3\pi x)}{3} + \dots + \frac{\sin(n\pi x)}{n}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \frac{1}{\sqrt{N}} \prod_{k=1}^N \frac{e^{-kx}}{k}.$$

Число N и значение аргумента x ввести с клавиатуры.

3. Найти среднее арифметическое всех целых чисел от 100 до b . Значение b вводится с клавиатуры: $b \geq 100$.

Вариант 29

1. Вычислить сумму первых n элементов ряда

$$S = -\frac{1}{x^2+1} + \frac{\sqrt{2}}{x^2+4} - \frac{\sqrt{3}}{x^2+9} + \dots + (-1)^n \frac{\sqrt{n}}{x^2+n^2}.$$

Число n и значение аргумента x ввести с клавиатуры.

2. Вычислить произведение

$$P = \prod_{k=1}^N \frac{\sqrt{|x-k|}}{k}.$$

3. Вычислить сумму $\sqrt{1 + \sqrt{2 + \sqrt{3 + \dots + \sqrt{50}}}}$.

Вариант 30

1. Вычислить сумму первых n элементов ряда

$$S = \frac{1}{1 - \ln 1} + \frac{1}{2 - \ln 2} + \dots + \frac{1}{n - \ln n}.$$

Число n ввести с клавиатуры.

2. Вычислить произведение

$$P = \sqrt{N} \prod_{k=1}^N \frac{k^2}{k+1} \sin x.$$

3. Напечатать таблицу умножения на 12:

$$1 \times 12 = 12$$

$$2 \times 12 = 24$$

...

$$9 \times 12 = 108$$

ЛАБОРАТОРНАЯ РАБОТА №4

Решение прикладных задач, основанных на использовании циклических алгоритмов с неизвестным количеством повторений цикла

Цель – приобрести навыки решения прикладных задач, основанных на использовании циклических алгоритмов с неизвестным количеством повторений цикла, в визуальной среде *Delphi*.

Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания приложения. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

Вычислить значение суммы S бесконечного ряда:

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \dots = \sum_{i=1}^{\infty} \frac{1}{i}. \quad (1)$$

Суммирование продолжать до тех пор, пока очередное слагаемое не станет меньше заданного числа $\varepsilon = 10^{-4}$, т.е.

$$\frac{1}{i} < \varepsilon. \quad (2)$$

Как следует из условия задачи, количество слагаемых в выражении (1) **не известно**. Процесс вычисления суммы (1) должен продолжаться до тех пор, пока не выполнится условие окончания цикла (2). Возможные варианты блок-схемы алгоритма, который выполняет необходимые действия, приведены на рисунке 1. Обращаем внимание на то, что проверка условия завершения цикла в первом варианте производится **до выполнения операторов цикла**, а во втором варианте – **после выполнения операторов цикла**.

Такие алгоритмы, в которых количество повторений цикла заранее **не известно** и может быть определено только в процессе вычислений, называются **итерационными** (итерация – последовательное приближение). Итерационные циклы программируют с использованием операторов **while** или **repeat**.

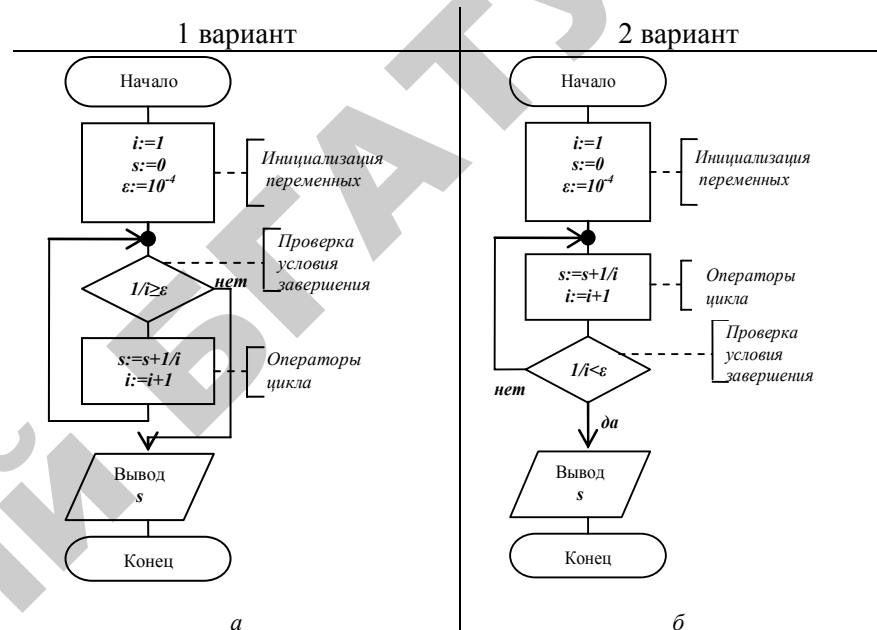


Рисунок 1. Блок-схемы алгоритма вычисления выражения (1)

Оператор while (оператор цикла с предусловием)

Используется в тех случаях, когда операторы цикла должны выполняться **после проверки условия завершения цикла**. Блок-схема, иллюстрирующая работу оператора **while**, показана на рисунке 2.

Оператор цикла **while** записывается в следующем виде:

```
while условие do
begin
    операторы
    цикла
end;
```

Если в цикле необходимо выполнить только один оператор, слова **begin** и **end** можно не указывать.

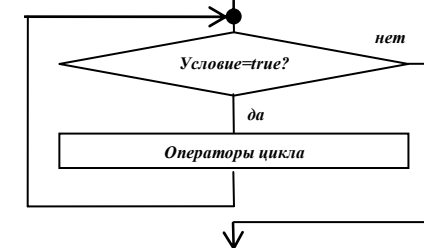


Рисунок 2. Блок-схема алгоритма работы оператора while

Здесь **условие** – логическое выражение. **Перед каждым выполнением операторов цикла** вычисляется логическое выражение – **условие завершения цикла**. Если результат его вычисления равен

true (истина), операторы цикла выполняются, если же результат вычисления логического выражения оказался равен **false** (ложь) – происходит выход из цикла.

Приведем последовательность операторов, реализующих вычисление выражения (1) в соответствии с рисунком 1, *a* с помощью оператора **while**:

```
s := 0; // инициализация s
i = 1; // инициализация i
eps := 1e-4; // инициализация ε
while 1/i >= eps do
begin
s := s+1/i; // операторы
i := i+1; // цикла
end;
...
```

Оператор repeat (оператор цикла с постусловием)

Используется, когда операторы цикла должны выполняться **до проверки условия завершения цикла**. Блок-схема, поясняющая работу оператора **repeat**, показана на рисунке 3.

Оператор цикла **repeat** записывается в следующем виде:

```
repeat
операторы
цикла
until условие;
```

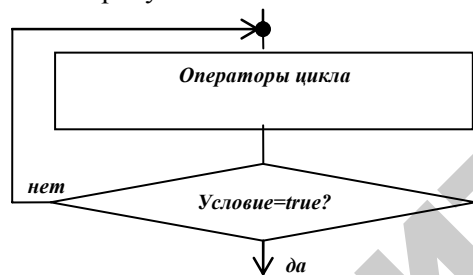


Рисунок 3. Блок-схема алгоритма работы оператора repeat

Здесь **условие** – логическое выражение. **После выполнения операторов цикла** вычисляется логическое выражение – **условие завершения цикла**. Если результат его вычисления равен **false** (ложь) операторы цикла выполняются еще раз. Если же результат вычисления логического выражения оказался равен **true** (истина) – происходит выход из цикла.

Таким образом, операторы цикла выполняются до тех пор, пока **условие завершения цикла** не станет истинным (**true**).

Приведем последовательность операторов, реализующих вычисление выражения (2) в соответствии с рисунком 2, *b* с помощью оператора **repeat**:

```
...
s := 0; // инициализация s
i = 1; // инициализация i
eps := 1e-4; // инициализация ε
repeat
s := s+1/i; // операторы
i := i+1; // цикла
until 1/i < eps;
...
```

Для более гибкого управления работой циклов могут использоваться операторы **break** и **continue**.

Оператор **break** прерывает выполнение цикла и осуществляет переход на оператор, расположенный за последним оператором цикла:

<pre>while условие1 do // начало цикла begin оператор1; оператор2; ... if условие2 then break; ... операторn; end; // конец цикла оператор;</pre>	<pre>repeat // начало цикла оператор1; оператор2; ... if условие2 then break; ... операторn until условие1; // конец цикла оператор;</pre>
---	--

2. Примеры создания приложения

Пример 1. Разработать *Windows*-приложение для вычисления *S* с заданной погрешностью ϵ значения суммы бесконечного ряда

$$s = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!}. \quad (3)$$

Суммирование продолжать до тех пор, пока очередное слагаемое по абсолютной величине не станет меньше заданного числа ϵ , т. е.

$$\left| \frac{1}{n!} \right| < \epsilon. \quad (4)$$

Вывести значение суммы S и, для контроля правильности вычислений, точное значение суммы ряда $S = 1/e$.

Разработка алгоритма

При решении подобных задач целесообразно использовать рекуррентную формулу для получения очередного члена ряда.

Рекуррентной называется формула, выражающая каждый член последовательности через предыдущие члены этой последовательности. Используется чаще всего в целях избавления от больших чисел при вычислении суммы членов последовательности. Как правило, рекуррентная формула имеет вид:

$$t_n = C \cdot t_{n-1}. \quad (5)$$

Из формулы (5), зная общий член последовательности, можно найти коэффициент C , на который нужно умножить каждый предыдущий член ряда, чтобы найти следующий:

$$C = \frac{t_n}{t_{n-1}}. \quad (6)$$

Запишем выражение для общего члена ряда:

$$t_n = (-1)^n \frac{1}{n!}. \quad (7)$$

Подставим в формулу (7) вместо n величину $n - 1$:

$$t_{n-1} = (-1)^{n-1} \frac{1}{(n-1)!}. \quad (8)$$

С учетом полученных выражений, найдем коэффициент C по формуле (6):

$$C = \frac{t_n}{t_{n-1}} = \frac{(-1)^n (n-1)!}{(n)! (-1)^{n-1}} = \frac{(-1)^{n-n+1} \cdot 1 \cdot 2 \cdot \dots \cdot (n-1)}{1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n} = -\frac{1}{n}. \quad (9)$$

Итак, получаем искомую рекуррентную формулу:

$$t_n = -\frac{1}{n} t_{n-1}. \quad (10)$$

Используя это соотношение можно последовательно вычислить все члены ряда. При этом не возникает необходимости в выполнении

в явном виде операций вычисления факториала и возведения в степень.

Блок-схема алгоритма вычисления суммы ряда приведена на рисунке 4. Для определения начального значения суммы s подставим в формулу общего члена (7) $n = 0$, тогда $s = 1$.

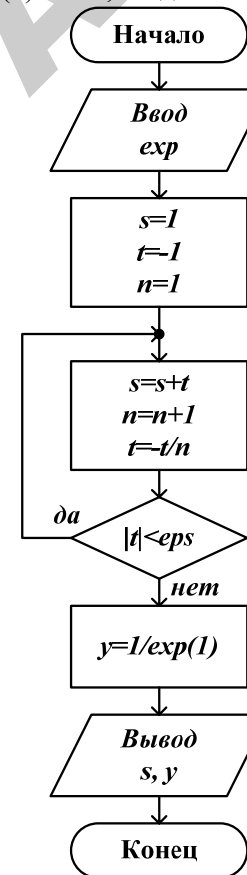


Рисунок 4. Блок-схема вычисления суммы бесконечного ряда

Методические рекомендации по созданию приложения

Сохранение проекта

1. Для нового проекта в своей рабочей папке создайте новую папку и назовите ее **LAB4_1**.
2. Сохраните проект.

1) **File**→ **Save Project As...** (**Файл** → **Сохранить Проект как ...**).

В окне диалога **Save Unit1 As (Сохранить Unit1 как)** найти папку **LAB4_1** и сохраните в ней модуль под именем **UnLab4_1 Delphi** по умолчанию присвоит этому файлу тип Delphi unit с расширением *.pas.

2) Затем в окне диалога **Save Project1 As... (Сохранить Project1 как...)** сохраните файл проекта в папке **LAB4_1** под именем **PrLab4_1** Здесь *Delphi* даст файлу тип Delphi project и расширение *.dpr.

Изменение заголовка Формы


Для изменения заголовка активизируйте окно Инспектора Объектов и на странице *Properties* в свойстве *Caption* замените заголовок **Form1** на **Лаб.работа № 4**. Убедитесь, что одновременно изменился заголовок окна Формы.

Размещение компонентов на Форме

Разместите компоненты на Форме согласно рисунку 5.

Для нанесения надписей на Форму используются компоненты Label. Для ввода исходных данных используются компоненты Edit.

Поместите на Форму кнопку, нажатие которой приведет к выводу таблицы значений *X* и *Y* в компонент Memo1. В свойстве *Caption* Инспектора Объектов заменить надпись на кнопке **Button1** на **Вычислить**

Для вывода результатов вычислений на экран будем использовать компонент **Memo**. Выберите в Палитре Компонентов на странице **Standart** пиктограмму  и поместите на Форму согласно рисунку 5. Для компонента **Memo** в свойстве **ScrollBars** установите значение **ssVertical** – появится вертикальная линейка прокрутки в компоненте **Memo**.

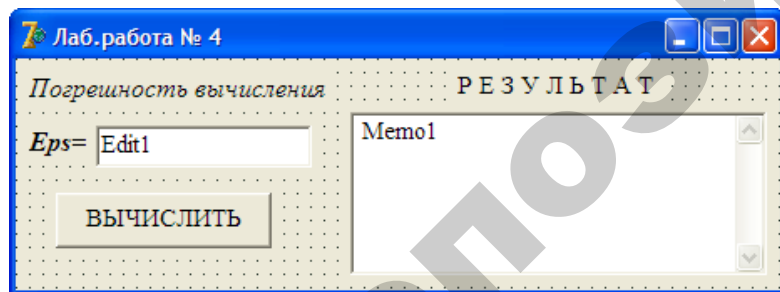



Рисунок 5. Форма с компонентами

Сохраните проект – для этого используйте пиктограмму  или опцию **Save All (Сохранить Все)** в меню **File (Файл)**.

Написание процедуры **Button1Click** обработки события нажатия кнопки **ВЫЧИСЛИТЬ**

Для создания процедуры-обработчика поместите курсор на изображение кнопки с надписью **ВЫЧИСЛИТЬ** и дважды щелкните по ней левой кнопкой мыши. На экране появится текст модуля UnLab4, в котором *Delphi* автоматически создает заготовку процедуры-обработчика события нажатия кнопки **Button1**:


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

Между операторами **begin** и **end** этой процедуры вставьте операторы, которые выполняют необходимые действия. После заголовка добавьте раздел объявлений **var**:

```
procedure TForm1.Button1Click(Sender: TObject);
var eps,s,t,y: extended;
    n: integer;

begin
eps:=StrToFloat(Edit1.Text);
s:=1; t:=-1; n:=1;
repeat
s:=s+t;
n:=n+1;
t:=-t/n;
until abs(t) < eps;
y:=1/exp(1);
Memo1.Lines.Add('S='+FloatToStrF(S, ffFixed, 9, 8));
Memo1.Lines.Add('Y='+FloatToStrF(y, ffFixed, 9, 8));
end;
```

Работа с приложением

Перед запуском приложения сохраните проект: пиктограмма  главного меню) или **File** → **Save All (Файл** → **Сохранить Все)**.

Для запуска созданного приложения нажмите пиктограмму или клавишу **F9**. При этом происходит компоновка проекта и создание выполняемого файла **PrLab4.exe**. На экране появляется панель интерфейса приложения, показанная на рисунке 6.

Пользуясь контрольным значением суммы, выясните, как изменяются значения **S** и **Y** при изменении значения **Eps** от 10^{-1} до 10^{-8}

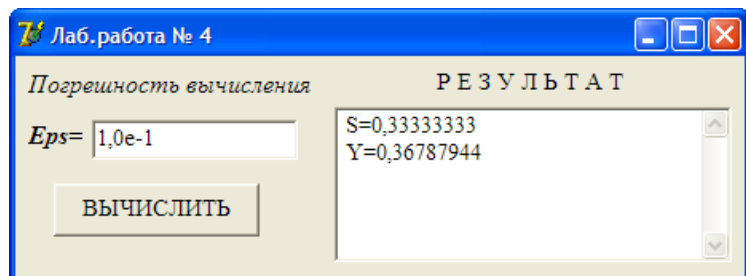


Рисунок 6. Панель интерфейса приложения

Пример 2. Разработать *Windows*-приложение для реализации следующей задачи.

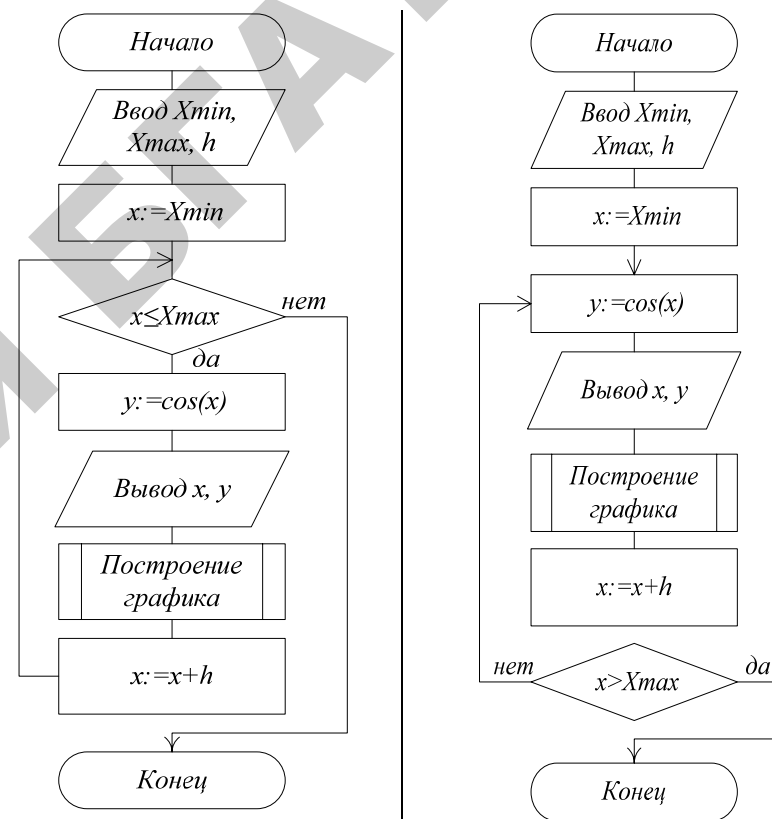
- 1) Вычислить и вывести на экран значения функции $y = \cos x$ при изменении аргумента от x_{\min} до x_{\max} с шагом h . Пределы изменения аргумента x и величину шага h задать самостоятельно.
- 2) Вывести на экран график функции $y = \cos x$.

Разработка алгоритма

Вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом называется **табулированием функции**. Необходимость в табулировании возникает при решении достаточно широкого круга задач. Например, необходимость в табулировании возникает при построении графиков функции на экране компьютера.

Итак, по условию задачи необходимо протабулировать функцию $y = \cos(x)$ на интервале $[x_{\min}, x_{\max}]$ с шагом h . Поскольку программа должна многократно вычислять значения одной и той же функции, то разумно составить циклический алгоритм. В данной постановке задачи наиболее удобны операторы WHILE или REPEAT ... UNTIL, поскольку для оператора FOR необходимо предварительно вычислить количество шагов.

На рисунке 7 представлены блок-схемы алгоритмов, которые используют для организации цикла операторы WHILE и REPEAT ... UNTIL.



а) с использованием оператора WHILE;

б) с использованием оператора REPEAT...UNTIL

Рисунок 7. Блок-схемы алгоритмов

Методические рекомендации по созданию приложения

Сохранение проекта

1. Для нового проекта в своей рабочей папке создайте новую папку и назовите ее **LAB4_2**.
2. Сохраните проект.

1) **File** → **Save Project As...** (**Файл** → **Сохранить Проект как ...**). В окне диалога **Save Unit1 As (Сохранить Unit1 как)** найдите папку **LAB4_2** и сохраните в ней модуль под именем **UnLab4_2.Delphi** по умолчанию присвоит этому файлу тип Delphi unit с расширением *.pas.

2) Затем в окне диалога **Save Project1 As... (Сохранить Project1 как...)** сохраните файл проекта в папке **LAB4_2** под именем **PrLab4_2**. Здесь *Delphi* даст файлу тип Delphi project и расширение *.dpr.

Изменение заголовка Формы

Для изменения заголовка активизируйте окно Инспектора Объектов и на странице **Properties** в свойстве **Caption** замените заголовок **Form1** на **Лаб. работа № 4**. Убедитесь, что одновременно изменился заголовок окна Формы.

Размещение компонентов на Форме

Разместите компоненты на Форме согласно рисунку 8.

Для нанесения надписей на Форму используются компоненты Label. Для ввода исходных данных используются компоненты Edit.

Поместите на Форму кнопку, нажатие которой приведет к выводу таблицы значений **X** и **Y** в компонент Memo1. В свойстве **Caption** Инспектора Объектов заменить надпись на кнопке **Button1** на **ВЫПОЛНИТЬ**.

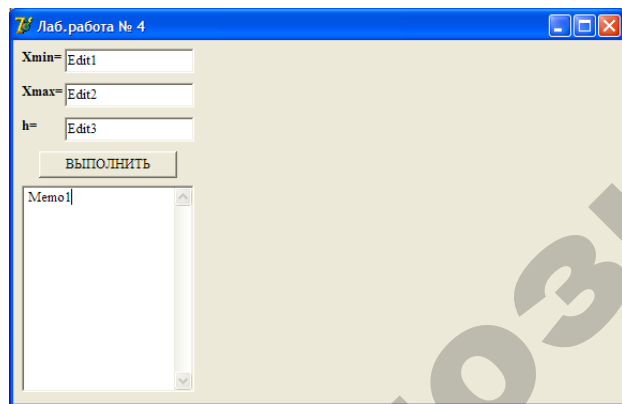



Рисунок 8. Размещение компонентов на Форме

Для вывода результатов вычислений на экран будем использовать компонент **Memo**. Выберите в Палитре Компонентов на странице **Standart** пиктограмму  и поместите на Форму согласно ри-


сунку 8. Для компонента **Memo** в свойстве **ScrollBars** установите значение **ssVertical** – появится вертикальная линейка прокрутки в компоненте **Memo**.

Сохраните проект – для этого используйте пиктограмму  или опцию **Save All (Сохранить Все)** в меню **File (Файл)**.

Построение графика функции $y = \cos x$

Для построения и вывода графиков и диаграмм в Delphi предназначен компонент **TChart**.

Построение графика производится после вычисления таблицы значений функции $y = f(x)$ на интервале $[X_{\min}, X_{\max}]$. Полученная таблица передается в объект **Series** компонента **TChart** с помощью метода **Add**. Компонент **TChart** выполняет всю работу по отображению графиков: строит и размечает оси, рисует координатную сетку, отображает переданную таблицу в виде графиков (так же диаграмм). С помощью встроенного редактора **EditingChart** компоненту **TChart** передаются данные о толщине, стиле и цвете линий, параметрах шрифта подписей, шагах разметки координатной сетки и другие настройки.

Поместите на Форму компонент **TChart**. Выберите в Палитре Компонентов на странице **Additional** пиктограмму  и поместите на Форму согласно рисунку 9.

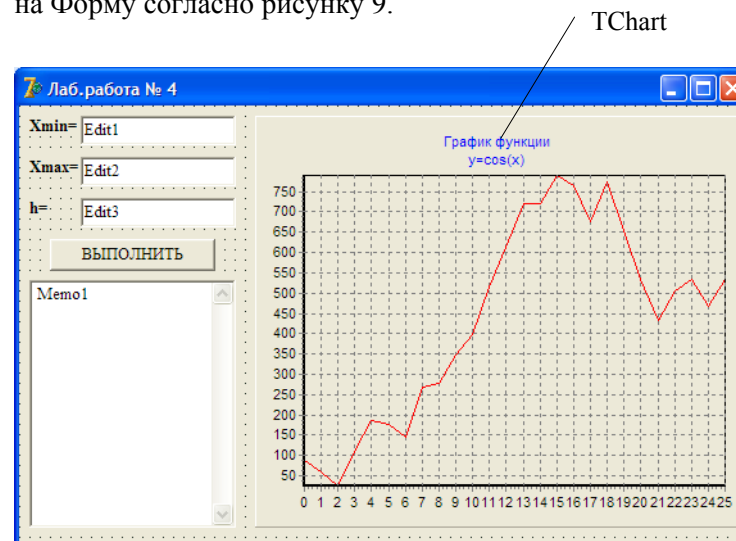


Рисунок 9. Форма с компонентом TChart

Работа с компонентом TChart

Дважды щелкните по компоненту **TChart**. Откроется окно редактирования **EditingChart1** (рисунок 10). Закладка **Series** открывает доступ к так называемым *сериям* – объектам, которые отображают наборы чисел в графическом виде.

Для создания новой серии **Series1** щелкните по кнопке **Add** на странице **Series**. В появившемся диалогом окне **TeeChart Gallery** выберите пиктограмму с надписью **Line** – график будет выводиться в виде линий. Отключите переключатель **3D**, тогда график не будет выводиться в трехмерном виде. Нажмите на кнопку **OK** – появится новая серия с названием **Series1**.

Измените название серии **Series1** – для этого нажмите кнопку **Title...** В появившемся редакторе набрать имя функции – **cos(x)**. Для изменения надписи над графиком на странице **Titles** в редакторе набрать **График функции $y = \cos(x)$** .

Нажимая различные кнопки меню, изучите другие возможности окна редактирования **EditingChart**.

После закрытия редактора **EditingChart** компонент **TChart** будет содержать примерный вид графика.

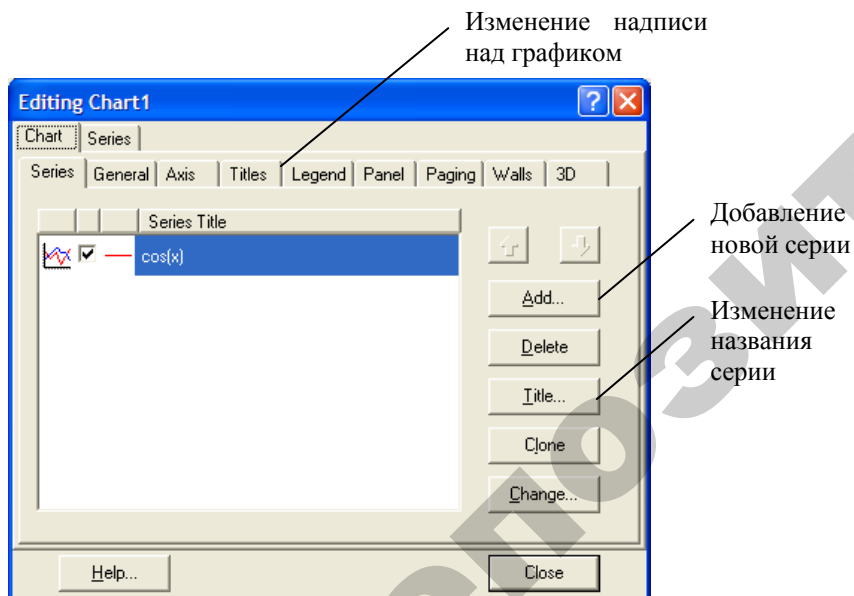


Рисунок 10. Окно редактирования EditingChart1

Написание процедуры Button1Click обработки события нажатия кнопки ВЫПОЛНИТЬ

Для создания процедуры-обработчика поместите курсор на изображение кнопки с надписью **Таблица** и дважды щелкните по ней левой кнопкой мыши. На экране появится текст модуля UnLab4, в котором *Delphi* автоматически создает заготовку процедуры-обработчика события нажатия кнопки **Button1**:

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

Между операторами **begin** и **end** этой процедуры вставьте операторы, которые выполняют необходимые действия. После заголовка добавьте раздел объявлений **var**:

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y, Xmin, Xmax, h: extended;
begin
  Xmin:=StrToFloat(Edit1.Text);
  Xmax:=StrToFloat(Edit2.Text);
  h:=StrToFloat(Edit3.Text);
  x:=Xmin;
  Memo1.Clear;           //Очистка компонента Memo1
  Series1.Clear;        //Очистка графика
  //Вывод в Memo1 поясняющего текста
  Memo1.Lines.Add('Таблица значений функции
y=cos(x)');
  While x<=Xmax do
  begin
    y:=cos(x);
    //Вывод в Memo1 таблицы значений X и Y
    Memo1.Lines.Add('прих='+FloatToStrF(x, ffFixed, 8, 2)+
      ' Y='+FloatToStrF(y, ffFixed, 8, 2));
    Series1.AddXY(x, y); //Вывод точки на график
    x:=x+h;
  end;
end;
```


Процедура `Series1.AddXY(x,y)` выводит в область `TChart` точки с координатами X и Y и строит по ним график функции $y = \cos(x)$.

Работа с приложением

Перед запуском приложения сохраните проект: пиктограмма (главного меню) или `File → Save All (Файл → Сохранить Все)`.

Для запуска созданного приложения нажмите пиктограмму или клавишу `F9`. При этом происходит компоновка проекта и создание выполняемого файла `PrLab4.exe`. На экране появляется панель интерфейса приложения, показанная на рисунке 11.

Внесите в поля компонентов `Edit1`, `Edit2`, `Edit3` начальное, конечное значения аргумента X и шаг изменения аргумента h . Убедитесь, что при нажатии на кнопку **Вычислить** в компонент `Memo1` выводится таблица значений и строится график функции $y = \cos(x)$.

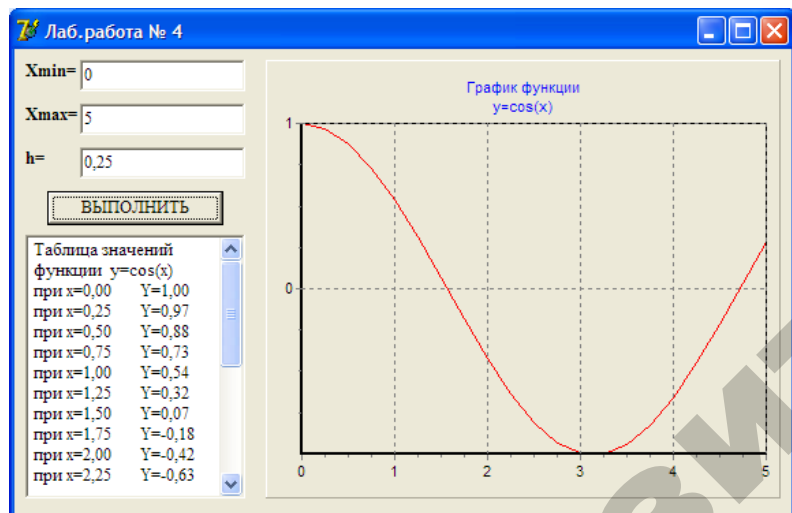


Рисунок 11. Результат выполнения задания

Вопросы для самоконтроля

1. Что такое итерационные алгоритмы?
2. Как записывается и как выполняется оператор `while`?
3. Как записывается и как выполняется оператор `repeat`?
4. В чем разница между операторами `while` и `repeat`?

Варианты индивидуальных заданий разного уровня:

- 1 уровень – выполните задание 1;
- 2 уровень – выполните задания 1, 2;
- 3 уровень – выполните задания 1, 2, 3.

Задание № 1

Разработать *Windows*-приложение для вычисления значения суммы S бесконечного ряда. Суммирование продолжать до тех пор, пока очередное слагаемое по абсолютной величине не станет меньше заданного числа ϵ . Значение ϵ принять равным ($\epsilon = 10^{-1}, 10^{-2}, \dots, 10^{-8}$).

Вывести значение суммы S и, для контроля правильности вычислений, точное значение суммы ряда. Проанализировать, как влияют на S различные значения ϵ , и сделать выводы.

Таблица 4.1 — Варианты индивидуальных заданий

Номер	S	Точное значение
1	$\sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$	e
2	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{2n-1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$	$\text{arctg } x$
3	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$	$\sin x$
4	$\sum_{n=0}^{\infty} (-1)^n \frac{x^n}{(2n)!} = 1 - \frac{x}{2!} + \frac{x^2}{4!} - \frac{x^3}{6!} + \dots$	$\cos \sqrt{x}$
5	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$	$\text{Ln } 2$
6	$\sum_{n=0}^{\infty} (-1)^n \frac{x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$	e^{-x}
7	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{\pi^{2n-1}}{(2n-1)!} = \frac{\pi}{1!} - \frac{\pi^3}{3!} + \frac{\pi^5}{5!} - \frac{\pi^7}{7!} + \dots$	$\sin \pi$

Продолжение таблицы 4.1

Номер	S	Точное значение
8	$\sum_{n=0}^{\infty} (-1)^n \frac{5^{2n} x^{2n}}{(2n)!} = 1 - \frac{25x^2}{2!} + \frac{625x^4}{4!} - \dots$	$\cos 5x$
9	$\sum_{n=0}^{\infty} (-1)^n \frac{1}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$	$\frac{\pi}{4}$
10	$\sum_{n=0}^{\infty} (-1)^n \frac{3^n x^n}{n!} = 1 - 3x + \frac{9x^2}{2!} - \dots$	e^{-3x}
11	$\sum_{n=0}^{\infty} \frac{(1/2)^n}{n!} = 1 + \frac{1}{2} + \frac{(1/2)^2}{2!} + \frac{(1/2)^3}{3!} + \dots$	\sqrt{e}
12	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{(\pi/10)^{2n-1}}{2n-1} = \frac{\pi}{10} - \frac{(\pi/10)^3}{3} + \frac{(\pi/10)^5}{5} - \dots$	$\operatorname{arctg} \frac{\pi}{10}$
13	$\sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \dots$	$\frac{1}{1-x}$
14	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n+2}}{2n-1} = x^4 - \frac{x^6}{3} + \frac{x^8}{5} - \dots$	$x^3 \operatorname{arctg} x$
15	$\sum_{n=0}^{\infty} (-1)^n \frac{1}{2^n n!} = 1 - \frac{1}{2} + \frac{1}{4 \cdot 2!} - \dots$	$\frac{1}{\sqrt{e}}$
16	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{4n-2}}{(2n-1)!} = \frac{x^2}{1!} - \frac{x^6}{3!} + \frac{x^{10}}{5!} - \dots$	$\sin x^2$
17	$\sum_{n=0}^{\infty} \frac{3^n x^n}{n!} = 1 + \frac{3x}{1!} + \frac{9x^2}{2!} + \dots$	e^{3x}
18	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{9^n}{n} = 9 - \frac{81}{1} + \frac{729}{2} - \dots$	$\ln 10$
19	$\sum_{i=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$	e^x

Окончание таблицы 4.1

Номер	S	Точное значение
20	$\sum_{n=1}^{\infty} \frac{1}{(2n-1)(2n+1)} = \frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \frac{1}{5 \cdot 7} + \dots$	$\frac{1}{2} - \frac{\pi}{8}$
21	$\sum (-1)^{n-1} \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$	$\ln(1+x)$
22	$\sum_{n=0}^{\infty} (-1)^{n+1} \frac{1}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$	$\operatorname{arctg} 1$
23	$\sum_{n=0}^{\infty} (-1)^n 2^n x^n = 1 - 2x + 4x^2 - \dots$	$\frac{1}{1+2x}$
24	$\sum_{n=0}^{\infty} (-1)^n \frac{x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$	$\frac{1}{e^x}$
25	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{(2n-1)!} = \frac{1}{1!} - \frac{1}{3!} + \frac{1}{5!} - \frac{1}{7!} + \dots$	$\sin 1$
26	$\sum_{n=0}^{\infty} (-1)^n x^n = 1 - x + x^2 - \dots$	$\frac{1}{1+x}$
27	$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{\pi^{2n-1}}{10^{2n-1} (2n-1)!} = \frac{\pi}{10} - \frac{\pi^3}{10^3 \cdot 3!} + \dots$	$\sin \frac{\pi}{10}$
28	$\sum_{n=0}^{\infty} (-1)^n \frac{3^n x^n}{n!} = 1 - 3x + \frac{9x^2}{2!} - \dots$	$\frac{1}{e^{3x}}$
29	$\sum_{n=0}^{\infty} (-1)^n \frac{1}{(2n)!} = 1 - \frac{1}{2!} + \frac{1}{4!} - \frac{1}{6!} + \dots$	$\cos 1$
30	$\sum_{n=0}^{\infty} (-1)^n \frac{x^{n+1}}{(2n)!} = x - \frac{x^2}{2!} + \frac{x^3}{6!} - \dots$	$x \cos \sqrt{x}$

Задание № 2

Протабулировать функцию f на заданном отрезке с шагом h и построить график этой функции. Исходные данные приведены в таблице 4.2. Границы отрезка и шаг изменения аргумента задать самостоятельно с учетом области определения функции.

Таблица 4.2 – Варианты индивидуальных заданий 2 уровня

Номер	Функция
1	$f(x) = 3 \sin \sqrt{x} - 3,8$
2	$f(x) = 0,25x^3 + x - 1,25$
3	$f(x) = \cos(x-1)e^{6x}$
4	$f(x) = \sqrt{0,4x^2} - \sin x$
5	$f(x) = \cos \frac{1}{x} - 2 \sin \frac{1}{x} + \frac{1}{x}$
6	$f(x) = 3x - 4 \ln x - 5$
7	$f(x) = \cos x - \sqrt{0,3x^3}$
8	$f(x) = 0,1x^2 - x \ln x$
9	$f(x) = \left(1 - \frac{x^2}{2}\right) \cos(x) - \frac{x}{2} \sin x$
10	$f(x) = x - \frac{1}{3 + \sin 3,6x}$
11	$f(x) = 5 \sin(x) \cos(3x+1)$
12	$f(x) = x^3 - 3x^2 + x + 1$
13	$f(x) = 3x - 14 + e^x$
14	$f(x) = \sqrt{1+x} - \operatorname{tg} x$
15	$f(x) = \cos x - e^{-\frac{x^2}{2}}$
16	$f(x) = 1 - x + \sin x - \ln(1+x)$

Окончание таблицы 4.2

Номер	Функция
17	$f(x) = 1,7x^2 + 6,9x - 4,6$
18	$f(x) = x - 2 + \sin 2x$
19	$f(x) = \frac{2 + \sin x}{1 + x^2}$
20	$f(x) = \sin(\ln x) - \cos(\ln x) + 2 \ln x$
21	$f(x) = 0,6 \cdot x^2 - 2,3x - 3$
22	$f(x) = \sqrt{1-x} - \cos \sqrt{1-x}$
23	$f(x) = 0,4 + \operatorname{arctg} x - x$
24	$f(x) = \left(\frac{x^2}{4} + \frac{x}{2} + 1\right) e^{\frac{x}{2}}$
25	$f(x) = x \operatorname{tg} x - \frac{1}{3}$
26	$f(x) = \ln x - x + 1,8$
27	$f(x) = 2(\cos^2 x - 1)$
28	$f(x) = 2x \sin x - \cos x$
29	$f(x) = x^2 - \ln(1+x) - 3$
30	$f(x) = \sin x^2 + \cos x^2 - 10x$

Задание № 3

Номер	Текст задания
1	Составить программу для перевода натурального числа A из десятичной системы счисления в двоичную систему счисления методом последовательного деления на основание системы счисления.
2	Дано натуральное число N . Вычислить произведение его цифр.

Номер	Текст задания
3	Составить алгоритм деления натурального числа n на натуральное число m методом последовательного вычитания с печатью частного и остатка.
4	Последовательность Фибоначчи образуется следующим образом: первый и второй члены последовательности равны 1, каждый следующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13, ...). Найти первое число в последовательности Фибоначчи, большее n . Значение n вводится с клавиатуры: $n > 1$.
5	Переменной t присвоить значение 1 или 0 в зависимости от того, является ли натуральное число k степенью 3.
6	В некотором году (назовем его первым) на участке в 100 гектар средняя урожайность ячменя составила 20 центнеров с гектара. После этого каждый год площадь участка увеличивалась на 5 %, а средняя урожайность на 2 %. Определить, в каком году урожайность превысит Q центнеров с гектара.
7	Подсчитать k – количество цифр в записи целого неотрицательного числа n .
8	Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, через сколько часов будет M клеток.
9	Дано натуральное число n . Изменить порядок следования цифр числа n на обратный.
10	Дана последовательность из вещественных чисел, оканчивающаяся числом K . Определить, образуют ли эти числа возрастающую последовательность.
11	Дано натуральное число N . Выбросить из записи числа цифры 0 и 3.
12	В этом году был открыт счет в банке на сумму S рублей. Процентная ставка по вкладу составляет k % годовых. Определить, в каком году размер вклада превысит R рублей?

Номер	Текст задания
13	Дано натуральное число N . Верно ли, что в записи числа присутствует цифра 5.
14	Дана последовательность ненулевых целых чисел, оканчивающаяся нулем. Определить, сколько раз в этой последовательности меняется знак. Например, в последовательности (10, -4, 12, 56, -4) знак меняется 3 раза.
15	Дана последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.
16	Предприниматель, начав дело, взял кредит размером k рублей под p процентов годовых и вложил его в свое дело. По прогнозам его дело должно давать прибыль r рублей в год. Сможет ли он накопить сумму, достаточную для погашения кредита, и если да, то через сколько лет?
17	Дана последовательность из n целых чисел. Определить, со скольких отрицательных чисел она начинается.
18	1 февраля был открыт счет в банке на сумму P рублей. Каждый месяц размер вклада увеличивается на 2 % от имеющейся суммы. Определить, через сколько месяцев размер вклада превысит T рублей.
19	Определить сумму первой и последней цифры натурального числа N .
20	Последовательность Фибоначчи образуется следующим образом: первый и второй члены последовательности равны 1, каждый следующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13, ...). Определить сумму всех членов этой последовательности, не превосходящих заданного числа N .
21	Дано натуральное число n . Выяснить, входит ли цифра 3 в запись числа n^2 .
22	Выяснить, является ли заданное число n членом арифметической прогрессии, первый член которой равен f , а знаменатель – z .

Номер	Текст задания
23	Составить программу для перевода натурального числа R из десятичной системы счисления в шестнадцатеричную систему счисления методом последовательного деления на основание системы счисления.
24	Дано натуральное число N . Вычислить сумму его цифр.
25	В течение месяца бригада за каждый день работы выпускает на r изделий больше предыдущего. В первый день было выпущено d изделий. Определить, через сколько дней бригада выполнит месячную норму a . Если в течение месяца (26 рабочих дней) норма не будет выполнена, то напечатать соответствующее сообщение.
26	Начав тренировки, лыжник в первый день пробежал 10 км. Каждый следующий день он увеличивал пробег на 13 % от пробега предыдущего дня. Определить, в какой день он пробежит больше L км.
27	В 2007 году средняя урожайность ячменя составила R центнеров с гектара. Каждые 2 года за счет применения передовых агротехнических приемов планируется увеличить урожайность на k % по отношению к предыдущему году. Определить, через сколько лет урожайность достигнет P центнеров с гектара ($P > R$).
28	Дана непустая последовательность целых чисел, оканчивающаяся отрицательным числом. Верно ли, что все элементы последовательности равны между собой?
29	Дано натуральное число d . Определить, какая цифра встречается в нем чаще – 0 или 9.
30	Дана последовательность положительных вещественных чисел, оканчивающаяся отрицательным числом. Определить, сколько чисел в последовательности.

ЛАБОРАТОРНАЯ РАБОТА №5

Решение прикладных задач, основанных на использовании алгоритмов с одномерными массивами

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании алгоритмов с одномерными массивами, в визуальной среде *Delphi*.

Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания приложения. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

Результаты измерений температуры воздуха в течение семи дней недели приведены в таблице.

День недели i	1	2	3	4	5	6	7
Температура t_i	11,2	10,5	12,8	13,1	12,3	11,4	10,6

Чему равна средняя температура воздуха t_{cp} ?

Значение средней температуры можно вычислить по следующей формуле:

$$t_{cp} = \frac{\sum_{i=1}^7 t_i}{7}. \quad (1)$$

Здесь i изменяется от 1 до 7, при этом происходит суммирование значений t_1, t_2, \dots, t_7 , представленных в таблице. При разработке алгоритмов таких задач, где данные представлены в табличном виде, удобно использовать массив.

Массив – это упорядоченная совокупность данных, состоящая из фиксированного количества элементов одинакового типа.

Положение элемента в массиве указывается с помощью индекса. Массивы указываются в разделе объявлений после слова **var**. На-

пример, для выделения памяти массиву из семи элементов действительного типа можно записать:

```
Var  
T : array [1..7] of extended;
```

Здесь:

1..7 – диапазон изменения индекса;

extended – тип элементов массива (действительные числа).

Объявленный массив *T* является *одномерным*, т.к. имеет только *один* индекс. Такие массивы используются при работе с *векторами*.

2. Примеры создания приложений

2.1. Вычисление суммы и произведения элементов массива

Пример 1. В заданном массиве *a*, состоящем из 10 целых значений, найти и вывести на экран сумму и произведение всех элементов массива.

Разработка алгоритма

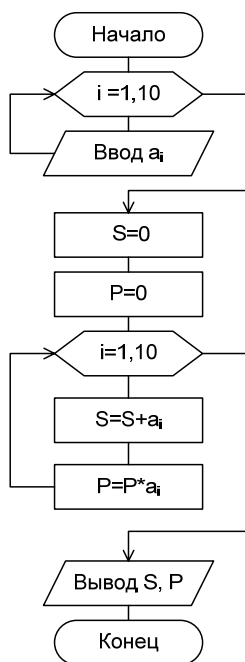


Рисунок 1. Блок-схема алгоритма решения задачи

На рисунке 1 представлена блок-схема алгоритма решения задачи.

Переменная *S* используется для получения суммы элементов массива *a*.

Переменная *P* – для вычисления произведения элементов массива *a*.

Методические указания по созданию приложения

Сохранение проекта


1. Для нового проекта в своей рабочей папке создайте новую папку **LAB5**.
2. Сохраните проект: **File**→ **Save Project As...**
В окне **Save Unit1 As** найдите папку **LAB51** и сохраните в ней модуль под именем **UnLab5.pas**. Затем в окне **Save Project1 As...** сохраните файл проекта в папке **LAB5** под именем **PrLab5.dpr**.

Изменение заголовка Формы

Измените заголовок Формы **Form1** на **Лаб.5-Пример 1**.

Размещение компонентов на Форме

Разместите компоненты на Форме в соответствии с рисунком 2.

Пиктограмма компонента **Memo**  находится на странице **Standard**. Добавим в компонент **Memo1** вертикальную линейку прокрутки. Для этого в окне Инспектора Объектов на странице **Properties** (Свойства) выберите в свойстве **ScrollBars** значение **ssVertical** – в компоненте **Memo1** появится вертикальная линейка прокрутки.

Компонент **Memo** будем использовать для ввода элементов массива. Каждая строка **Memo** нумеруется, начиная с нуля (в нулевой строке размещена надпись **Memo1**). Доступ к строке **Memo** осуществляется с помощью свойства **Lines**, в квадратных скобках указывают номер нужной строки, например, **Memo1.Lines[2]**.

При использовании компонента **Memo** для ввода массива значение каждого элемента массива следует вводить в отдельной строке и после ввода каждого элемента массива нажимать клавишу **<Enter>**.

При нажатии на кнопку **Вычислить** в **Memo1** будет выводиться сумма и произведение элементов массива.

Написание процедуры **Button1Click** обработки события нажатия кнопки **ВЫЧИСЛИТЬ**

Для создания процедуры-обработчика поместите курсор на изображение кнопки с надписью **ВЫЧИСЛИТЬ** и дважды щелкните по ней левой кнопкой мыши. Листинг процедуры-обработчика приведен ниже.

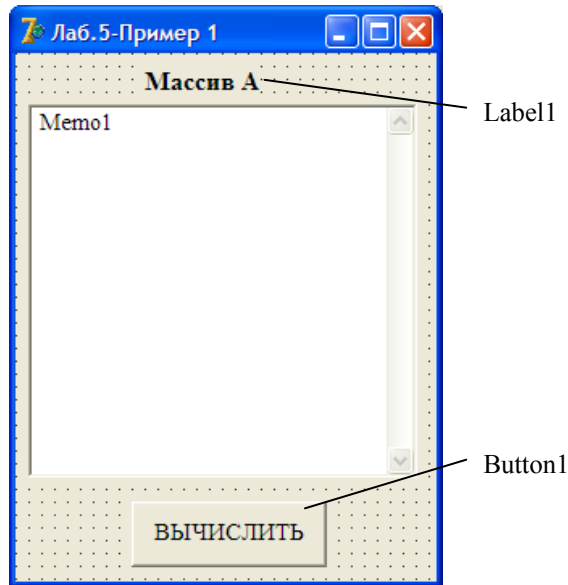


Рисунок 2. Форма с компонентами

```

procedure TForm1.Button1Click(Sender: TObject);
Var S,P,i:integer;
        a:array[1..10] of integer;
begin
    //Заполнение массива
    for i:=1 to 10 do
        a[i]:=StrToInt(Memo1.Lines[i]);
    //Инициализация переменных
    S:=0;
    P:=1;
    //Получение суммы и произведения элементов массива
    for i:=1 to 10 do
        begin
            S:=S+a[i];
            P:=P*a[i];
        end;
    //Вывод результатов
    Memo1.Lines.Add('S='+IntToStr(S));
    Memo1.Lines.Add('P='+IntToStr(P));
end;

```

Работа с приложением

Перед запуском приложения сохраните проект: пиктограмма (главного меню) или **File → Save All (Файл → Сохранить Все)**.

Для запуска созданного приложения нажмите пиктограмму или клавишу **F9**. При этом происходит компоновка проекта и создание выполняемого файла **PrLab5.exe**. На экране появляется панель интерфейса приложения, показанная на рисунке 3.

Сценарий работы приложения

1. В строке, следующей за словом Memo1, набрать значение 1-го элемента массива.
2. Нажать клавишу Enter (Ввод);
3. Повторить пп 1), 2) для всех элементов массива.
4. Для получения результата щелкнуть на кнопке **ВЫЧИСЛИТЬ**.

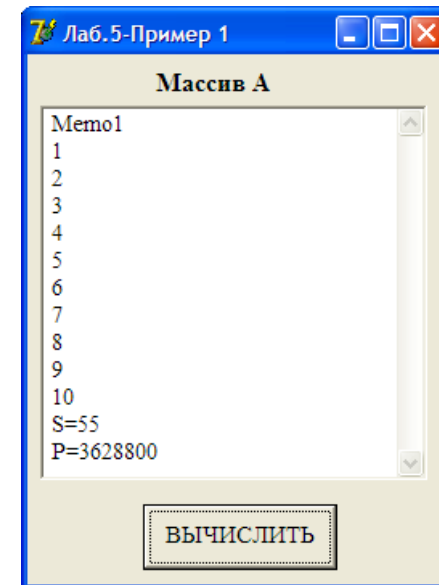


Рисунок 3. Панель интерфейса приложения

2.2. Формирование массива констант

Пример 2. В массиве t содержатся результаты измерений температуры воздуха, которые проводились ежедневно в течение недели. Вычислить значение средней температуры t_{cp} по формуле (1). В интерфейсе приложения предусмотреть окно для вывода элементов массива.

Разработка алгоритма

Блок-схема алгоритма решения задачи может иметь вид, показанный на рисунке 4.

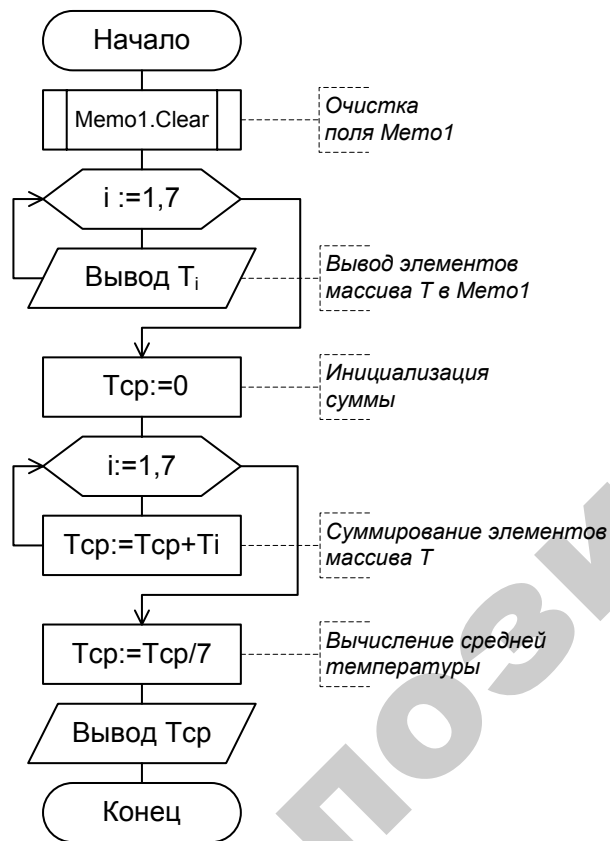


Рисунок 4. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Создание интерфейса

Массив значений температуры будем вводить не с клавиатуры, а зададим непосредственно в программе. Тогда один из возможных вариантов панели интерфейса создаваемого приложения может выглядеть следующим образом (рисунок 5).

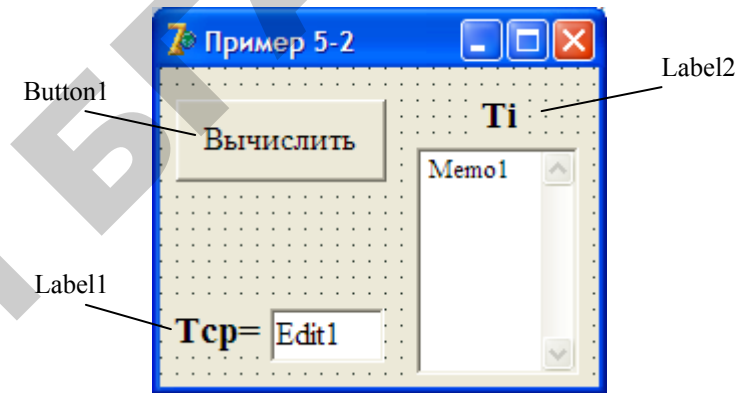


Рисунок 5. Форма с компонентами

Здесь для вывода значений элементов массива T используется компонент **Memo** — многострочный редактор. Пиктограмма компонента **Memo** находится на странице **Standard**. Для удобства использования компонента **Memo** его снабжают линейками прокрутки. Чтобы обеспечить просмотр всех элементов массива T добавим в компонент **Memo1** вертикальную линейку прокрутки. Для этого щелчком левой кнопки мыши на компоненте **Memo1** сделайте его активным, перейдите в окно Инспектора Объектов и на странице **Properties** (Свойства) выберите в свойстве **ScrollBars** значение **ssVertical** — в компоненте **Memo1** появится вертикальная линейка прокрутки.

Методические указания по созданию приложения

Сохранение проекта

Назовите проект **PrLab52**, а модуль проекта – **UnLab52** и сохраните их в папке **Lab52**.

Создание процедуры Button1Click обработки события нажатия кнопки **Вычислить**

Для облегчения тестирования программы часто численные значения элементов массивов задают в разделе **Const**, например, для нашей задачи можно записать:

Const

```
T:array[1..7] of extended=(11.2,10.5,12.8,13.1,12.3,11.4,10.6);
```

Для создания процедуры-обработчика поместите курсор на изображение кнопки **Вычислить** и дважды щелкните левой кнопкой мыши. Полученную заготовку процедуры дополните необходимыми операторами так, как показано ниже.

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

Const

```
T: array[1..7] of extended=(11.2,10.5,12.8,13.1,12.3,11.4,10.6);
```

Var


```
  i :integer;  
  Tcp:extended;
```

begin

```
  Mem1.Clear;           // очистка Memo1  
  for i:=1 to 7 do      // вывод массива T в Memo1  
    Mem1.Lines.Add(FloatToStrF(T[i], ffFixed, 6, 1));  
  Tcp:= 0;             // инициализация суммы  
  for i:=1 to 7 do  
    Tcp:=Tcp+T[i];     // суммирование элементов массива T  
  Tcp:=Tcp/7;         // вычисление среднего значения Tcp  
  Edit1.Text:=FloatToStrF(Tcp, ffFixed, 8, 1);
```

end;

Работа с приложением

Для запуска созданного приложения нажмите пиктограмму  главного меню или клавишу **F9**. Выполните действия, предусмотренные сценарием работы приложения. На экране появится панель интерфейса приложения, показанная на рисунке 6.

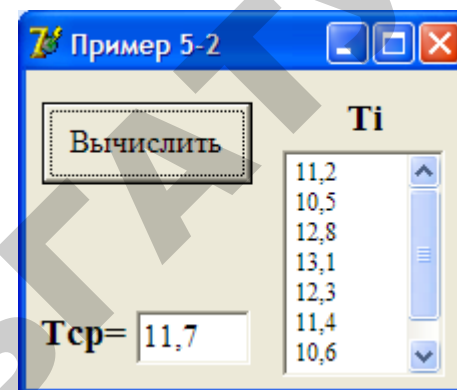


Рисунок 6. Панель интерфейса приложения

2.3. Изменение порядка следования элементов в массиве

Пример 3. Разработать *Windows*-приложение, которое вводит с клавиатуры одномерный массив из 10-ти элементов целого типа, а затем изменяет порядок следования элементов в этом массиве на противоположный. Вспомогательные массивы не использовать.

Методические указания по созданию приложения

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке 7. Здесь для ввода значений элементов массива **T** используется компонент **Memo1**, а для вывода – компонент **Memo2**.

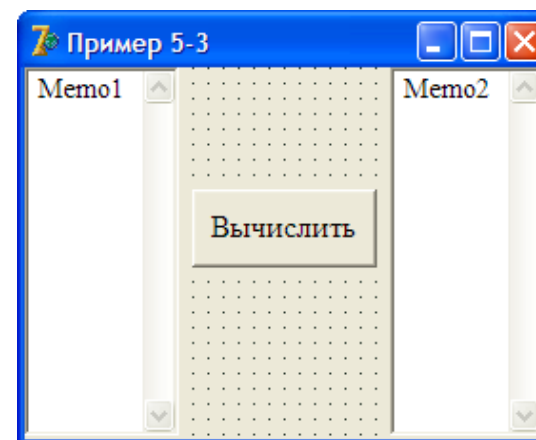


Рисунок 7. Форма с компонентами

Блок-схема алгоритма процедуры обработки события нажатия кнопки **Вычислить** может иметь вид, показанный на рисунке 8.

Назовите проект **PrLab53**, а модуль проекта – **UnLab53** и сохраните их в папке **Lab53**.

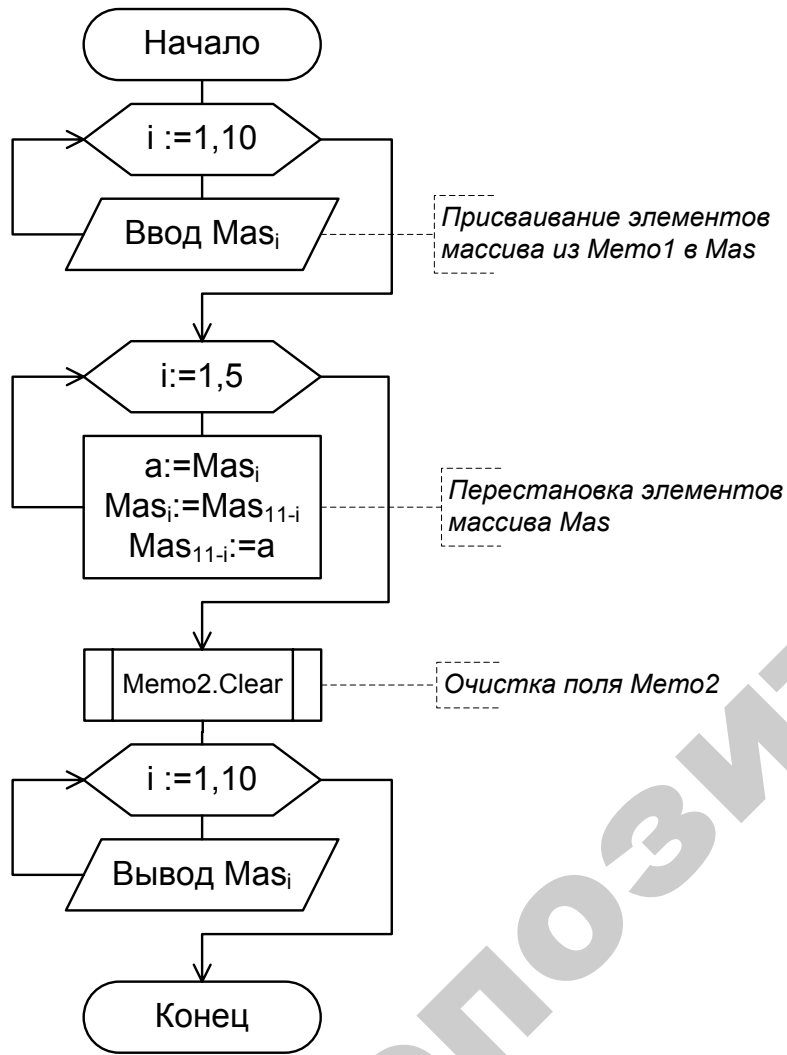


Рисунок 8. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Листинг процедуры **Button1Click** обработки события нажатия кнопки **Вычислить** представлен ниже.

```

procedure TForm1.Button1Click(Sender: TObject);
var
    a,i :integer;
    Mas :array[1..10] of integer;
begin
    for i:=1 to 10 do           {присваивание элементов массива}
        Mas[i]:=StrToInt(Memo1.Lines[i]);    {из поля Мемо1 в Mas}
        for i:=1 to 5 do      // цикл повторяется 5 раз
            begin
                a:=Mas[i];           { для перестановки элементов }
                Mas[i] :=Mas[11-i];  { массива Mas используется }
                Mas[11-i]:=a;        { дополнительная переменная a}
            end;
        Memo2.Clear;           // очистка Мемо2
        for i:=1 to 10 do
            Memo2.Lines.Add(IntToStr(Mas[i]));
        end;

```

Результаты выполнения программы представлены на рисунке 9.

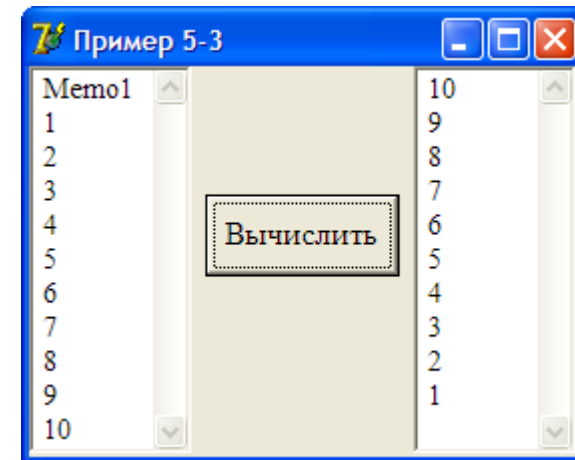


Рисунок 9. Панель интерфейса приложения

2.4. Программное формирование массива

В некоторых задачах элементы массива не вводятся с клавиатуры, а формируются в программе. Рассмотрим такие примеры.

Пример 4. Разработать *Windows*-приложение, в котором формируется массив $a = \{a_1, a_2, \dots, a_{21}\}$ из членов последовательности $\cos x, \cos(x+h), \dots, \cos(x+20h)$. Сформированный массив вывести в панель интерфейса. Найти сумму тех элементов массива, которые по абсолютной величине больше, чем 0,5.

Методические указания по созданию приложения

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке 10. Здесь компонент **Мемо** используется для визуального контроля элементов сформированного массива.

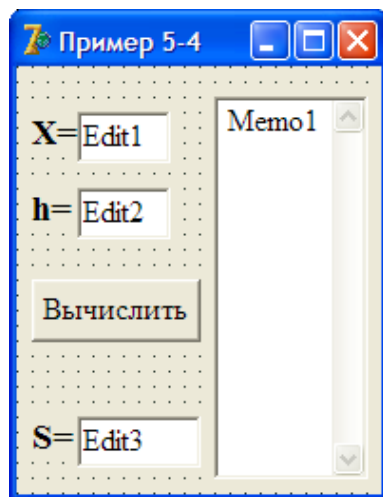


Рисунок 10. Форма с компонентами

Сценарий работы приложения:

- 1) занести в компонент Edit1 численное значение x ;
- 2) занести в компонент Edit2 численное значение h ;
- 3) щелчок на кнопке **Вычислить**.

Блок-схема алгоритма процедуры обработки события нажатия кнопки **Вычислить** может иметь вид, показанный на рисунке 11.

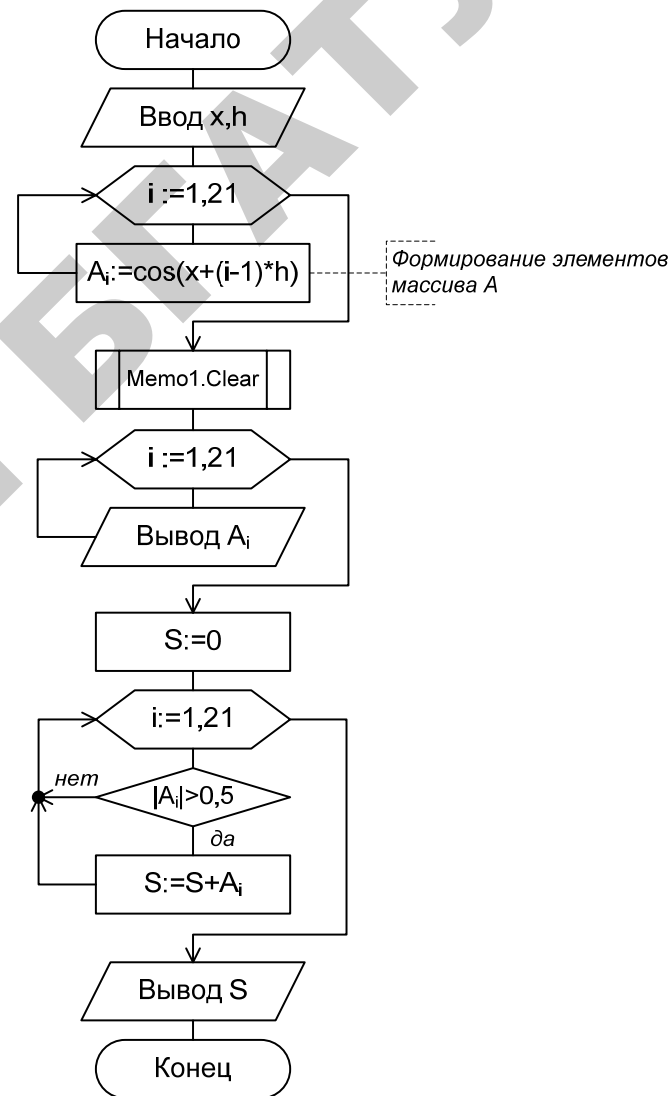


Рисунок 11. Блок-схема процедуры обработки события нажатия кнопки Вычислить

Листинг процедуры-обработки события нажатия кнопки **Вычислить** представлен ниже.

```

procedure TForm1.Button1Click(Sender: TObject);
var
  i      :integer;
  x,h,s:extended;
  A      :array[1..21] of extended;
begin
  x:=StrToFloat(Edit1.Text);
  h:=StrToFloat(Edit2.Text);
  for i:=1 to 21 do
    A[i]:=cos(x+(i-1)*h);
  Mem1.Clear;
  for i:=1 to 21 do
    Mem1.Lines.Add(FloatToStrF(A[i], ffFixed, 8, 3));
  s:=0;
  for i:=1 to 21 do
    if abs(A[i])>0.5 then
      s:=s+A[i];
  Edit3.Text:=FloatToStrF(s, ffFixed, 8, 3);
end;

```

Результат работы программы представлен на рисунке 12.

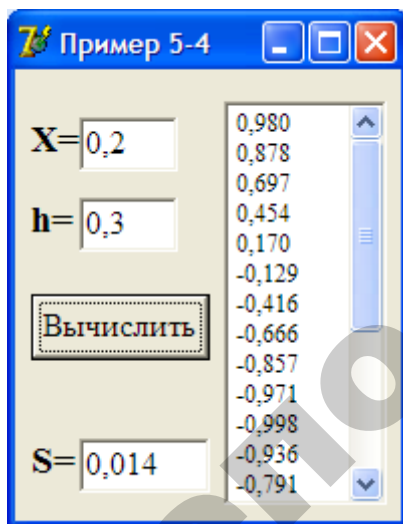


Рисунок 12. Панель интерфейса приложения

2.5. Задание значений элементов массива случайным образом

Пример 5. Разработать *Windows*-приложение, в котором формируется одномерный массив из 15-ти различных положительных элементов целого типа в диапазоне от 0 до 100. В сформированном массиве осуществляется поиск и вывод минимального и максимального значений.

Методические указания по созданию приложения

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке 13.

При нажатии на кнопку **Вычислить** в Memo1 будет выведен массив, сформированный случайным образом, а в компоненты Edit1 и Edit2 – значения минимального и максимального значений массива соответственно.

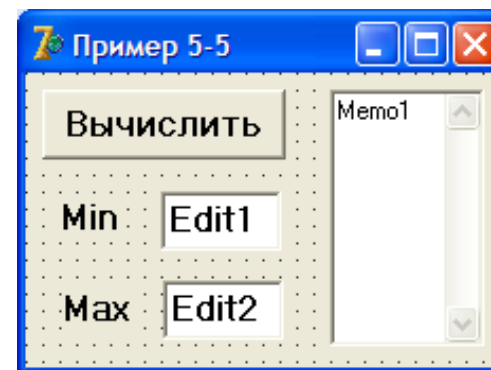


Рисунок 13. Форма с компонентами

Для формирования массива из различных элементов можно использовать датчик случайных чисел, к которому легко обратиться при написании операторов процедуры-обработчика. Блок-схема алгоритма процедуры обработки события нажатия кнопки **Вычислить** может иметь вид, показанный на рисунке 14.

Назовите проект **PrLab55**, а модуль проекта – **UnLab55** и сохраните их в папке **Lab55**.

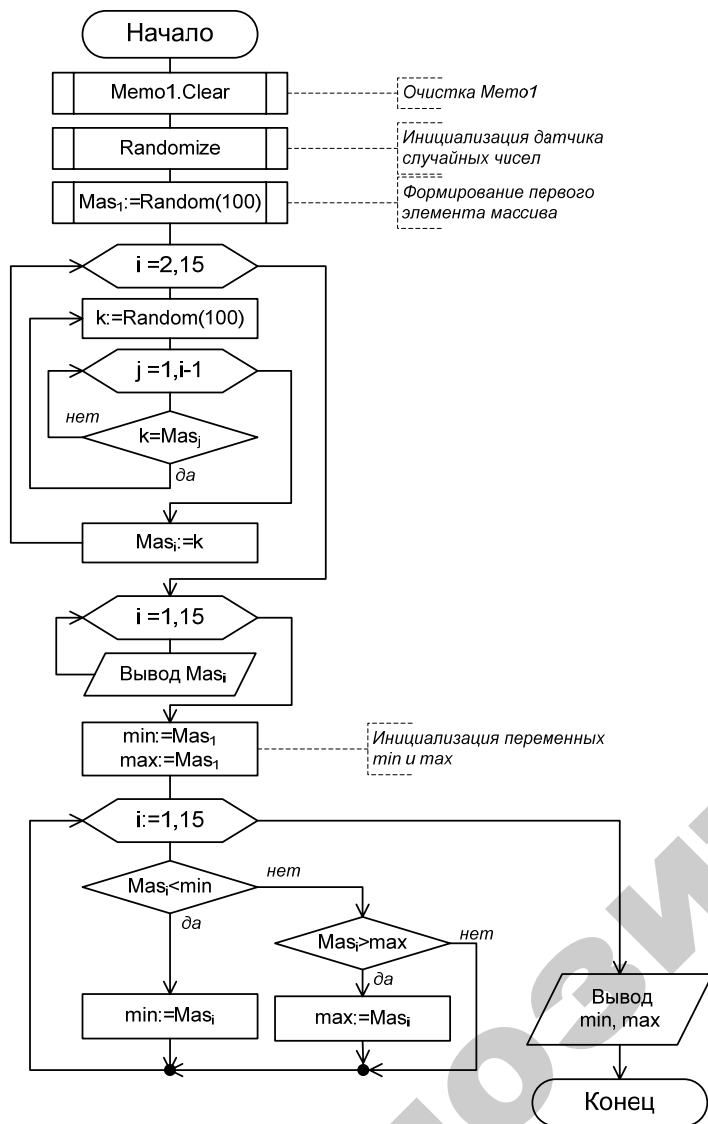


Рисунок 14. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Листинг процедуры обработки события нажатия кнопки **Вычислить** представлен ниже. Каждое нажатие кнопки **Вычислить**

приводит к формированию нового массива случайных чисел и выводу значений min и max.

```

procedure TForm1.Button1Click(Sender: TObject);
Label 1;
var
    i,j,k,min,max: integer;
    Mas: array[1..15] of integer;
begin
    Memo1.Clear; // очистка Мемо1
    Randomize; // инициализация датчика случайных чисел
    Mas[1]:=Random(100); // в 1-й элемент заносится случайное число
    for i:=2 to 15 do { формирование }
    begin { массива }
    1: k:=Random(100); { случайных }
    for j:=1 to i do { чисел от 0 до 100 }
    if k=Mas[j] then { если такое число есть в массиве, тогда }
    goto 1; { перейти к генерации другого случайного числа }
    Mas[i]:= k; //занести случайное число в массив
    end;
    for i:=1 to 15 do
    Memo1.Lines.Add(IntToStr(Mas[i]));
    Min:=Mas[1];
    Max:=Mas[1];
    for i:=1 to 15 do
    if Mas[i]<min then
    min:=Mas[i]
    else
    if Mas[i]>max then
    max:=Mas[i];
    Edit1.Text:=IntToStr(min);
    Edit2.Text:=IntToStr(max);
end;
  
```

Вопросы для самоконтроля

1. Как в программе объявляются одномерные массивы?
2. Каким образом можно присвоить значения элементам массива?
3. Что такое индекс? Каким требованиям он должен удовлетворять?
4. Как осуществляется доступ к элементам массива?

Варианты индивидуальных заданий разного уровня:

1 уровень – выполните задание 1;

2 уровень – выполните задания 1, 2;

3 уровень – выполните задания 1, 2, 3.

Вариант 1

1. Для массива $a_1, a_2, a_3, \dots, a_{10}$ получить сумму положительных элементов и количество отрицательных.

2. Даны координаты 7 точек $(x_1, y_1), (x_2, y_2), \dots, (x_7, y_7)$. Вывести на экран координаты точки, расположенной ближе всех к началу координат. Расстояние точки с координатами (x, y) до начала координат вычисляется по формуле $L = \sqrt{x^2 + y^2}$.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне от 0 до 90. Поменять местами максимальный и минимальный элементы массива. Вывести на экран исходный и полученный массивы.

Вариант 2

1. Для массива $x_1, x_2, x_3, \dots, x_{10}$ получить среднее арифметическое отрицательных элементов.

2. Дан целочисленный массив $b_1, b_2, b_3, \dots, b_{10}$. Найти наибольший элемент массива и определить, является ли он четным числом. Вывести на экран соответствующее сообщение.

3. Переменная x изменяется в интервале от $x_{\text{нач}}$ до $x_{\text{кон}}$ с шагом dx . Сформировать массив Y , содержащий 30 вещественных чисел, по формуле:

$$y_i = \begin{cases} \cos x, & x \leq 0; \\ \ln x, & x > 0. \end{cases}$$

Упорядочить массив Y в порядке возрастания элементов.

Вариант 3

1. В массиве $c_1, c_2, c_3, \dots, c_9$ найти количество элементов, больших среднего арифметического элементов заданного массива. Вывести на экран результаты вычислений.

2. В массиве хранится информация о росте 15 человек. Определить рост самого высокого человека.

3. Дан массив целых чисел из 30 элементов, заполненный случайным образом числами в диапазоне от -10 до 10. Записать элементы массива в обратном порядке. Полученный массив вывести на экран.

Вариант 4

1. Заданы координаты 8 точек $(x_1, y_1), (x_2, y_2), \dots, (x_8, y_8)$. Определить количество точек, лежащих на оси абсцисс. Если таких точек нет, вывести на экран соответствующее сообщение.

2. Задан массив a_1, a_2, \dots, a_{11} . Вывести на экран номер наименьшего элемента массива.

3. Задан массив Y из n целых чисел. Сформировать массив Z таким образом, чтобы в начале шли отрицательные элементы массива Y , затем положительные, и, наконец, нулевые.

Вариант 5

1. Задан массив $a_1, a_2, a_3, \dots, a_{12}$. Все отрицательные элементы массива увеличить на 5, а все неотрицательные заменить на 1. Вывести на экран полученный массив.

2. В массиве t_1, t_2, \dots, t_{12} найти минимальный и максимальный элементы и получить их среднее арифметическое. Вывести на экран результаты вычислений.

3. Из элементов массива D сформировать массив A той же размерности по правилу: если номер четный, то значение элемента находится по формуле $A_i = D_i^2$, а если нечетный, то по формуле $A_i = D_i / i$. Определить произведение элементов массива A , расположенных между минимальным и максимальным элементами массива.

Вариант 6

1. Имеется 10 шаров диаметрами d_1, d_2, \dots, d_{10} . Определить, сколько из них пройдет сквозь круглое отверстие площадью S .

2. В массиве хранится информация о результатах 15 спортсменов, участвовавших в лыжной гонке. Определить результат и номер спортсмена-победителя гонки.

3. Дан массив целых чисел из 20 элементов. Удалить из массива элемент с номером m .

Вариант 7

1. Для массива $a_1, a_2, a_3, \dots, a_{15}$ получить сумму и количество элементов, кратных 3.

2. Пусть даны вещественные числа d_1, d_2, \dots, d_{12} . Умножить все элементы массива d на его наименьший элемент. Вывести на экран полученный массив.

3. Переменная x изменяется в интервале от $x_{\text{нач}}$ до $x_{\text{кон}}$ с шагом h . Сформировать массив R по формуле:

$$R_i = \begin{cases} x \cos x, & x < 0; \\ 1 + x^2, & 0 \leq x \leq 2,5; \\ x - 4, & x > 2,5. \end{cases}$$

Расположить элементы массива R в обратном порядке. Полученный массив вывести на экран.

Вариант 8

1. Дан массив y_1, y_2, \dots, y_{11} . Найти произведение ненулевых элементов массива и количество элементов, равных нулю.

2. Дан массив вещественных чисел x_1, x_2, \dots, x_{10} . Найти отношение максимального элемента массива к минимальному элементу.

3. Массив D заполнен случайным образом в диапазоне $[-20, 50]$. Подсчитать количество положительных элементов массива, кратных заданному числу K .

Вариант 9

1. Получить произведение тех элементов массива c_1, c_2, \dots, c_{10} , которые превышают заданное число d . Число d вводится с клавиатуры.

2. В массиве хранится информация о годе рождения каждого из 15 человек. Определить порядковый номер самого старшего по возрасту человека.

3. Дан массив X , состоящий из n элементов. Напишите программу для подсчета числа элементов массива, равных максимальному элементу. Вывести на экран элементы массива X и результаты вычислений.

Вариант 10

1. Заданы координаты 10 точек $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Вывести на печать: количество точек, которые попадают в круг радиуса R с центром в начале координат, и номера точек, не попавших в круг.

2. В массиве $a_1, a_2, a_3, \dots, a_{10}$ найти минимальный элемент и его порядковый номер (в массиве нет совпадающих элементов) и поменять его местами с первым элементом массива. Вывести на экран полученный массив.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне $[0, 30]$. Заменить элементы массива с нечетными номерами на квадраты их номеров. Т.е. вместо 1-го элемента записать 1, вместо 3-го элемента – 9 и т.д.

Вариант 11

1. Даны действительные числа a_1, a_2, \dots, a_{10} . Определить каких элементов в массиве больше отрицательных или положительных.

2. В массиве хранится информация о результатах 22 спортсменов, участвовавших в соревнованиях по бегу на 100 м. Определить результаты спортсменов, занявших первое и последнее места.

3. Из элементов массива D сформировать массив B той же размерности по правилу: если номер четный, то значение элемента находится по формуле $B_i = i \cdot D_i^2$, а если нечетный, то по формуле $B_i = D_i / (i + 1)$. Подсчитать число чередований знака соседних элементов массива B с «+» на «-» и наоборот.

Вариант 12

1. Даны координаты 10 точек $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Определить количество точек, лежащих в 4-ой четверти, если таких точек нет – выдать соответствующее сообщение.

2. Имеются сведения о выработке электроэнергии одной из турбин электростанции по месяцам (с января по декабрь) R_1, R_2, \dots, R_{12} . Определить среднемесячную выработку электроэнергии турбиной и номер месяца, в который выработали наибольшее количество электроэнергии.

3. Дан массив целых чисел из m элементов, заполненный случайным образом в диапазоне $[-100, 50]$. Изменить знак у макси-

мального по модулю элемента массива. Вывести на экран исходный и полученный массивы.

Вариант 13

1. Массив G содержит информацию о возрасте каждого из 20 студентов одной группы. Определить средний возраст студентов группы и вывести номера студентов младше 18 лет.

2. Даны координаты 10 точек $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Вывести на печать координаты точки, расположенной дальше всех от начала координат. Расстояние точки с координатами (x, y) до начала координат вычисляется по формуле $L = \sqrt{x^2 + y^2}$.

3. Массив R сформирован случайным образом и содержит целые числа в диапазоне от 0 до 100. Удалить из массива все элементы кратные 5. Вывести на экран полученный массив.

Вариант 14

1. В массиве $a_1, a_2, a_3, \dots, a_{10}$ вместо отрицательных элементов записать их модули, вместо неотрицательных элементов вычислить их квадратный корень. Вывести на экран полученный массив.

2. В массиве r_1, r_2, \dots, r_8 найти среднее арифметическое элементов массива без наибольшего элемента.

3. В области 20 районов. Площади, засеянные пшеницей (в гектарах), и урожай, собранный в каждом районе (в центнерах), хранятся в двух массивах. Определить среднюю урожайность по каждому району и по области в целом.

Вариант 15

1. Среди точек с координатами $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$ найти количество точек, лежащих в I-ой четверти. Если таких точек нет, вывести на экран соответствующее сообщение.

2. Для массива $a_1, a_2, a_3, \dots, a_{15}$ найти разность между средним арифметическим элементов массива и наименьшим из них.

3. Из элементов массива P сформировать массив M той же размерности по правилу: первый и последний элементы равны нулю, а все остальные находятся по формуле $M_i = -P_i \cdot (i + 1)$. Вывести на экран исходный и полученный массивы.

Вариант 16

1. Дан массив d_1, d_2, \dots, d_{12} , состоящий из целых чисел. Найти сумму и количество элементов кратных 10 и вывести на экран элементы, не кратные 10. Вывести на экран результаты вычислений.

2. В вещественном массиве $b_1, b_2, b_3, \dots, b_{10}$ поменять местами минимальный элемент и первый.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне от 1 до 100. Найти количество простых чисел в этом массиве. *Натуральное число N называется простым, если оно делится нацело без остатка только на 1 и на N .*

Вариант 17

1. В массиве $a_1, a_2, a_3, \dots, a_{10}$ заменить на 5 все элементы, равные и большие среднего арифметического, и на 0 – меньшие среднего арифметического всех элементов массива. Вывести на экран полученный массив.

2. Заданы координаты 8 точек $(x_1, y_1), (x_2, y_2), \dots, (x_8, y_8)$. Вывести на экран порядковый номер точки, которая расположена ближе всех от начала координат. Расстояние точки с координатами (x, y) до начала координат вычисляется по формуле $L = \sqrt{x^2 + y^2}$.

3. Массив D заполнен случайным образом числами в диапазоне от -100 до 100. Заменить элементы массива между минимальным и максимальным на те же элементы в обратном порядке.

Вариант 18

1. Заданы координаты 8 точек $(x_1, y_1), (x_2, y_2), \dots, (x_8, y_8)$. Определить количество точек, лежащих на оси ординат. Если таких точек нет, вывести на экран соответствующее сообщение.

2. В массиве $a_1, a_2, a_3, \dots, a_{10}$ найти максимальный элемент и его порядковый номер и поменять его местами с последним элементом массива. Преобразованный массив вывести на печать.

3. Из элементов массива D сформировать массив A той же размерности по правилу: элементы с 3-го по 12-й находятся по формуле $A_i = -D_i^2$, остальные по формуле $A_i = D_i - 1$. Найти произведение четных элементов, значения которых по модулю меньше 5.

Вариант 19

1. Задан массив X , состоящий из 12 целых элементов. Определить количество элементов, меньших среднего арифметического элементов заданного массива.

2. Задан массив C_1, C_2, \dots, C_{12} . Отрицательные элементы массива C записать в массив D . Найти в массиве D максимальный элемент и его порядковый номер. Вывести на экран исходный и полученный массивы.

3. Дан массив целых чисел из 30 элементов, заполненный случайным образом числами в диапазоне $[-50, 150]$. Найти в массиве среднее арифметическое тех элементов массива, которые расположены между минимальным и максимальным элементами массива.

Вариант 20

1. Дан массив x_1, x_2, \dots, x_{15} . Определить сумма каких элементов больше – четных или нечетных. Вывести на экран соответствующее сообщение.

2. В массиве D хранится информация о годе рождения каждого из 20 человек. Определить возраст самого молодого по возрасту человека.

3. Переменная x изменяется в интервале $[-4; 4]$ с шагом $dx = 0,2$. Сформировать массив y по формуле:

$$y_i = \begin{cases} 1 - x^2, & x \leq 0; \\ 1 + \sqrt{\cos x}, & x > 0. \end{cases}$$

Определить, есть ли пара соседних элементов с суммой, равной заданному числу Z . Число Z задается с клавиатуры.

Вариант 21

1. Имеется список 15 членов бригады с указанием их возраста. Определить средний возраст членов бригады и количество работников, чей возраст старше 55 лет.

2. Наибольший элемент массива t_1, t_2, \dots, t_{12} уменьшить в 2 раза. Вывести на экран полученный массив.

3. Массив P сформирован случайным образом и содержит 30 чисел в диапазоне от -120 до 100. Вывести на экран положительные элементы массива, делящиеся на заданное число k с остатком 2. Если таких элементов нет, выдать соответствующее сообщение.

Вариант 22

1. Задан массив x_1, x_2, \dots, x_{15} . Определить произведение элементов массива, принадлежащих отрезку $[-5, 5]$. Элементы массива, не принадлежащие отрезку $[-5, 5]$ вывести на экран.

2. Задан массив a_1, a_2, \dots, a_{10} . Определить, является ли наименьший элемент массива положительным или отрицательным числом. Вывести на экран соответствующее сообщение.

3. Из элементов массива C сформировать массив A той же размерности по правилу: если номер i элемента четный, то $A_i = C_i^2$, если нечетный, то $A_i = 2C_i$. Упорядочить массив y в порядке убывания элементов.

Вариант 23

1. Даны координаты 10 точек: $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Определить, сколько точек попадает в кольцо с внутренним радиусом r_1 и внешним r_2 , если центр кольца находится в начале координат.

2. В массиве d_1, d_2, \dots, d_7 умножить минимальный элемент на 100 (в массиве нет совпадающих элементов). Вывести на экран полученный массив.

3. Дан массив целых чисел из n элементов, заполненный случайным образом в диапазоне $[-200, 200]$. Каждый отрицательный элемент заменить суммой тех двух элементов, которые стоят справа и слева. Вывести на экран исходный и полученный массивы.

Вариант 24

1. В массиве A хранятся оценки 20 студентов по информатике. Определить количество неуспевающих студентов и средний балл успевающих.

2. Дан массив вещественных чисел g_1, g_2, \dots, g_{11} . Вывести на экран порядковые номера максимального и минимального элементов массива.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне $[0, 60]$. Определите количество элементов массива, являющихся квадратами четных чисел. Число n задается с клавиатуры.

Вариант 25

1. Заданы координаты 12 точек $(x_1, y_1), (x_2, y_2), \dots, (x_{12}, y_{12})$. Найти количество и вывести номера точек, лежащих в I или IV четвертях.

2. Дан массив вещественных чисел c_1, c_2, \dots, c_{11} . заменить минимальный элемент массива последним элементом. Преобразованный массив вывести на печать.

3. Из элементов массива P сформировать массив M той же размерности по правилу: если номер четный, то $M_i = i \cdot P_i$, если нечетный, то $M_i = -P_i$. Вывести номера элементов массива, делящихся на 5 с остатком 2. Если элементов, удовлетворяющих условию нет, вывести на экран соответствующее сообщение.

Вариант 26

1. Рост 15 учеников класса представлен в виде массива. Рост мальчиков условно задан отрицательными числами, рост девочек – положительными. Определить средний рост мальчиков и средний рост девочек.

2. В массиве X , содержащем 12 вещественных чисел, заменить максимальный элемент массива числом 1000. Вывести на экран полученный массив.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне от 1 до 200. Элементы массива, расположенные между минимальным элементом и максимальным, заменить нулями.

Вариант 27

1. Для массива $a_1, a_2, a_3, \dots, a_{15}$ получить среднее арифметическое четных элементов массива. Если четных элементов в массиве нет, вывести на экран соответствующее сообщение.

2. В массиве d_1, d_2, \dots, d_{11} заменить нулями элементы массива между минимальным и максимальным, кроме их самих.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне от -40 до 50. Удалить из него все элементы, в записи которых есть цифра 5.

Вариант 28

1. В массивах X и Y хранится информация о количестве студентов в 10 группах I и II курсов соответственно. Определить, на каком курсе студентов больше. Вывести на экран соответствующее сообщение.

2. Задан массив a_1, a_2, \dots, a_9 . Заменить последний элемент массива на минимальный.

3. Задан массив X , содержащий 12 вещественных чисел. Переписать элементы массива X в массив Y , сдвинув элементы массива X вправо на три позиции. При этом три элемента с конца массива X перемещаются в начало: $(y_1, y_2, \dots, y_{12}) = (x_{10}, x_{11}, x_{12}, x_1, \dots, x_9)$.

Вариант 29

1. Дан массив a_1, a_2, \dots, a_{12} . Все элементы массива, принадлежащие отрезку $[-2, 5]$ заменить нулями. Вывести на экран полученный массив.

2. Для массива $a_1, a_2, a_3, \dots, a_{15}$ найти разность между средним арифметическим элементов массива и наибольшим из них.

3. Дан массив целых чисел из n элементов, заполненный случайным образом числами в диапазоне от -200 до 200. Определить количество различных элементов массива (повторяющиеся элементы считать один раз).

Вариант 30

1. Имеется 7 точек с координатами $(x_1, y_1), (x_2, y_2), \dots, (x_7, y_7)$. Определить количество точек, лежащих во 2-ой четверти.

2. В массиве $a_1, a_2, a_3, \dots, a_{10}$ вывести на экран номера максимального и минимального элементов массива.

3. Переменная x изменяется в интервале $[-2; 1]$ в шаге $dx = 0,1$. Сформировать массив y по формуле:

$$y[i] = \begin{cases} e^{x+1}, & x < -1; \\ -1, & x = -1; \\ \cos 2x, & x > -1. \end{cases}$$

Удалить из массива y все числа, превышающие среднее арифметическое элементов массива.

ЛАБОРАТОРНАЯ РАБОТА №6

Решение прикладных задач, основанных на использовании алгоритмов с двумерными массивами

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании алгоритмов с двумерными массивами, в визуальной среде *Delphi*.

Порядок выполнения лабораторной работы

1. Изучить раздел «Краткие теоретические сведения».
2. Выполнить пример создания приложения. Результаты показать преподавателю.
3. Выполнить индивидуальное задание.
4. Оформить отчет по лабораторной работе.

1. Краткие теоретические сведения

Если $m \times n$ выражений расставлены в прямоугольной таблице из m строк и n столбцов:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

то говорят о **матрице** размера $m \times n$, или сокращенно об $m \times n$ матрице; выражения a_{ik} называются **элементами** матрицы. Положение элемента в таблице характеризуется двойным индексом: первый индекс означает номер строки, второй – номер столбца, на пересечении которых стоит элемент (нумерация строк производится сверху вниз, а столбцов – слева направо).

Матрица размера $n \times n$ называется **квадратной матрицей порядка n** . Элементы a_{ii} , т.е. элементы, стоящие в таблице на диагонали квадрата, проходящей из левого верхнего угла в правый нижний (на **главной диагонали** матрицы) называются **главными диагональными элементами** или просто **диагональными элементами**. Элементы $a_{i,n+1-i}$ ($i = 1, \dots, n$), т.е. элементы, стоящие на диагонали, которая проходит от правого верхнего угла в левый нижний (**по-**

бочная диагональ матрицы), называются **побочными диагональными элементами**.

Сумма главных диагональных элементов называется **следом** (Spur, Trace) матрицы **Sp A** или **Tr A**.

В языке ОР для представления матриц используются **двумерные массивы**, у которых **два** индекса: первый – для указания номера строки, а второй – для номера столбца. Например, для работы с матрицей, содержащей 2 строки и 3 столбца действительных чисел, можно объявить двумерный массив Mas следующим образом:

var

```
Mas : array [1..2,1..3] of extended;
```

Когда по условию задачи матрица считается заданной, численные значения элементов массива можно задать в разделе **Const**, например:

Const

```
A:array[1..2,1..3] of integer=((11,12,13),  
(21,22,23));
```

2. Примеры создания приложений


Пример 1. Создать *Windows*-приложение для построения вектора $x = \{x_1, x_2, \dots, x_6\}$, равного p -й строке матрицы $A = \{a_{ij}\}$ ($x_j = a_{pj}$, $j = 1, 2, \dots, 6$) и вектора $y = \{y_1, y_2, \dots, y_4\}$, равного q -му столбцу матрицы $A = \{a_{ij}\}$ ($y_i = a_{iq}$, $i = 1, 2, \dots, 4$). Элементы матрицы A ввести с клавиатуры.

Методические рекомендации

Создание интерфейса

Один из возможных вариантов панели интерфейса создаваемого приложения приведен на рисунке 1, где показана Форма с компонентами.

При работе с массивами ввод и вывод информации на экран удобно организовывать с помощью компонента **StringGrid**, пикто-

грамма  которого находится на странице **Additional** Палитры Компонентов.

Компонент **StringGrid** используется для отображения информации в виде таблицы. Таблица содержит две зоны – фиксированную и рабочую.

Фиксированная зона служит для вывода наименований строк и столбцов рабочей зоны и управления их размерами с помощью мыши. Фиксированная зона выделена другим цветом и в нее запрещен ввод информации с клавиатуры. Количество строк и столбцов фиксированной зоны устанавливается в свойствах **FixedRows** и **FixedCols** соответственно.

Рабочая зона содержит **RowCount-1** строк и **ColCount-1** столбцов информации, которую можно изменять как программно, так и с помощью мыши или клавиатуры.

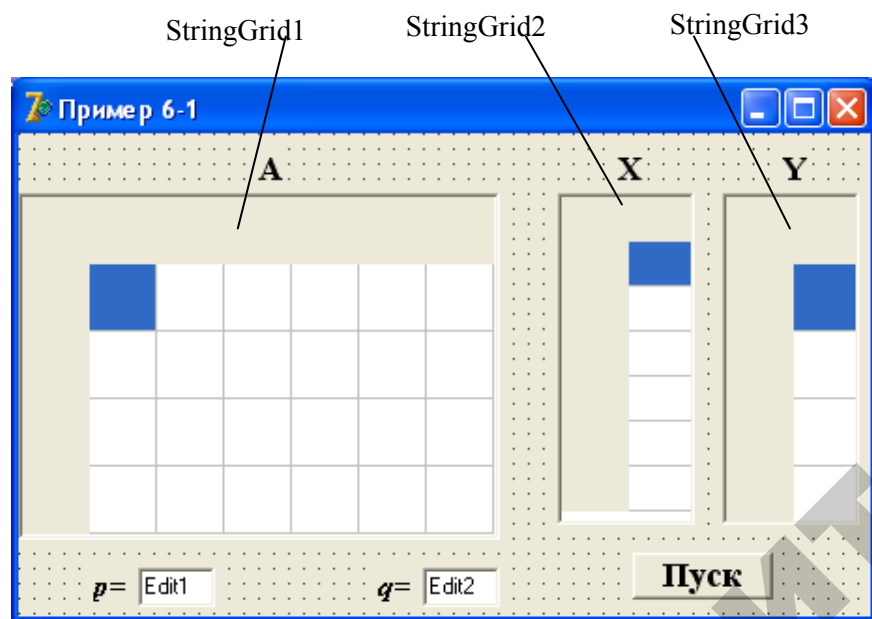


Рисунок 1. Форма с компонентами

Для всех компонентов **StringGrid** количество строк и столбцов фиксированной зоны сделаем равным единице. Для этого в Инспекторе Объектов значения свойств **FixedCols** и **FixedRows** установите равными 1. Значения **RowCount** строк и **ColCount** столбцов рабочей зоны устанавливается с учетом количества строк **FixedRows** и количества столбцов **FixedCols** фиксированной зоны. Поэтому, в соответствии с заданием, для компонента **StringGrid1** установите значения количества строк **RowCount=5** и количества столбцов **ColCount=7**. Для компонента **StringGrid2** количество столбцов

ColCount=2, а количество строк – **RowCount=7**; для компонента **StringGrid3 ColCount=2, RowCount=5**.

По умолчанию в компонент **StringGrid** запрещен ввод информации с клавиатуры. Поэтому, для того чтобы иметь возможность вводить и редактировать информацию в компоненте **StringGrid**, необходимо активизировать компонент **StringGrid1**, затем перейти в Инспектор Объектов и дважды щелкнуть мышью на символе + свойства + **Options**, где в открывшемся списке опций установить значение **goEditing** в **True**.

Сценарий работы приложения:

- 1) ввод с клавиатуры численных значений коэффициентов A_{ij} ;
- 2) ввод с клавиатуры численного значения p ;
- 3) ввод с клавиатуры численного значения q ;
- 4) щелчок на кнопке **Пуск**.

Разработка алгоритма

Блок-схема алгоритма процедуры обработки события нажатия кнопки **Пуск** может иметь вид, показанный на рисунке 2.

Сохранение проекта

Назовите проект **PrLab61**, а модуль проекта **UnLab61** и сохраните их в папке **Lab61**.

Создание процедуры **Button1Click** обработки события нажатия кнопки **Пуск**

Для создания процедуры-обработчика поместите курсор на изображение кнопки **Пуск** и дважды щелкните левой кнопкой мыши. Полученную заготовку процедуры дополните необходимыми операторами так, как показано ниже.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  A:array[1..4,1..6] of extended;
  X:array[1..6] of extended;
  Y:array[1..4] of extended;
  i,j,p,q:integer;
begin
  // Ввод значений из таблицы в массив A
  for i:=1 to 4 do
    for j:=1 to 6 do
      A[i,j]:=StrToFloat(StringGrid1.Cells[j,i]);
```

```

p:=StrToInt(Edit1.Text);
q:=StrToInt(Edit2.Text);
// Формирование массива X и вывод его значений в таблицу
for j:=1 to 6 do
begin
X[j]:=A[p,j];
StringGrid2.Cells[1,j]:=FloatToStrF(X[j],ffFixed,3,1);
end;
// Формирование массива Y и вывод его значений в таблицу
for i:=1 to 4 do
begin
Y[i]:=A[i,q];
StringGrid3.Cells[1,i]:=FloatToStrF(Y[i],ffFixed,3,1);
end;
end;

```

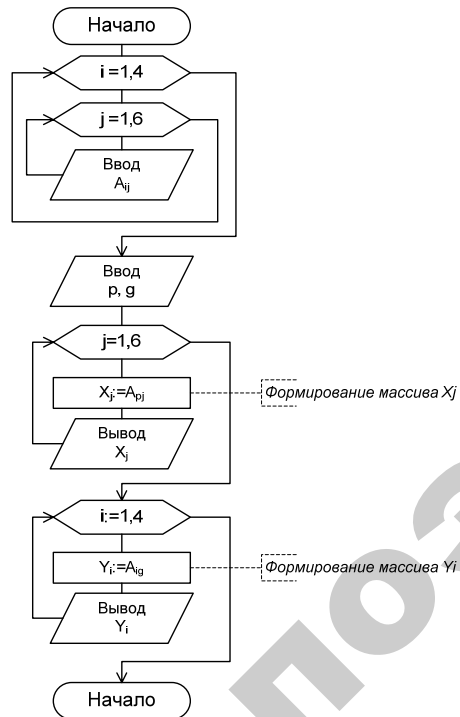



Рисунок 2. Блок-схема алгоритма процедуры обработки события нажатия кнопки **Пуск**

Работа с приложением

Для запуска созданного приложения нажмите пиктограмму  главного меню или клавишу **F9**. При этом происходит создание выполняемого файла **PrLab61.exe**. На экране появляется панель интерфейса приложения, показанная на рисунке 3.

В соответствии со сценарием работы приложения введите исходные данные в матрицу **A**, как показано на рисунке 3. Внесите в поля компонентов **Edit1** и **Edit2** численные значения переменных **p** и **q** соответственно.

На панели интерфейса нажмите кнопку **Пуск**. В результате будут сформированы массивы **X** и **Y**. Массив **X** содержит значения 2-ой строки ($p = 2$), массив **Y** – значения 3-его столбца ($q = 3$). Изменяя значения **p** и **q**, убедитесь в правильном функционировании приложения.

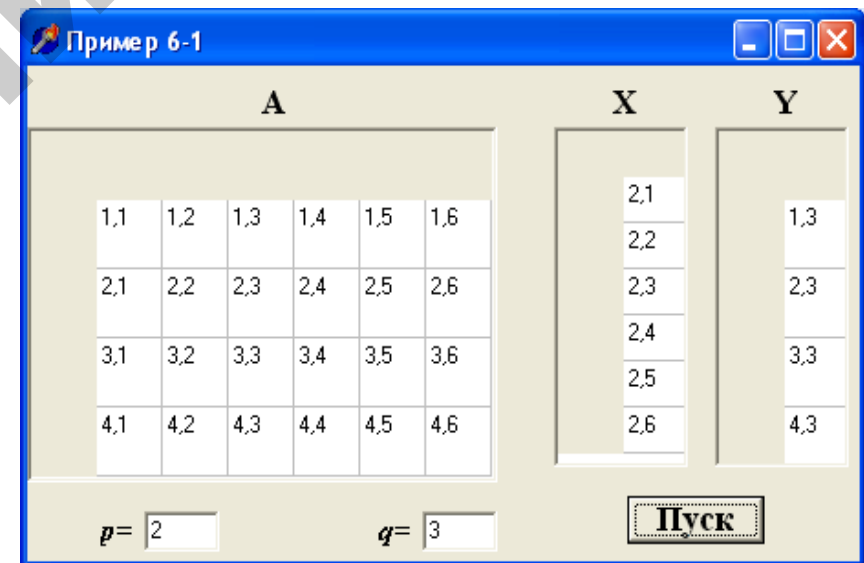


Рисунок 3. Панель интерфейса приложения

В случае нештатного функционирования приложения восстановить первоначальный режим работы с проектом можно выбором в меню **Run (Запуск)** опции **ProgramReset (Сброс программы)** или нажатием клавиши **Ctrl+F2**.

Пример 2. Задана квадратная матрица $A(5, 5)$ из целочисленных элементов. Вычислить след матрицы $Sp A$.

Методические рекомендации Создание интерфейса

Элементы матрицы $A(5, 5)$ зададим непосредственно в программе. Тогда один из возможных вариантов панели интерфейса создаваемого приложения может выглядеть так, как показано на рисунке 4. Здесь для вывода значений элементов массива A используется компонент **Мемо** — многострочный редактор.

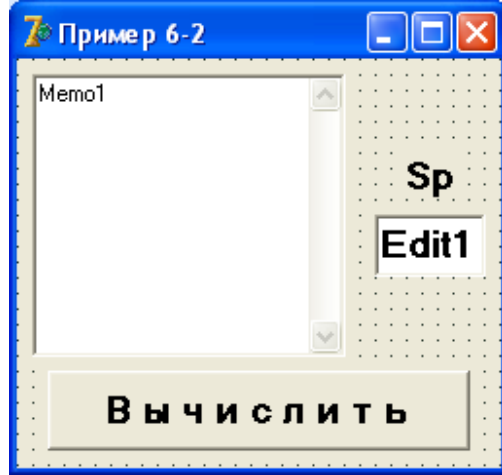


Рисунок 4. Форма с компонентами

Разработка алгоритма

Блок-схема алгоритма процедуры обработки события нажатия кнопки **Вычислить** может иметь вид, показанный на рисунке 5.

Сохранение проекта

Назовите проект **PrLab62**, а модуль проекта – **UnLab62** и сохраните их в папке **Lab62**.

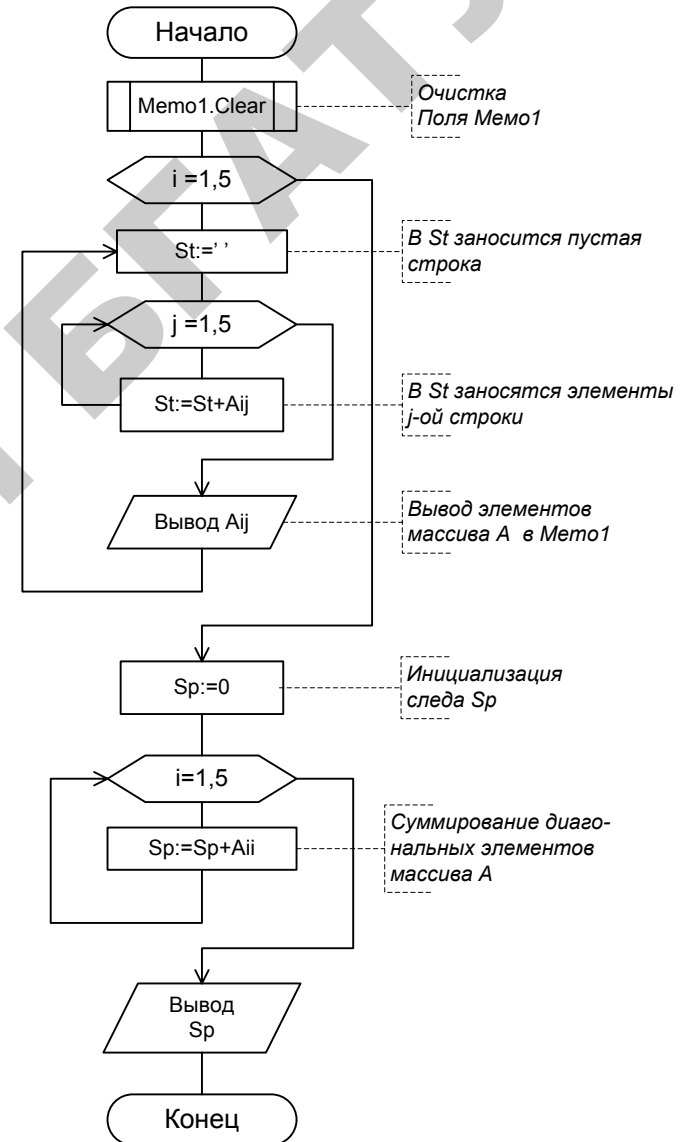


Рисунок 5. Блок-схема процедуры обработки события нажатия кнопки **Вычислить**

Создание процедуры Button1Click обработки события нажатия кнопки Вычислить

Для создания процедуры-обработчика поместите курсор на изображение кнопки **Вычислить** и дважды щелкните левой кнопкой мыши. Полученную заготовку процедуры дополните необходимыми операторами так, как показано ниже.

```
procedure TForm1.Button1Click(Sender: TObject);
Const
  A:array[1..5,1..5] of integer=((11,12,13,14,15),
                                (21,22,23,24,25),
                                (31,32,33,34,35),
                                (41,42,43,44,45),
                                (51,52,53,54,55));

Var
  i,j,Sp:integer;
  St:string;
begin
  Mem1.Clear;           //очистка поля Mem1
  for i:=1 to 5 do
    begin
      St:='';           // в St заносится пустая строка
      for j:=1 to 5 do
        St:=St+IntToStr(A[i,j])+ ' ';
      Mem1.Lines.Add(St); //St выводится в Mem1
    end;
  Sp:=0;
  for i:=1 to 5 do
    Sp:=Sp+A[i,i];
  Edit1.Text:=IntToStr(Sp) ;
end;
```

Работа с приложением

Запустите созданное приложение и на панели интерфейса нажмите кнопку **Вычислить**. Результат выполнения приложения показан на рисунке 6.

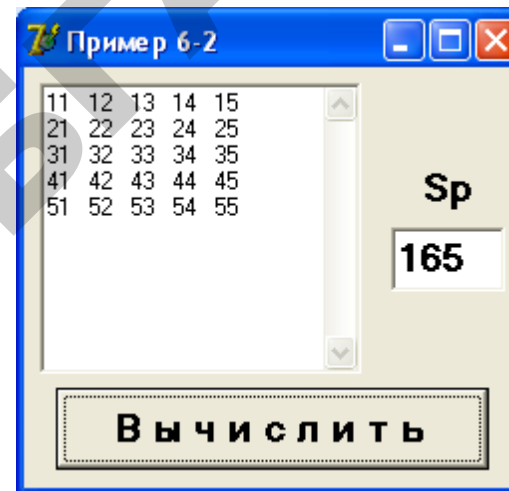


Рисунок 6. Результат выполнения приложения

Вопросы для самоконтроля

1. Что такое матрица? Приведите пример.
2. Что такое след матрицы?
3. Как объявить двумерный массив в программе?
4. Какие компоненты удобно применять при работе с двумерными массивами?
5. Как в среде Delphi осуществляется ввод и вывод элементов массива?
6. Что следует сделать, чтобы в компоненте StringGrid можно было вводить данные?
7. Как используется свойство Cells компонента StringGrid?

Варианты индивидуальных заданий разного уровня:

1 уровень – выполните задание 1;

2 уровень – выполните задания 1, 2;

3 уровень – выполните задания 1, 2, 3.

Вариант 1

1. Задан двумерный массив A размером 4×5 . Найти произведение элементов каждого столбца матрицы A .

2. Задана квадратная матрица B ($5, 5$) из целочисленных элементов. Определить среднее арифметическое элементов расположенных выше главной диагонали.

3. Для заданной матрицы размером $n \times m$ вывести на экран все ее седловые точки. Элемент матрицы называется 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.

Вариант 2

1. Задана матрица A размером 5×4 . Определить количество и сумму элементов каждой строки матрицы, кратных 5.

2. Задана квадратная матрица D размером 6×6 . Найти максимальный элемент массива, расположенный ниже главной диагонали, вывести на экран его координаты.

3. Задана целочисленная матрица A размером $n \times m$. Получить массив B , присвоив его k -му элементу значение 0, если элементы k -го столбца матрицы нулевые, и значение 1 в противном случае ($k = 1, 2, \dots, M$).

Вариант 3

1. Задан двумерный массив A размером 5×3 . В каждой строке матрицы A найти количество элементов, удовлетворяющих условию $1 \leq A_{ij} \leq 4$.

2. Задана квадратная матрица B ($3, 3$). Найти сумму тех строк матрицы, в которых элемент главной диагонали равен нулю.

3. Дан двумерный массив размером $n \times m$. Найти элемент матрицы, являющийся максимальным в своей строке и минимальным в своем столбце. Если такой элемент отсутствует, вывести соответствующее сообщение.

Вариант 4

1. Задан двумерный массив A размером 3×5 . Заменить минимальный элемент матрицы нулем. Полученную матрицу вывести на экран.

2. В квадратной матрице B ($5, 5$) найти сумму элементов каждой строки, кроме элемента, стоящего на главной диагонали.

3. Задана целочисленная матрица A размером $n \times m$. Получить массив B , присвоив его k -му элементу значение 1, если k -ая строка матрицы симметрична, и значение 0 в противном случае ($k = 1, 2, \dots, N$).

Вариант 5

1. Дана действительная матрица X размером 5×3 . В каждой строке матрицы найти сумму положительных элементов и количество отрицательных.

2. В квадратной матрице B ($3, 3$) найти произведение элементов каждого столбца, кроме элемента, стоящего на главной диагонали.

3. Задана целочисленная матрица размером $n \times m$. Определить k – количество «особых» элементов матрицы, считая элемент «особым», если он больше суммы остальных элементов своего столбца.

Вариант 6

1. Дана вещественная матрица A размером 5×4 . В каждом столбце матрицы подсчитать количество элементов, больших заданного числа F .

2. Все элементы квадратной матрицы C ($5, 5$), расположенные выше главной диагонали, переписать по строкам, начиная с первой, в одномерный массив B . Вывести на экран полученный массив.

3. Дан двумерный массив размером $n \times m$. Удалить все строки, содержащие ноль.

Вариант 7

1. Задана матрица D размером 4×5 . Найти минимальный элемент матрицы и вывести на экран элементы строки, в которой он расположен.

2. В квадратной матрице A ($5, 5$) заменить нулями элементы, расположенные на главной диагонали и выше ее.

3. Дан двумерный массив размером $n \times m$, заполненный случайным образом. Поменять местами k -ый и последний столбцы матрицы. Значение k задается с клавиатуры пользователем.

Вариант 8

1. Дана вещественная матрица B размером 5×4 . Подсчитать количество столбцов, сумма элементов которых положительна.
2. В целочисленной квадратной матрице C ($4, 4$) найти минимальный элемент, расположенный ниже главной диагонали.
3. Задан двумерный массив размером $m \times n$, заполненный случайным образом числами в диапазоне от -20 до 50 . Удалить строку и столбец, на пересечении которых расположен наибольший по модулю элемент массива.

Вариант 9

1. Задан двумерный массив B размером 4×5 . В каждой строке матрицы найти среднее арифметическое отрицательных элементов.
2. В квадратной матрице D ($4, 4$) подсчитать количество элементов, равных нулю, расположенных выше главной диагонали.
3. Даны вещественные числа x_1, x_2, \dots, x_8 . Получить квадратную матрицу R порядка 8, образованную по правилу:

$$R_{ij} = \begin{bmatrix} x_1 & x_2 & \dots & x_8 \\ x_1^2 & x_2^2 & \dots & x_8^2 \\ \dots & \dots & \dots & \dots \\ x_1^8 & x_2^8 & \dots & x_8^8 \end{bmatrix}.$$

Вариант 10

1. Задан двумерный массив C размером 4×3 . Заменить максимальный элемент матрицы на противоположный по знаку. Вывести на экран полученный массив.
2. Для каждой строки матрицы B ($4, 4$) подсчитать количество элементов, совпадающих с элементом, стоящим на главной диагонали соответствующей строки.
3. Пусть дано натуральное число n . Матрица A размером $n \times n$ сформирована по правилу:

$$a_{ij} = \sin\left(\frac{i^2 - j^2}{N}\right).$$

Определить, сколько положительных элементов содержит матрица A . Вывести на экран матрицу A .

Вариант 11

1. Задан двумерный массив A размером 5×3 . В каждом столбце матрицы найти количество отрицательных элементов и сумму неотрицательных.
2. В квадратной матрице C ($4, 4$) найти максимальный элемент среди элементов матрицы, расположенных выше главной диагонали.
3. Преобразовать квадратную матрицу A размером $n \times n$ по правилу – столбец с номером p поменять со строкой с номером q . Числа n, p, q задаются с клавиатуры.

Вариант 12

1. Задан двумерный массив X размером 3×4 . Заменить минимальный элемент матрицы нулем.
2. В квадратной матрице R размером 5×5 найти сумму элементов главной диагонали и произведение элементов, расположенных на побочной диагонали.
3. Из двух квадратных матриц A ($n \times n$) и B ($n \times n$) получить новую матрицу сложением элементов каждого столбца первой матрицы с наименьшим элементом соответствующего столбца второй матрицы.

Вариант 13

1. Задан двумерный массив C размером 4×6 . В каждом столбце массива найти произведение четных элементов.
2. В квадратной матрице D размером 4×4 найти максимальный элемент, расположенный выше главной диагонали, и заменить на его квадрат.
3. Задана матрица размером $n \times m$, заполненная случайным образом числами в диапазоне от -20 до 30 . Получить новую матрицу, выбросив из исходной строку с минимальным элементом.

Вариант 14

1. Задан двумерный массив Y размером 3×6 . Для каждой строки матрицы подсчитать количество тех элементов, значения которых удовлетворяют условию $1 \leq Y_{ij} \leq 1,5$.
2. Задана квадратная матрица B размером 6×6 . Определить, сумма каких элементов больше, – расположенных на главной диагонали или находящихся на побочной диагонали.

3. В зрительном зале 25 рядов, в каждом ряду 50 мест. Информация о проданных билетах хранится в двумерном массиве, номера строк которого соответствуют номерам рядов, а номера столбцов – номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, в противном случае – 0. Определить общее число свободных мест в каждом ряду зрительного зала. Вывести на экран соответствующее сообщение.

Вариант 15

1. Задан двумерный массив A размером 5×3 . Подсчитать количество строк массива, у которых первый элемент – отрицательное число.

2. В квадратной матрице G размером 4×4 определить координаты минимального элемента, расположенного выше главной диагонали.

3. Пусть дана вещественная квадратная матрица порядка n . Постройте последовательность вещественных чисел a_1, \dots, a_n по правилу: если в i -ой строке матрицы элемент, принадлежащий главной диагонали, отрицателен, то элемент a_i равен сумме положительных элементов i -ой строки, в противном случае a_i равен произведению отрицательных элементов i -ой строки.

Вариант 16

1. Задан двумерный массив D размером 3×4 . Заменить минимальный элемент каждого столбца матрицы нулем.

2. Задана квадратная матрица F размером 5×5 . Найти сумму всех элементов матрицы, кроме элементов, стоящих на главной диагонали.

3. Задана целочисленная матрица A размером $n \times m$. Получить массив B , присвоив его k -му элементу значение 1, если элементы k -ой строки матрицы упорядочены по убыванию, и значение 0 в противном случае ($k = 1, 2, \dots, N$).

Вариант 17

1. Задан двумерный массив B размером 4×5 . Для каждой строки матрицы найти сумму и количество элементов, кратных 3.

2. Все элементы квадратной матрицы C ($6, 6$), расположенные ниже главной диагонали, переписать в одномерный массив X .

3. Задана матрица размером $n \times m$. Определить количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз).

Вариант 18

1. Задан вещественный массив Y размером 3×6 . Найти среднее арифметическое тех строк матрицы, у которых первый элемент положительный.

2. Задана квадратная матрица B ($4, 4$). Определить среднее арифметическое элементов, расположенных ниже главной диагонали.

3. Определить, является ли заданная квадратная матрица n -го порядка симметричной относительно главной диагонали.

Вариант 19

1. Задан двумерный массив C размером 4×6 . Определить количество столбцов, в которых первый элемент равен нулю.

2. В квадратной матрице B ($4, 4$) определить количество ненулевых элементов, расположенных ниже главной диагонали.

3. В матрице n -го порядка переставить строки так, чтобы на главной диагонали матрицы были расположены элементы, наибольшие по абсолютной величине.

Вариант 20

1. Задан двумерный массив A размером 3×6 . В каждой строке матрицы найти произведение положительных элементов. Полученные результаты вывести на экран.

2. В квадратной матрице B размером 5×5 найти сумму тех строк, в которых элемент главной диагонали равен единице.

3. В заданной действительной матрице размером $n \times m$ поменять местами строку, содержащую элемент с наибольшим значением со строкой, содержащей элемент с наименьшим значением.

Вариант 21

1. Задан двумерный массив B размером 6×5 . Вывести на экран номера тех строк матрицы, сумма элементов которых – положительное число.

2. В квадратной матрице C размером 6×6 найти минимальный элемент, расположенный выше главной диагонали.

3. Из заданной целочисленной матрицы n -го порядка получить матрицу порядка $n - 1$ путем удаления из исходной матрицы строки и столбца, на пересечении которых расположен элемент с наибольшим значением (предполагается, что такой элемент единственный).

Вариант 22

1. Задан двумерный массив C размером 3×6 . Определить, есть ли в данном массиве строка, в которой ровно два отрицательных элемента.

2. В квадратной матрице B размером 4×4 найти произведение элементов матрицы, кроме элементов расположенного на главной диагонали.

3. Даны массив $A = (a_1, a_2, \dots, a_{10})$ и матрица B ($n \times n$). Заменить нулями в матрице B те элементы, для которых имеются равные среди элементов массива A .

Вариант 23

1. Задан двумерный массив D размером 4×3 . Найти количество строк матрицы, сумма элементов которых положительное число.

2. В квадратной матрице S (**6, 6**) найти количество элементов, принадлежащих интервалу $[-5, 5]$ и расположенных выше главной диагонали квадратной матрицы.

3. Определите, является ли заданная целая квадратная матрица n -го порядка магическим квадратом. Магическим квадратом называют матрицу, в которой суммы элементов во всех строках и столбцах одинаковы.

Вариант 24

1. Задан двумерный массив X размером 5×3 . В каждом столбце матрицы найти количество элементов, кратных 10.

2. В квадратной матрице Z размером 5×5 определить, сумма каких элементов больше – расположенных выше главной диагонали или находящихся ниже главной диагонали.

3. Дан двумерный массив размером $n \times m$, заполненный случайным образом. Поменять местами ее строки так, чтобы их минимальные элементы образовывали возрастающую последовательность.

Вариант 25

1. Дана матрица A размера 5×6 . В заданной матрице найти количество и вывести на экран номера строк матрицы, сумма элементов которых больше 20.

2. В квадратной матрице C размером 5×5 найти наибольший элемент, расположенный на главной диагонали.

3. В поезде 8 вагонов, в каждом из которых 36 мест. Информация о проданных на поезд билетах хранится в двумерном массиве, номера строк которых соответствуют номерам вагонов, а номера столбцов – номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, противном случае – 0. Составить программу, определяющую число свободных мест в любом из вагонов поезда.

Вариант 26

1. Дана матрица B размера 4×5 . В каждом столбце матрицы найти и вывести на экран наименьшей элемент.

2. Задана квадратная матрица Y размером 5×5 . Определить где четных элементов больше – выше главной диагонали или ниже главной диагонали.

3. Дан двумерный массив размером $n \times m$, заполненный случайным образом. Поменять местами второй и предпоследний столбцы. Полученный массив вывести на экран.

Вариант 27

1. Дана матрица C размера 4×5 . Вывести на экран номера строк, содержащих больше положительных элементов, чем отрицательных.

2. В квадратной матрице U размера 6×6 заменить все отрицательные элементы, расположенные выше главной диагонали, на 0.

3. Дана целочисленная матрица размера $m \times n$. Найти количество ее строк, все элементы которых различны.

Вариант 28

1. Дана матрица D размера 4×3 . Определить, расположены наибольший и наименьший элементы массива в одной строке или нет. Вывести на экран соответствующее сообщение.

2. Задана квадратная матрица C размером 5×5 . Найти количество отрицательных элементов, расположенных выше главной диагонали, сумму элементов, расположенных на главной диагонали, и произведение положительных элементов, расположенных ниже главной диагонали.

3. Дана целочисленная матрица размера $m \times n$. Найти элемент, являющийся максимальным в своей строке и минимальным в своем столбце. Если такой элемент отсутствует, то вывести на экран соответствующее сообщение.

Вариант 29

1. Дана матрица A размера 4×8 . Вывести на экран номера строк, в которых равное количество положительных и отрицательных элементов.

2. Задана квадратная матрица C размером 5×5 . Вместо максимального элемента, лежащего на главной диагонали, записать число 55555.

3. Задан двумерный массив размером $m \times n$, заполненный случайным образом числами в диапазоне от -20 до 50. Удалить строку и столбец, на пересечении которых расположен наименьший элемент массива.

Вариант 30

1. Задан двумерный массив B размера 6×4 . Найти и вывести на экран среднее арифметическое элементов каждого столбца матрицы.

2. В квадратной матрице D размером 6×6 определить произведение положительных элементов, расположенных выше главной диагонали.

3. Задана квадратная матрица R ($n \times n$). Определить в данном массиве столбец с максимальной суммой элементов.

ЛАБОРАТОРНАЯ РАБОТА №7

Решение прикладных задач, основанных на использовании алгоритмов численных методов вычисления интегралов

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании алгоритмов численных методов вычисления интегралов, в визуальной среде *Delphi*.

Порядок выполнения работы

1. Ознакомиться с разделом «Краткие теоретические сведения».
2. Пользуясь методическими рекомендациями выполнить раздел «Пример выполнения задания».
3. Результаты предъявить преподавателю.
4. Исправить замечания и сохранить полученные результаты.
5. Выполнить свой вариант из раздела «Индивидуальные задания».
6. Оформить отчет.

1. Краткие теоретические сведения

Определенный интеграл от непрерывной функции $f(x) \geq 0$ в пределах от a до b представляет площадь криволинейной трапеции S , ограниченной кривой $f(x)$, осью абсцисс и прямыми $x = a$, $x = b$ (рисунок 1).

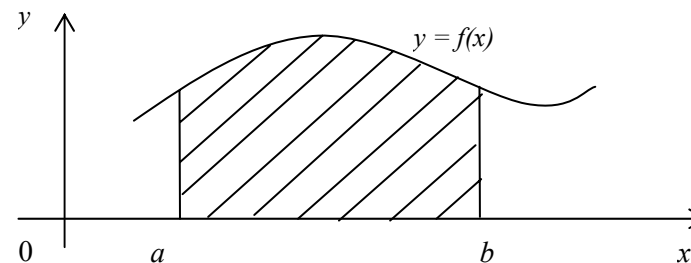


Рисунок 1. Определенный интеграл – площадь криволинейной трапеции

Из курса высшей математики известно, что

$$S = \int_a^b f(x) dx = F(x) \Big|_a^b = F(b) - F(a),$$

где $F(x)$ – первообразная для $f(x)$ на отрезке $[a, b]$, т.е. $F'(x) = f(x)$ на отрезке $[a, b]$. Если $f(x) < 0$ на отрезке $[a, b]$, то в формуле $S < 0$, но $|S|$ равно площади криволинейной трапеции, находящейся под осью абсцисс.

Однако на практике приведенной формулой часто нельзя воспользоваться по двум основным причинам:

1) вид функции $f(x)$ не допускает непосредственного интегрирования, т.е. первообразную нельзя выразить в элементарных функциях;

2) значения функции $f(x)$ заданы только на фиксированном конечном множестве точек x_i , т.е. функция задана в виде таблицы.

В этих случаях используются методы численного интегрирования. Они основаны на аппроксимации подынтегральной функции некоторыми более простыми выражениями, например, многочленами нулевой ($y = c$), первой ($y = cx + d$) или второй ($y = cx^2 + dx + k$) степени, а численные методы вычисления определенного интеграла, основанные на подобной аппроксимации, называются соответственно методами прямоугольников, трапеций и Симпсона (парабол).

Пусть требуется приближенно вычислить значение интеграла $S = \int_a^b f(x) dx$. В методе прямоугольников (рисунок 2) криволинейная

трапеция разбивается на n частей, каждая из которых представляет собой прямоугольник, основание которого равно шагу интегрирования $h = \frac{b-a}{n}$, а длины сторон соответственно $Y_0 = f(x_0)$, $Y_1 = f(x_1)$,

$Y_n = f(x_n)$, где $x_0 = a$, x_1, \dots, x_{n-1} , $x_n = b$ – точки деления отрезка $[a, b]$ на n равных частей.

Различают методы правых, левых и средних прямоугольников, в зависимости от расположения начальной точки x_0 при вычислении площади элементарного прямоугольника. При этом, если за высоту каждого прямоугольника принимается **левая** ордината (y_0, y_1, y_2, \dots), то вычисление интеграла будет производиться по методу **левых** прямоугольников; если **правая** ордината (y_1, y_2, y_3, \dots), то по методу **правых** прямоугольников; если за высоту принимается **середина** интервала h , то будет применяться метод **средних** прямоугольников. Основанием всех прямоугольников будет являться величина шага интегрирования h .

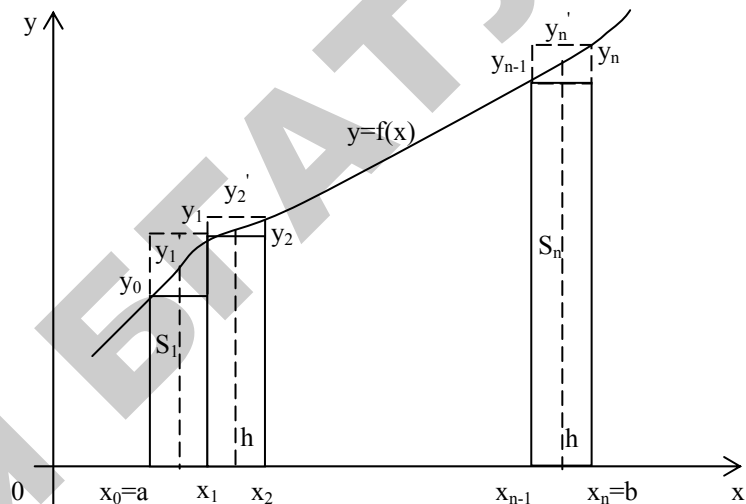


Рисунок 2. К выводу формул вычисления определенных интегралов методами прямоугольников

Тогда при методе левых прямоугольников

$$\int_a^b f(x) dx = S_1 + S_2 + \dots + S_n = y_0 \cdot h + y_1 \cdot h + \dots + y_{n-1} \cdot h,$$

при методе правых прямоугольников

$$\int_a^b f(x) dx = S_1 + S_2 + \dots + S_n = y_1 \cdot h + y_2 \cdot h + \dots + y_n \cdot h,$$

при методе средних прямоугольников

$$\int_a^b f(x) dx = S_1 + S_2 + \dots + S_n = y'_1 \cdot h + y'_2 \cdot h + \dots + y'_n \cdot h.$$

Как видно из рисунка 2, первоначальное значение x при методе левых прямоугольников $x = a$, правых — $x = a + h$, средних — $x = a + h/2$. Последующие значения x будут получаться через операцию присваивания $x = x + h$, а элементарные площади $S_1, S_2,$

..., S_n будут вычисляться по формуле $St = f(x) \cdot h$. Сумма этих площадей дает значение интеграла. Изложенное выше реализует алгоритм (рисунок 3), где St – значения элементарных площадей, а их сумма S – значение интеграла.

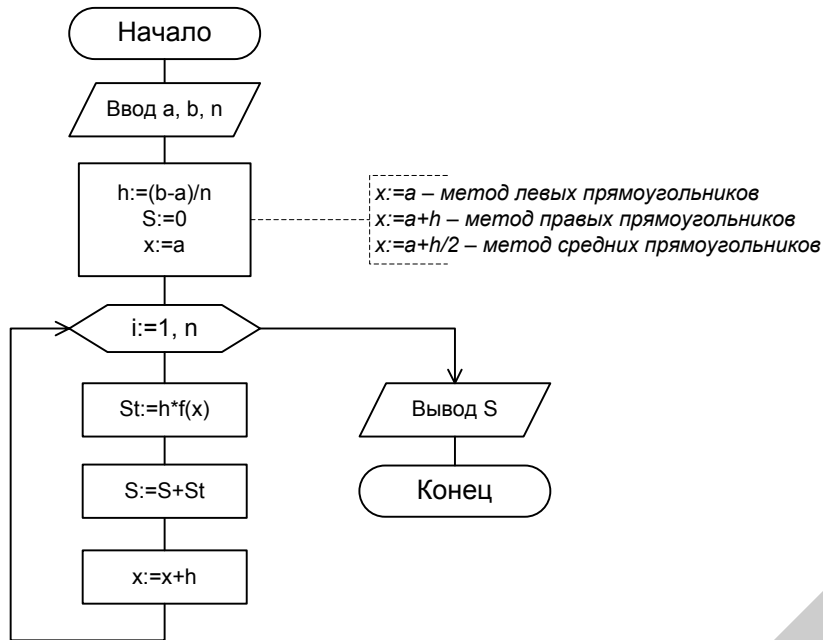


Рисунок 3. Схема алгоритма вычисления интеграла методом левых прямоугольников

Более точное значение интеграла получается при вычислении его методом трапеций, когда ординаты $y_0, y_1, y_2, \dots, y_n$ подынтегральной функции соединяют отрезками прямых и искомую площадь заменяют суммой площадей трапеций, высотой которых является шаг h , а основаниями y_0 и y_1 для S_1 , y_1 и y_2 для S_2 (рисунок 2).

Тогда

$$\int_a^b f(x) dx = S_1 + S_2 + \dots + S_i + \dots + S_n = h \frac{y_0 + y_1}{2} + h \frac{y_1 + y_2}{2} + \dots + h \frac{y_{n-1} + y_n}{2},$$

где $h = (b - a) / n$, а $y_0, y_1, y_2, \dots, y_n$ равны значениям функции $f(x)$ при соответствующих значениях аргумента x .

Поскольку $y_0 = f(a)$, $y_1 = f(a + h)$, $y_2 = f(a + 2h)$, ..., то схема алгоритма примет вид, приведенный на рисунке 4, где St – значения площадей трапеций, а их сумма S – значение интеграла.

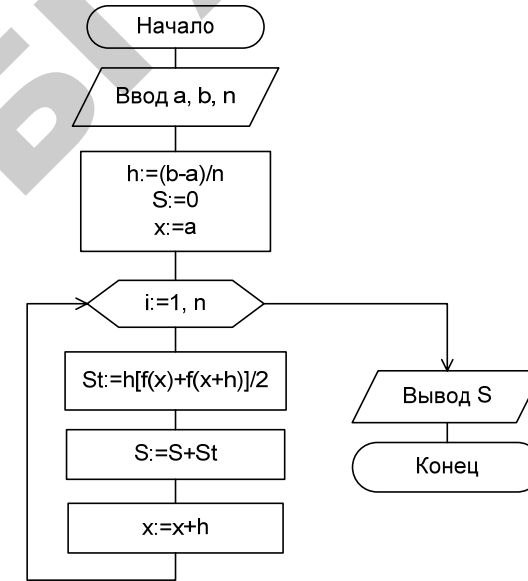


Рисунок 4. Схема алгоритма вычисления интеграла методом трапеций

Еще точнее значение интеграла получается при вычислении его методом Симпсона, когда на каждой паре соседних участков ординаты $y_0, y_1, y_2; y_2, y_3, y_4; y_4, y_5, y_6 \dots$ подынтегральной функции соединяют параболы, проходящими через 3 точки, и искомую площадь заменяют суммой площадей криволинейных трапеций.

Тогда

$$\int_a^b f(x) dx = S_1 + S_2 + \dots + S_i + \dots + S_n = h \frac{y_0 + 4y_1 + y_2}{3} + h \frac{y_2 + 4y_3 + y_4}{3} + \dots + h \frac{y_{2n-2} + 4y_{2n-1} + y_{2n}}{3},$$

где $h = \frac{(b-a)}{2n}$, а $y_0, y_1, y_2, \dots, y_n$ равны значениям функции $f(x)$ при соответствующих значениях аргумента x .

Поскольку $y_0 = f(a)$, $y_1 = f(a+h)$, $y_2 = f(a+2h)$, ..., то схема алгоритма примет вид, приведенный на рисунке 5, где St – значения площадей криволинейных трапеций, а их сумма S – значение интеграла.

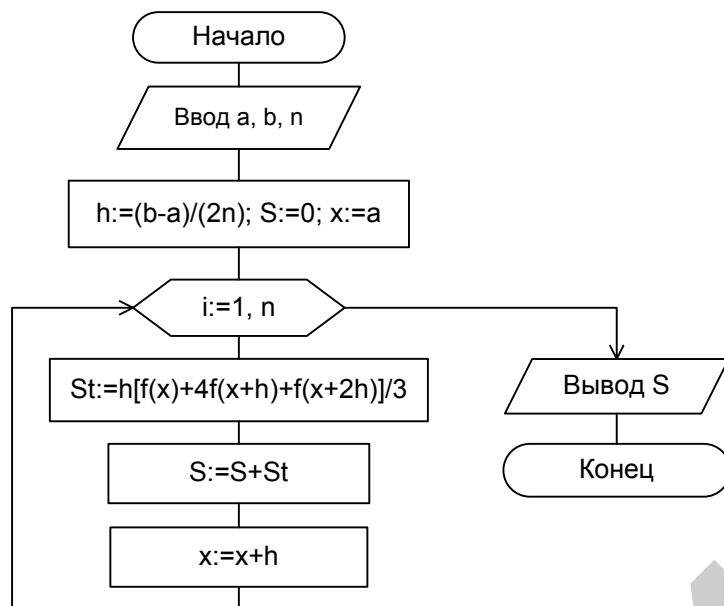


Рисунок 5. Схема алгоритма вычисления интеграла методом Симпсона

2. Пример выполнения задания

Разработать приложение для вычисления $\int_a^b \frac{e^x}{x+1} dx$ методом трапеций.

Методические рекомендации

Создание интерфейса

Один из возможных вариантов панели интерфейса создаваемого приложения может выглядеть следующим образом.

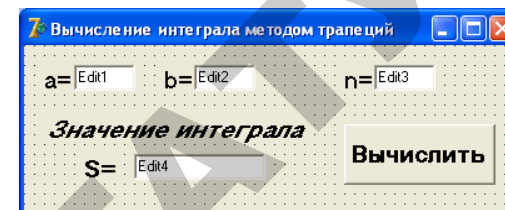


Рисунок 6. Форма с компонентами

Сценарий работы приложения:

- 1) ввод значения **a** в поле **Edit1**;
- 2) ввод значения **b** в поле **Edit2**;
- 3) ввод значения **n** в поле **Edit3**;
- 4) щелчок на кнопке **Вычислить**.

Разработка алгоритма

Блок-схема алгоритма нажатия кнопки **Вычислить** приведена на рисунке 4.

Написание процедуры Button1Click обработки события нажатия кнопки **Вычислить**

```

procedure TForm1.Button1Click(Sender: TObject);
var
  a, b, h, S, St, x: extended;
  i, n: integer;
begin
  a:=StrToFloat(Edit1.Text);
  b:=StrToFloat(Edit2.Text);
  n:=StrToInt(Edit3.Text);
  h:=(b-a)/n;
  S:=0;
  x:=a;
  for i:=1 to n do
  begin
    St:=h*(exp(x)/(x+1)+exp(x+h)/(x+h+1))/2;
    S:=S+St;
    x:=x+h;
  end;
  Edit4.Text:=FloatToStrF(S, ffFixed, 8, 3);
end;
  
```

Работа с приложением

Запустите созданное приложение и введите исходные данные, как показано на рисунке 7. На панели интерфейса нажмите кнопку **Вычислить**. Результат выполнения приложения показан на рисунке 7.

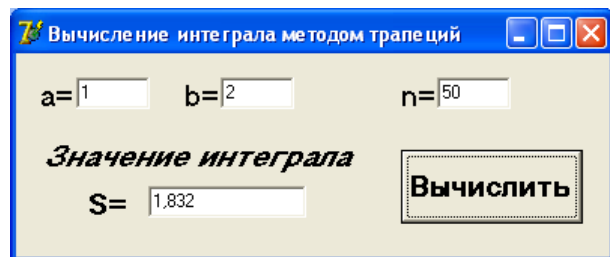


Рисунок 7. Панель интерфейса приложения

Вопросы для самоконтроля

1. В чем сущность численных методов вычисления определенного интеграла?
2. От чего зависит точность вычисления интеграла?
3. В чем отличия методов прямоугольников, трапеций и Симпсона?
4. Чем объясняется различная точность при вычислении интеграла методами прямоугольников, трапеций и Симпсона?

Варианты индивидуальных заданий разного уровня:

- 1 уровень** – вычислите интеграл методом прямоугольников;
2 уровень – вычислите интеграл методом прямоугольников и методом трапеций;
3 уровень – вычислите интеграл методами прямоугольников, трапеций и Симпсона.

Вариант 1	$\int_0^{\pi} \frac{\sin x}{\sqrt{5-4\cos x}} dx$	Вариант 16	$\int_0^1 x \ln(1+x) dx$
Вариант 2	$\int_0^1 \frac{7-x}{(x+1)^2} dx$	Вариант 17	$\int_1^{\pi/2} \frac{1}{1-\cos 2x} dx$
Вариант 3	$\int_0^{\pi/4} \cos 5x \sin 3x dx$	Вариант 18	$\int_0^2 \frac{3x^2 - 2x}{(x^2 + 1)^2} dx$
Вариант 4	$\int_0^1 \frac{1-x^2}{(x+1)^2} dx$	Вариант 19	$\int_0^{\pi/4} 2 \cos x \sin 3x dx$
Вариант 5	$\int_0^{\pi/2} \frac{1}{1+\cos x} dx$	Вариант 20	$\int_0^5 \frac{x}{\sqrt{1+3x}} dx$
Вариант 6	$\int_1^3 \frac{8(x-1)}{(x+1)^2} dx$	Вариант 21	$\int_0^{\pi/2} \cos \frac{x}{2} \cos \frac{3x}{2} dx$
Вариант 7	$\int_0^{\pi/2} \frac{1}{1+\sin x} dx$	Вариант 22	$\int_{\pi/4}^{\pi/2} \frac{1}{\sin^2 x} dx$
Вариант 8	$\int_0^2 \frac{2x^2 + 2x + 1}{(2x+1)^2} dx$	Вариант 23	$\int_1^2 (x+1) \ln x dx$
Вариант 9	$\int_0^{\pi/2} 6 \sin x \cos^2 x dx$	Вариант 24	$\int_0^{\pi/3} \frac{\sin 2x}{\cos^3 x} dx$
Вариант 10	$\int_0^1 \frac{3x(x+2)}{(x+1)^2} dx$	Вариант 25	$\int_{\pi/2}^{\pi} \sqrt{1+\sin(x)} dx$

Вариант 11	$\int_{\pi/4}^{\pi/2} 8 \sin x \cos^2 x dx$	Вариант 26	$\int_0^{\pi/4} \frac{4}{(\sin x + \cos x)^2} dx$
Вариант 12	$\int_{-3}^0 \frac{9}{\sqrt{25+3x}} dx$	Вариант 27	$\int_0^1 x \operatorname{arctg} x dx$
Вариант 13	$\int_0^{\pi/3} \frac{3 \sin x}{\cos x} dx$	Вариант 28	$\int_0^{\pi} \cos \frac{x}{2} \cos \frac{x}{3} dx$
Вариант 14	$\int_1^3 \frac{x^2 + 2x + 3}{(x+1)^2} dx$	Вариант 29	$\int_{-1}^0 (x+1)e^{-2x} dx$
Вариант 15	$\int_0^{\pi/4} 3 \sin^2 2x dx$	Вариант 30	$\int_0^{\pi/2} \frac{1}{(0,5 \sin x + \cos x)^2} dx$

ЛАБОРАТОРНАЯ РАБОТА №8

Решение прикладных задач, основанных на использовании алгоритмов численных методов решения нелинейных уравнений

Цель лабораторной работы – приобрести навыки решения прикладных задач, основанных на использовании алгоритмов численных методов решения нелинейных уравнений, в визуальной среде *Delphi*.

Порядок выполнения работы

1. Ознакомиться с разделом «Краткие теоретические сведения».
2. Пользуясь методическими рекомендациями, выполнить раздел «Пример выполнения задания».
3. Результаты предъявить преподавателю.
4. Исправить замечания и сохранить полученные результаты.
5. Выполнить свой вариант из раздела «Индивидуальные задания».
6. Оформить отчет.

1. Краткие теоретические сведения

При решении любое нелинейное уравнение, например $x^3 - 3x = 1$, предварительно приводится к виду $f(x) = 0$, т.е. $x^3 - 3x - 1 = 0$, при этом функции $f(x)$ могут быть алгебраическими, у которых над аргументом x выполняются лишь арифметические операции, и трансцендентными, которые содержат, кроме того, показательные, логарифмические, тригонометрические функции аргумента x . Такие уравнения далеко не всегда можно решить точно, а для практических расчетов точное решение не всегда является обязательным. Поэтому при решении практических задач часто применяют численные методы приближенного вычисления корней.

Численные методы основаны на последовательном уточнении значения корня от какого-то начального значения до достижения требуемой точности. Каждое повторное уточнение корня называется *итерацией*. Количество итераций, которое необходимо сделать, заранее не известно и зависит от удачного выбора начального значения корня, вида функции $y = f(x)$, требуемой точности вычисления корня и, наконец, от выбранного численного метода.

В общем случае задача решения нелинейного уравнения состоит из двух этапов:

1) **отделение корней**, т.е. выделение таких отрезков $[a, b]$, в каждом из которых находится **только один корень** уравнения;

2) **уточнение корней**, т.е. нахождение их на найденных отрезках с заданной степенью точности.

Для составления алгоритма решения задачи отделения корней рассмотрим поведение гладкой функции $y = f(x)$ на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом h (рисунок 1).

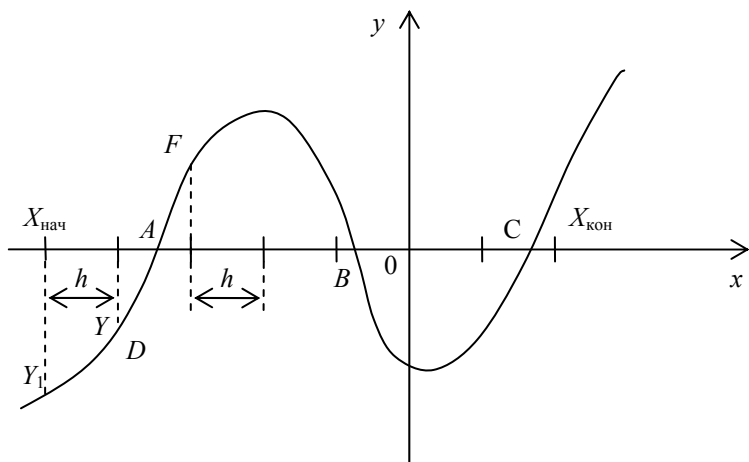


Рисунок 1. График функции $y = f(x)$ при наличии корней

Как видно из рисунка, корни уравнения находятся в точках A, B и C , где $f(x) = 0$. Данные точки можно обнаружить, рассчитывая значения функции $y = f(x)$ от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом h и наблюдая за изменением ее знака. Так, в точке A значение функции изменяется с « $-$ » на « $+$ », а в точке B — с « $+$ » на « $-$ » и т.д. Обнаружить изменение знака функции можно путем перемножения двух ее значений, вычисленных от аргументов x , находящихся в соседних точках на расстоянии h одно от другого. Для точки A , например, это будут значения функции D и F .

Отделение корней по рассмотренному методу реализует алгоритм, представленный на рисунке 2.

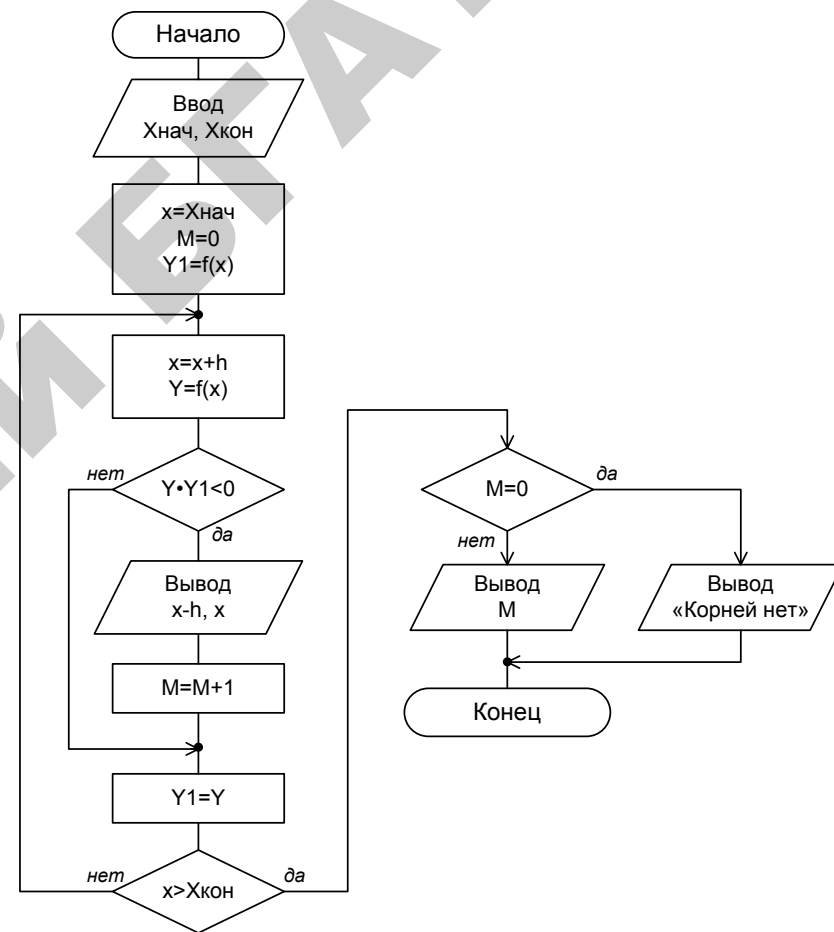


Рисунок 2. Схема алгоритма отделения корней функции $f(x)$ на отрезке $[X_{\text{нач}}, X_{\text{кон}}]$

2. Пример выполнения задания

Найти все действительные корни нелинейного уравнения

$$y = f(x) = x^3 - 3x^2 + x + 1 = 0$$

с точностью $\varepsilon = 10^{-5}$.

Методические рекомендации

1-й этап: отделение корней

Создание интерфейса приложения

Один из возможных вариантов панели интерфейса приложения для отделения корней приведен на рисунке 3, где показана Форма с компонентами.

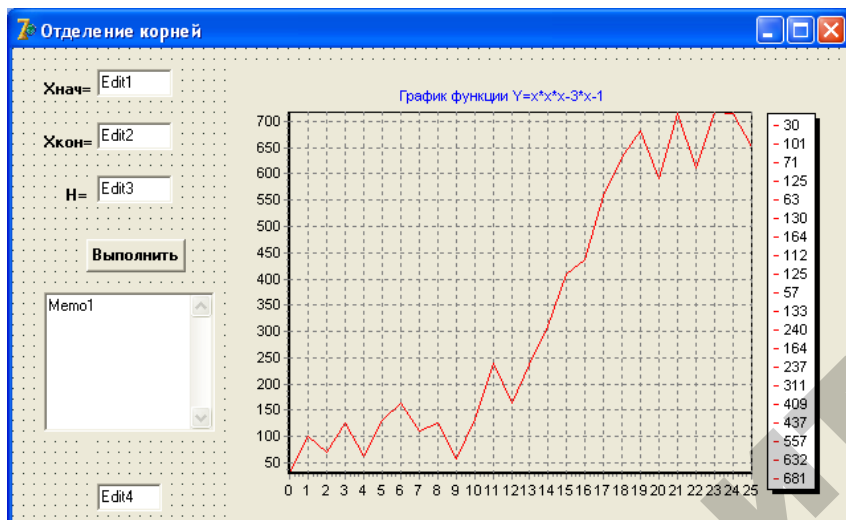


Рисунок 3. Форма с компонентами

Сценарий работы приложения:

- 1) ввод значения *Хнач* в поле **Edit1**;
- 2) ввод значения *Хкон* в поле **Edit2**;
- 3) ввод значения *h* в поле **Edit3**;
- 4) щелчок на кнопке **Выполнить**.

Разработка алгоритма

Блок-схема алгоритма нажатия кнопки **Выполнить** приведена на рисунке 2 (процедура построения графика на блок-схеме отсутствует).

Написание процедуры **Button1Click** обработки события нажатия кнопки **Выполнить**

```
procedure TForm1.Button1Click(Sender: TObject);
var
  x, Xn, Xk, h, Y, Y1: extended;
  M: integer;
begin
  Xn:=StrToFloat(Edit1.Text);
  Xk:=StrToFloat(Edit2.Text);
  h:=StrToFloat(Edit3.Text);
  x:=Xn; M:=0;
  Series1.Clear;
  Y1:=x*x*x-3*x-1;
  repeat
    x:=x+h;
    Y:=x*x*x-3*x-1;
    Series1.AddXY(x, Y);
    if Y*Y1<0 then
      begin
        Memo1.Lines.Add('A'+FloatToStrF(x-h, ffFixed, 8, 2)+
          ' B'+FloatToStrF(x, ffFixed, 8, 2));
        M:=M+1;
      end;
    Y1:=Y;
  until x>Xk;
  if M=0 then
    Label4.Caption:='Корней нет'
  else
    begin
      Label4.Caption:='Корней: ';
      Edit4.Text:=IntToStr(M);
    end;
end;
```

Работа с приложением

Запустите созданное приложение и введите исходные данные, как показано на рисунке 4. На панели интерфейса нажмите кнопку **Выполнить**.

Результат выполнения приложения показан на рисунке 4: уравнение имеет три корня, расположенных на отрезках $[-1,6;-1,5]$, $[-0,4;-0,3]$, $[1,8;1,9]$.

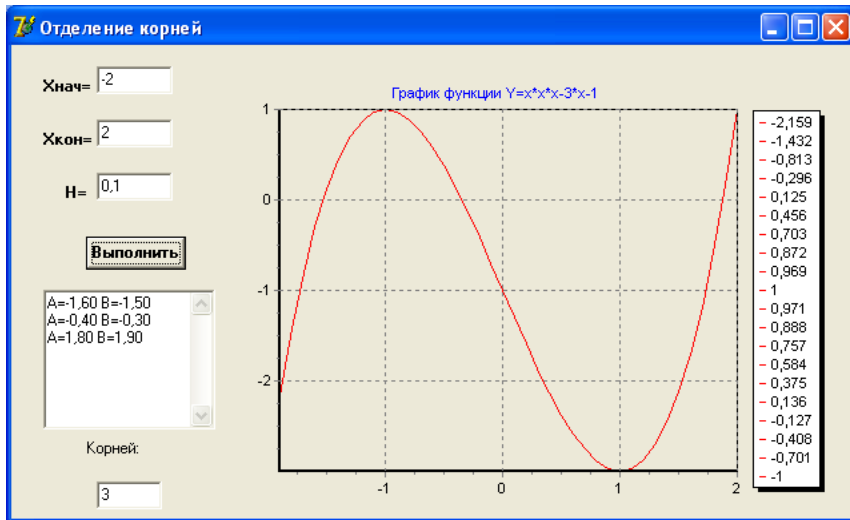


Рисунок 4. Панель интерфейса приложения

2-й этап: уточнение корней

Уточнение корня, найденного на отрезке $[a, b]$, осуществляется одним из следующих методов: деления отрезка пополам, хорд (секущих), касательных (Ньютона), итераций.

Рассмотрим метод деления отрезка пополам (рисунок 5). Отрезок $[a, b]$ делится пополам и в найденной точке $c = (a + b) / 2$ вычисляется значение функции $y = f(c)$. Если $|y| \leq e$, где e – заданная точность, то c является корнем уравнения, т.к. при полученном значении c функция $y = f(c)$ пересекает ось OX . В противном случае выбираем один из отрезков или $[a, (a + b)/2]$ (рисунок 5, а) или $[(a + b)/2, b]$ (рисунок 5, б), на концах которого $f(x)$ имеет противоположные знаки. Выбранный отрезок снова делим пополам $c = (a + b) / 2$ и вычисляем значение функции $y = f(c)$. Процесс повторяется до тех пор, пока не будет получено значение $|y| \leq e$.

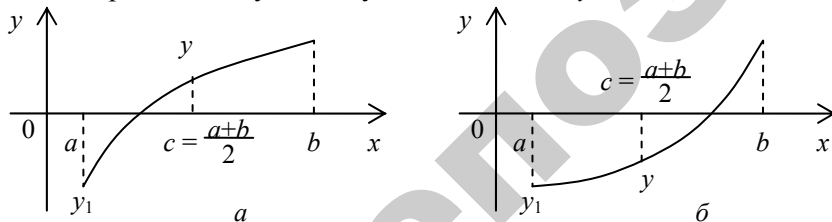


Рисунок 5. Уточнение корня методом деления отрезка пополам

Схема алгоритма уточнения корня методом деления отрезка пополам представлена на рисунке 6.

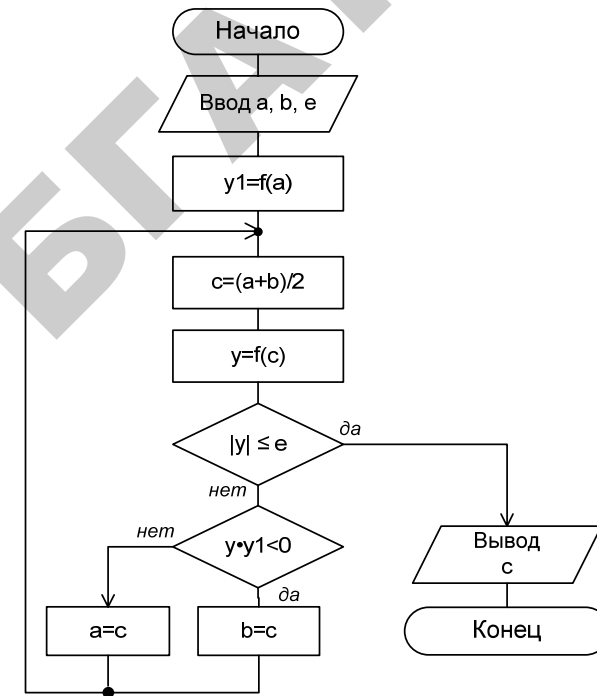


Рисунок 6. Схема алгоритма уточнения корня методом деления отрезка пополам

Рассмотрим уточнение корня методом касательных (рисунок 7). Пусть функция $y = f(x)$ имеет на отрезке $[a, b]$ единственный корень, а также на этом отрезке первая и вторая производные не меняют своего знака, т.е. кривая либо выпукла, либо вогнута.

Проведем касательную к графику функции $y = f(x)$ в точке x_1 в том конце отрезка $[a, b]$, в котором знаки $f(x)$ и $f''(x)$ совпадают, т.е. для вогнутой функции выбираем тот конец отрезка, где она положительна, для выпуклой – отрицательна. Уравнение касательной может быть записано в виде $y - f(x_1) = f'(x_1)(x - x_1)$. Найдем координату точки x_2 – точки пересечения касательной с осью OX . Точка x_2 будет расположена ближе к корню уравнения, чем точка x_1 . Так

ЛИТЕРАТУРА

1. *Архангельский, А. Я.* Приемы программирования в Delphi: учебник / А. Я. Архангельский. – Москва: Бинوم-Пресс, 2004. – 848 с.
2. *Архангельский, А. Я.* Программирование в Delphi 7: учебник / А. Я. Архангельский. – Москва: Бинوم-Пресс, 2004. – 1152 с.
3. *Вирт, Н.* Алгоритмы + структуры данных + программы : учебник / Н. Вирт. – Москва: Мир, 1985. – 218 с.
4. Информатика. Базовый курс / С. В. Симонович [и др.]; под ред. С. В. Симоновича. – Санкт-Петербург: Питер, 1999. – 419 с.
5. *Кнут, Д.* Искусство программирования для ЭВМ: монография в 3-х т. / Д. Кнут. – Москва: Мир, 1978. – Т 3. – 167 с.
6. *Культин, Н. В.* Основы программирования в Delphi 8 для Microsoft: учебник/ Н. В. Культин. – Санкт-Петербург: BHV, 2004. – 400 с.
7. *Прищепов, М. А.* Программирование на языках Basic, Pascal и Object Pascal в среде Delphi : учеб. пособие / М. А. Прищепов, Е. В. Севернева, А. И. Шакирин ; под общ. ред. М. А. Прищепова. – Мн. : ТетраСистемс, 2006. – 320 с.
8. *Прищепов, М. А.* Экзамен по информатике. Основы алгоритмизации и программирования: справ. пособие / М. А. Прищепов, В. П. Степанцов, Е. В. Севернева. – Минск: ТетраСистемс, 2001. – 192 с.
9. Программирование на языке Object Pascal в визуальной среде Delphi: методические указания к лабораторным занятиям по дисциплине «Основы информатики и вычислительной техники» в 2-х частях / сост. А. И. Шакирин. – Минск: БГАТУ, 2001. – Ч. 1. – 75 с.
10. Программирование на языке Object Pascal в визуальной среде Delphi: методические указания к лабораторным занятиям по дисциплине «Основы информатики и вычислительной техники» в 2-х частях / сост. А. И. Шакирин. – Минск: БГАТУ, 2001. – Ч. 2. – 45 с.
11. *Фаронов В. В.* Система программирования в Delphi в подлиннике: учебник / В. В. Фаронов. – Санкт-Петербург: BHV, 2003. – 912 с.
12. *Шакирин, А. И.* Основы программирования на алгоритмическом языке Object Pascal в визуальной среде Delphi : конспект лекций по дисциплине «Основы информационных технологий» / А. И. Шакирин. – Минск: БГАТУ, 2005. – 142 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ЛАБОРАТОРНАЯ РАБОТА №1	
Решение прикладных задач, основанных на использовании линейных алгоритмов	4
ЛАБОРАТОРНАЯ РАБОТА №2	
Решение прикладных задач, основанных на использовании разветвляющихся алгоритмов	31
ЛАБОРАТОРНАЯ РАБОТА № 3	
Решение прикладных задач, основанных на использовании циклических алгоритмов с известным количеством повторений цикла	61
ЛАБОРАТОРНАЯ РАБОТА №4	
Решение прикладных задач, основанных на использовании циклических алгоритмов с неизвестным количеством повторений цикла	83
ЛАБОРАТОРНАЯ РАБОТА №5	
Решение прикладных задач, основанных на использовании алгоритмов с одномерными массивами	106
ЛАБОРАТОРНАЯ РАБОТА №6	
Решение прикладных задач, основанных на использовании алгоритмов с двумерными массивами	133
ЛАБОРАТОРНАЯ РАБОТА №7	
Решение прикладных задач, основанных на использовании алгоритмов численных методов вычисления интегралов	152
ЛАБОРАТОРНАЯ РАБОТА №8	
Решение прикладных задач, основанных на использовании алгоритмов численных методов решения нелинейных уравнений	162
ЛИТЕРАТУРА	171

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕТОК

РЕПОЗИТОРИЙ БГАТУ

ДЛЯ ЗАМЕТОК

Учебное издание

Исаеня Николай Васильевич, **Силкович** Юрий Николаевич,
Шакирин Анатолий Иванович, **Львова** Ольга Михайловна

РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ
НА АЛГОРИТМИЧЕСКОМ ЯЗЫКЕ
Object Pascal

Лабораторный практикум

Ответственный за выпуск *Ю. Н. Силкович*
Редактор *Н. А. Антипович*
Компьютерная верстка *А. И. Стебуля*

Подписано в печать 7.05.2010 г. Формат 60×84¹/₁₆.

Бумага офсетная. Ризография.

Усл. печ. л. 10,23. Уч.-изд. л. 8,12. Тираж 75 экз. Заказ 239.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный аграрный технический университет».

ЛИ № 02330/0552841 от 14.04.2010.

ЛП № 02330/0552743 от 02.02.2010.

Пр. Независимости, 99–2, 220023, Минск.