

Федеральное агентство по образованию РФ  
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
(ГОУВПО «АмГУ»)

Т.А. Макаруч

## **Основы программирования в Delphi**

Лабораторный практикум



**Благовещенск**

**2005**

*Печатается по решению  
редакционно-издательского совета  
факультета математики и информатики  
Амурского государственного университета*

*Макарчук Т.А. (составитель)*

Основы программирования в Delphi. Лабораторный практикум. –  
Благовещенск: Амурский гос. ун-т, 2005.

Лабораторный практикум является дополнением к учебно-методическому пособию автора «Основы программирования в Delphi» и содержит систематизированное изложение практических основ применения языков и систем программирования в области экономики. Для студентов высших учебных заведений РФ, обучающихся по экономическим специальностям, а также аспирантов, молодых преподавателей и научных сотрудников, изучающим основы программирования для решения экономических задач.

*Рецензенты:*

Зав. кафедрой информатики БГПУ, доцент, к.ф.-м.н. Е.Ф. Алутина

Доцент кафедры ИУС, к.т.н. Т.А. Галаган

@ Амурский государственный университет, 2005

**СОДЕРЖАНИЕ**

Введение .....	4
Тема №1. Знакомство с системой программирования Delphi .....	5
Тема №2. Программирование вычислительных задач .....	13
Тема №3. Программы с ветвлением. Компонент TCheckBox .....	20
Тема №4. Программы выбора из списка. Класс TStrings.....	31
Тема №5. Программы циклической структуры .....	42
Тема №6. Программы работы с векторами .....	56
Тема №7. Программы работы с матрицам .....	65
Тема №8. Программы построения графиков .....	73
Литература.....	79
Приложение А. Типовые экзаменационные задачи.....	80

## Введение

В современном информационном обществе специалисты, в том числе и экономисты, должны не только обладать определенными знаниями, умениями и творческими навыками в области фундаментальных наук, но и владеть методами, способами и практическими навыками использования информационно-компьютерных технологий в своей профессиональной деятельности.

Согласно ГОС ВПО обучение принципам программирования и создания алгоритмов является обязательной составляющей общей фундаментальной подготовки экономистов и является основой для дальнейшего усвоения принципов создания, эксплуатации, развития современных информационных систем.

Лучшим языком для изучения программирования является Pascal с системой программирования Turbo Pascal для MS DOS. После появления графической операционной системы Windows 3.1 разработчики систем программирования, прежде всего корпорации Microsoft и Borland, не замедлили выпустить соответствующие средства: в 1991 г. Borland выпускает Turbo Pascal for Windows, в 1992 г. усовершенствованную версию этой системы программирования - Borland Pascal with Object 7.0, в 1993 г. вышла первая визуальная среда программирования Visual Basic фирмы Microsoft, в 1995 г. корпорация Borland выпустила первую версию Delphi, а затем, с интервалом в один год еще несколько версий: 2, 3, 4, 5, 6 и 7.

Данное пособие предназначено для студентов экономических специальностей, которые самостоятельно или под руководством преподавателя могут научиться основам программирования в Delphi.

Лабораторный практикум содержит восемь лабораторных работ, каждая из которых содержит этапы выполнения типового упражнения темы, задания для самостоятельной работы и варианты тестов для самоконтроля знаний. Также в практикум включен перечень примерных экзаменационных задач.

## Лабораторная работа №1

### Знакомство с системой программирования Delphi

#### Цель занятия

##### **узнать:**

- два вида интерфейса MDI и SDI;
- структуру интерфейса Delphi;
- свойства: Font, Color, Visible, Name, Top, Width, Height, Left, Caption;
- событие OnClick;
- компоненты TLabel, TPanel, TButton;
- правило задания названия процедуры - реакции на события.

##### **научиться:**

- размещать компоненты на форме;
- задавать свойства и события объекта;
- описывать реакцию на событие в виде процедуры;
- сохранять и открывать программу: файлы \*.dpr, \*.pas, \*.dfm, \*.exe;
- запускать программу на исполнение и прекращать выполнение программы.

**Задача:** Программа «Светофор» для изменения цвета надписи.

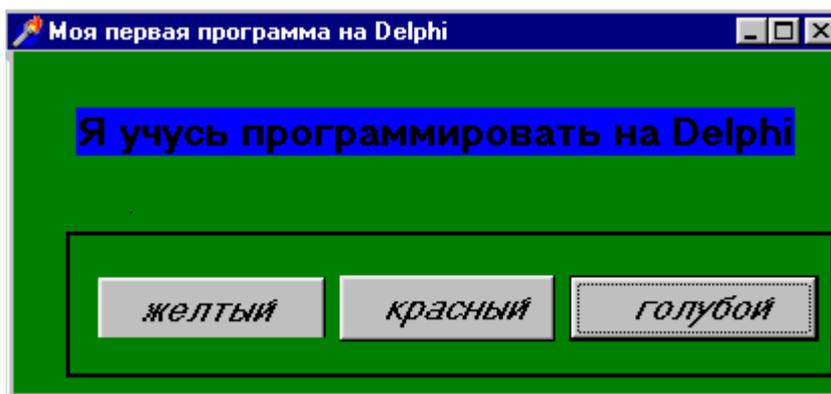


Рис.1.1. Пример окна программы «Светофор».

#### Этапы выполнения задачи

##### **Этап 1. Структура среды программирования**

Внешний вид системы программирования Delphi отличается от многих приложений Windows. Все программы Microsoft Office (Word, Excel и др.) являются MDI приложениями. MDI (Multiple Document Interface) определяет особый способ управления несколькими дочерними окнами внутри одного большого окна. Delphi следует другой спецификации, называемой Single Document Interface (SDI) и состоит из нескольких отдельно расположенных окон. SDI близок интерфейсу, используемому в Windows. Окна могут перемещаться по экрану, частично или полностью перекрывая друг друга.

Интерфейс Delphi (рис. 1.2) представлен следующими окнами:

1. главное окно (Project1);
2. окно формы (Form1);
3. окно Инспектора Объектов (Object Inspector);
4. окно кода программы (Unit1.pas).

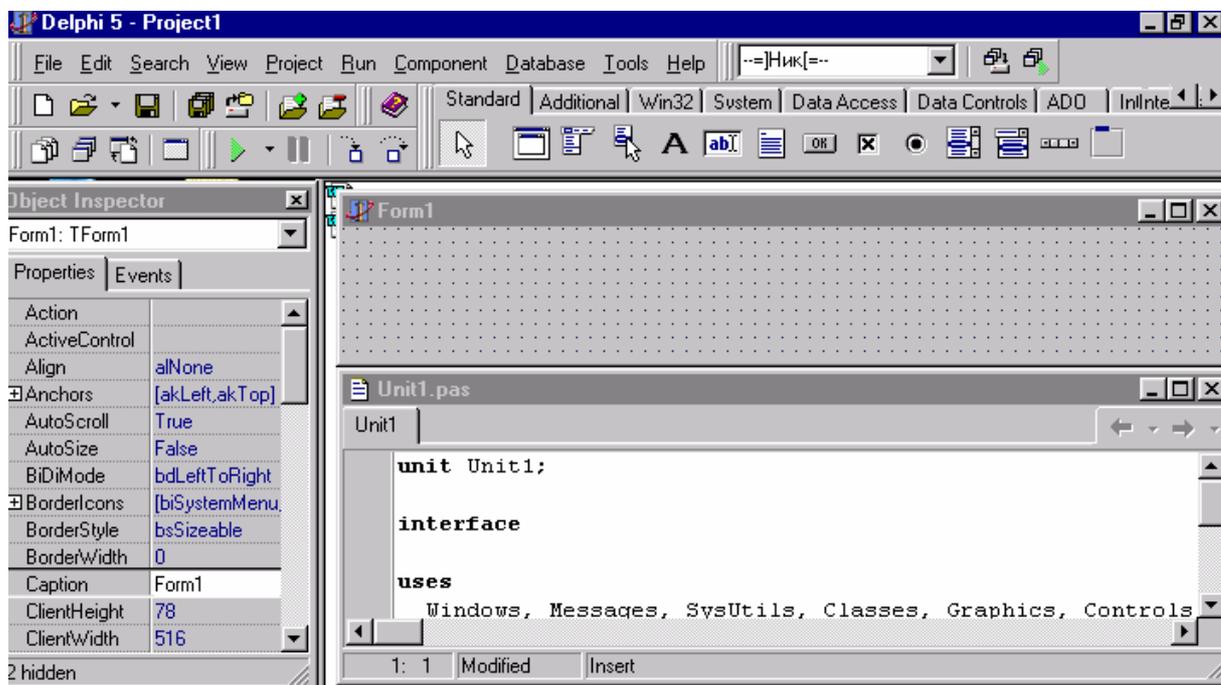


Рис. 1.2. Основные окна Delphi.

Главное (горизонтальное) окно и располагается в верхней части экрана. Оно содержит строку меню, панель инструментов и палитру компонентов. Палитра компонентов (рис. 1.3) представлена группой отдельных страниц, каждая из которых снабжена закладкой. При подведении курсора мыши к компоненту появляется подсказка - название компонента.

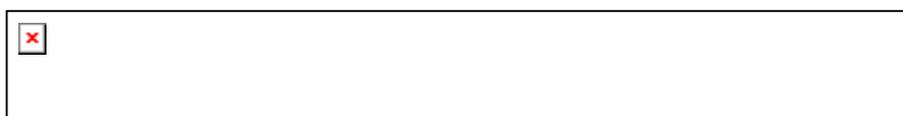


Рис. 1.3. Палитра визуальных компонент

Первоначально окно кода перекрыто окном формы, переключение между которыми осуществляется клавишей **F12**. Чтобы упорядочить окна так, как они показаны на рисунке, их размеры и местоположение можно изменить вручную. Расположение окон может быть произвольное, их расположение не влияет на работу Delphi.

## Задание 1.

- 1.1. Создайте папку на диске C: для сохранения файлов программы.
- 1.2. Запустите систему программирования Delphi.
- 1.3. Расположите окна Delphi согласно рис. 1.2, найдите каждое окно.

### Этап 2. Форма. Инспектор Объектов

**Форма** - окно будущей программы. Окно формы представляет собой проект окна Windows: имеет заголовок и системные кнопки. На форме размещаются компоненты, тем самым, формируя окно будущей программы.

Форма относится к классу TForm. Конкретное окно в программе является реализацией класса - объектом Form1. Имя объекта формируется путем задания название компонента (Form) и его порядкового номера (1). То есть, вторая форма получит имя Form2, далее – Form3 и т.д.

**Окно Инспектора Объектов** служит для задания свойств и событий объекта *Properties (Свойства)* и *Events (События)*.

Основными **свойствами** компонентов являются:

Name - имя переменной, задаются по умолчанию. Имя объекта можно изменить, используя только английские буквы и цифры.

**Запомните!** Свойство Name обязательно, это имя объекта, используемое в программе.

Font - шрифт, позволяет выбрать параметры шрифта.

Caption – текстовая надпись, заголовок.

Visible - видимость, принимает значение True и False.

COLOR – цвет объекта, выбирается при помощи меню.

Width - ширина объекта.

Height - высота объекта.

## Задание 2:

- 2.1. Для формы задайте свойства: имя - F1, размер - 500\*300, цвет - зеленый, шрифт - жирный курсив, заголовок (caption) - "Моя первая программа".
- 2.2. В окне кода программы посмотрите описание объекта формы.

### ЭТАП 3. Компоненты TLabel, TButton, TPanel

Для размещения компонента на форме нужно среди палитры компонентов щелкнуть мышью на данном компоненте, после щелкнуть на форме. Компонент можно удалить при помощи клавиши Delete.

**Компонент TLabel** - метка, служит для отображения текста на экране. Если цвет метки совпадает с цветом фона, то при сокращении надписи до нулевой длины, она может исчезнуть с экрана. В этом случае разыскать ее можно, перейдя в окно Инспектора Объектов, развернув список компонент и выбрав ее из этого списка.

**Компонент TButton** - командная кнопка. TButton позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы.

**Компонент TPanel** - контейнер общего назначения. Панель имеет свойства:

- BevelOuter (внешняя рамка) =
  - BevelInner (внутренняя рамка) =
- $$\left. \begin{array}{l} \text{bvRaised (выпуклая)} \\ \text{bvRaised (вдавленная),} \\ \text{bvNone (отсутствует).} \end{array} \right\}$$
- BevelWidth (ширина рамки).

**Запомните!** Определяя свойства компонента, его необходимо выделить.

#### Задание 3.

3.1. Разместите на форме TLabel. По умолчанию свойство name=Label1. Укажите надпись "Я учусь программировать на Delphi". Задайте жирный шрифт 14 размера. Выберите для метки, задаваемые в программе цвета: желтый, красный, голубой, выпишите названия цветов в тетрадь. Установите серый цвет метки.

3.2. Расположите ниже компонент TPanel. Задайте внешнюю и внутреннюю рамки, ширину рамки так, чтобы панель казалась вдавленной.

3.3. На панели разместите три командные кнопки с именами Button1, Button2, Button3 и надписями "Желтый", "Красный", "Голубой".

3.4. Откройте в окне Инспектора объектов список объектов, размещенных на форме. Проверьте наличие каждого компонента на форме.

3.5. В окне кода программы найдите созданное описание всех компонентов.

#### Этап 4. События. Реакция на события. Окно кода программы

**События** устанавливаются в окне Инспектора Объектов на странице Events (События). Наиболее часто используемым событием Windows является *OnClick* - событие щелчок мыши.

В нашей программе при нажатии мыши (событие *OnClick*) на командные кнопки будет соответственно изменяться цвет метки.

Для задания события *OnClick* нужно два раза щелкнуть мышью на компоненте. После выбора события автоматически открывается окно кода программы.

**Окно кода программы** предназначено для создания и редактирования текста программы на языке программирования. В системе программирования Delphi используется язык программирования Object Pascal.

После задания события в окне кода программы автоматически создается процедура обработки события. Заголовок процедуры формируется:

*Procedure* *имя* формы. *Имя объекта*Событие ();

**Procedure** TF1.Button1Click (Sender:TObject); {событие нажатие мыши на Button1}

**Begin**

**End;**

Между **Begin** **End** указывается **реакция на события** - перечень совершаемых действий.

При нажатии на командные кнопки Button1 ("желтый"), Button2 («красный»), Button3 («голубой») цвет метки должен изменяться соответственно.

В этом случае в процедурах описываются следующие действия:

<b>Procedure</b> TF1.Button1Click (..);	<b>Procedure</b> TF1.Button2Click (..);	<b>Procedure</b> TF1.Button3Click (..);
<b>Begin</b>	<b>Begin</b>	<b>Begin</b>
Label1.color:=clyellow;	Label1.color:=_____;	Label1.color:=_____;
<b>End;</b>	<b>End;</b>	<b>End;</b>

**Задание 4.** Создайте поочередно события и реакцию на события для командных кнопок. Названия цветов были выписаны на этапе 3.

## Этап 5. Отладка, запуск, сохранение программы

Сохранение программы осуществляется в два этапа:

- **File/Save as ...**- сохраняет модуль Pascal с расширением .pas. Имя файла задается английскими буквами.
- **File/Save project as ...**- сохраняет проект, основную программу, с расширением .dpr.

Имя модуля и проекта не должно совпадать. Поэтому удобно задавать имена файлов следующим образом:

- 1) имя модуля – lab1.pas;
- 2) имя проекта - P\_lab1.dpr.

При сохранении программы автоматически создаются несколько файлов:

- \*.dfm - файл формы;
- \*.exe - выполняемый файл, создается после запуска программы на исполнение.

**File/Open** – открытие файла программы. **Запомните!** Для открытия программы в Delphi выбирается файл проекта. При открытии файла проекта автоматически открываются файл модуля и формы.

**Run/Run** – запуск программы на исполнение. При запуске программы на исполнение программа компилируется – переводится в машинный код. Обнаруженные в процессе компиляции синтаксические ошибки выделяются красным цветом, *курсор помещается на символ, следующим за ошибкой*. В строке состояния (нижней части окна) ошибка поясняется.

**Run/ Program Reset** - прекращение работы программы.

### Задание 5:

- 5.1. Сохраните модуль и проект в папку, созданную на 1 этапе.
- 5.2. Запустите программу на исполнение, исправляя возникшие ошибки.
- 5.3. Прекратите выполнение программы. Закройте Delphi (главное окно).
- 5.6. Откройте Вашу папку и найдите созданные файлы (\*.pas, \*.dpr, \*.dfm,\*.exe).
- 5.7. Запустите выполняемый файл программы. Откройте файл проекта.

## Дополнительный этап: ОКНА СООБЩЕНИЯ

Для открытия окна сообщения существует команда:

*ShowMessage ('текст сообщения');*

Например, мы хотим вывести сообщение о том, кто разрабатывал данную программу.

Создайте командную кнопку Button4 - «Автор». Процедура обработки события примет вид:

**Procedure** TF1. Button4Click (Sender:TObject);

**Begin**

ShowMessage ( ' программа разработана Ивановым Сергеем');

**End;**

**Задание 6.** Создайте кнопку «Автор», событие и реакцию на данное событие.

### Упражнения

**1. Составьте программу, которая при выборе соответствующей командной кнопки, изменяет:**

*уровень начальный*

а) цвет **формы** на голубой, фиолетовый, серый;

*уровень средний*

б) текстовую надпись на "Язык программирования", "Система программирования", "Алгоритм";

в) размер формы на "крупный" (600\*800), "средний" (400\*600), "мелкий" (200\*400);

*уровень высокий*

г) вид панели "выпуклая", "вдавленная", "отсутствует";

д) видимость – невидимость текстовой надписи.

**Дополнительно. Создайте окно сообщения:**

а) об условии задачи;

б) о назначении компонентов при щелчке мыши на данном компоненте.

## Тест самоконтроля знаний

### Вариант 1

1. **Delphi** реализует интерфейс управления окнами:
  - a) SDI (Single Document Interface);
  - b) MDI (Multiple Document Interface);
  - c) RDI (Reformative DI).
2. **Окно формы** используется для:
  - a) проектирования окна программы;
  - b) выбора визуальных компонентов;
  - c) создания текста программы.
3. **В инспекторе объектов задаются:**
  - a) заголовок формы;
  - b) свойства и события;
  - c) программа и алгоритм.
4. **Страница Properties** используется:
  - a) для задания свойств объекта;
  - b) для задания событий объекта";
  - c) в меню главного окна.
5. **TButton** является:
  - a) командной кнопкой;
  - b) меткой;
  - c) свойством.
6. **Свойство Font** определяет:
  - a) параметры шрифта;
  - b) палитру фона;
  - c) характеристики формы.
7. **Свойства размера объекта:**
  - a) Width, Height;
  - b) Width, Top;
  - c) Left, Top.
8. **TButton** не имеет свойства:
  - a) Caption;
  - b) Color;
  - c) Left.
9. **Событие щелчок мыши:**
  - a) OnClick;
  - b) OnCreate;
  - c) OnClose.
10. **Верное описание события:**
  - a) Procedure Form1.Button1Click();
  - b) Procedure Form1.OnClick(Button1);
  - c) Procedure Button1.Click().
11. **Команда прекращения программы:**
  - a) Project/Compile;
  - b) Run/Run;
  - c) Run/Reset.
12. **Выполняемый файл \*.exe** создается:
  - a) при компиляции сохраненной программы;
  - b) при алгоритмизации программы;
  - c) во время сохранения программы.

### Вариант 2

1. **В Delphi** нет окна:
  - a) окна компиляции;
  - b) окна формы;
  - c) окна инспектора объектов.
2. **Окно кода программы** служит:
  - a) для выбора объектов;
  - b) для запуска программы;
  - c) для создания текста программы.
3. **Местоположение окон в Delphi:**
  - a) строго упорядоченно;
  - b) можно изменить вручную;
  - c) изменяется алгоритмом.
4. **TLabel** является:
  - a) командной кнопкой;
  - b) меткой;
  - c) свойством.
5. **Свойство name** для первой метки:
  - a) TLabel;
  - b) TLabel1;
  - c) Label1.
6. **Свойство Caption** определяет:
  - a) имя объекта;
  - b) заголовок объекта;
  - c) размер объекта.
7. **В свойствах кнопка [...]** позволяет:
  - a) не задавать значение свойства;
  - b) задать значение в окне диалога;
  - c) выбрать значение в меню.
8. **Реакция на событие описывается:**
  - a) в окне инспектора объектов;
  - b) в окне кода программы;
  - c) в окне событий.
9. **Заголовок процедуры события:**
  - a) Procedure Класс формы.  
Имя объекта Событие();
  - b) Procedure Имя объекта Событие  
(имя формы);
  - c) Procedure Событие. Имя кнопки().
10. **Задание красного цвета метке:**
  - a) Label1:=color(clred);
  - b) Label1.Color:=clred;
  - c) Color.clred:=Label1.
11. **При сохранении программы:**
  - a) создается один файл \*.pas;
  - b) появляются несколько файлов;
  - c) формируется один файл \*.exe.
12. **Для открытия программы в Delphi:**
  - a) выбирается файл проекта \*.dpr;
  - b) выполняется файл \*.exe;
  - c) используется файл модуля \*.pas.

## Лабораторная работа №2

### Программирование вычислительных задач

#### Вопросы теоретического контроля

1. Понятие алгоритма. Линейная алгоритмическая структура. Блок-схема.
2. Понятие языка программирования: оператор, ключевые слова, идентификатор.
3. Типы данных Object Pascal: integer, real, string.
4. Функции перевода числа в строку, строку в число.
5. Математические функции.
6. Оператор присваивания.
7. Структура программных единиц: проекта, модуля, подпрограмм.

#### Цель занятия

##### узнать:

- этапы решения вычислительной задачи;
- свойства компонента TEdit;
- этапы выполнения вычислительных задач.

##### научиться:

- рисовать линейные блок-схемы;
- составлять простейшие программы вычисления;
- формировать структуры проекта.

**Задача:** Зависимость спроса  $y$  (ед.) в сутки на 1000 жителей от цены  $x$  (руб.) выражается степенной функцией\*:  $y=3,4*x^{0,4}$ . Составить программу оценки спроса на товар.

#### Этапы выполнения задачи

##### Этап 1. Анализ требований программы

В задаче вычисления значения выражения используются данные:

- *входная переменная:  $x$*
- *выходная переменная:  $y$*

Пусть переменная  $x$  будет целочисленна. Результат  $y=3,4*x^{0,4}$  будет вещественным числом. Описание переменных примет вид:

- *$x$ : целое число;*
- *$y$ : вещественное число;*

**Задание 1.** Запишите задание и переменные программы в тетрадь.

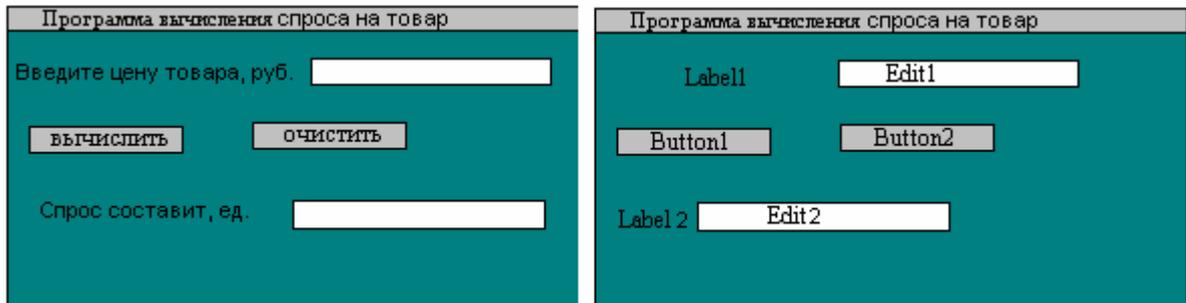
---

\* Эконометрика / под ред. И.И.Елисеевой. – М.: «Финансы и статистика», 2002 г. – С.70  
 Модель построена студентами 3 курса 270 гр. по данным для г. Благовещенска 2004 г.

## Этап 2. Окно будущей программы

*Для визуального проектирования* зарисовываются два окна. В первом окне определяется внешний облик окна, во втором окне указываются используемые элементы управления (в частности, компоненты).

Окно программы вычисления спроса удобно представлять:



В окне исполняемой программы «ответ» первоначально должен быть невидим. Поэтому для компонентов вывода Label2 и Edit2 устанавливается свойство видимости `visible=false`. Напомним, что для всех компонентов по умолчанию свойство `visible=true` (компоненты видны на форме).

Свойства объектов формы:

**Форма:** name=S1, caption=Вычисление спроса на товар, color ....

**Метка:** name=Label1, caption= Введите цену товара, руб., font (шрифт) = 12 пт

name=Label2, caption=Спрос составит, ед., font (шрифт) = 12 пт, visible=false

**Командная кнопка** name Button1, caption=вычислить

name Button2, caption=очистить

**Однострочное окно** name=Edit1, text □ (пустая строка)

name=Edit2, text □ (пустая строка), visible=false

**Задание 2.** Создайте в Delphi окно будущей программы.

## Этап 3. Описание событий

После визуального проектирования окна формы определяются события программы. В программе будут происходить события:

- ✓ щелчок мыши на командной кнопке "Вычислить".
- ✓ ДОПОЛНИТЕЛЬНО: щелчок мыши на "Очистить".

Для каждого события в Delphi создаются процедуры:

**Procedure** TS1.Button1Click(); **Procedure** TS1.Button2Click();

**Задание 3.** Создайте события программы.

## ЭТАП 4. Алгоритмизация и программирование

На этапе алгоритмизации составляется алгоритм каждого события.

Алгоритмы реализуются на языке программирования.

Процедура «Вычислить» примет вид:

**Procedure** TS1.Button1Click();

**Var**

x:Integer; y: real;

**begin**

x:=StrToInt(edit1.text);

y:=3.4\*exp(0,4\*ln(x));

label2.visible:=true;

edit2.visible:=true;

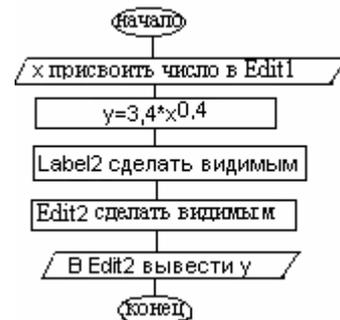
edit2.text:=FloatToStr(S);

**end;**

**Заголовок** подпрограммы

**Переменные**

x:целое; y:вещественное;



Алгоритм и процедура «Очистить» следующие:

<p><b>Заголовок</b> подпрограммы</p> <pre> graph TD     Start([Начало]) --&gt; Edit1[очистить Edit1]     Edit1 --&gt; Edit2[Очистить Edit2]     Edit2 --&gt; Label[Сделать невидимым Label2]     Label --&gt; Edit2[Сделать невидимым Edit2]     Edit2 --&gt; Cursor[Курсор поместить в Edit1]     Cursor --&gt; End([Конец])   </pre>	<p><b>Procedure</b> TS1.Button2Click();</p> <p><b>begin</b></p> <p>edit1.clear;</p> <p>edit2.clear;</p> <p>label2.visible:=false;</p> <p>edit2.visible:=false;</p> <p>edit1.SetFocus;</p> <p><b>end;</b></p>
--	--

### Дополнительный этап: Очистка компонентов формы

Очистка компонентов формы осуществляется следующим алгоритмом:

<pre> graph TD     Start([Начало]) --&gt; Edit1[очистить Edit1]     Edit1 --&gt; Edit2[Очистить Edit2]     Edit2 --&gt; Edit3[Очистить Edit3]     Edit3 --&gt; Edit3[Сделать невидимым Edit3]     Edit3 --&gt; Cursor[Курсор поместить в Edit1]     Cursor --&gt; End([Конец])   </pre>	<p><b>Procedure</b> TS1.Button2Click();</p> <p><b>begin</b></p> <p>edit1.clear; edit2.clear;</p> <p>edit3.clear;</p> <p>edit3.visible:=false;</p> <p>edit1.SetFocus;</p> <p><b>end;</b></p>
---	---

Для программирования используются методы компонента TEdit:

**Clear** -удаляет весь текст.

**SetFocus** - устанавливает фокус ввода (курсор).

**Задание 4.** Зарисуйте алгоритм событий программы. Реализуйте процедуры.

### ЭТАП 5. Тестирование и отладка программы

Напомним, тестирование – обнаружение ошибок, отладка – их исправление. Ошибки бывают синтаксические и структурные. Как уже говорилось, при запуске программы осуществляется компиляция, в процессе которой обнаруживаются синтаксические ошибки.

После запуска программы работающую программу проверяют на наличие структурных ошибок. Например, в нашем задании необходимо на бумаге задать значение  $x$ . При помощи калькулятора посчитать значение выражения. Ввести эти данные в программу и проверить вычисленный результат с рассчитанным Вами.

Пусть цена товара  $x=50$  руб., тогда спрос  $y=3,4*50^{0,4} = 16$  (ед.).

**Задание 5.** Приведите свой пример проверки результатов программы.

### ЭТАП 6. Исполняемая программа

Конечный программный продукт должен быть подготовлен к реализации – в нашем случае к демонстрации студенческой группе и преподавателю. Для этого:

- a) Необходимо сохранить программу. Например, сохраним модуль с именем *Spros.pas*. Файл проекта удобно формировать путем добавления начальной буквы *P* – *Pspros.dpr*
- b) Убедиться, что файл модуля прикреплен к проекту. Команды *Project/View source*, *Project/Add to project*, *Project/Remove from project*.
- c) Желательно подготовить сопроводительную информацию: ФИО разработчика, формулировка задачи и др. Например, сопровождение программы *лаб./работы №1*.

**Структура модуля** в окне кода программы примет вид:

```

unit Sproc; {имя модуля совпадает с именем файла Sproc.pas}
interface
{список модулей, с которыми устанавливается связь}
uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
type Ts1 = class(TForm) {объявления типа TS1 класса формы}
    Label1: TLabel; {перечень компонентов формы типа TS1}
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
procedure Button1Click(Sender: TObject); {объявления заглавия процедур}
private {переменные раздела private доступны в пределах данного модуля }
public {переменные раздела public доступны в пределах модуля и за его пределами}
    end; {завершение описания типа}
var s1: Ts1; x:Integer; y: real; {глобальные переменные модуля}
implementation {$R *.DFM}
procedure Ts1.Button1Click(Sender: TObject);
var x:Integer; y: real; {переменные доступны внутри процедуры – локальные переменные
begin подпрограммы}
x:=StrToInt(edit1.text);
y:=3.4*exp(0.4*ln(x));
label2.visible:=true; edit2.visible:=true;
edit2.text:=FloatToStr(S);
end; end.

```

Обратите внимание, что переменные можно описывать в разных разделах, в зависимости от требований программы.

Текст проекта (*Project/View source*) принимает вид:

```

program Psproc;
uses Forms,
    Sproc in 'Sproc.pas' {s1}; {подключили модуль Sproc и форму S1 файла 'Sproc.pas'}
{$R *.RES}
begin
    Application.Initialize;
    Application.CreateForm(Ts1, s1);
    Application.Run;
end.

```

## Тест самоконтроля знаний

### Вариант 1

- Этап алгоритмизация следует после:**
  - этапа программирования;
  - этапа тестирования;
  - этапа методологии решения.
- Компонент TEdit представляет:**
  - окно выбора события;
  - строку сообщения;
  - однострочное окно ввода-вывода.
- Функция FloatToStr переводит:**
  - целое число в строку;
  - строку в вещественное число;
  - вещественное число в строку.
- Переменные описывают в разделе:**
  - var;
  - begin;
  - type.
- Выберите верную команду:**
  - S:=FloatToInt(edit1.text);
  - S:=FloatToStr(edit1.text);
  - S:=StrToInt(edit1.text);
- Функция возведение  $x$  в квадрат:**
  - SQRT( $x$ );
  - SQR( $x$ );
  - не существует.
- Функция ( $r \bmod 2$ ) находит:**
  - модуль числа;
  - целую часть от деления;
  - остаток от деления.
- ABS((5-x)+sqr(x)) при  $x=9$  равно:**
  - 1;
  - 77;
  - 19.
- Этап отладки осуществляется:**
  - после анализа результатов тестирования;
  - перед тестированием программы;
  - на этапе алгоритмизации.
- Ошибки программы бывают:**
  - структурные и алгоритмические;
  - синтаксические и структурные;
  - синтаксические и орфографические.
- Структура программной единицы:**
  - заголовок, описание переменных, описание событий;
  - заголовок, начало и окончание программы;
  - заголовок, описания данных, раздел операторов.

### Вариант 2

- Данные входа-выхода определяют на этапе:**
  - программирования;
  - указания требований программы;
  - алгоритмизации.
- Компонент TEdit не имеет свойство:**
  - Text;
  - Color;
  - Caption.
- Функция StrToInt переводит:**
  - целое число в строку;
  - строку в вещественное число;
  - строку в целое число.
- Числовые типы данных:**
  - Integer, Real;
  - Float, Integer;
  - Char, Real.
- Выберите верную команду:**
  - edit1.text:=IntToStr(S);
  - edit1.text:=S(IntToStr);
  - edit1:=StrToInt(S.text);
- Функция вычисления модуля  $x$ :**
  - ABS( $x$ );
  - ABC( $x$ );
  - mod( $x$ ).
- Функция ( $r \div 3$ ) находит:**
  - корень кубический из  $r$ ;
  - целую часть от деления;
  - остаток от деления.
- Sqrt( $x+\text{sqr}(2*x)+10$ ) при  $x=3$  равно:**
  - 1123;
  - 7;
  - 4.
- При тестировании выполняется:**
  - на бумаге расчеты частных результатов программы;
  - запуск программы, всегда показывающей верный результат;
  - ровно одно вычисление результата работы программы.
- Компиляция не завершится при:**
  - структурных ошибках программы;
  - алгоритмических ошибках;
  - синтаксических ошибках.
- В окне кода программы пишется:**
  - проект;
  - модуль;
  - отдельная процедура.

### Упражнения

1. Производственная функция Кобба-Дугласа имеет вид:  $P=5*L^{0,4}*K^{0,6}$ , где  $P$ -объем продукции,  $L$ -затраты труда,  $K$ -величина капитала. Составить программу вычисления объема продукции.
2. Объем  $V$  выработки молока фермерского хозяйства описывается моделью:  $V_{t0(1997)} = 5,6*e^{3(t-t_0)}$ . Составить программу вычисления  $V$  в году  $t$ .
3. Спрос на получения кредита  $S$  (руб.) семьи в год для семьи со среднемесячный доход семьи на одного человека  $D$  (руб.) и количество детей в семье  $m$  описывается моделью:  $S = 3,4+5^m +4,3\text{Ln}D$ . Составить программу вычисления спроса.
4. Для вклада  $S$  руб. под  $n\%$  годовых рассчитать сумму через год.
5. Программа вычисления стоимости телефонного разговора. Запрашивает стоимость минуты и длительность разговора, а затем вычисляет стоимость разговора.
6. Для суммы  $S$  (руб.) делается скидка  $n\%$ . Рассчитать скидку в руб. и сумму оплаты.

## Лабораторная работа №3

### Программы с ветвлением. Компонент TCheckBox

#### Вопросы теоретического контроля

1. Понятие ветвления
2. Типы ветвления
3. Блок-схема полного и неполного ветвления
4. Синтаксис оператора ветвления.
5. Математические операции для записи условия
6. Назначение компонента TCheckBox.

#### Цель занятия

узнать:

- правила записи сложных условий неравенств типа  $0 < x < 10$  и др.;
- возможные состояния флажка TCheckBox, его свойства;
- понятие независимый переключатель.

научиться:

- программировать задачи нахождения значения функции на промежутках;
- составлять программы нахождения максимального элемента;
- записывать условия проверки состояния флажка.

#### **Задача 1. Программа вычисления значения функции на промежутках**

$$y = \begin{cases} x^3 + 3x + 4, & \text{если } x \in [0; 1], \\ (x^3 + 3x + 4)^2, & \text{если } x < 0, \\ -4, & \text{если } x > 1. \end{cases}$$

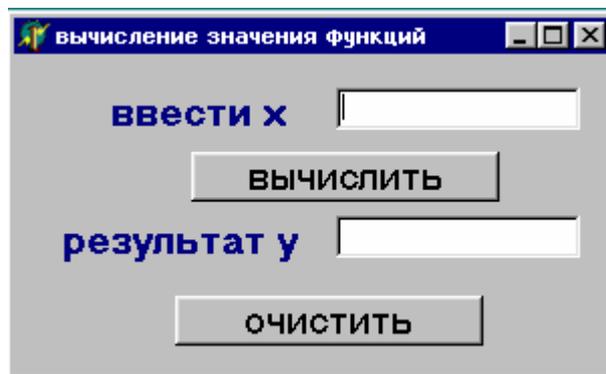


Рис. 3.1. Окно программы задачи 1.

#### Этапы выполнения задачи 1

##### **Этап 1. Определение требований программы**

В программе используются две переменные: входная  $x$ , выходная  $y$ .

**Задание 1.** В тетради запишите условие задачи 1. На оси  $Ox$  отметьте промежутки, на которых функция описывается различными уравнениями.

## Этап 2. Окно будущей программы

Для решения задачи форма формируется согласно рис.3.1. Для компонентов TForm, TEdit, TLabel задаются уже известные свойства.

Форма: name = Form1, Caption = Вычисление значения функции.

Метка: name = Label1, Caption = Введите X, Autosize=true.

Аналогично Label2.

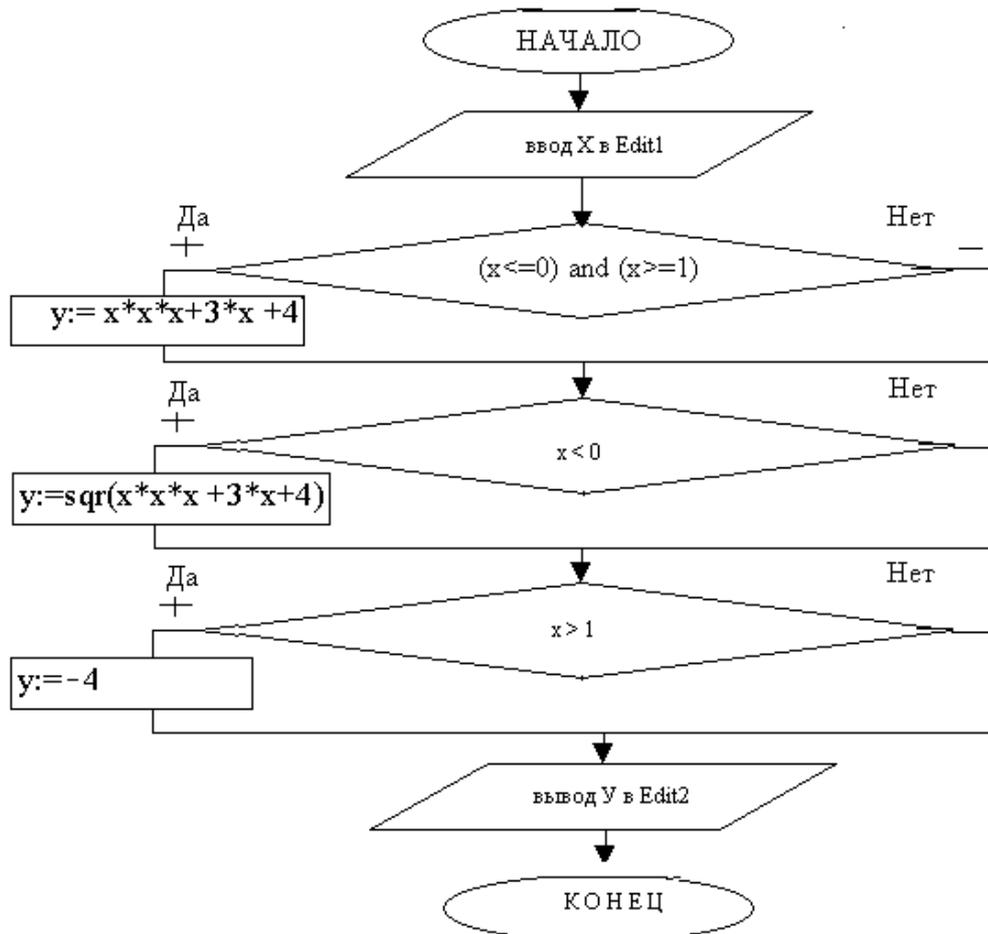
Однострочное окно: name = Edit1, Text .

Аналогично Edit2.

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

## Этап 3. Алгоритмизация

Алгоритм **вычисления** значения функции определяет промежуток, к которому принадлежит введенное значение  $x$ .



В алгоритме проверяются условия:

1 условие:  $x < 0$

2 условие:  $x > 1$

3 условие:  $x \in [0; 1]$  - сложное, состоит из двух частей  $(x \leq 0)$  and  $(x \geq 1)$ .

Блок-схема решения задачи содержит три неполных ветвления проверки каждого условия.

**Задание 3.** Зарисуйте блок-схему в тетрадь. Выделите три блока ветвления.

#### Этап 4. Программирование

В программе будут происходить два события:

1. Щелчок мыши по «Вычислить»: **Procedure** TForm1.Button1Click();
2. Дополнительно «Очистить»: **Procedure** TForm1.Button2Click();

Программирование алгоритма «Вычислить» записывается:

```
procedure TForm1.Button1Click(Sender: TObject);
  var x,y:real;
begin
  x:=strtofloat(edit1.text);
  if (x>=0) and (x<=1) then y:= x*x*x+3*x +4;
  if (x<0) then y:=sqr(x*x*x +3*x+4);
  if (x>0) then y:=-4;
  edit2.text:=floattostr(y);
end;
```

#### Задание 4.

- 4.1. Создайте событие "Вычислить" и запишите реакцию на него.
- 4.2. Дополнительно: Напишите реакцию на событие "Очистить" самостоятельно (см. лаб./работу №2).

#### Этап 5. Тестирование

Наличие структурных ошибок проверяется тремя частными вариантами.

- 1) Ввести значение  $x \in [0; 1]$ . Например, при  $x=0,5$   $y=x^3 + 3*x + 4=5,625$ .
- 2) Ввести значение  $x < 0$ . Например, при  $x=-2$   $y=(x^3 + 3*x + 4)^2=100$ .
- 3) Ввести значение  $x > 1$ . Например, при  $x=2$   $y=-4$ .

Расчетные результаты должны совпадать с результатами работающей программы. Несовпадение результатов показывает наличие структурных ошибок алгоритмизации или программирования.

**Задание 5.** Сохраните модуль и проект. Запустите программу на исполнение. Осуществите тестирование и отладку программы.

## Задача 2. Программа нахождения максимального значения

$$\text{Max} (5x^2 - 4y; e^y; \ln(x/2))$$

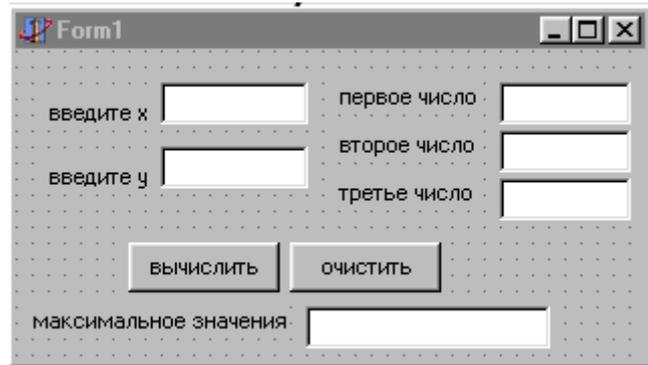


Рис. 3.2. Окно программы задания 2.

### Этапы выполнения задачи 2

#### Этап 1. Определение требований программы

Для нахождения  $\text{max} (5x^2 - 4y; e^y; \ln(x/2))$  рассчитываются три значения:

- ✓  $a1 = (5x^2 - 4y)$  - первое значение;
- ✓  $a2 = e^y$  - второе значение;
- ✓  $a3 = \ln(x/2)$  – третье значение,

среди которых находится максимальное.

Входными данными в задаче будут переменные  $x$ ,  $y$ .

Промежуточными данными являются переменные  $a1$ ,  $a2$ ,  $a3$ .

Выходом будет переменная  $\text{max}$ .

**Задание 1.** Опишите в тетради входящие и выходящие данные.

#### Этап 2. Окно формы

На рис. 3.3 показано расположение компонентов в окне формы.

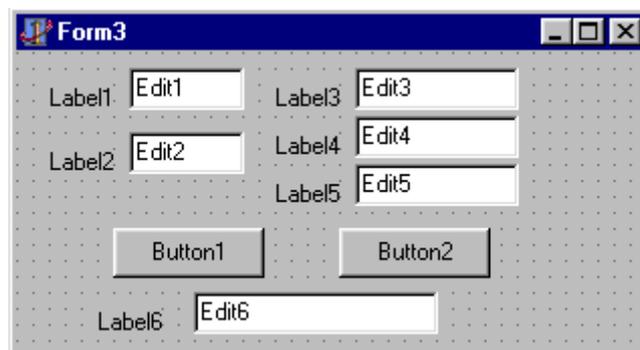


Рис. 3.3. Проектирование окна формы задачи 2.

В данной задаче не рассматривается задание свойств компонентов (подробно эти этапы рассматривалось в лабораторных работах №1, №2).

Усложним форму указанием видимости компонентов. Компоненты задания входных данных (Edit1, Edit2) сделаем видимыми при открытии формы: установим свойство `visible=true`. Компоненты отображения результатов (Edit3 – Edit6) первоначально невидимы: установим свойство `visible=false`.

**Задание 2.** В Delphi на оформите окно формы. Сохраните модуль и проект.

### Этап 3. События и реакция на события программы

В программе событие «Вычислить» реализуется процедурой:

```

procedure TForm1.Button1Click(Sender: TObject);
    var x,y,a1,a2,a3, max: real;
begin
    x:=strtofloat(edit1.text);
    y:= strtofloat(edit2.text);
    a1:=5*x*x-4*y*y; Edit3.text:=floattostr(a1);
    Edit3.visible:=true;
    a2:=exp(y); Edit4.text:=floattostr(a2);
    Edit4.visible:=true;
    a3:=ln(x/2); Edit5.text:=floattostr(a3);
    Edit5.visible:=true;
    if (a1>a2) and (a1>a3) then max:=a1;
    If (a2>a1) and (a2>a3) Then max:=a2;
    IF ( a3>a1) and (a3> a2) then max:=a3;
    Edit6.text:=floattostr(max);
    Edit6.visible:=true;
end;

```

Обратите внимание, что максимальное значение присваивается переменной `max`.

Типичная ошибка `a1:=max` *изменяет значение переменной a1* и нигде не записывает максимальное значение (`max` - неопределенно).

**Задание 3.** Создайте событие «Вычислить» и реакцию на него.

### ЭТАП 4. Тестирование

Для тестирования программы решим задачу  $\max(5x^2-4y; e^y; \ln(x/2))$  при  $x=2, y=3$ . Для этого рассчитываются значения `a1`, `a2`, `a3`, например, с

помощью программу "Калькулятор" в Пуск/Программы/Стандартные. Среди  $a_1$ ,  $a_2$ ,  $a_3$  находится максимальное значение.

**Задание 4.** Сравните полученные Вами значения с результатом программы.

**Задание 5.** Алгоритмизация: Зарисуйте в тетради блок-схему решения задачи.

**Дополнительное задание.** Создайте кнопку «Условие задачи» с выводом окна (рис. 3.3).

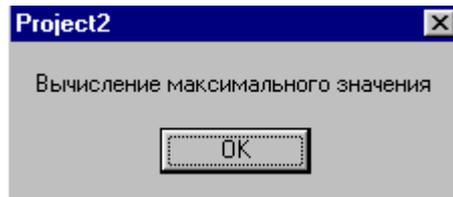


Рис. 3.3. Окно сообщение задачи 2.

### Задача 3. Программа "Флажок"

Вычисление квадрата  
и модуля числа

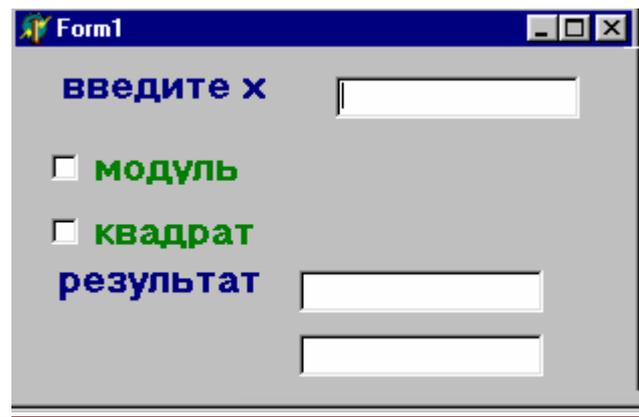


Рис. 3.4. Окно задачи 3.

### Этапы выполнения задачи 3

#### Этап 1. Определение требований программы

В рассматриваемой задаче:

- ✓ вход:  $x$  (целое число);
- ✓ выход:  $y_1$  (модуль числа  $x$ ) и  $y_2$  (квадрат числа  $x$ ).

**Задание 1.** В тетради опишите тип переменных  $x$ ,  $y_1$ ,  $y_2$ . Запишите функции нахождения модуля и квадрата числа.

#### Этап 2. Окно формы

В окне формы компоненты располагаются, как показано на рис. 3.5.

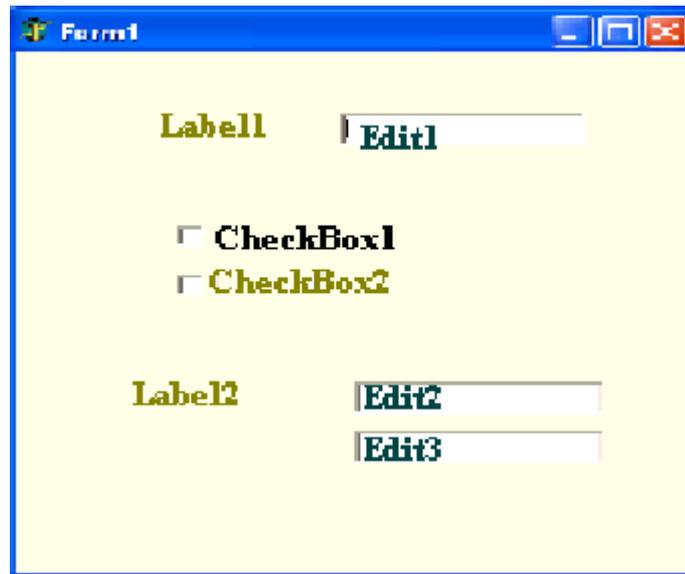


Рис. 3.5. Проектирование окна формы задачи 3.

Для компонентов TForm, TEdit, TLabel задаются уже известные Вам свойства.

Форма: name = Form1, Caption = Программа вычисления модуля и квадрата.

Метка: name = Label1, Caption = Введите X, Autosize = true.

Аналогично Label2, Label3.

Однострочное окно: name = Edit1, Text .

Аналогично Edit2, Edit3. Установите окна результата невидимыми.

### Флажок

Кроме известных компонентов на форму помещается новый компонент флажок. Флажок может принимать три состояния:

- ДА – cbChecked;
- НЕТ – cbUnchecked;
- НЕ ЗНАЮ – cbGrayed.

Состояние флажка задается свойством **State** = [cbChecked, cbUnchecked, cbGrayed].

Работая в Windows, Вы встречались с флажком и замечали, что чаще всего флажок принимает только два состояния ДА-НЕТ. Свойство **AllowGrayed**=[true, false] включает/выключает третье состояние - НЕ ЗНАЮ.

### Задание 2.

2.1. Запишите в тетради свойства компонента TCheckBox: AllowGrayed и State.

2.2. Оформите окно формы Delphi.

2.3. Изучите состояния флажков. Поэкспериментируйте: для CheckBox1 задайте включен (ДА), для CheckBox2 - НЕ ЗНАЮ.

2.4. Установите для флажков свойства:

Флажок: name = CheckBox1 (CheckBox2)

Caption = модуль (квадрат)

State - флажки не включены.

### Этап 3. События изменения состояния флажков

Событие щелчок мыши по кнопке «модуль» - CheckBox1Click, находит абсолютное значение числа X, записанное в Edit1, и выводит его в Edit2.

**Procedure** TForm1.CheckBox1Click (Sender: TObject);

var x,y1: integer;

**begin**

x:=StrToInt(edit1.text);

y1:=abs(x);

edit2.text:= IntToStr(y1);

if checkbox1.state=cbChecked

then edit2.visible=true

else edit2.visible=false;

**end;**

Обратите внимание на условие для флажка: *checkbox1.state=cbChecked*. Условие истинно при включенном флажке. В этом случае пользователю выводится результат. При ложном условии флажок выключен, следовательно, результат не отображается на экране.

**Задание 3.** Создайте события для двух флажков, реакции на эти события.

### Этап 4. Тестирование и отладка

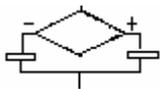
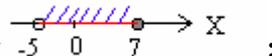
Введите отрицательное значение x, например, x=-2. Выберите флажок "модуль" и должны получить результат. Выключите флажок - результат исчезнет. Аналогично протестируйте флажок "квадрат".

Обратите внимание, Вы можете включить два флажка вместе, можете отключить все флажки, включать только один. Говорят, что состояние одного флажка не зависит от состояния других, поэтому **флажок - независимый переключатель**.

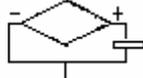
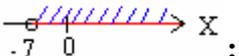
**Задание 4.** Выпишите в тетрадь понятия независимый переключатель. Создайте исполняемую программу. Не забудьте сохранить программу.

## Тест самоконтроля знаний

### Вариант 1

1. Ветвление бывает:
    - a) полное/неполное/частичное;
    - b) однозначное/многозначное;
    - c) полное/ неполное.
  2. Краткая форма оператора IF:
    - a) If <условие> else <оператор>;
    - b) If <условие> then <оператор1> else <оператор2>;
    - c) If <условие> then <оператор>.
  3. Для связи условий используются:
    - a) арифметических операций;
    - b) логических операций;
    - c) операций сравнения.
  4. Условие  $(x \geq 5)$  or  $(x < -10)$  при  $x = -20$  равно:
    - a) boolean;
    - b) false;
    - c) true.
- Блок-схема  :
5.
    - a) вариантного ветвления;
    - b) полного ветвления;
    - c) неполного ветвления.
  6. В Pascal условие принадлежности промежутку  :
    - a)  $(-5 < x \leq 7)$ ;
    - b)  $(x > -5)$  or  $(x \leq 7)$ ;
    - c)  $(x > -5)$  and  $(x \leq 7)$ .
  7. Свойство, включающее флажок:
    - a) Caption;
    - b) Color;
    - c) State.
  8. У TCheckBox отключают состояние:
    - a) Нет;
    - b) Не знаю;
    - c) Ложь.
  9. If  $(x < 4)$  and  $(x > -2)$  then  $y := x + 2$  else  $y := x - 5$ ; при  $x = 5$  вернет значение:
    - a)  $y = x - 5$ ;
    - b) 7;
    - c) 0.
  10. Независимый переключатель:
    - a) включает флажки только поочередно;
    - b) включает любое количество флажков;
    - c) включает один флажок при выключенных других.

### Вариант 2

1. В ветвление выполняется:
    - a) истинная цепочка действий;
    - b) только одна цепочка действий;
    - c) две цепочки действий.
  2. Полная форма оператора IF:
    - a) If <условие> else <оператор1> then <оператор2>;
    - b) If <условие> then <оператор1> else <оператор2>;
    - c) If <условие> <оператор1> and <оператор2>.
  3. Используются логические операции:
    - a) And/or/not;
    - b) And/or/then;
    - c) And/or/if.
  4. Условие  $(x \geq 5)$  and  $(x < -10)$ :
    - a) всегда ложно;
    - b) ложно/истинно;
    - c) всегда истинно.
- Блок-схема  :
5.
    - a) вариантного ветвления;
    - b) полного ветвления;
    - c) неполного ветвления.
  6. В Pascal условие принадлежности промежутку  :
    - a)  $\text{not}(x < -7)$ ;
    - b)  $\text{not}(x > -7)$ ;
    - c)  $\text{not}(x \leq -7)$ .
  7. Компонент TCheckBox - это:
    - a) флажок;
    - b) радиокнопка;
    - c) командное окно.
  8. TCheckBox по-другому называется:
    - a) автоматический переключатель;
    - b) независимый переключатель;
    - c) зависимый переключатель.
  9. If  $\text{not}(a \geq 4)$  then  $c := \text{sqr}(a)$  else  $c := a - 1$ ; при  $a = -5$  вернет значение:
    - a) 25;
    - b) -6;
    - c) нет решений.
  10. Событие щелчок мыши на компоненте CheckBox1 на форме F1:
    - a) Procedure TF1. CheckBox1Click ();
    - b) Procedure F1. CheckBox1OnClick ();
    - c) Procedure CheckBox1Click ().

Упражнения

1. Вычислить  $Y$  – размер ВВП страны в зависимости от:

$a$  – индекса расходов на сельское хозяйство к бюджету страны (от 0 до 1)

$b$  – индекса расходов на промышленность к бюджету страны (от 0 до 1)

$z$  – индекс расходов на образование к общему бюджету страны (от 0 до 1)

$x = a - b$  – разница индексов расходов на сельское хозяйство и промышленность

			<i>Уровень средний</i>
<b>A</b>	$y = \begin{cases} 100z^2, & \text{если } z \leq 0,3 \\ 120z^2 + 5 \ln z, & \text{если } 0,3 < z < 0,7 \\ 200z^3, & \text{если } z \geq 0,7 \end{cases}$	<b>E</b>	$y = \begin{cases} a^2 + b^2, & \text{если } a^2 + b^2 \leq 1, \\ (a + b)^2, & \text{если } a^2 + b^2 > 1 \text{ и } a \geq b, \\ 0,5, & \text{если } a^2 + b^2 > 1 \text{ и } a < b. \end{cases}$
<b>B</b>	$y = \begin{cases} x^2, & \text{если } -0,5 \leq x \leq 0,5, \\ 4, & \text{если } x > 0,5, \\ \sqrt{-8x}, & \text{если } x < -0,5. \end{cases}$	<b>F</b>	$y = \begin{cases} 1/4, & \text{если } a = b \\ 2(a^2 + b^2), & \text{если } a > b \\ \operatorname{tg}(a + b), & \text{если } a < b \end{cases}$
<b>C</b>	$y = \begin{cases} 0, & \text{если } x \geq 1, \\ x, & \text{если } 0 < x \leq 1, \\ x^4, & \text{в противном случае.} \end{cases}$	<b>G</b>	$y = \begin{cases} 0, & \text{если } x \leq 0, \\ x^2 - x, & \text{если } 0 < x \leq 1, \\ x^2 - \sin \pi z^2 - 1, & \text{в противном случае} \end{cases}$
<b>D</b>	$y = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 2, \\ \frac{1}{x^2 + 4x + 5}, & \text{в противном случае.} \end{cases}$		

*Уровень высокий.* Дополнить программу флажками. Задать три флажка, соответствующие каждому промежутку. При выводе результата включать соответствующий флажок.

2. Даны  $x, y, z$ . Составить программу нахождения максимального элемента

	<i>Уровень начальный</i>		<i>Уровень средний</i>
<b>A</b>	$\max(x + y + z, 3x, y - 4z)$	<b>F</b>	$\min(\sin(x^2 + y^2), \operatorname{tg}(z), x/y)$
<b>B</b>	$\min(x^2 + y^2,  z , xyz)$	<b>G</b>	$\max(5x^2 - 4y^2, e^y, \ln x/2 )$
<b>C</b>	$\max(x^2 + y^2 - z^2, xyz, (x - y)^4)$	<b>I</b>	$\max(\sin(x + y), \operatorname{tg}(10 - 2z), 6y^3)$
<b>D</b>	$\min( x + y , (z/x)^2, 45/x^3)$	<b>J</b>	$\max(\ln x - y , \operatorname{tg}(z), x/y)$
<b>E</b>	$\min(\sin(x + y), 5 - 6y, \operatorname{tg}(z))$	<b>K</b>	$\max(\sin(x - y), \operatorname{tg}^2(z), x/y)$

**Творческие задания**

1. Напишите программу расчета скидки на покупку. Если сумма покупки составляет более 500 р., то покупателю предоставляется скидка  $n\%$ .

2. Известна величина прожиточного минимума на человека в месяц. Напишите программу, которая для семьи  $n$  человек с доходом  $S$  включает флажок выше или ниже прожиточного минимума. Дополнительно: предусмотреть возможность равенства значений.

3. Некоторый товар можно приобрести в трех торговых точках. В каждом магазине известна цена товара  $S_1, S_2, S_3$  и предоставляемые скидки  $r_1, r_2, r_3$ . Найти минимальную цену с указанием магазина. Вывести реальную цену покупки в каждом магазине.

4. Стоимость печати стандартной визитки  $S$  рублей. При печати визитки на глянцевой бумаге стоимость увеличивается на 10%. Нанесение логотипа на визитку увеличивает стоимость печати на 5%. Печать более 100 экземпляров уменьшает стоимость на 8%. Написать программу расчета стоимости печати одной визитки. В программе выбор параметров осуществляется флажками «Глянцевая бумага», «Логотип», «Объем более 100 экз.».

## Лабораторная работа №4

### Программы выбора из списка. Класс TStringList

#### Вопросы теоретического контроля

1. Понятие и синтаксис записи оператора выбора.
2. Принцип работы ключа выбора.
3. Компоненты класса TStringList: TRadioGroup, TMemo, TListBox.
4. Индексация строк компонентов класса TStringList.
5. Методы класса TStringList: Clear, SetFocus, Add, Delete, LoadFromFile, SaveToFile и др.

#### Цель занятия

##### УЗНАТЬ:

- правила записи констант выбора оператора CASE;
- представление набора строк в TListBox и TRadioGroup;
- возможности выбора радиокнопок в TRadioGroup, понятие зависимый переключатель;
- различия в использовании флажка и радиокнопки;
- отличия компонентов TMemo и TListBox, TMemo и TEdit, TRadioGroup и TCheckBox, TListBox и TRadioGroup.

##### НАУЧИТЬСЯ:

- составлять программы с множественным выбором решения;
- использовать компоненты TRadioGroup и TListBox для выбора данных;
- выделять результат в списке через компоненты TRadioGroup и TListBox;
- выводить результат в TMemo.

#### **Задача 1:** Программа выбора прессы в TListBox по дню недели

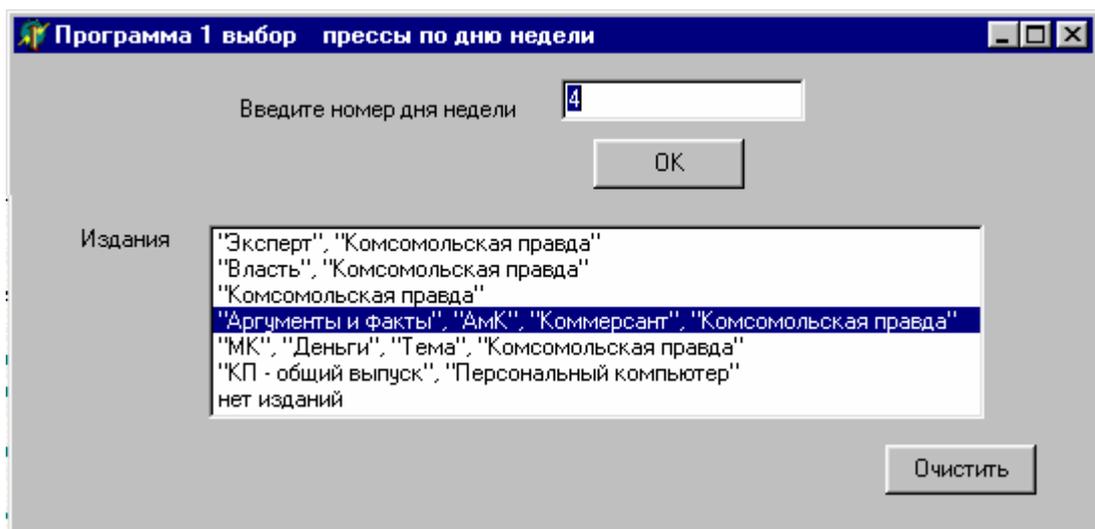


Рис.4.1. Пример окна программы «Периодические издания»-1.

## Этапы выполнения задачи 1

### Этап 1. Определение требований программы

В задаче 1:

✓ *входная переменная:*

n: целого типа (вводится в Edit1 - номер дня недели).

✓ *результат:*

действие - выделение строки в списке изданий TListBox.

**Задание 1.** В тетради опишите этап 1.

### Этап 2. Окно будущей программы

Окно формы представлено на рис. 4.1.

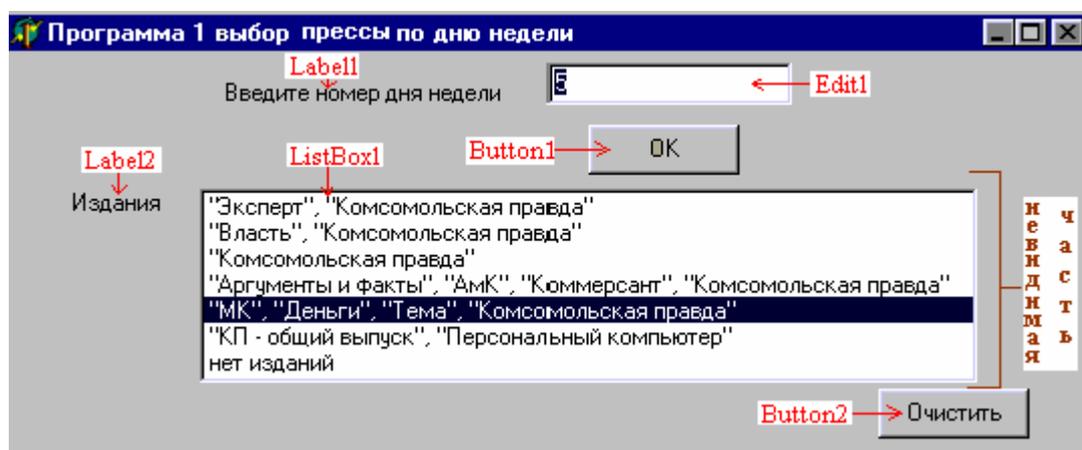


Рис. 4.2. Пример визуального проектирования окна программы 1.

В окне формы располагаются известные TEdit, TLabel, TButton. Зададим свойства ЭТИХ КОМПОНЕНТОВ.

Форма: name = F1, Caption = Программа 1 выбора прессы по дню недели.

Метки: name = Label1, Caption = Введите номер дня недели.

name = Label2, Caption = Издания в этот день.

Окно ввода: name = Edit1, Text □.

Командная кнопка: name = Button1, Caption = ОК.

Разместим на форме новый компонент с именем ListBox1 в виде прямоугольного окна. Размер компонента можно изменять при помощи мыши.

TListBox представляет список строк, для обращения к которому используется свойство .

Items (список строк) относится к классу TStrings и ему присущи все свойства и методы данного класса. Свойство Items имеет кнопку [...] для открытия окна списка строк (рис.4.3).

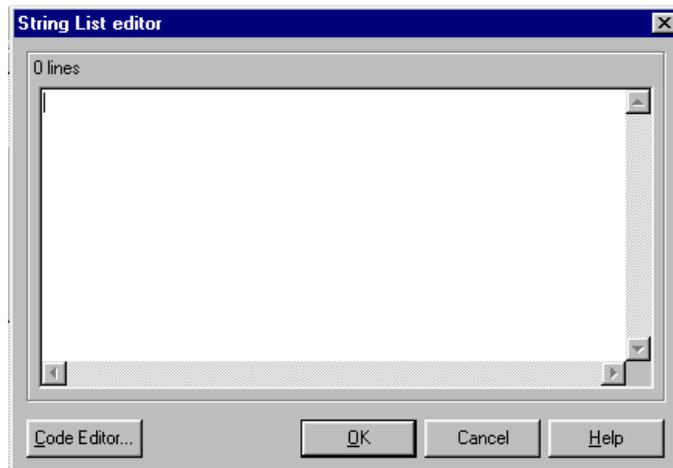


Рис. 4.3. Окно списка строк компонента TListBox.

В окне списка набираются строки, каждая из которых **индексируется, начиная с нуля.**

В нашей программе у компонента ListBox1 свойство *Items* равно:

```
«Эксперт», «КП»;
«Власть», «КП»;
«КП»;
«КП», «АиФ», «АмК», «Коммерсант»;
«КП», «МК», «Деньги», «Тема»;
«КП – общий выпуск», «Персональный компьютер»;
нет изданий
```

Для выделения строки в списке служит свойство *itemindex*:

- = -1 при отсутствии выделения;
- = 0, 1, 2, ... - номер выделенной строки.

Первоначально окно строк невидимо, поэтому `visible = false`.

## Задание 2.

- 2.1. В Delphi на форме расположите указанные выше компоненты.
- 2.2. Для ListBox1 выделите первую строку (какой номер она будет иметь?).
- 2.3. Сохраните модуль и проект.
- 2.4. Запишите в тетради свойства TListBox: *Items*, *ItemIndex*.

### Этап 3. Алгоритмизация программирование

Оператор выбора не является классической алгоритмической структурой, поэтому не имеет установленной блок-схемы. При реализации алгоритма будем использовать словесной способ его представления.

<p><i>Алгоритм события "ОК"</i></p> <p><b>переменная:</b> n: целого типа;</p> <p><b>начало</b></p> <p>ввод данных в Edit1 (номер недели) - переменная n;</p> <p><b>выбор</b> значения переменной n из списка:</p> <p>1 (понедельник): в ListBox1 выделить строку [0];</p> <p>2 (вторник): в ListBox1 выделить строку [1];</p> <p>3 (среда): в ListBox1 выделить строку [2];</p> <p>4 (четверг): в ListBox1 выделить строку [3];</p> <p>5 (пятница): в ListBox1 выделить строку [4];</p> <p>6 (суббота): в ListBox1 выделить строку [5];</p> <p>7 (воскресенье): в ListBox1 выделить строку [6];</p> <p><b>иначе</b> снять выделение (неверный день недели);</p> <p><b>конец выбора;</b></p> <p>сделать видимым ListBox;</p> <p><b>конец</b></p>	<pre> <b>Procedure</b> TF1.Button1Click (); <b>var</b> n: integer; <b>begin</b> n:=StrToInt(edit1.text); <b>case</b> n <b>of</b>   1:ListBox1.itemindex:=0;   2:ListBox1.itemindex:=1;   3:ListBox1.itemindex:=2;   4:ListBox1.itemindex:=3;   5:ListBox1.itemindex:=4;   6:ListBox1.itemindex:=5;   7:ListBox1.itemindex:=6;   <b>else</b> ListBox1.itemindex:= -1; <b>end;</b> ListBox1.visible:=true; <b>end;</b> </pre>
<p><b>Дополнительное событие</b></p>	
<p><i>Алгоритм события "Очистить"</i></p> <p><b>начало</b></p> <p>Очистить Edit1;</p> <p>Поместить курсор в Edit1;</p> <p>Снять выделение строк в ListBox1;</p> <p>Сделать невидимым ListBox1;</p> <p><b>Конец</b></p>	<pre> <b>Procedure</b> TF1.Button2Click(...); <b>begin</b> Edit1.clear; Edit1.setfocus; ListBox1.itemindex:=-1; ListBox1.visible:=false; <b>end;</b> </pre>

**Задание 3.** В тетради запишите алгоритм и программу события «ОК». Создайте событие «ОК». По-возможности, создайте событие «Очистить».

### Этап 4. Тестирование и отладка

После запуска программы на исполнение программа проверяется на наличие структурных ошибок. При вводе дня недели понедельник (1)

выделяется первая строка, воскресенья (7) - седьмая строка списка изданий и т.д. При вводе неверного дня недели (8 и др.) - выделение снимается.

**Особенность работы TListBox** изучается при работе с выполняемой программой. TListBox – набор строк, поэтому щелчок мыши на строке приводит к ее выделению. Строки нельзя редактировать и удалять.

**Задание 4.** Завершить построение исполняемой программы. Изучите особенности компонента TListBox.

**Задача 2.** Программа вывода дня выхода выбранного издания.

Компоненты TRadioGroup, TMemo.

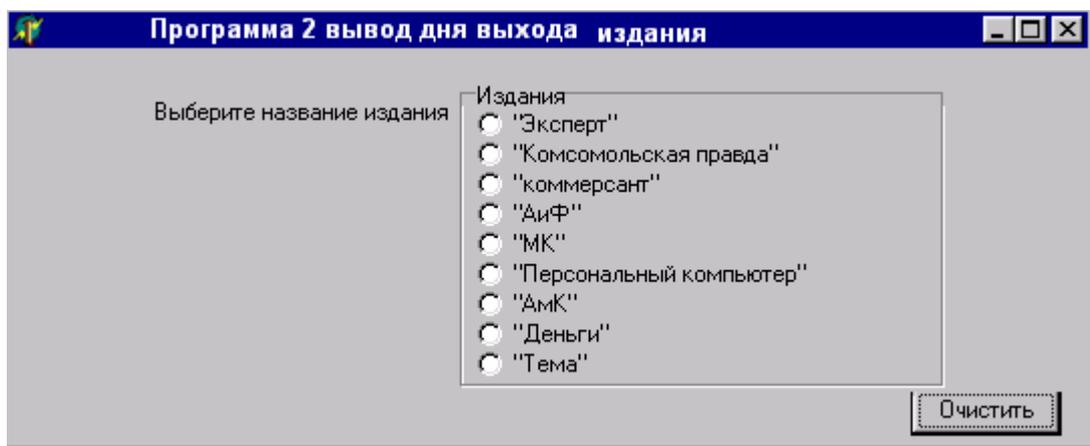


Рис.4.4. Пример окна программы 2 «Периодические издания».

## Этапы выполнения задачи 2

### Этап 1. Определение требований программы

*Входные данные* выбираются в TRadioGroup. Переменная *n*: *целого типа* содержит номер выделенной радиокнопки.

*Выходные данные* отражаются в TMemo. Переменная *d*: *строка* (*d:String*) определяет день выхода издания.

Дни выхода изданий:

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| ✓ «Эксперт» - понедельник, четверг; | ✓ «Персональный компьютер» - среда; |
| ✓ «КП» - вторник, пятница;          | ✓ «АмК» - понедельник;              |
| ✓ «Коммерсант» - суббота;           | ✓ «Деньги» - пятница;               |
| ✓ «АиФ» - понедельник;              | ✓ «Тема» - вторник.                 |
| ✓ «МК» - среда, пятница;            |                                     |

**Задание 1.** В тетради опишите переменные задачи 2 и их тип.

## Этап 2. Окно будущей программы

Для решения задачи расположим компоненты на окне формы (рис. 4.5).

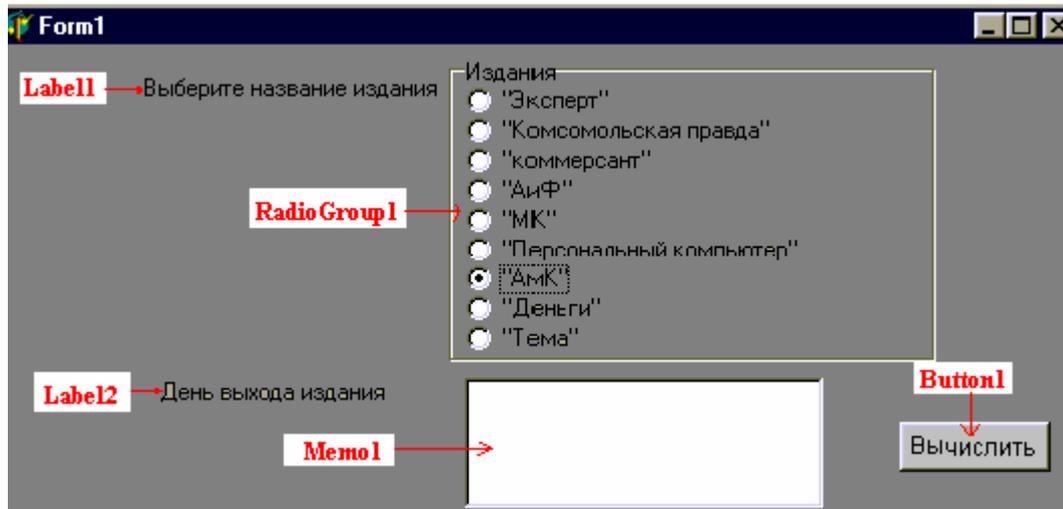


Рис. 4.5. Пример визуального проектирования окна программы 2.

В окне будущей программы располагаются компоненты TМемо, TLabel, TButton, TRadioGroup.

Форма: name = F2, Caption = Программа 2 вывод дня выхода издания

Метки: name = Label1, Caption = Выберите название издания

Name = Label2, Caption = День выхода издания

Командная кнопка: name = Button1, Caption = Вычислить

Список радиокнопок TRadioGroup аналогичен TListBox. Список радиокнопок задается свойством **Items:Tstrings**, каждая радиокнопка индексируется, начиная с нуля. Свойство Items содержит строки:

```
«Эксперт»
«КП»
«Коммерсант»
«АиФ»
«МК»
«Персональный компьютер»
«АМК»
«Деньги»
«Тема»
```

Многострочное окно ввода-вывода TМемо представляет небольшой текстовый редактор. В окне TМемо можно выводит, вводить, редактировать текст. В нашей программе Мемо1 служит для вывода результата.

Для обращения к строкам служит свойство `Lines:Tstrings`, при обращении к которому открывается окно строк. Окно вывода первоначально невидимо, т.е. `Visible=false`.

Свойства `RadioGroup1.Items` и `Memo1.Lines` относятся к классу `TStrings`.

**Задание 2.** Сформируйте окно формы. В тетради запишите назначение `TRadioGroup` (свойства `ItemIndex`, `Items`) и `TMemo` (свойства `Lines`).

### Этап 3. Алгоритмизация и программирование

<p><i>Алгоритм события «Вычислить»</i></p> <p><b>переменные</b> n: целого типа; d: строка;</p> <p><b>начало</b> ввод данных n = номер выделенной радиокнопки;</p> <p><b>выбор</b> значения переменной n из списка:  [0]: d:= 'понедельник, четверг';  [1]: d:= 'вторник, пятница';  [2]: d:= 'суббота';  [3]: d:= 'понедельник';  [4]: d:= 'среда, пятница';  [5]: d:= 'среда';  [6]: d:= 'понедельник';  [7]: d:= 'пятница';  [8]: d:= 'вторник';</p> <p><b>конец выбора;</b> сделать видимым <code>Memo1</code>;</p> <p><b>конец</b></p>	<pre> <b>Procedure</b> TF2.Button1Click (...); <b>var</b> n: integer; d:string; <b>begin</b> n:=RadioGroup1.itemindex; <b>case</b> n of 0: d:= 'понедельник, четверг'; 1: d:= 'вторник, пятница'; 2: d:= 'суббота'; 3: d:= 'понедельник'; 4: d:= 'среда, пятница'; 5: d:= 'среда'; 6: d:= 'понедельник'; 7: d:= 'пятница'; 8: d:= 'вторник'; <b>end;</b> Memo1.visible:=true; <b>end;</b> </pre>
<p><b>Дополнительное событие</b></p>	
<p><i>Алгоритм для события "Очистить"</i></p> <p><b>начало</b> снять выделение радиокнопок в <code>RadioGroup1</code>; очистить <code>Memo1</code>; сделать невидимым <code>Memo1</code>;</p> <p><b>конец</b></p>	<pre> <b>procedure</b> TF2.Button1Click(...); <b>begin</b> RadioGroup1.itemindex:=-1; Memo1.Lines.clear; Memo1.visible:=false; <b>end;</b> </pre>

**Задание 3.** В тетради запишите алгоритм и программу события "Вычислить". Создайте данное событие и реакцию на него. По-возможности создайте событие «Очистить».

## Этап 4. Тестирование и отладка

Правильно написанная программа должна использовать для ввода – Radiogroup1, для вывода – Memo1.

Например, при выборе радиокнопки «КП» в окне вывода должны отражаться дни «вторник, пятница». Для издания «АиФ» результатом является «понедельник».

**Особенности работы TRadioGroup** удобно изучать после запуска программы. Если первоначально включить радиокнопку третьей строке, а далее включить радиокнопку пятой строки, то при этом третья радиокнопка автоматически выключится. *В данный момент времени может быть включена максимально одна радиокнопка.* Радиокнопка - "зависимый переключатель", состояние одной радиокнопки влияет на состояние других.

По-существу, радиокнопки и флажки выполняют одинаковые функции: выбор данных (параметров). Однако, принято использовать флажок как независимый, а радиокнопку как зависимый переключатели. Если одна радиокнопка включена, то остальные выключены. При включение другой, предыдущая автоматически отключается.

**Особенности работы TМемо:** 1) щелчок мыши внутри компонента (текстовой области) приводит к появлению курсора; 2) текст TМемо можно редактировать: выделять фрагмент, копировать (CTRL+INSERT), вставлять (SHIFT+INSERT), удалять и т.д.

### Задание 5.

5.1. Запустите программу на исполнение. Изучите особенности работы TRadiogroup. Запишите в тетрадь понятие «зависимый переключатель».

5.2. В программе Word выберите Формат/Шрифт и Вид/Масштаб. Объясните, почему для задания эффектов шрифта используются флажки, а для указания масштаба - радиокнопки.

5.3. Изучите особенности TМемо и запишите в тетрадь.

5.4. Перечислите различия между TМемо и TListBox, TRadioGroup и TCheckBox, TМемо и TEdit, TListBox и TRadioGroup.

### Тест самоконтроля знаний

- |  |   |
|--|---|
| <p><b>1. Оператор выбора позволяет:</b></p> <p>a) выделить строку текста;</p> <p>b) выбрать одну цепочку действий;</p> <p>c) выбрать две цепочки действий;</p> <p><b>2. В операторе выборе между CASE и OF:</b></p> <p>a) проверка условия;</p> <p>b) ключ выбора;</p> <p>c) операция выбора.</p> <p><b>3. В операторе выбора промежутки значений от 1 до 10 записывается:</b></p> <p>a) 1...10;</p> <p>b) 1-10;</p> <p>c) 1..10.</p> <p><b>4. Класс TStringList – это:</b></p> <p>a) компоненты TListBox, TMemo и др.;</p> <p>b) визуальный объект - текстовый редактор;</p> <p>c) абстрактный класс работы со строками.</p> <p><b>5. Метод ДОБАВИТЬ класса TStringList:</b></p> <p>a) Paste(&lt;строка&gt;);</p> <p>b) Add (&lt;строка&gt;);</p> <p>c) Insert (&lt;строка&gt;).</p> <p><b>6. Компонент Tmemo является:</b></p> <p>a) панелью переключателей;</p> <p>b) многострочным окном ввода-вывода;</p> <p>c) окном выбора строк.</p> <p><b>7. Вывод переменной y:Real в Memo1:</b></p> <p>a) Memo1.Lines.Add('y');</p> <p>b) Memo1.Caption:=FloatToStr(y);</p> <p>c) Memo1.Lines.Add(FloatToStr(y)).</p> <p><b>8. Компонент TListBox является:</b></p> <p>a) панелью переключателей;</p> <p>b) многострочным окном ввода-вывода;</p> <p>c) списком строк.</p> <p><b>9. Выделение строки в ListBox1:</b></p> <p>a) ListBox1.ItemIndex(0);</p> <p>b) ListBox1.Items:=1;</p> <p>c) ListBox1.ItemIndex:=0.</p> <p><b>10. Свойство список строк TRadioGroup:</b></p> <p>a) Lines;</p> <p>b) ItemIndex;</p> <p>c) Items.</p> <p><b>11. Радиокнопка называется:</b></p> <p>a) автоматический переключатель;</p> <p>b) независимый переключатель;</p> <p>c) зависимый переключатель.</p> <p><b>12. Количество строк в RadioGroup1:</b></p> <p>a) n:=RadioGroup1.Items.Count;</p> <p>b) n:=RadioGroup1.Count;</p> <p>c) RadioGroup1.ItemIndex.Count(n).</p> | <p><b>1. Ключевые словами оператора выбора:</b></p> <p>a) Case then else end;</p> <p>b) Case of else end;</p> <p>c) If of else end;</p> <p><b>2. Внутренняя часть оператора выбора записывается:</b></p> <p>a) ключ выбора = оператор;</p> <p>b) константа выбора: оператор;</p> <p>c) оператор = n(индекс).</p> <p><b>3. В операторе выбора перечень значений [1,4,5,8] записывается</b></p> <p>a) 1,4,5,8;</p> <p>b) 1:4:5:8;</p> <p>c) 1;4;5;8.</p> <p><b>4. TMemo имеет свойство класса TStringList:</b></p> <p>a) String;</p> <p>b) Items;</p> <p>c) Lines.</p> <p><b>5. Свойство COUNT класса TStringList есть:</b></p> <p>a) получение всего текста списка;</p> <p>b) обращение к строке;</p> <p>c) количество строк в наборе.</p> <p><b>6. Щелчок мыши на Tmemo приводит:</b></p> <p>a) к появлению курсора;</p> <p>b) к выделению строки;</p> <p>c) к выбору радиокнопки.</p> <p><b>7. Команда очистки компонента Memo1:</b></p> <p>a) Memo1.Delete;</p> <p>b) Memo1.Lines.Clear;</p> <p>c) Memo1.Clear;</p> <p><b>8. Свойство список строк TListBox:</b></p> <p>a) Lines;</p> <p>b) ItemIndex;</p> <p>c) Items.</p> <p><b>9. Запись строки РЕЗУЛЬТАТ в ListBox1:</b></p> <p>a) ListBox1.ItemIndex:=FloatToStr('результат');</p> <p>b) ListBox1.ItemIndex(результат);</p> <p>c) ListBox1.Items.Add('результат').</p> <p><b>10. Отключение радиокнопок RadioGroup1:</b></p> <p>a) RadioGroup1.ItemIndex:=false;</p> <p>b) RadioGroup1.ItemIndex:=-1;</p> <p>c) RadioGroup1.Items.Clear.</p> <p><b>11. Щелчок мыши на RadioGroup1:</b></p> <p>a) Procedure TForm1. RadioGroup1Click ();</p> <p>b) Procedure TForm1. Button1Click ();</p> <p>c) Procedure TRadioGroup1 OnClick ();</p> <p><b>12. Команда скрывающее RadioGroup1:</b></p> <p>a) RadioGroup1.ItemIndex.Clear;</p> <p>b) RadioGroup1.Visible:=false;</p> <p>c) RadioGroup1.Items.visible:=false.</p> |
|--|---|

## Упражнения

1. Составить программу, в которой **вывод данных** отображается включением радиокнопки:

### *Уровень начальный*

а) Человек в течение своей жизни проходит несколько возрастных этапов (младенец, ребенок, подросток, ...). Требуется по введенному количеству лет определить возрастной промежуток для данного возраста.

б) В пятиэтажном доме на первом этаже 3 квартиры, на следующих этажах по четыре квартиры. Нумерация квартир с 1 по 19. Требуется по номеру квартиры указать номер этажа, на котором она расположена.

в) По введенному номеру месяца выделить номер квартала.

### *Уровень средний*

г) По введенному количеству лет вывести возможные виды кредита:

до 1 года – экспресс-кредитование;

до 5 лет – кредит на неотложные нужды;

до 15 лет – ипотечное кредитование.

Для предлагаемого кредита в ТМето выводятся условия его получения.

### *Уровень высокий*

д) Средний ежеквартальный расход электроэнергии -  $P$  кВт/час. При этом:

в 1 квартале расход на 15% выше среднего,

во 2 и 4 кварталах - на 5% выше среднего,

в 3 квартале - на 8% ниже среднего.

Вводится номер квартала. Выделяется корректирующий сезонный процент и вычисляется расход электроэнергии в данном квартале.

2. Составить программу, в которой **ввод данных** осуществляется выделением строки списка.

### *Уровень начальный*

а) Для выбранного дня недели вывести часы приема врача:

▪ понедельник 8-00 – 10-00

▪ четверг 8-00 – 10-00

▪ вторник 11-00 – 13-00

▪ пятница 11-00 – 13-00

▪ среда 9-00 - 11-00

▪ суббота 9-00 – 10-00

- воскресенье выходной

б) Программа – киноафиша: на экране список фильмов, при нажатии на одно из названий, появляется время сеансов.

в) Для выбранного автомобиля вывести данные мощности двигателя.

*Уровень средний*

г) Оклад служащего составляет  $X$  рублей. Для выбранного района (центральный, дальневосточный, северный) вычисляется з/плата. Выводится процентная надбавка (0%, 50%, 100%) и зарплата за месяц.

д) Напишите программу расчета стоимости телефонного разговора. Известно, что тариф днем, вечером и на выходных разный.

День – 100%

Вечер – скидка 20%

Ночь – скидка 50%.

Пользователь вводит количество минут и стоимость 1 минуты разговора, выбирает в списке строк категорию времени, получает результат - стоимость разговора. Результат вывести в TListBox.

*Уровень высокий*

е) Для суммы  $S$  в списке радиокнопок выбирается вид вклада («Пенсионный» - 17% годовых, «До востребования» - 3% годовых, «Молодежный» - 14% годовых и др.). Указать сумму вклада через один, два, три года с учетом начисляемых процентов.

ж) Составьте программу стоимости услуг парикмахерской. Пользователь в предложенном списке строк выбирает вид услуги, после нажатия «ОК» высвечивается стоимость услуги. Предусмотреть флажок «Скидка», для утренних клиентов 5%.

## Лабораторная работа №5

### Программы циклической структуры

#### Вопросы теоретического контроля

1. Понятие цикла.
2. Виды циклических операторов.
3. Цикл с предусловием: принцип работы, блок-схема, оператор.
4. Цикл с постусловием: принцип работы, блок-схема, оператор.
5. Цикл с параметром: принцип работы, блок-схема, оператор, два способа записи.
6. Два различия использования циклов с пост - и предусловиями.

#### Цель занятия

##### УЗНАТЬ:

- переменные, используемые в вычислительных циклических задачах;
- начальные значения суммы ряда;
- два различия между циклами с пост- и предусловиями;
- особенность цикла с параметром.

##### НАУЧИТЬСЯ:

- вычислять сумму и среднее значение числовых рядов;

**Задача:** Программа вычисления числового ряда. Стоимость производственной базы (ПБ) в 20003 г. -  $V$  тыс. руб. Ежегодный темп прироста  $n\%$ . Рассчитать стоимость производственной базы на 5 лет вперед и среднее значение ряда.

Пример программы 1

Введите стоимость ПБ ( $V$ )

Введите темп прироста ( $n$ )

Постусловие      Предусловие      Параметр

элементы ряда

Постусловие      Предусловие      Параметр

средняя стоимость за 5 лет

Рис. 5.1. Пример окна программы вычисления числового ряда.

## Этапы выполнения задачи

### Этап 1. Определение требований программы

#### Входные данные

$v$ : вещественного типа – элемент ряда (стоимость ПБ).

$n$ : целого типа – темп пророста.

#### Промежуточные переменные

$i$ : целого типа - параметр цикла.

#### Результат

$S$ : вещественного типа - сумма ряда.

$c$ : вещественного типа – среднее значение ряда.

#### ВЫЧИСЛЕНИЕ ЭЛЕМЕНТА РЯДА.

Стоимость производственной базы  $V$  в следующем году будет рассчитываться:  $V=V*(1+n/100)$ , где

$1+n/100$  – темп роста,

$V$ (стоимость ПБ в новом году) =  $V$ (стоимости ПБ в прошлом году)\* темп роста.

Пусть  $V_{1999}=3$  тыс. руб. и  $n=5\%$ . Темп роста составит  $1+5/100 = 1,05$ .

$V_{2000} = V_{1999} * 1,05$ .

**Задание 1.** Опишите в тетради переменные задачи, их тип и назначение.

Вспомните определение темпа роста и прироста.

### Этап 2. Окно будущей программы

Окно формы программы 1 создается согласно рис. 5.1. и 5.2.

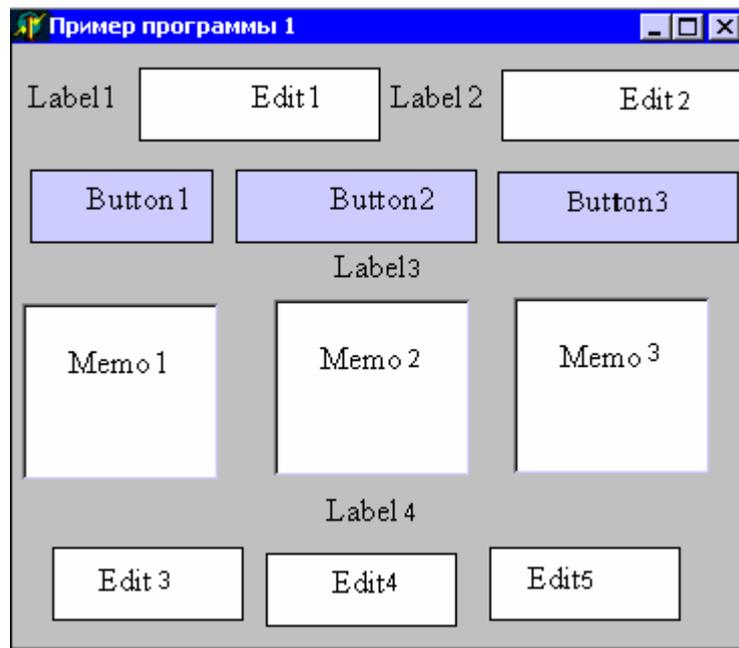


Рис. 5.2. Пример визуального проектирования окна программы 1.

**Задание 2.** В Delphi на форме расположите компоненты программы 1.

### Этап 3. Алгоритмизация и программирование

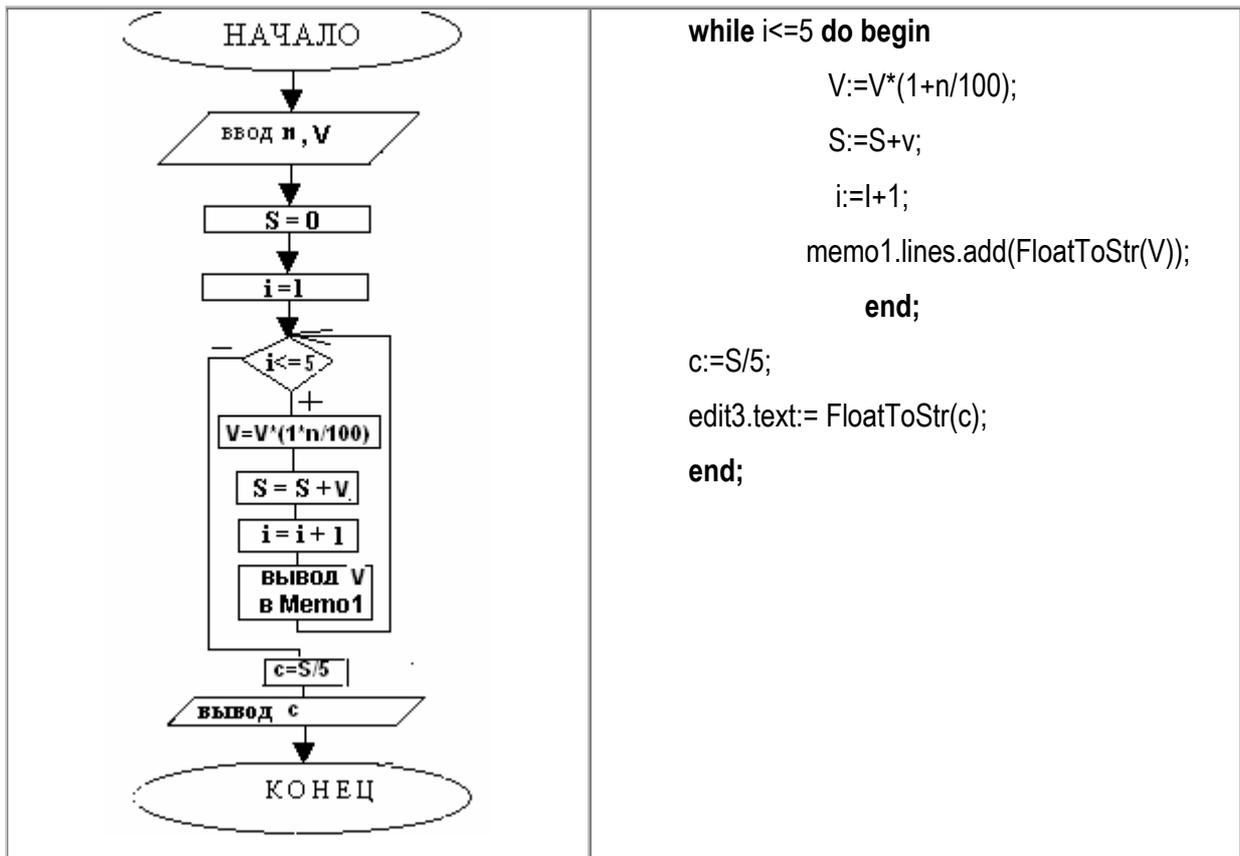
Алгоритм вычисления числового ряда содержит действия:

- 1) ввод данных;
- 2) задание первоначальных значений переменных;
- 3) тело цикла:
  - вычисление элемента ряда;
  - прибавление его к сумме;
  - изменение параметра цикла;
- 4) вывод результата

Запомните! Первоначальное значение суммы  $S=0$ , произведения  $P=1$ .

Данные значения не влияют на дальнейшие вычисления.

<p><i>Алгоритм цикла с предусловием</i></p>	<pre> <b>Procedure</b> TForm1.Button1Click(); <b>var</b> n,i:Integer;         V,c,S:real; <b>begin</b> V:=StrToInt (edit1.text); n:=StrToInt (edit2.text); S:=0 ; i:=1; </pre>
---	--



Циклы с пред- и постусловием имеют отличия:

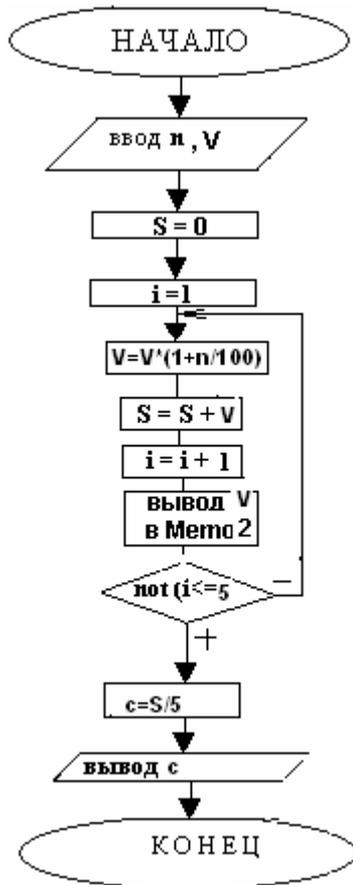
Цикл с предусловием:

- условие проверяется в начале цикла;
- тело цикла выполняется в случае истинности условия.

Цикл с постусловием:

- условие проверяется в конце цикла;
- тело цикла выполняется в случае ложности условия.

Алгоритм цикла с постусловием



```
Procedure TForm1.Button2Click();
```

```
var n,i:Integer;
```

```
V,c,S:real;
```

```
begin
```

```
V:=StrToInt(edit1.text);
```

```
n:=StrToInt(edit2.text);
```

```
S:=0;
```

```
i:=1;
```

```
Repeat
```

```
V:=V*(1+n/100);
```

```
S:=S+v;
```

```
i:=i+1;
```

```
memo2.lines.add(FloatToStr(v));
```

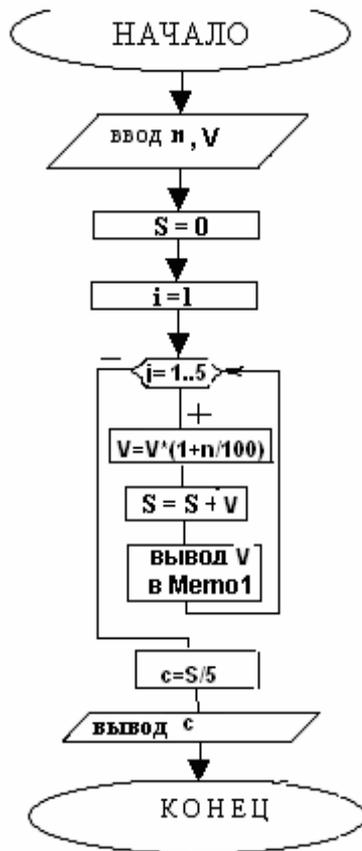
```
Until not (i<=5);
```

```
C:=S/5;
```

```
Edit4.text:= FloatToStr(c);
```

```
end;
```

Цикл с параметром не является классической алгоритмической структурой, поэтому для него нет определенной блок-схемы.



Цикл с параметром имеет два отличия:

- параметр цикла изменяется строго на 1;
- изменение параметра цикла происходит автоматически.

**Procedure TForm1.Button3Click();**

**var**

n,i:Integer;

V,c,S:real;

**begin**

V:=StrToInt (edit1.text);

n:=StrToInt (edit2.text);

S:=0; i:=1;

**For i:=1 to 5 do begin**

V:=V\*(1+n/100);

S:=S+V;

memo3.lines.add(FloatToStr(V));

{ i:=i+1; - отсутствует, параметр цикла изменяется автоматически}

**end;**

c:=S/5;

edit5.text:= FloatToStr(c);

**end;**

**For i:=5 downto 1 do begin**

V:=V\*(1+n/100);

S:=S+V;

memo3.lines.add(FloatToStr(V));

**end;**

**Задание 3.**

- 3.1. Зарисуйте блок-схемы и программы трех видов циклов в тетрадь.
- 3.2. Создайте события программы и реакцию на него.
- 3.3. Выпишите отличия циклов с пред- и постусловиями, особенность цикла с параметром.
- 3.4. Для цикла с параметром реализуйте один из двух способов записи.

**Этап 4. Тестирование и отладка**

Задача 1 решается тремя способами с использованием различных видов циклов. При этом результаты всех вариантов решения должны совпадать.

При тестировании задаются входные данные и рассчитывается результат.

Пусть  $V_{1999}=30$  тыс. руб.,  $n=7\%$ .

$V_{2000}=\underline{\hspace{2cm}}$

$V_{2001}=\underline{\hspace{2cm}}$

...

$V_{2004}=\underline{\hspace{2cm}}$

Сумма за 5 лет равна  $S=\underline{\hspace{2cm}}$

Средняя стоимость ПБ  $c=\underline{\hspace{2cm}}$

**Задание 4.**

- 4.1. Рассчитайте на бумаге результат задачи при  $V=20$ ,  $n=5\%$ .
- 4.2. Запустите программу на исполнение. Все алгоритмы задачи должны выводить одинаковые результаты, согласно Вашим расчетам.

## Тест самоконтроля знаний

### 1. Циклический оператор содержит:

- a) константу выбора;
- b) две и более цепочки действий;
- c) повторяющиеся команды.

### 2. Условие проверяется в начале цикла:

- a) Repeat ...;
- b) во всех циклических операторах;
- c) While ....

### 3. Выберите верное утверждение:



- a) блок-схема цикла с предусловием;
- b) TRUE, FALSE нужно поменять местами;
- c) оператор2 веточки FALSE лишний.

### 4. Цикл с предусловием имеет вид:

- a) Repeat (<условие>) do <оператор>;
- b) While (<условие>) then <оператор>;
- c) While (<условие>) do <оператор>.

### 5. Выберите верное утверждение:



- a) блок-схема цикла с постусловием;
- b) TRUE, FALSE нужно поменять местами;
- c) отсутствует второй оператор.

### 6. Параметр цикла с постусловием:

- a) обязателен и изменяется автоматически;
- b) задается и изменяется программистом;
- c) изменяется каждый раз только на один.

### 7. Тело цикла с параметром содержит:

- a) любое количество операторов;
- b) только один оператор;
- c) не более двух операторов.

### 8. a:=3; S:=0; For i:=2 to 5 do S:=S+a:

- a) результат S=15;
- b) результат S=12;
- c) результат S=S+a.

### 9. Начальное значение произведения:

- a) 1; б) -1; в) 0.

### 10. Отличие циклов с параметром и постусловием:

- a) параметр цикла изменяется автоматически/оператором;
- b) тело цикла выполняется n/(n-1) раз;
- c) условие записывается в одну/две строки.

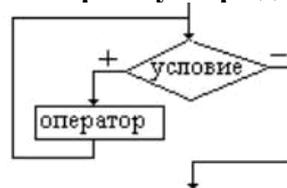
### 1. Циклические операторы бывают:

- a) с параметром/ пред-/ постусловиями;
- b) простые/сложные/многовариантные;
- c) начальным/повторяющимся условиями.

### 2. Условие проверяется в конце цикла:

- a) Repeat ...;
- b) For ...;
- c) While ....

### 3. Выберите верное утверждение:



- a) блок-схема цикла с предусловием;
- b) TRUE, FALSE нужно поменять местами;
- c) отсутствует оператор веточки FALSE.

### 4. Выход из цикла с предусловием:

- a) при ложности условия;
- b) при истинности условия;
- c) каждый раз после выполнения действий.

### 5. Выберите верное утверждение:



- a) блок-схема цикла с постусловием;
- b) отсутствует веточка повтора;
- c) отсутствует оператор для веточки TRUE.

### 6. Цикл с постусловием имеет вид:

- a) Repeat (<условие>) do <оператор>;
- b) While (<оператор>) then <условие>;
- c) Repeat (<оператор>) do <условие>.

### 7. Параметр цикла с параметром:

- a) обязателен и изменяется автоматически;
- b) задается и изменяется программистом;
- c) часто используется.

### 8. a:=3; t:=1; For i:=2 to 5 do t:=t\*a:

- a) результат t=120;
- b) результат t=5;
- c) результат t=81.

### 9. Начальное значение суммы ряда:

- a) S; б) 1; в) 0.

### 10. Отличие циклов с предусловием и постусловием:

- a) изменение параметра цикла автоматически/оператором;
- b) противоположностью условий цикла;
- c) записью в одну/несколько строк.

## Упражнения

### *Уровень начальный*

1. Начальный объем выпуска предприятия составляет  $S$  ед. Ежегодное увеличение выпуска составляет  $k\%$  к предыдущему году. Рассчитать предполагаемый выпуск предприятия на  $t$  лет вперед и суммарный выпуск предприятия.
2. Объем продажи авиабилетов в 2002 г. составил  $X$  тыс. ед. Ежегодный темп прироста составляет  $n\%$ . Рассчитать объем продажи до 2005 г. и средний ежегодный объем продаж.
3. В магазине №1 спрос на хозяйственные товары в 2001 г. составил  $m$  тыс. руб. Ежегодный прирост  $n\%$ . Рассчитать спрос на 3 года вперед и средний спрос.

### *Уровень средний*

4. Объем продажи авиабилетов в 1 квартале 2002 г. составил  $X$  тыс. ед. Ежеквартальный темп прироста составляет  $n\%$ . Рассчитать объем продажи до конца 2002 г. и средний ежеквартальный объем продаж.
5. Человек открыл счет в банке  $S$  руб. под  $n\%$  годовых. Начисление процентов происходит в конце года. Рассчитать сумма вклада на 3 года вперед и среднее значение вклада за 3 года.
6. Первоначальная стоимость основного фонда предприятия составляет  $S$  млн. руб. Амортизация в год составляет  $n\%$ : ежегодная стоимость уменьшается на  $n\%$ . Рассчитать стоимость основного фонда предприятия на  $t$  лет вперед.

### *Уровень высокий*

7. Для кредита на сумму  $S$  на  $t$  лет под  $n\%$  годовых рассчитать сумму выплат: погашение основного кредита + выплата процентов. Известно, что сумма основного долга выплачивается равными долями ежемесячно, ежемесячный процент  $n/12$  рассчитывается на оставшуюся сумму долга. Указать общую сумму выплаты и сумму переплаты по отношению к начальной сумме.

8. В 1 квартале 2000 г. цена на телевизор «SAMSUNG-2536NY» составила  $x$  тыс. руб. Ежеквартальное увеличение цены составляет  $n\%$ . Вычислить ежеквартальный спрос на TV -  $y$  единиц, если  $y=3,4*x^{0,4}$ .

**Дополнительные задания на повторение**

Дополните выполненное упражнение:

*Уровень начальный*

- а) панелью радиокнопок с выбором цвета формы;
- б) набором строк с выбором размера формы;
- в) панелью радиокнопок с выбором заглавия формы.

*Уровень средний*

- г) флажком с выбором «курсив» в Tmemo;
- д) флажком с выбором «красный цвет шрифта» в TLabel;

*Уровень высокий*

- е) набором строк с выбором «Удалить строку в мемо1», «Сохранить мемо2 в файл», «Добавить строку 'ЭЛЕМЕНТ' в мемо3», «Вывести окно диалога в ФИО автора».

**Творческие задания**

1. Начальный объем выпуска предприятия составляет  $S$  ед. Ежегодное увеличение выпуска составляет  $k\%$  к предыдущему году:  $S_t = S_{t-1} * k$ . Рассчитать предполагаемый выпуск предприятия на  $t$  лет вперед.
2. Для кредита на сумму  $S$  на  $t$  лет под  $n\%$  годовых рассчитать сумму выплат: погашение основного кредита и выплата процентов. Известно, что сумма основного долга выплачивается равными долями ежемесячно, ежемесячный процент  $n/12$  рассчитывается на оставшуюся сумму долга.
3. Первоначальная стоимость основного фонда предприятия составляет  $S$  млн. руб. Амортизация в год составляет  $n\%$ : ежегодная стоимость уменьшается на  $n\%$ . Рассчитать стоимость основного фонда предприятия на  $t$  лет вперед.

## Лабораторная работа №6. Работа со строками

### Вопросы теоретического контроля

1. Укажите название строкового типа данных. В какие знаки заключаются строки?
2. Каково назначение [n] при описании типа String[n]. В каких случаях [n] отсутствует?
3. Как выглядит обращение к символу строки?
4. Укажите знак, используемый для сложения строк.
5. Укажите функцию отыскания позиции подстроки в строке. Какого типа данных результат данной функции?
6. Запишите функцию вычисления длины строки и тип результат данной функции.
7. Назовите форматы функций удаления, копирования, вставки подстроки в строке.

### Цель занятия

#### УЗНАТЬ:

- запись команд присвоения строковым переменным значения текстовых компонент;
- правила замены в тексте одной подстроки на другую;
- присвоение текстовым компонентам значения строковых переменных;

#### НАУЧИТЬСЯ:

- описывать строковые переменные нужной длины;
- определять длину строки;
- обращаться к символу строки;
- находить и удалять в тексте данное слово;
- вставлять в текст подстроку;
- копировать часть текста.

### Задача: Программа замены слога в тексте. Подсчет количества замен.

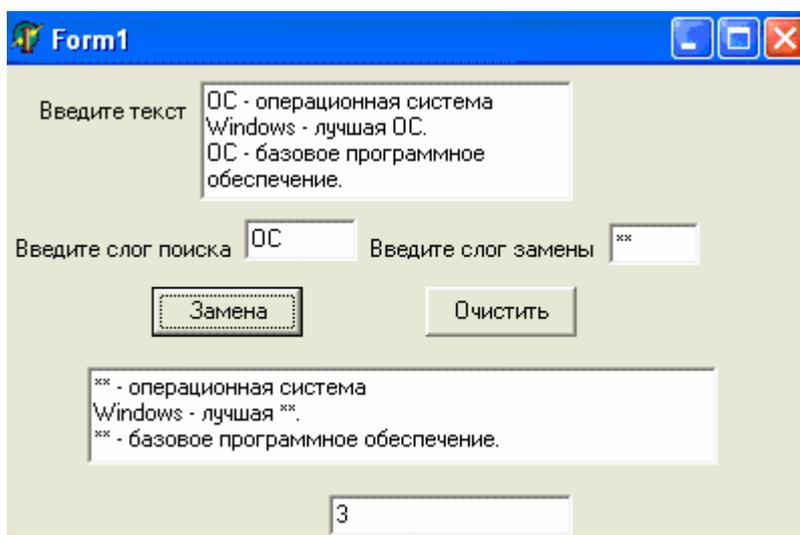


Рис. 6.1. Окно программы задачи 1.

## Этапы выполнения задачи 1

### Этап 1. Определение требований программы

В задаче необходимо ввести текст, искомый слог и слог замены. Для этого будем использовать входные переменные:

S – исходный текст; a - искомый слог; b - слог замены.

Результатом будет являться:

S - измененный текст; k - количество замен.

Промежуточные переменные определим в ходе алгоритмизации.

**Задание 1.** Опишите в тетради переменные:

S:string; - максимально текст может содержать 256 символов.

a,b:string[4]; - слог не может быть больше 4 символов.

k:integer; - количество замен целое число.

### Этап 2. Окно формы

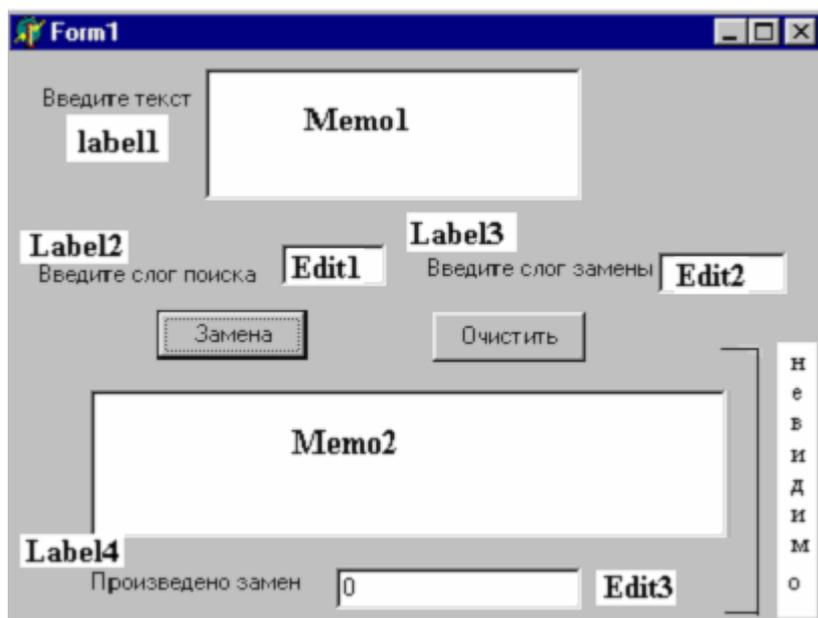


Рис. 6.2. Проектирование окна формы задачи 1.

Особенностями окна формы задачи 1 являются:

- при запуске программы результирующая часть окна невидна, поэтому для данных компонентов устанавливается свойство `visible=false`;
- текстовое поле компонентов Memo1 и Memo2 очищается с использованием свойства `Lines`.

**Задание 2.** Запустите Delphi. Сформируйте окно формы.

### Этап 3. Алгоритмизация и программирование

#### Алгоритм замены

Начало	Begin
1. Водной переменной S задать значения Memo1.	S:=memo1.text;
2. Входной переменной a задать значения Edit1.	a:=Edit1.text;
3. Входной переменной b задать значения Edit2.	b:=Edit2.text;
4. Количеству замен k задать начальное значение 0.	k:=0;
5. Найти позицию слога a в тексте b (переменная i = позиция слога a в тексте b).	i:=Pos(a,S);
6. Цикл: пока позиция не равна нулю ( $i < > 0$ ) будем делать:	<b>While</b> $i < > 0$ <b>do begin</b>
а) удалить в тексте с позиции i количество символов слога a;	delete(S,i,lenght(a));
б) вставить в текст S с позиции i слог b;	Insert (b,S,i);
в) увеличить количество замен (переменную k) на 1;	K:=k+1;
г) найти в измененном тексте S новую позицию слога a;	i:=Pos(a,S); <b>end;</b>
7. Вывод результата S в Memo2.	Memo2.text:=S;
8. Вывод результата k в Edit3.	Edit3.text:=k;
9. Результат сделать видимыми.	memo2.visible:=true; Label4.visible:=true; Edit3.visible:=true;
<b>Конец</b>	<b>End</b>

#### Обратите внимание!

Функции численно-строкового перевода не используются при задании:

- текстовой компоненте значения строковой переменной;
- строковой компоненте значения компоненты.

Например, Memo2.text:=S; S:=Memo1.text.

#### Алгоритм очистки

Начало	Begin
1. Очистить Memo1. Поместить курсор в Memo1.	Memo1.Lines.Clear; Memo1.Lines.SetFocus;
2. Очистить Edit1.	Edit1.Clear;
3. Очистить Edit2.	Edit2.Clear;
4. Очистить и скрыть Memo2.	Memo2.Lines.Clear; Memo1.visible:=false;
5. Очистить и скрыть Edit3.	Edit3.Clear; Edit3.visible:=false;
6. Скрыть Label3.	Label3.visible:=false;
<b>Конец</b>	<b>End</b>

**Задание 3.** В тетради составьте блок-схемы алгоритмов. В программе создайте события для клавиш Button1 и Button2, опишите реакцию на события.

**Тест самоконтроля знаний****Упражнения**

Задан текст: "Суздаль – областной центр. В центре города Суздаль расположен Кремль. Суздаль – один из первых русских городов. Город Суздаль находится на Северо-Западе России".

Написать программу, которая:

*уровень начальный*

- а) заменяет слово «Суздаль» словом «Новгород»;
- б) заменяет точки в тексте восклицательными знаками;
- в) определяет количество точек.

*уровень средний*

- г) заменяет слово «Суздаль» на название заданного города;
- д) рассчитывает количество встреч в тексте заданного слога;

*уровень высокий*

- е) подсчитывает количество слов, начинающихся с «о»;
- ж) подсчитывает количество слов в тексте.

## Лабораторная работа №6. Программы работы с векторами

### Вопросы теоретического контроля

1. Понятие одномерного массива (вектора).
2. Описание переменной типа вектор. Индексация элементов вектора.
3. Обращение к элементу вектора.

### Цель занятия

#### узнать:

- способы ввода-вывода вектора;
- задание значений элементов вектора;
- алгоритм сравнения элементов вектора.

#### научиться:

- суммировать элементы вектора;
- находить количество элементов с заданными свойствами;
- находить max (min) элемент вектора;
- формировать вектор-результат.

### Задача: Программа «Курс доллара 2003 г.». Вычисление:

- среднегодового курса доллара;
- количества месяцев курса выше 30 руб.;
- максимальное значение курса доллара в году;
- вектора-результата как разница между средним и текущим значениями \$.

Курс доллара

январь	февраль	март	апрель	май	июнь	июль	август	сентябрь	октябрь	ноябрь	декабрь
33,6	33,4	33,1	29,8	31,2	31,6	31,8	31,1	29,9	29,7	29,6	30,4

записать вектор

среднее значение

курс > 30 руб. кол-во месяцев

Месяца:

максимальный курс

значение

месяц

Новый вектор

Разница между средним и текущим курсами \$

--	--	--	--	--	--	--	--	--	--	--	--

Рис.6.1. Окно программы задачи «Курс доллара 2003 г.».

## Этапы выполнения задачи 1

### Этап 1. Определение требований программы

В программе используются переменные:

Входные: a:array [1..12] of real; - вектор ежемесячных значений курса \$ (12 месяцев).

Промежуточные: i:integer; - счетчик.

Выходные:

- ✓ S, c:real; - сумма и среднее элементов вектора;
- ✓ k:integer; - количество месяцев с курсом \$ более 30 руб.;
- ✓ max:real; - максимальный курс \$;
- ✓ nmax:integer; - номер месяца максимального курса.
- ✓ b:array [1..12] of real; - вектор-результат.

**Задание 1.** В тетради запишите условие задачи и переменные.

### Этап 2. Окно будущей программы

Для решения задачи форма формируется согласно рис.6.1 и 6.2. Вывод результатов каждой подзадачи осуществляется на отдельную панель.

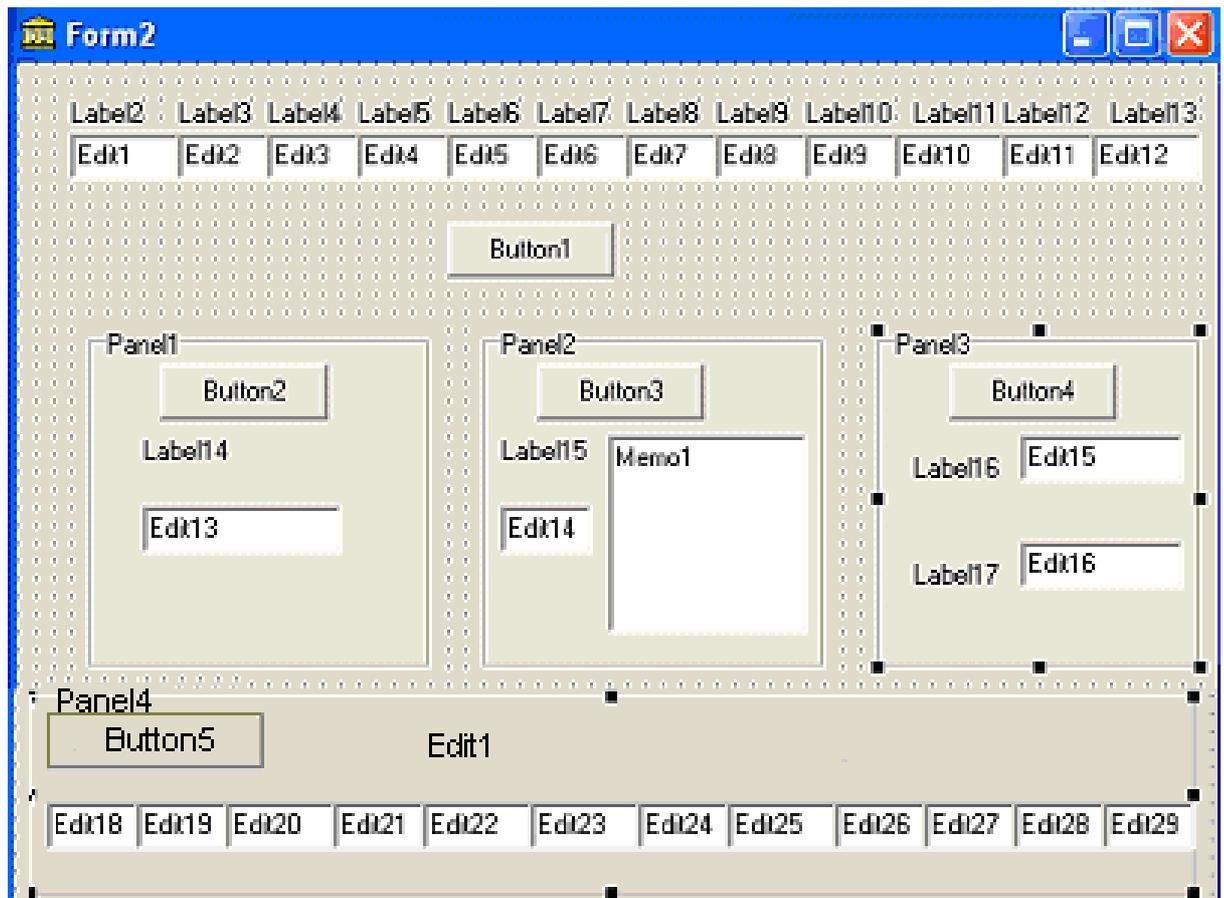


Рис.6.2. Компоненты окна формы задачи «Курс доллара 2003 г.».

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

### **Этап 3. Алгоритмизация и программирование** *Локальное и глобальное описание переменных.*

В ранее рассматриваемых задачах переменные описывались внутри каждой процедуры – локальное описание. *Локальные переменные известны только внутри данной процедуры.*

Как уже рассматривалось в лаб. работе №2 , глобальное описание данных осуществляется в первой части модуля - INTERFACE. *Глобальные переменные действительны для всех процедур модуля.*

В настоящей задаче исходный вектор является глобальным данным. Он будет использоваться во всех процедурах модуля. Переменная среднегодового значения курса \$ также будет глобальной – будет использоваться в двух процедурах. Мы можем все переменные описать как глобальные, при этом оперативная память, используемая работающей программой, увеличится на количество байт, необходимых для хранения значений глобально описанных переменных. В целях экономии оперативной памяти при написании программных продуктов количество данных (в т.ч. переменных) минимизируется.

**Unit ...**

**Interface**

...

**procedure** TForm1.Button1Click(Sender: TObject); («Записать вектор»)

**procedure** TForm1.Button2Click(Sender: TObject); («Средний курс \$»)

**procedure** TForm1.Button3Click(Sender: TObject); («Количество месяцев с курсом более 30 руб.»)

**procedure** TForm1.Button4Click(Sender: TObject); («Максимальный курс»)

**procedure** TForm1.Button5Click(Sender: TObject); («Новый вектор»)

**var** Form1: TForm1;

  a:array [1..12] of real;

  S,c: real;

  i:integer; k:integer;

  max:real; nmax:integer;

  b:array [1..12] of real;

Переменные описаны в секции INTERFACE

Все переменные являются глобальными

## Implementation

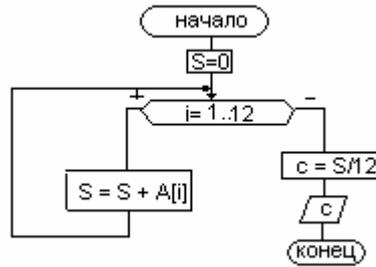
<pre> <b>procedure</b> TForm1.Button1Click(Sender: TObject); <b>begin</b> a[1]:=StrToFloat(Edit2.text); a[2]:=strToFloat(edit3.text); a[3]:=strToFloat(edit4.text); a[4]:=strToFloat(edit5.text); a[5]:=strToFloat(edit6.text); a[6]:=strToFloat(edit7.text); a[7]:=strToFloat(edit8.text); a[8]:=strToFloat(edit9.text); a[9]:=strToFloat(edit10.text); a[10]:=strToFloat(edit11.text); a[11]:=strToFloat(edit12.text); a[12]:=strToFloat(edit13.text); <b>end;</b> </pre>	<p><b>«Записать вектор»</b></p> <p><b>начало</b></p> <p>Элементам вектора присвоить значения с компонентов TEdit.</p> <p><b>конец;</b></p>
<pre> <b>procedure</b> TForm1.Button2Click(Sender: TObject); <b>begin</b> S:=0; <b>for</b> i:=1 <b>to</b> 12 <b>do</b>            S:=S+a[i]; c:=S/12; edit13.text:=FloatToStr(c); <b>end;</b> </pre>	<p><b>«Средний курс \$»</b></p> <p><b>начало</b></p> <p>первоначально сумма \$ S:=0;</p> <p>Цикл: перебирая все месяца</p> <p>к сумме прибавляем текущий курс \$;</p> <p>Среднее c:=S/12;</p> <p>В edit 13 вывести результат c - среднее;</p> <p><b>Конец</b></p>
<pre> <b>procedure</b> TForm1.Button3Click(Sender: TObject); <b>begin</b> k:=0; <b>for</b> i:=1 <b>to</b> 12 <b>do</b>   <b>if</b> a[i]&gt;30 <b>then begin</b>            k:=k+1;            memo1.Lines.Add(IntToStr(i));            <b>end;</b> edit14.Text:=IntToStr(k); <b>end;</b> </pre>	<p><b>«Количество месяцев с курсом более 30 руб.»</b></p> <p><b>начало</b></p> <p>Первоначально кол-во найденных месяцев k=0;</p> <p>Цикл: перебирая все месяца,</p> <p><b>если</b> курс i-месяца a[i]&gt;30, <b>то начало</b></p> <p>увеличиваем кол-во найденных месяцев k:=k+1;</p> <p>выводим номер месяца в Мемо1;</p> <p><b>конец;</b></p> <p>в Edit 14 выводится результат k;</p> <p><b>конец</b></p>

<pre> <b>procedure</b> TForm1.Button4Click(Sender: TObject); <b>begin</b> max:=a[1]; nmax:=1; <b>for</b> i:=1 to 12 <b>do</b>   <b>if</b> a[i]&gt;max <b>then</b>     <b>begin</b>       max:=a[i]; nmax:=i;     <b>end</b>; edit15.Text:=FloatToStr(max); edit16.Text:=IntToStr(nmax); <b>end</b>; </pre>	<p><b>«Максимальный курс»</b></p> <p><b>начало</b> Первоначально считается: max курс \$ в 1 месяце, номер месяца nmax =1.</p> <p><b>Цикл:</b> перебирая все месяца, <b>если</b> курс \$ i-месяца a[i] &gt; max, <b>то</b></p> <p><b>начало</b> значения max, nmax переписываются;</p> <p><b>конец;</b></p> <p>в Edit 15 выводится max курс \$ в Edit 16 выводится номер месяца</p> <p><b>конец;</b></p>
<pre> <b>procedure</b> TForm1.Button5Click(Sender: TObject); <b>var</b> B: array [1..12] of real; <b>begin</b> <b>for</b> i:=1 to 12 <b>do</b> B[i]:=c-a[[i];  edit17.Text:=FloatToStr(B[1]); ... edit29.Text:=FloatToStr(B[12]); <b>end</b>; </pre>	<p><b>«Новый вектор»</b></p> <p>переменная B – новый вектор</p> <p><b>начало</b></p> <p><b>Цикл:</b> перебирая все месяца, B[i-месяца] = среднееC – A[i-месяца]</p> <p>в Edit 17 выводится B[1] ...</p> <p>в Edit 29 выводится B[12]</p> <p><b>конец;</b></p>

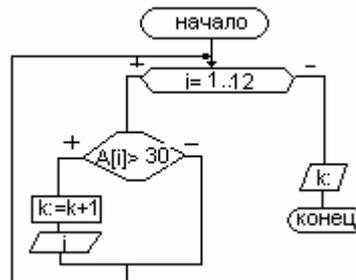
### Блок-схемы подзадач

Графическое представление линейных, циклических и разветвляющих алгоритмов Вам уже известны. Именно эти структуры используются для решения любых задач, в том числе работы с векторами. Рассмотрим две наиболее часто используемые блок-схемы решения подзадач работы с векторами.

«Средний курс \$» – блок-схема содержит циклический алгоритм.



«Количество месяцев с курсом более 30 руб.» – блок-схема содержит циклический алгоритм с вложенным неполным ветвлением.



### Задание 3.

- 3.1. Создайте события и реакции на события программы.
- 3.2. Зарисуйте блок-схемы подзадач «Запомнить», «Максимальный» и «Новый вектор».

### Этап 4. Тестирование и отладка

Данные курса \$ известны и представлены на рис. 6.1. Для проверки структуры программы рассчитывают:

- ✓ S – сумму всех элементов;
- ✓ C – среднее значение;
- ✓ K – количество элементов более 30 руб. и номера этих месяцев;
- ✓ Max – максимальный курс \$ и номер месяца максимального курса \$.

**Задание 4.** Рассчитайте на бумаге данные результата. Сравните с результатом программы.

### Тест самоконтроля знаний

**11. Тип данных массив относится к:**

- a) структурированным типам данных;
- b) простым типам данных;
- c) объектным типам данных.

**12. Вектор объема продаж квартир А (2000-2004 гг.) описывается:**

- a) A: array [2000-2004] of Integer;
- b) A: array [2000..2004] of real;
- c) A:=IntToStr ([2000,2004]).

**13. Обращение к 2001 г. вектора А задания 2 записывается:**

- a) A(1);
- b) A[2001];
- c) A(2).

**14. For i:=2000 to 2002 do S:=S+A[i]:**

- a) находится сумму значений за три года;
- b) находится количество значений;
- c) находится среднегодовое значение.

**15. for i:=2000 to 2004 do**

**edit11.text:=IntToStr(i):**

- a) выведет на экран номер года;
- b) выведет на экран объем продаж i-года;
- c) выведет количество лет вектора.

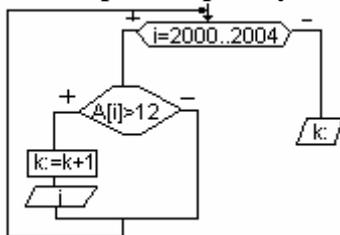
**16. k:=5; A[2000]:=14; A[2001]:=12;  
A[2002]:=13; A[2003]:=16; A[2004]:=17;  
For i:=2000 to 2004 do**

**If A[i]>14 then k:=k-1;**

**Выберите конечное значение k:**

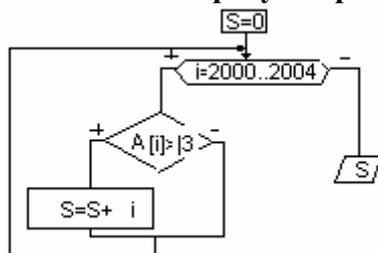
- a) k = 1; b) k = 3; c) k = 2.

**17. Выберите верное утверждение:**



- a) K выводится после выполнения цикла;
- b) K выводится при ложности ветвления;
- c) K выводится при истинности ветвления.

**18. Согласно вопросу 6 S равно:**



- a) S=47; b) S=3; c) S=6007

**11. Одномерный массив - это:**

- a) множество равных элементов;
- b) структура однотипных элементов;
- c) совокупность разнотипных элементов.

**12. Вектор ежеквартальных доходов В (1-4 кв.) описывается:**

- a) B: array [1..4] of real;
- b) B: =[1,4];
- c) B: =FloatToStr(1..4);

**13. Обращение к 1 кварталу вектора В задания 2 записывается:**

- a) B[0];
- b) B (0);
- c) B[1].

**14. For i:=1 to 4 do B[i]:=A[0] + A[i]:**

- a) выражение не имеет смысла;
- b) находится годовая сумма (1-4 кв.);
- c) рассчитывается новый вектор B[1-4 кв.].

**15. for i:=1 to 4 do**

**edit11.text:=IntToStr(B[i]):**

- a) выведет номер года;
- b) выведет ЛОЖЬ;
- c) выведет на экран доход i-квартала.

**16. k:=0;**

**B[1]:=4; B[2]:=2; B[3]:=5; B[4]:=3;**

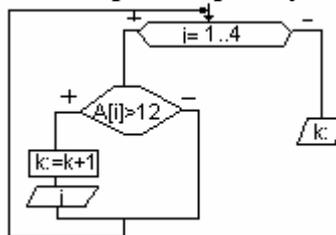
**For i:=1 to 4 do**

**If B[i]>4 then k:=k+1;**

**Выберите конечное значение k:**

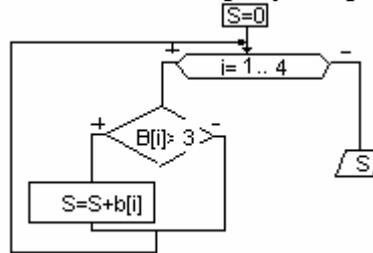
- a) k = 2; b) k = 4; c) k = 1.

**17. Выберите верное утверждение:**



- a) i выводится после выполнения цикла;
- b) i выводится при ложности ветвления;
- c) i выводится при истинности ветвления.

**18. Согласно вопросу 6 S равно:**



- a) S=5; b) S=9; c) S=14.

### Упражнения

1. Для вектора объема продаж квартир А (2000-2004 гг.) найти:
  - среднегодовой объем продаж квартир;
  - максимальный объем продаж и номер года максимального объема продаж;
  - количество лет, в которых объем продаж квартир был ниже среднего;
  - получить новый вектор, как разница между максимальным объемом и текущим годовым.
2. Для вектора расходов на оплату услуг мобильной связи В (2001-2005 гг.) найти:
  - суммарный расход за 2001 – 2005 гг.;
  - минимальный расход и номер года минимального расхода;
  - количество лет, в которых расход был выше минимального;
  - получить новый вектор, как процент текущего года от суммы.
3. Для вектора доходов от реализации Интернет-услуг С (2001-2005 гг.) найти:
  - суммарный доход за 2001 – 2005 гг.;
  - максимальный и минимальный доход;
  - количество лет с доходом выше  $\frac{1}{4}$  суммы доходов за 2001-2005 гг.;
  - получить новый вектор, в котором значения лет с доходом ниже  $\frac{1}{4}$  суммы доходов за 2001-2005 гг. заменить нулями.

### Творческие задания

#### *уровень начальный*

1. В векторе объема удоя молока фермерского хозяйства А (2000-2004 гг.) найти максимальное значение. Указать количество максимальных элементов и года их встречи.
2. В векторе объема продаж мороженого С (2000-2004 гг.) найти среднегодовое значение. Получить новый вектор как разницу между средним и текущим годом.

*уровень средний*

3. Для вектора выпуска бытовой техники А (2000-2004 гг.) найти года выпуска, значения которых больше среднего.
4. В векторе спроса на ЭКСПЕРЕСС-кредит С (2000-2004 гг.) найти суммарное и максимальное значения. Указать процент максимального значения от суммарного.

*уровень высокий*

5. Для вектора объема сбора урожая В (1995-2004гг.) найти средние значения XX(1995-2000 гг.) и XXI(2001-2004 гг.) и указать в каком веке наибольшее.
6. Для вектора В (1999-2004гг.) найти разницу между минимальным и максимальным значениями. Указать год с минимальным значением.
7. Для вектора объема продаж авиабилетов А (2000-2002 гг.) получить новый вектор как разницу между объемом 2000 г. и текущим годовым значением.

## Лабораторная работа №7. Программы работы с матрицами

### Вопросы теоретического контроля

4. Понятие двумерного массива (матрицы).
5. Описание переменной типа матрица. Индексация элементов матрицы.
6. Обращение к элементу матрицы.

### Цель занятия

#### узнать:

- способы ввода-вывода матрицы;
- задание значений элементов матрицы;
- алгоритм сравнения элементов матрицы.

#### научиться:

- суммировать элементы матрицы;
- находить количество элементов с заданными свойствами;
- находить max (min) элемент матрицы;
- формировать матрицу-результат.

### Задача: Программа «Поквартальный курс доллара 2003-2005 гг.».

#### Вычисление:

- среднеквартального курса доллара;
- количества кварталов каждого года выше 30 руб.;
- максимальное значение курса доллара;
- матрицы-результата с заменой курса ниже среднеквартального на 0.

Курс доллара 2003-2005 гг.

	2003	2004	2005
1 квартал	28,3	33,6	33,4
2 квартал	33,1	29,8	31,2
3 квартал	31,6	31,8	31,1
4 квартал	29,9	29,7	29,6

запомнить матрицу

Значения ниже среднего заменены нулями

	2003	2004	2005
1 квартал	28,3	33,6	33,4
2 квартал	33,1	0	31,2
3 квартал	31,6	31,8	31,1
4 квартал	29,9	29,7	29,6

Новая матрица

среднеквартальное значение: 28,60833333

курс > 30 руб.

кол-во кварталов 2003 г. 2

кол-во кварталов 2004 г. 2

кол-во кварталов 2005 г. 3

максимальный курс

значение: 33,6

год: 2004

квартал: 1

Рис. 7.1. Окно программы задачи «Курс доллара 2003-2005 гг.».

## Этапы выполнения задачи 1

### Этап 1. Определение требований программы

В программе используются переменные:

Входные: `a:array [1..4, 2003..2005] of real;` - матрица 1-4 строки, 2003-2005 столбцы.

Промежуточные: `i:integer;` - счетчик кварталов;

`g:integer;` - счетчик лет.

Выходные:

- ✓ `S, c:real;` - сумма и среднее элементов матрицы;
- ✓ `k2003, k2004, k2005:integer;` - количество кварталов года с курсом \$ более 30 руб.;
- ✓ `max:real;` - максимальный курс \$ матрицы;
- ✓ `kvmax:integer;` - номер квартала максимального курса.
- ✓ `godmax:integer;` - номер года максимального курса.
- ✓ `b:array [1..4, 2003.. 2005] of real;` - матрица-результат.

**Задание 1.** В тетради запишите условие задачи и переменные. Укажите какие переменные глобальные, какие локальные для подзадач «Запомнить», «Среднеквартальное», «Количество», «Максимальное», «Новая матрица».

### Этап 2. Окно будущей программы

Для решения задачи форма формируется согласно рис.7.1 и 7.2. Вывод результатов каждой подзадачи осуществляется на отдельную панель.



Рис.7.2. Компоненты окна формы задачи «Курс доллара 2003 г.».

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

### Этап 3. Алгоритмизация и программирование

Unit ...

**Interface** (Секция описания данных)

...

**procedure** TForm1.Button1Click(Sender: TObject); («Запомнить матрицу»)

**procedure** TForm1.Button2Click(Sender: TObject); («Среднеквартальный курс матрицы»)

**procedure** TForm1.Button3Click(Sender: TObject); («Количество кварталов года с 1\$ > 30 руб.»)

**procedure** TForm1.Button4Click(Sender: TObject); («Максимальный курс»)

**procedure** TForm1.Button5Click(Sender: TObject); («Новая матрица»)

**var** Form1: TForm1;

**a:array [1..4, 2003..2005] of real;**

**i,g:integer;**

**c:real;**

Переменные секции INTERFACE глобальные

Матрица А используется во всех процедурах

Счетчики кварталов I и лет G часто используются

C- среднее используется для расчета

**Implementation** (Секция реализации)

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

**begin**

a[1,2003]:=StrToFloat(Edit1.text);

a[1,2004]:=strToFloat(edit2.text);

a[1,2005]:=strToFloat(edit3.text);

a[2,2003]:=strToFloat(edit4.text);

a[2,2003]:=strToFloat(edit4.text);

a[2,2005]:=strToFloat(edit6.text);

a[3,2003]:=strToFloat(edit7.text);

a[3,2004]:=strToFloat(edit8.text);

a[3,2005]:=strToFloat(edit9.text);

a[4,2003]:=strToFloat(edit10.text);

a[4,2004]:=strToFloat(edit11.text);

a[4,2005]:=strToFloat(edit12.text);

**end;**

**«Запомнить матрицу»**

**начало**

Элементам матрицы присвоить значения с компонентов TEdit.

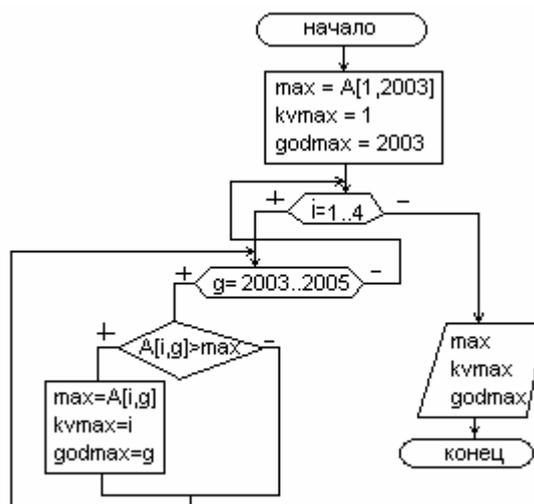
**конец;**

<pre> <b>procedure</b> TForm1.Button2Click(Sender: TObject); <b>var</b> S:real; <b>begin</b> S:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2003 <b>to</b> 2005 <b>do</b>     S:=S+a[i,g]; c:=S/12; edit13.text:=FloatToStr(c); <b>end</b>; </pre>	<p><b>«Среднеквартальный курс \$»</b></p> <p>локальная переменная S - сумма</p> <p><b>начало</b></p> <p>первоначально сумма \$ S:=0;</p> <p><b>Цикл</b> I=1..4: перебирая все кварталы</p> <p><b>Цикл</b> G=2003..2005: перебирая все года к сумме прибавляем текущий курс \$;</p> <p>Среднее c:=S/12;</p> <p>В edit 13 вывести результат с - среднее;</p> <p><b>Конец</b></p>
<pre> <b>procedure</b> TForm1.Button3Click(Sender: TObject); <b>var</b> k2003, k2004,k2005:integer; <b>begin</b> k2003:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2003 <b>to</b> 2003 <b>do</b>     <b>if</b> a[i,g]&gt;30 <b>then</b> k2003:=k2003+1;  edit30.Text:=IntToStr(k2003);  k2004:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2004 <b>to</b> 2004 <b>do</b>     <b>if</b> a[i,g]&gt;30 <b>then</b> k2004:=k2004+1; edit31.Text:=IntToStr(k2004);  k2005:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2005 <b>to</b> 2005 <b>do</b>     <b>if</b> a[i,g]&gt;30 <b>then</b> k2005:=k2005+1; edit32.Text:=IntToStr(k2005); <b>end</b>; </pre>	<p><b>«Количество месяцев года с 1\$ &gt; 30 руб.»</b></p> <p>Локальные переменные количества каждого года</p> <p><b>начало</b> Первоначально кол-во найденных кварталов 2003 года k2003=0;</p> <p><b>Цикл</b> I=1..4: перебирая все кварталы</p> <p><b>Цикл</b> G=2003..2005: перебирая все года</p> <p><b>если</b> курс i-квартала, g-года a[i,g]&gt;30,</p> <p><b>то</b> увеличиваем кол-во k2003:=k2003+1;</p> <p>в Edit 30 выводится результат k2003;</p> <p>...</p> <p><b>конец</b></p> <p>Эту же процедуру модно было реализовать другими способами. Например,</p> <p><b>Begin</b> k2003:=0; k2004:=0; k2005:=0;</p> <p><b>for</b> i:=1 <b>to</b> 4 <b>do begin</b></p> <p><b>if</b> a[i,2003]&gt;30 <b>then</b> k2003:=k2003+1;</p> <p><b>if</b> a[i,2004]&gt;30 <b>then</b> k2004:=k2004+1;</p> <p><b>if</b> a[i,2005]&gt;30 <b>then</b> k2005:=k2005+1;</p> <p><b>end</b>;</p> <p>edit30.Text:=IntToStr(k2003);</p> <p>edit31.Text:=IntToStr(k2004);</p> <p>edit32.Text:=IntToStr(k2005);</p> <p><b>end</b>;</p>

<pre> <b>procedure</b> TForm1.Button4Click(Sender: TObject); <b>var</b> max:real; kvmax, Godmax:integer; <b>begin</b> max:=a[1,2003]; kvmax:=1; Godmax:=2003; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2003 <b>to</b> 2005 <b>do</b>     <b>if</b> a[i,g]&gt;max       <b>Then begin</b>         max:=a[i,g];         kvmax:=i;         godmax:=g;       <b>end;</b> edit14.Text:=FloatToStr(max); edit15.Text:=FloatToStr(godmax); edit16.Text:=IntToStr(kvmax); <b>end;</b> </pre>	<p><b>«Максимальный курс»</b></p> <p>Переменные max-курс, kvmax-квартал, godmax-год</p> <p><b>начало</b> Первоначально считается:  max курс \$ в 1 квартале 2003 года,  max-квартал lmax =1.  max-год Godmax =2003.</p> <p><b>Цикл</b> l=1..4: перебирая все кварталы</p> <p><b>Цикл</b> G=2003..2005: перебирая все года</p> <p><b>если</b> курс \$ i-месяца g-года a[l,g] &gt; max,  <b>то начало</b></p> <p>значения max, kvmax, godmax переписываются;</p> <p><b>конец;</b></p> <p>в Edit 14 выводится max-курс \$  в Edit 15 выводится max-квартал  в Edit 16 выводится max-год</p> <p><b>конец;</b></p>
<pre> <b>procedure</b> TForm1.Button5Click(Sender: TObject); <b>var</b> B: array[1..4,2003..2005] of real; <b>begin</b> <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2003 <b>to</b> 2005 <b>do</b>     <b>if</b> a[i,g]&lt;c <b>then</b> b[i,g]:=0 <b>else</b> b[i,g]:=a[i,g];  edit18.text:=FloatToStr(b[1,2003]); edit19.text:=FloatToStr(b[1,2004]); ... edit29.text:=FloatToStr(b[4,2005]); <b>end;</b> </pre>	<p><b>«Новая матрица»</b></p> <p>Локальная переменная B – новая матрица</p> <p><b>начало</b></p> <p><b>Цикл</b> l=1..4: перебирая все кварталы</p> <p><b>Цикл</b> G=2003..2005: перебирая все года</p> <p><b>если</b> курс i-месяца g-года &lt; среднего (c),  <b>то</b> в новой матрице на этом месте пишем 0  <b>иначе</b> в новой матрице значение не меняется</p> <p>в Edit 18 выводится B[1,2003]  в Edit 19 выводится B[1,2004]  ...  в Edit 29 выводится B[4,2005]</p> <p><b>конец;</b></p>

### Блок-схемы подзадач

Блок-схема «Максимальный курс \$» содержит 3 структуры: в цикл для  $I=1..4$  вложен цикл для  $G=2003$  для 2005, который в свою очередь содержит ветвление.



В результате выполнения данного алгоритма:

For  $i:= 1$  to 4 do

For  $g:= 2003$  to 2005 do ...

просмотр матрицы будет осуществляться следующим образом:

для  $i=1$

$g=2003$

$g=2004$

$g=2005$

для  $i=2$

$g=2003$

$g=2004$

$g=2005$

...

для  $i=4$

$g=2003$

$g=2004$

$g=2005$

1,2003	1,2004	1,2005	→
2,2003	2,2004	2,2005	→
3,2003	3,2004	3,2005	→
4,2003	4,2004	4,2005	→

### Задание 3.

3.1. Создайте события и реакции на события программы.

3.2. Зарисуйте блок-схемы «Количество» и «Среднеквартальное».

### Упражнения

1. Для матрицы объема продаж квартир А (1-4 кв., 2002-2004 гг.) найти:
  - средней объем продаж квартир каждого года;
  - максимальный объем продаж всей матрицы, номер квартала и года максимального объема продаж;
  - количество кварталов всей матрицы, в которых объем продаж квартир был ниже среднего;
  - получить новую матрицу, как разница между максимальным объемом и текущим кварталом.
2. Для матрицы расходов на оплату услуг мобильной связи В (1-2 полугодия, 2001-2005 гг.) найти:
  - суммарный расход за 2001 – 2005 гг.;
  - минимальный расход каждого года, номер полугодия;
  - количество полугодий матрицы, в которых расход был выше минимального;
  - получить новую матрицу, как процент полугодия от суммы.
3. Для матрицы доходов от реализации Интернет-услуг С (1-4 кв., 2003-2005 гг.) найти:
  - суммарный доход всей матрицы за 2003 – 2005 гг.;
  - максимальный доход каждого года;
  - количество кварталов матрицы с доходом выше  $\frac{1}{4}$  суммы доходов за 2003-2005 гг.;
  - получить новую матрицу, в которой значения кварталов с доходом ниже  $\frac{1}{4}$  суммы доходов за 2003-2005 гг. заменить нулями.

### Творческие задания

#### *уровень начальный*

1. В матрице объема удоя молока фермерского хозяйства А (1-4 кв., 2000-2004 гг.) найти максимальное значение. Указать количество максимальных элементов, года и кварталы их встречи.

2. Для матрицы выпуска бытовой техники А (1-4 кв., 2002-2004 гг.) найти кварталы выпуска, значения которых больше среднего.

*уровень средний*

3. Для матрицы В (1-4 кв., 2001-2004гг.) найти разницу между минимальным и максимальным значениями. Указать год с минимальным значением.

4. В матрице спроса на ЭКСПЕРЕСС-кредит С (1-2 полугодия, 2000-2004 гг.) найти суммарное и максимальное значения. Указать процент максимального значения от суммарного.

*уровень высокий*

5. Для матрицы объема сбора урожая В (1-2 полугодия, 1995-2004гг.) найти средние значения XX(1995-2000 гг.) и XXI(2001-2004 гг.) и указать в каком веке наибольшее.

6. В матрице объема продаж мороженого С (1-2 полугодия, 2000-2004 гг.) найти среднеполугодовые значения S1 и S2. Получить новую матрицу как процент текущего полугодия к соответствующей сумме.

7. Для матрицы объема продаж авиабилетов А (1-4 кв., 2000-2002 гг.) получить новую матрицу как разницу между первым и текущим кварталами соответствующего года.

## Лабораторная работа №8. Программы построения графиков

### Цель занятия

#### узнать:

- компонент построения графиков TChart;

#### научиться:

- графически представлять данные задачи.

### Задача: Графическое изменение курса \$ с 2003 по 2005 гг.

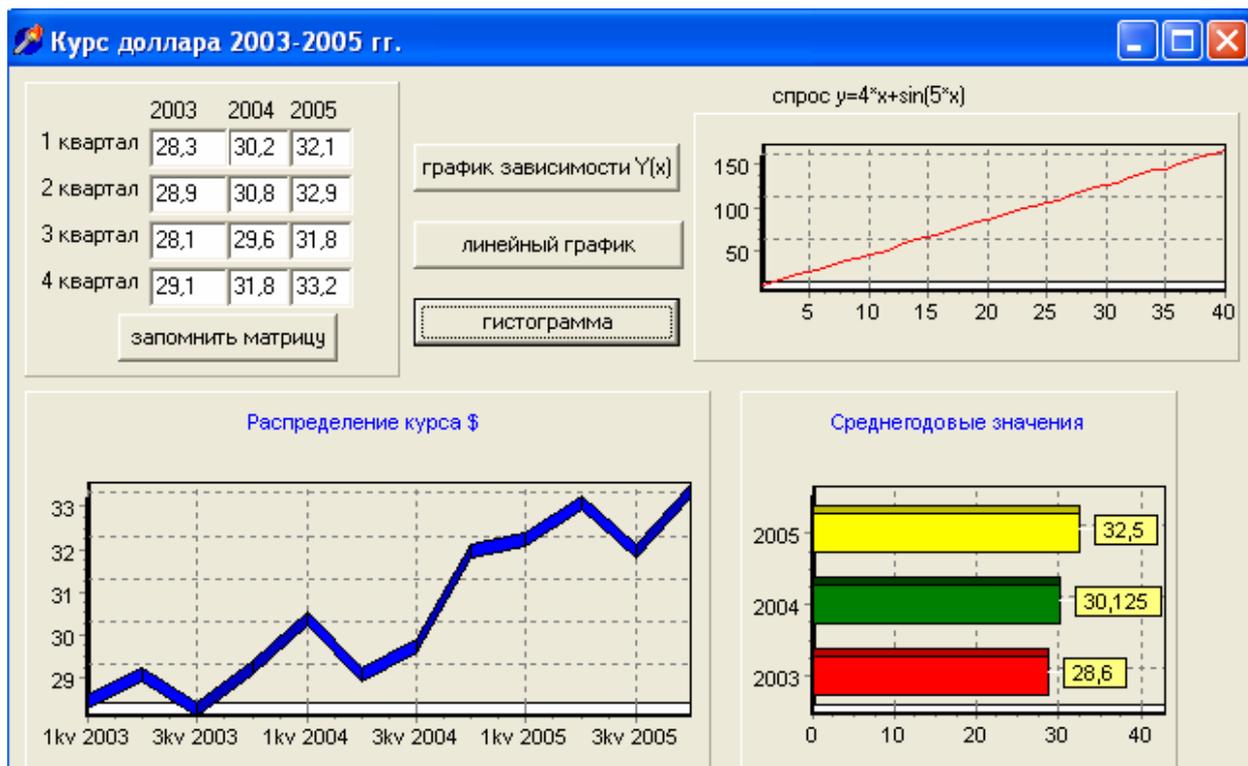


Рис.8.1. Пример окна программы построения графиков.

### Этапы выполнения задачи

#### Этап 1. Окно формы. Компонент TCHART

 **TChart** – построитель графиков. Компонент предназначен для графического представления числовых данных (рис.8.2).

Компонент расположен на странице Additional (Дополнительная) палитры компонентов.

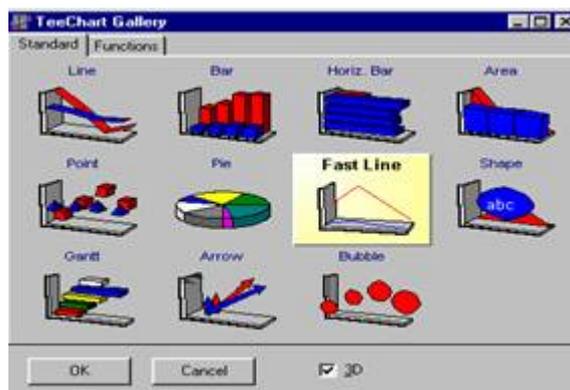


Рис. 8.2. Возможные виды графиков компонента TChart.

После размещения компонента на форме двойной щелчок мыши (или контекстное меню *Edit Chart*) вызывает окно редактирования компонента (рис.8.3).

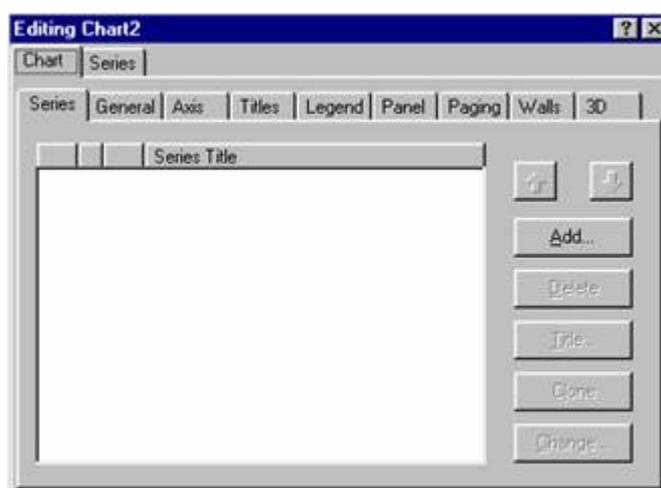


Рис. 8.3. Окно редактирования графиков компонента TChart.

В окне редактирования нужно нажать кнопку *Add* и выбрать подходящий тип графика (рис.8.2).

Каждый выбранный график называется серией, которые нумеруются, начиная с нуля. Доступ к сериям осуществляется с помощью закладки *Series*.

Окно редактирования *Series* (рис. 8.4) имеет закладки:

- General / Show in Legend (отображение легенды);
- Marks / Visible (подписи данных).

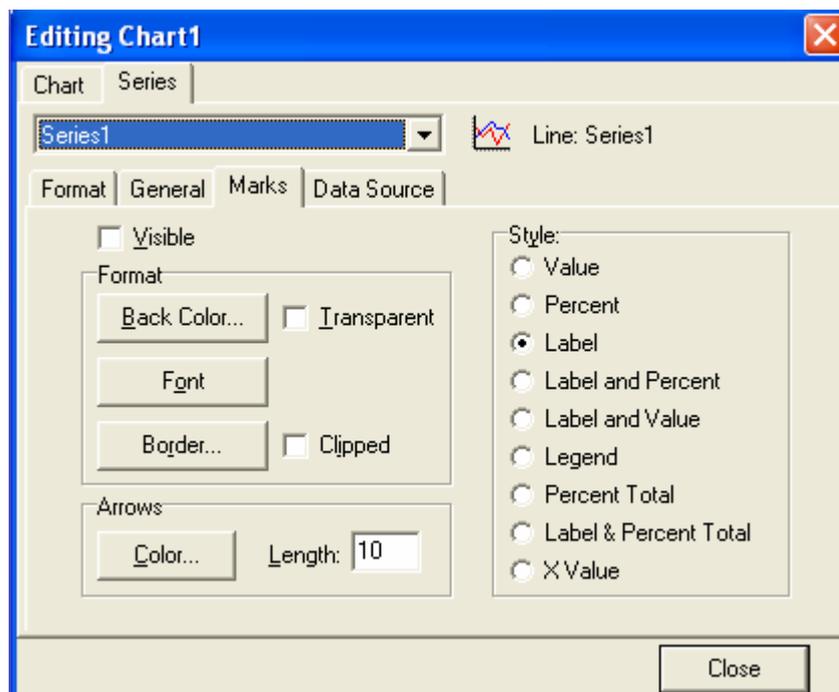


Рис. 8.4. Окно редактирования серий компонента TChart.

После закрытия окна редактора компонент будет содержать примерный вид графика. Однако его реальный вид зависит от фактических данных, которые создаются в работающей программе. В программе обращаются к компоненту Chart1 обращаются:

`Chart1.SeriesList[0].AddXY(х, у, подпись данных, цвет),`

где SeriesList [<номер серии>] – список серий; первая созданная серия имеет индекс 0, вторая 1 и т.д. AddX, AddY, AddXY – добавить данные по X, Y, XY в зависимости от типа графика.

### **Задание 1.**

1.4. Создайте окно формы как на рис. 8.1.

1.5. Задайте свойства графиков:

- ✓ для линейного графика отключите легенду и добавьте подписи данных: шрифт 8 пт, стиль=метки (Label);
- ✓ для гистограммы отключите легенду и добавьте подписи данных: стиль=значения (value);
- ✓ для графика зависимости  $Y(X)$  отключите легенду и подписи данных.

## Этап 2. Программирование

Программа основывается на задаче лабораторной работы 7 «Курс \$ 1-4 кварталы, 2003-2005 гг.». Реализация процедур основана на данной задаче.

### Описание глобальных переменных:

```
Var a:array [1..4, 2003..2005] of real;
i,g:integer;
```

### Процедура «Запомнить»

```
begin
a[1,2003]:=StrToFloat(Edit1.text);
...
a[4,2005]:=strToFloat(edit12.text);
end;
```

### Процедура «Линейный график – Временной ряд»

```
var x,y:integer; d:string;
begin
for g:=2003 to 2005 do
  For i:= 1 to 4 do begin
    d:=IntToStr(i)+'kv '+IntToStr(g); - подпись данных строка d = номер квартала + номер года
    Chart1.SeriesList[0].addY(a[i,g],d,clblue);
  end;
end;
```

### Процедура «Гистограмма среднегодовых значений»

```
var S2003,S2004,S2005:real; c2003,c2004,c2005:real;
begin S2003:=0; For i:= 1 to 4 do s2003:=s2003+a[i,2003];
c2003:=S2003/4;
Chart2.SeriesList[0].addX(c2003,'2003',clred);           подпись данных - '2003', красная
S2004:=0; For i:= 1 to 4 do s2004:=s2004+a[i,2004];
c2004:=S2004/4;
Chart2.SeriesList[0].addX(c2004,'2004',clgreen);        подпись данных – '2004', зеленая
S2005:=0; For i:= 1 to 4 do s2005:=s2005+a[i,2005];
c2005:=S2005/4;
Chart2.SeriesList[0].addX(c2005,'2005',clyellow);      подпись данных – '2005', желтая
End;
```

**Процедура «График зависимости Y(X)»**

```
var x:integer; y:real;
```

```
begin
```

```
For x:= 1 to 40 do
```

```
  begin
```

```
    y:=4*x+sin(5*x);
```

```
    Chart3.SeriesList[0].addXY(x,y,"clred); - подписи данных не задаются (")
```

```
  end;
```

```
end;
```

**Задания 2.** Создайте события и реакции на события. Поэкспериментируйте с графиками: цвет, заголовок, стиль и др.

**Упражнения***уровень начальный*

1.Спрос  $y$  (ед.) в сутки на 1000 жителей от цены товара  $x$  (руб.) выражается степенной функцией  $y=3,4*x^{0,4}$ . Составить программу, в которой для данной цены  $X$  вычисляется спрос  $Y$ . Изобразить график зависимости  $Y(X)$ .

2. Спрос на получения кредита  $S$  (руб.) семьи в год для семьи со среднемесячный доход семьи на одного человека  $m$  (руб.) описывается моделью:  $S = 5^m - 3,4m$ . Составить программу вычисления спроса для введенного  $m$ . Построить график зависимости  $S(m)$ .

3. Телевизор Samsung 34KL стоит  $x_1, x_2, x_3$  руб. соответственно в магазине №1, №2, №3. В каждом магазине при покупке TV дается скидка  $r_1\%, r_2\%, r_3\%$  соответственно. Вычислить реальную стоимость покупки в каждом магазине, отобразите расчеты в виде гистограммы.

4. Известен средний ежеквартальный расход электроэнергии  $P$  кВт/час. При этом известно:

в 1 квартале расход на 15% выше среднего,

во 2 и 4 кварталах - на 5% выше среднего,

в 3 квартале - на 8% ниже среднего.

Вывести расход электроэнергии в каждом квартале  $p_1, p_2, p_3, p_4$ . Изобразите ежеквартальный расход электроэнергии в виде круговой диаграммы.

*уровень средний*

1. Первоначальная стоимость основного фонда предприятия составляет  $S$  млн. руб. Амортизация в год составляет  $n\%$  (ежегодная стоимость уменьшается на  $n\%$ ). Рассчитать стоимость основного фонда предприятия на  $t$  лет вперед и отобразить уменьшение в виде гистограммы. Используйте цикл с параметром.
2. Человек открыл счет в банке  $S$  руб. под  $n\%$  годовых. Начисление процентов происходит в конце года. Рассчитать сумма вклада на 3 года вперед и отобразить увеличение суммы вклада в виде графика. Используйте цикл с предусловием.
3. Для вектора объема сбора урожая  $B$  (1995-2004гг.) найти средние значения  $XX$ (1995-2000 гг.) и  $XXI$ (2001-2004 гг.), которые отобразить на гистограмме.
4. В матрице объема продаж мороженого  $C$  (1-2 полугодия, 2000-2004 гг.) найти минимальное значение. Построить график временного ряда.

*уровень высокий*

1. Для кредита на сумму  $S$  на  $t$  лет под  $n\%$  годовых рассчитать сумму выплат: погашение основного кредита + выплата процентов. Известно, что сумма основного долга выплачивается равными долями ежегодно, годовой процент рассчитывается на оставшуюся сумму долга. Указать общую сумму выплаты и сумму переплаты по отношению к начальной сумме. Ежегодные суммы выплат отобразить в виде линейного графика.
2. Для вектора выпуска бытовой техники  $A$  (2000-2004 гг.) найти суммарное значение. Получить новую матрицу как процент каждого года от суммы. Новую матрицу отобразить в виде круговой диаграммы.
3. Для матрицы объема сбора урожая  $B$  (1-4 кв., 2001-2004гг.) найти среднеквартальное значение  $C_{2001}$ ,  $C_{2002}$ ,  $C_{2003}$ ,  $C_{2004}$  каждого года. Изобразить круговую диаграмму данных среднеквартальных значений.

### Дополнительная литература

1. Волков В. Б. Самоучитель Delphi. –СПб: Изд-во "Питер", 2005 г.
2. Дэн Оузер, и др. Delphi. Освой самостоятельно /Пер. с англ. Архангельского А.Я. - М.: «Издательство БИНОМ», 1998 г.
3. Кетков А.Ю. Практика программирования: Visual Basic, C++ Builder, Delphi (монография). – СПб.: БХВ-Санкт-Петербург, 2002.- 464 с
4. Кнут Д. Искусство программирования для ЭВМ, т. 1. Основные алгоритмы (монография). — СПб.: Вильямс, 2000.
5. Макаручук Т.А. Основы программирования в Delphi 5. Учебно-методическое пособие. – Благовещенск, АмГУ, 2001.
6. Овсянников А., Овсянникова Т., Марченко А., Прохоров Р. Избранные задачи олимпиад по информатике. — М.: Тривант, 1997.
7. Тюкачев Н., Свиридов Ю. Delphi 5: Создание мультимедийных приложений. — М.: Нолидж, 2000.
8. Фаронов В.В. Delphi 5. Учебный курс. - М.: Нолидж, 2000.
9. Форум «Delphi для всех» - <http://www.delphi.vov.ru>

## Приложение А. Перечень типовых экзаменационных задач

1. *Линейная структура*: нахождение значения  $P$  – спроса на бытовую технику в  $n$  году в экономической модели:  $P_t = 3,4 + 5 \ln(Z) + e^{0,3t}$ , где  $Z$  – средних доход семьи в месяц.

2. *Ветвление*:

А) нахождение значения  $S$  – стоимости покупки на промежутках:

$$S = \begin{cases} S \text{ не изменяется, при } S < 500(\text{руб.}) \\ S \text{ уменьшается на } 5\% \text{ при } 500 < S < 699(\text{руб.}) \\ S \text{ уменьшается на } 10\% \text{ при } S > 700(\text{руб.}) \end{cases}$$

Б) При наличной сумме  $S$  определить возможность покупки стоимостью  $D$ . При положительном ответе вывести остаток.

В) Флажок. Для числа  $x$  при выборе флажка «Целочисленное деление на 2» вывести результат.

3. *Выбор*:

А) вывод годового % для выделенного вида вклада.

Б) для введенного % банковского вклада выделить радиокнопку ближайшего вида вклада.

4. *Цикл*:

А) расчет суммы вклада на  $t$  лет вперед при  $n$  % годовых. Три вида цикла.

Б) расчет выплат суммы кредита на  $t$  лет при  $n$  % годовых. Три вида цикла.

5. *Строки*:

А) В тексте заменить точки восклицательными знаками;

Б) В тексте заменить слог  $f1$  на слог  $f2$ ;

В) В тексте с позиции  $K1$  до позиции  $K2$  скопировать (удалить) слоги:

Г) В тексте вставить слово  $D$  в конце текста.

Д) В тексте последние 10 символов удалить.

**6. Вектор:**

- А) В векторе (2000г.-2004г.) найти среднее и min значения, разницу между ними.
- Б) В векторе (2000г.-2004г.) найти max-значение. Получить новый вектор, в котором значения ниже max заменить 0.
- В) В векторе (1995г.-2004г.) найти количество лет, значения которых больше значения 1995 г. Вывести найденные года.
- Г) Дан вектор (2001г.-2004г.). Найти сумму значений. Получить новый вектор, как % каждого года от суммы.

**7. Матрицы:**

- А) В матрице (1кв.-4 кв.,2002г.-2004г.) найти сумму значений каждого года.
- Б) В матрице (1кв.-4 кв.,2002г.-2004г.) найти max-значение каждого года.
- В) В матрице (1кв.-4 кв.,2002г.-2004г.) найти среднее и количество значений больше среднего.
- Г) Дана матрица (1кв.-4 кв.,2002г.-2004г.). Получить новую матрицу, в которой значения 1 квартала заменить нулями, а для последующих кварталов найти разницу между данным кварталом и предыдущим.