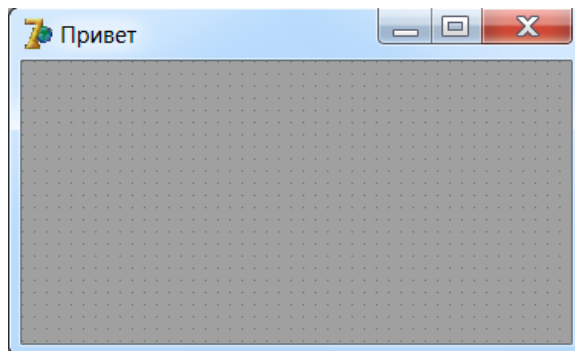



Лабораторная работа 1

Создание простого приложения

1. **Запустите Delphi.** Поработаем с формой в инспекторе объектов (Object Inspector).
2. **Измените** заголовок окна формы с Form1 на **Привет:** для этого свойству **Caption** задайте значение **Привет** (введите с клавиатуры).
3. **Измените** цвет формы на другой: в окне инспектора объектов (Object Inspector) установите для свойства **Color**, например, значение **clBtnShadow**.
4. Задайте свойство **Position = poScreenCenter**. Тогда при запуске приложения форма появится в центре экрана.



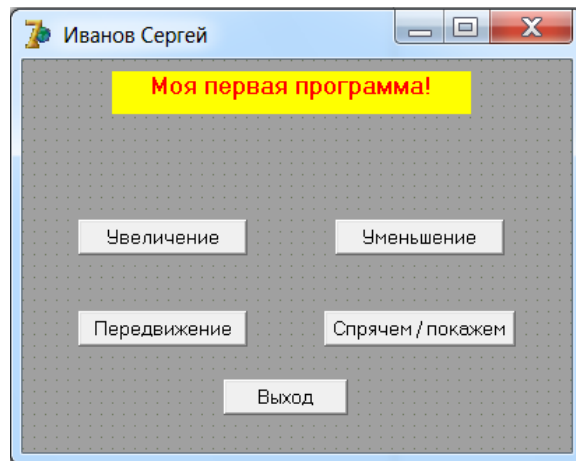
5. **Сохраните** форму и проект на диске:
 - ✓ Выберите меню: **File – Save Project As;**
 - ✓ С помощью Проводника создайте на диске свою папку (можно указать свои инициалы), а в ней создайте папку для этого проекта, например, **Лаб1**.
 - ✓ В окне **Имя Файла** вместо **Unit1** введите **ULab1**, автоматически файлу будет присвоено расширение **.pas**.
 - ✓ Далее в окне **Имя Файла** вместо **Project1** введите **PLab1**, автоматически файлу будет присвоено расширение **.dpr**.
6. **Выполните** приложение:
 - ✓ выберите меню: **Run – Run;**
 - ✓ или нажмите **F9;**
 - ✓ или нажмите кнопку .
7. **Измените** размеры окна с помощью мыши.
8. **Поэкспериментируйте** со стандартными кнопками минимизации и максимизации окна.
9. **Завершите** работу приложения, закрыв его окно.
10. Вы заметите, что в папке с проектом создано приложение **PLab1.exe**
11. **Изменяя** значения свойства **BorderStyle**, запустите проект и наблюдайте, что при этом происходит с формой:
 - ✓ bsDialog;
 - ✓ bsNone;
 - ✓ bsSingl;

- ✓ bsSizeable;
- ✓ bsSizeToolWin;
- ✓ bsToolWindow.

12. **Закройте** проект.

Лабораторная работа 2

Создание простого приложения с использованием кнопок



1. Для создания нового приложения необходимо выполнить команду **File/New Application**.
2. Измените **заголовок** окна формы с Form1 на свою **Фамилию** и **имя**: в окне инспектора объектов (Object Inspector) установите для свойства **Caption** значение, например, **Иванов Сергей**.
3. Поместите объект **TLabel** в окно формы **Form1**:
На вкладке **Standard** Палитры Компонент выберите пиктограмму **Label** и поместите её на форму, этот компонент автоматически получит имя **Label1**.
4. Измените свойства объекта **Label1**:
В окне Инспектора Объектов (Object Inspector) установите следующие значения для свойств объекта:

<i>Объект</i>	<i>Свойство</i>	<i>Значение</i>
<i>Label1</i>	<i>Caption</i>	<i>Моя первая программа!</i>
	<i>Font</i>	<i>12n, красный</i>
	<i>Alignment</i>	<i>taCenter</i>
	<i>Color</i>	<i>clYellow</i>
	<i>AutoSize</i>	<i>False</i>

5. Сохраните форму и проект на диске:
 - ✓ Имя папки **Лаб2**;
 - ✓ Вместо **Unit1** введите **ULab2**;
 - ✓ Вместо **Project1** введите **PLab1**.
6. Поместите объект **Button** на **Form1**:

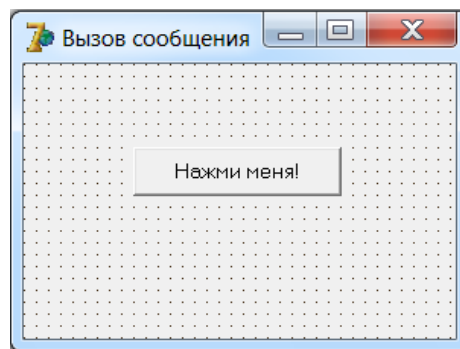
На вкладке **Standard** Палитры Компонент выберите пиктограмму **Button** и поместите её на форму, этот компонент автоматически получит имя **Button1**. Измените его размеры.

7. Установите свойство **Caption** объекта **Button1** в значение «Увеличение».
 8. Напишите код для события **Click** на объекте **Button1**:
 - ✓ Выполните двойной щелчок по объекту **Button1**;
 - ✓ образуется процедура обработки события: *procedure TForm1.Button1Click(Sender: TObject);*
 - ✓ в ней между **Begin** и **End** напишите следующий код:
Label1.Font.Size:=Label1.Font.Size+2;
 9. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Увеличение».
 10. Добавьте кнопку для уменьшения размера шрифта в тексте.
 11. Добавьте кнопку для того, чтобы двигать текст.
Код: *Label1.Left := Label1.Left + 10;*
Label1.Top := Label1.Top + 10;
 12. Добавьте кнопку для управления видимостью текста («спрячем / покажем»).
Код: *Label1.visible := not Label1.visible;*
 13. Добавьте кнопку для выхода из работы программы.
Код: *Close;*
- Выполните проект.

Лабораторная работа 3

Вызов окна сообщения

1. Откройте новый проект.
2. На форму поместите компонент **Button**, изменив его свойство **Caption** на «Нажмите меня!».



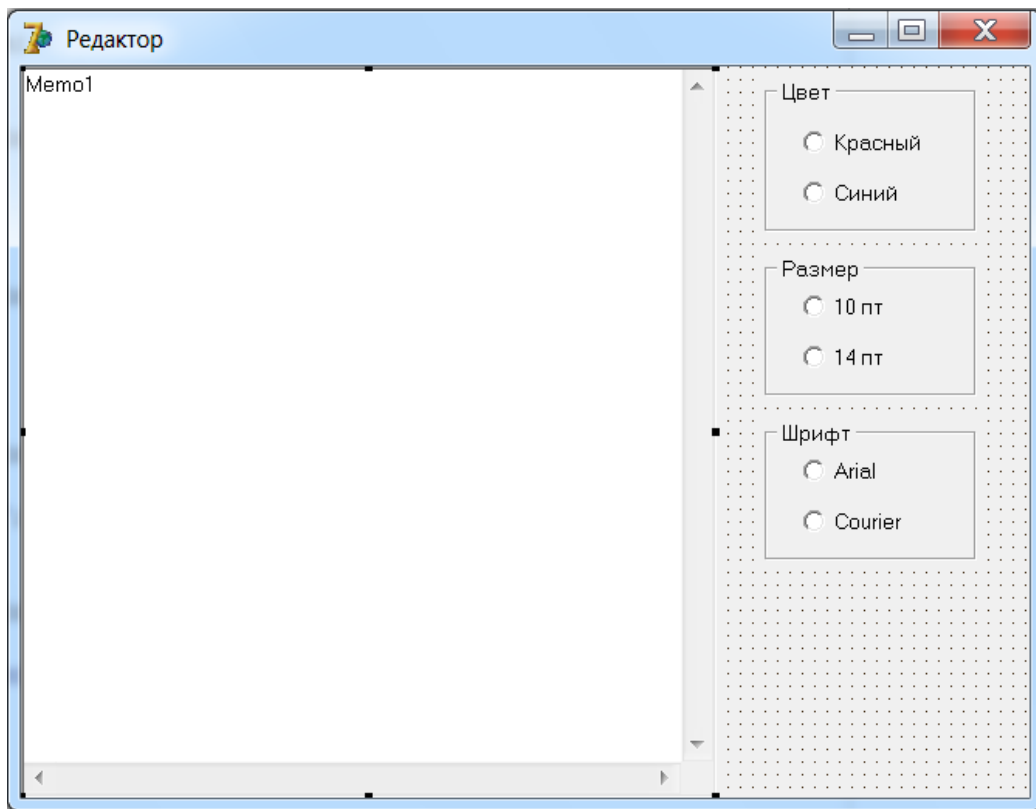
3. Напишите код для события **Click** на объекте **Button1**:
procedure TForm1.Button1Click(Sender: TObject);
begin
MessageDlg('Вы здесь?', mtConfirmation, mbYesNoCancel, 0);
end;

4. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
 5. Измените код для события **Click** на объекте **Button1**:
MessageDlg('Осторожно!', mtWarning, [mbOk], 0);
 6. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
 7. Измените код для события **Click** на объекте **Button1**:
MessageDlg('Ха-ха!', mtError, [mbCancel], 0);
 8. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
 9. Измените код для события **Click** на объекте **Button1**:
MessageDlg('Не забудьте пообедать.', mtInformation, [mbOk], 0);
 10. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
 11. Измените код для события **Click** на объекте **Button1**:
MessageDlg('Вы выполнили действие?', mtConfirmation, [mbYes, mbNo], 0);
 12. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
 13. Измените код для события **Click** на объекте **Button1**:
MessageDlg('Хорошо сидим!', mtCustom, [mbYes], 0);
 14. Выполните приложение. Обратите внимание на то, что происходит при нажатии кнопки с надписью «Нажмите меня!».
- Закройте проект.

Лабораторная работа 4

Создание текстового редактора

1. Вызовите текстовый редактор **Блокнот** и создайте в нем текстовый файл **My_text.txt** с содержанием:
 - Button* - кнопка
 - RadioButton* - переключатель
 - Label* - надпись
 - Edit* - строка текста
 - Memo* - редактор текста
 Сохраните файл в папку **ЛР4**.
2. Начните новый проект и сразу сохраните его в папке **ЛР4**.
3. Свойству **Caption** формы присвойте значение «Редактор».
4. Поместите компонент **Memo** на форму.



5. В окне **Инспектора Объектов** (Object Inspector) установите следующие значения для свойств объекта:

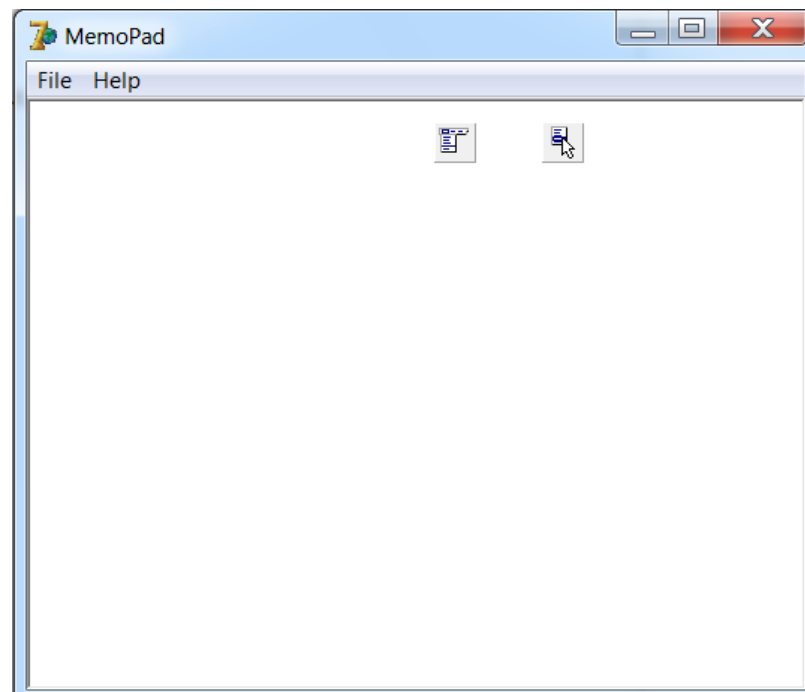
<i>Объект</i>	<i>Свойство</i>	<i>Значение</i>
<i>Мемо</i>	<i>Align</i>	<i>alLeft</i>
	<i>ScrollBars</i>	<i>ssBorth</i>
	<i>ShowHint</i>	<i>True</i>

6. Запишите код для **загрузки** файла при **создании формы**:
 Для создания процедуры обработки события в объекте **Form1** на вкладке **Events** (в Object Inspector) задействуйте событие **OnCreate**:
`procedure TForm1.FormCreate(Sender: TObject);`
`begin`
`Memo1.Lines.LoadFromFile('My_text.txt');`
`end;`
7. Запишите код, позволяющий **сохранить** файл при **закрытии формы**:
`procedure TForm1.FormClose...`
`begin`
`Memo1.Lines.SaveToFile('My_text.txt');`
`end;`
8. Запустите приложение. Добавить что-нибудь в появившийся текст.
9. Закройте приложение.
10. Запустите её еще раз и убедитесь, что загружается откорректированный текст.
11. Поместите на форму объект **GroupBox**. Свойству **Caption** присвойте значение **Цвет**.



12. Добавьте на объект **GroupBox1** два переключателя **RadioButton** для выбора **цвета** шрифта. Свойству **Caption** присвойте значения: *красный, синий*.
13. Двойным щелчком мыши по объекту **RadioButton1** создайте событие **Click** и напишите код для него:
Memo1.Font.Color:=clRed;
14. Напишите код для события **Click** на объекте **RadioButton2**:
Memo1.Font.Color:=clBlue;
15. Добавить группу переключателей для выбора **размера** шрифта.
16. Напишите код для события **Click** на объекте **RadioButton3**:
Memo1.Font.Size:=10;
17. Напишите код для события **Click** на объекте **RadioButton4**:
Memo1.Font.Size:=14;
18. Добавить группу переключателей для выбора **вида** шрифта.
19. Напишите код для события **Click** на объекте **RadioButton5**:
Memo1.Font.Name:='Arial';
20. Напишите код для события **Click** на объекте **RadioButton6**:
Memo1.Font.Name:='Courier';
21. Выполните приложение.

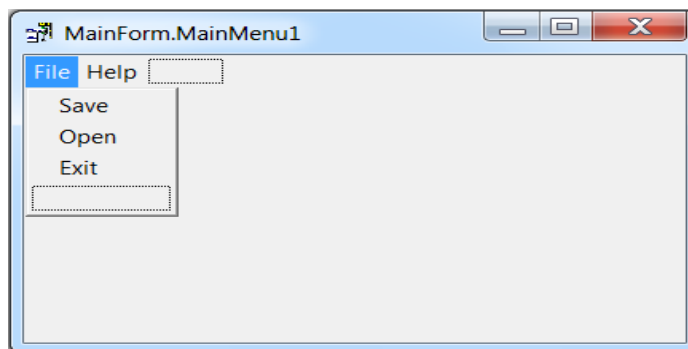
Лабораторная работа 5

Создание приложения MemoPad с использованием меню

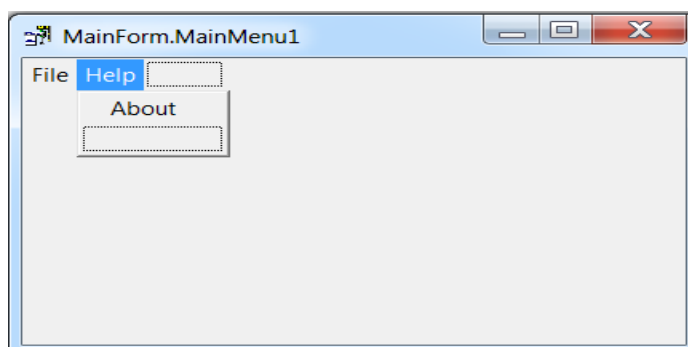


1. Откройте новое приложение.
2. Сохраните его на диске:
 - ✓ Создайте и откройте новую папку под именем **MemoPad**;
 - ✓ Сохраните **Unit1** под именем **Main, Project1 – MemoPad**.

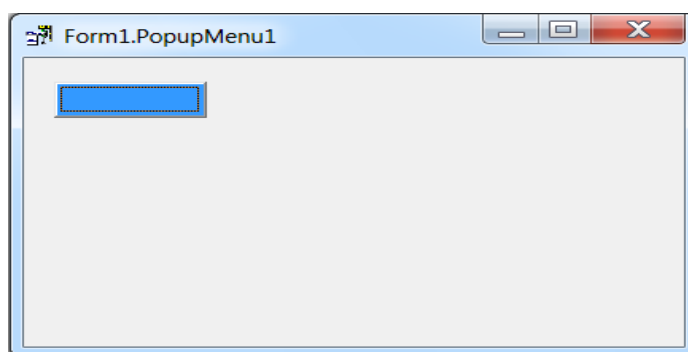
3. Измените значение свойства **Name** формы на **MainForm**.
4. Свойству **Caption** формы присвойте значение **MemoPad**
5. На вкладке **Standart** выберите объекты **MainMenu**  и **PopupMenu**  поместите их на форму.
6. Дважды щелкните на пиктограмме **MainMenu1**. Откроется окно **меню дизайнер (Menu Designer)**.
7. Для создания меню **Файл**:
 - ✓ В поле **Caption** введите **File**, нажмите **Enter**;
 - ✓ В следующем ниже— **Save**
 - ✓ Для других введите: **Open**; **Exit**.



8. Перейдите в поле справа от опции **File**. Свойству **Caption** присвойте значение **Help**.
9. Для пункта подменю введите **About**.



10. Закройте окно меню дизайнера.
11. Посмотрите, как выглядит меню на форме.
12. Дважды щелкните на пиктограмме **PopupMenu1**. Откроется окно **меню дизайнер**.



13. Введите в появившееся меню **Open, Save, Exit**.

14. Закройте дизайнер.
15. Укажите, как использовать всплывающее меню:
 - ✓ Щелкните на свободном месте формы, в верхней части окна Object Inspector должна появиться запись: «MainForm1: TMainForm1»;
 - ✓ Задайте свойству **PopupMenu** значение **PopupMenu1**.
16. Установить на экранную форму объект **Memo**. Чтобы он занял всё свободное пространство на форме, укажите ему свойство **Align = alClient**.
17. В инспекторе объектов выберите свойство **Lines** и щелкните по кнопке **построителя** (кнопка с многоточием).
18. В открывшемся диалоговом окне **String List Editor** (однострочный текстовый редактор) удалите надпись: «Memo1». Нажмите **Ok**.
19. Напишите код для пункта меню **File – Open**:
 - ✓ Щелкните по **Open**;
 - ✓ В процедуре TMainForm.Open1Click между словами **Begin** и **End** вставьте код:
`Memo1.Lines.LoadFromFile('memos.txt');`
20. Напишите код для пункта меню **File – Save**:
 - ✓ Щелкните по **Save**;
 - ✓ В процедуре TMainForm.Save1Click между словами **Begin** и **End** вставьте код:
`Memo1.Lines.SaveToFile('memos.txt');`
21. Напишите код для пункта меню **File – Exit**:
 - ✓ Щелкните по **Exit**;
 - ✓ В процедуре TMainForm.Exit1Click между словами **Begin** и **End** вставьте код:
`Close;`
22. Напишите код для пункта меню **Help – About**:
 - ✓ Щелкните по **About**;
 - ✓ В процедуре TMainForm.About1Click между словами **Begin** и **End** вставьте код:
`MessageDlg('MemoPad'#13#10'@2021 by FIO'#13#10, mtInformation, [mbOk], 0);`
(Вместо FIO напишите свои Ф.И.О.)

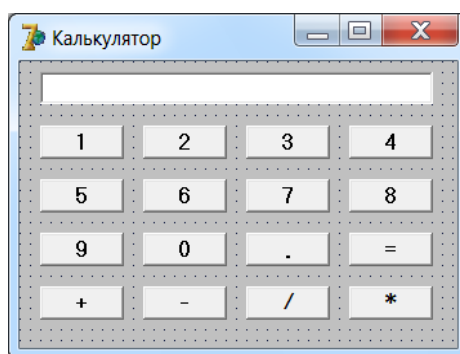
Приступите к программированию команд всплывающего меню.

23. Дважды щелкните на объекте **PopupMenu1**. Откроется окно **Меню дизайнер**.
24. Щелкните на корешке вкладки **Object Inspector**.
25. Выделите команду **Open** в меню. Установите для события **OnClick** событие **Open1.Click**.
26. Выделите команду **Save** в меню. Установите для события **OnClick** событие **Save1.Click**.
27. Выделите команду **Exit** в меню. Установите для события **OnClick** событие **Exit1.Click**.
28. Выполните приложение.

Лабораторная работа 6

Создание приложения «Калькулятор»

1. Откройте новый проект.
2. Создайте интерфейс как показано на рисунке. При этом в Инспекторе объектов укажите для формы свойство **BorderStyle = bsSingle** – чтобы размеры формы не могли изменяться, и желательно выключить кнопку режима Maximize – в **BorderIcons**.



3. Сохраните проект в папке **Калькулятор**. Имя файла **Unit1** замените на **UCalc, Project1** – на **PCalc**.
4. Добавьте глобальные переменные в раздел описания глобальных переменных (после **var Form1: TForm1**):

```
i, p, r: real; //они будут сохранять числа введенные пользователем  
mode: integer; //режим калькулятора (умножить, разделить и т.д)
```

5. Выделите, используя клавишу **Shift**, десять цифровых кнопок (0 ÷ 9).
6. Перейдите на вкладку **Events**.
7. Щелкните два раза напротив **OnClick**.
8. Введите код:

```
Edit1.Text:=Edit1.Text + (Sender as TButton).Caption;
```

9. Напишите код для события **Click** для объекта **Button11** (где *точка*):

```
Edit1.Text:=Edit1.Text + ',';
```

10. Напишите код для события **Click** для объектов **Button13** – **Button16**. Переменная **mode** будет содержать числовое значение действия: 1 – прибавление, 2 – вычитание, 3 – деление, 4 – умножение.

✓ Код для **Button13**:

```
Procedure TForm1.Button13Click(Sender: TObject);  
Begin  
    i:=StrToFloat(Edit1.Text);  
    mode:=1;  
    Edit1.Text:=' '  
end;
```

✓ Код для **Button14**:

```
Procedure TForm1.Button14Click(Sender: TObject);  
Begin
```

```
        i:=StrToFloat(Edit1.Text);
        mode:=2;
        Edit1.Text:=' ';
    end;
```

✓ Код для **Button15**:

```
Procedure TForm1.Button15Click(Sender: TObject);
Begin
    i:=StrToFloat(Edit1.Text);
    mode:=3;
    Edit1.Text:=' ';
end;
```

✓ Код для **Button16**:

```
Procedure TForm1.Button16Click(Sender: TObject);
Begin
    i:=StrToFloat(Edit1.Text);
    mode:=4;
    Edit1.Text:=' ';
end;
```

11. Напишите код для события **Click** для объекта **Button12** (кнопки с обозначением « \Leftrightarrow »):

```
Procedure TForm1.Button12Click(Sender: TObject);
Begin
    Case mode of
        1: begin
            p:=StrToFloat(Edit1.Text);
            r:=i+p;
            Edit1.Text:=FloatToStr(r);
            end;
        2: begin
            p:=StrToFloat(Edit1.Text);
            r:=i-p;
            Edit1.Text:=FloatToStr(r);
            end;
        3: begin
            p:=StrToFloat(Edit1.Text);
            r:=i/p;
            Edit1.Text:=FloatToStr(r);
            end;
        4: begin
            p:=StrToFloat(Edit1.Text);
            r:=i*p;
            Edit1.Text:=FloatToStr(r);
            end;
    end;
end;
```

12. Выполните приложение.

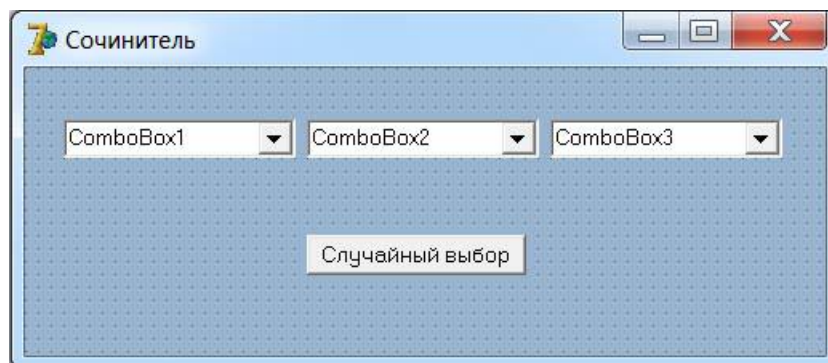
13. Закройте проект.

Попробуйте самостоятельно добавить кнопку «C» для очистки результата.

Лабораторная работа 7

Случайный выбор из списка

Цель работы – создать программу, которая может случайным образом составлять предложения из трёх слов, используя три списка слов.



Для этого будем использовать три текстовых файла, их можно приготовить в «Блокноте» – стандартном приложении Windows. В начале работы программы будем загружать списки из файлов.

Например, в файле nouns.txt – существительные:

Кошка
Змея
Кузнечик
Дельфин
Черепашка
Ласточка

в файле adverbs.txt – наречия:

быстро
высоко
медленно
сильно
хорошо
плохо

в файле verbs.txt – глаголы:

плавает
бегает
летает
ползает
прыгает
прячется

Описание разработки программы

1. Откройте новый проект.
2. Разместите на форме три компонента **ComboBox** и командную кнопку **Button**.

3. В Object Inspector выполните следующие настройки формы **Form1**: в свойствах (Properties) укажите имя формы (Caption) «Сочинитель», BorderStyle = bsSizeToolWin, Position = poScreenCenter.

4. Для компонентов **ComboBox** свойство Style выберите значение csDropDownList из раскрывающегося списка.

5. Во вкладке Events для **Form1** задействуйте событие OnCreate, где следует организовать чтение текстовых файлов и заполнение списков в **ComboBox**.

```
procedure TForm1.LoadList(fName:string; ComboBox:TComboBox);
var
  Txt: TextFile;
  s: string;
begin
  ComboBox.Clear;
  ComboBox.ItemIndex:= -1; // выбор отсутствует
  AssignFile(Txt, fName);
  try // начало защищённого блока
    reset(Txt); // открываем файл для чтения
    while not eof(Txt) do begin
      readln(Txt, s);
      if s <> '' then
        ComboBox.Items.Add(s)
    end;
    CloseFile(Txt);
  except // не удалось открыть файл - возможно, его нет
    ShowMessage('Проблемы при открытии файла ' + fName);
  end; // конец защищённого блока
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  LoadList('nouns.txt', ComboBox1);
  LoadList('adverbs.txt', ComboBox2);
  LoadList('verbs.txt', ComboBox3);
end;
```

Поскольку здесь выполняется три раза практически одно и то же действие, то удобнее это оформить в виде общей процедуры LoadList. Заголовок процедуры следует внести в **interface** в описание класса **TForm1** в раздел **private**.

6. Для кнопки **Button1** создадим обработчик события OnClick:

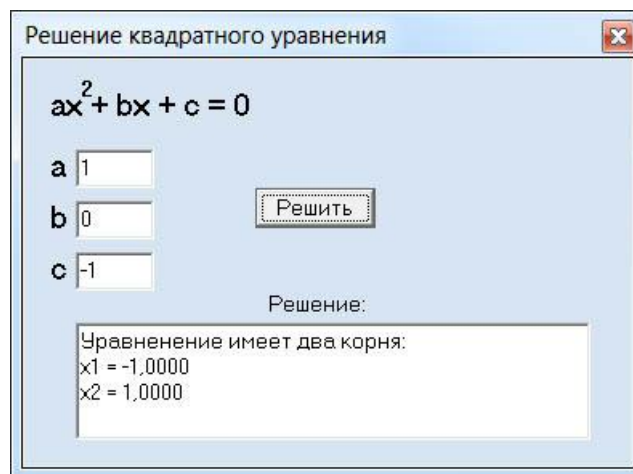
```
procedure TForm1.Button1Click(Sender: TObject);
begin
  // установка датчика случайных чисел в произвольное начальное состояние
  Randomize;
  // для каждого ComboBox выбирается случайная позиция
  ComboBox1.ItemIndex:= Random(ComboBox1.Items.Count);
  ComboBox2.ItemIndex:= Random(ComboBox2.Items.Count);
  ComboBox3.ItemIndex:= Random(ComboBox3.Items.Count);
end;
```

Лабораторная работа 8

Решение квадратного уравнения

Создать приложение для решения квадратного уравнения вида

$$ax^2 + bx + c = 0$$



1. Поместите на форму три компонента **Edit**, через которые программа будет получать значения коэффициентов уравнения **a**, **b** и **c**. Текст решения будет выводиться в поле компонента **Memo**. Потребуется ещё кнопка **Button** и несколько компонентов **Label**.

2. Установить свойства формы, как и в предыдущей задаче.

3. Запуск решения – через обработку события **OnClick** кнопки **Button1**. Данные брать из полей **Edit**, при этом учитывать, что мы имеем дело с вещественными числами. Т.е. например,

```
a := StrToFloat(Edit1.Text);
```

Из школьного курса алгебры берём алгоритм решения уравнения:

дискриминант $d = b^2 - 4ac$

если $d < 0$, то действительных корней нет;

если $d = 0$, то решение имеет один корень: $x = -b/(2a)$;

если $d > 0$, то решение имеет два корня:

$$x_1 = (-b - \sqrt{d})/(2a), \quad x_2 = (-b + \sqrt{d})/(2a).$$

По данному алгоритму составляем программу решения уравнения. Функция извлечения квадратного корня `sqrt(d)`. Полученный текст решения можно в несколько строк внести в компонент **Memo1** – пример:

```
Memo1.Clear; // очищаем окно
```

```
Memo1.Lines.Add(s); // добавляем строку s.
```

Используйте для вещественного **N** форматированный вывод, например:

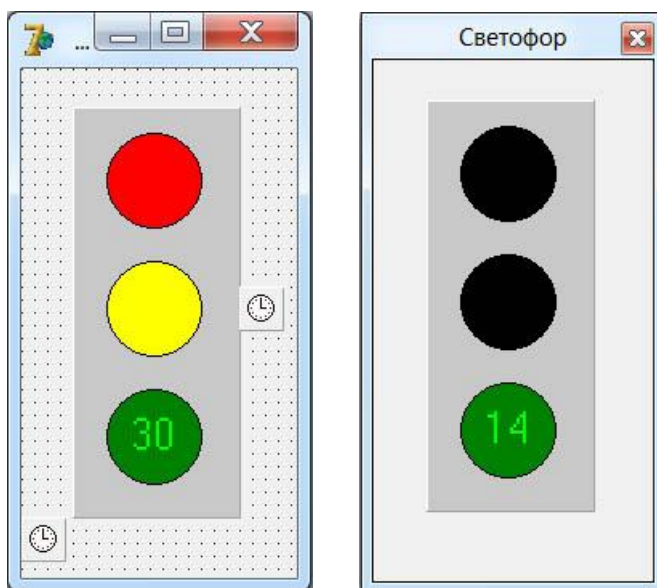
```
Memo1.Lines.Add(format('x1 = %.4f', [N]));
```

Добавьте в алгоритм анализ нулевых значений коэффициентов – с соответствующим сообщением.

Лабораторная работа 9

Светофор

Создать приложение, изображающее работающий светофор.



То есть, при запуске программы горит только красный сигнал, затем через некоторое время включается мигающий жёлтый, а красный гаснет. Зелёная ячейка дополнительно показывает оставшееся время (до включения жёлтого сигнала) – всё, как в реальном светофоре.

Описание разработки программы

1. Поместите на форму панель – сам светофор, её можно выделить цветом – сделать темнее, чем форма. Форму настройте так же, как и в предыдущей задаче. Подберите размеры.

2. На панели разместите три компонента **Shape** (вкладка **Additional**) одинакового размера квадратной формы. У каждого свойство **Shape** задайте **stCircle**. Это будут секции светофора. Верхней секции зададим красный цвет, а остальным – в программе (используем свойство **Brush.Color**).

3. На зелёную секцию поместите компонент **Label**, чтобы показывать секунды, оставшиеся до окончания зелёного сигнала. Постарайтесь отрегулировать размеры компонента и установить его на центр круга. При этом и текст – тоже выровнять по центру **Label** (используйте **Alignment**). А чтобы размер **Label** сам не изменялся при работе программы, установите свойство **AutoSize = False**.

4. Вам понадобятся два компонента **Timer** (вкладка **System**) – один будет работать на весь светофор, а второй – для жёлтого мигающего сигнала. Первому ставим **Interval = 1000**, второму – **100**.

5. Переходим к тексту программы. Нам понадобятся следующие константы и переменные:

Periods – сколько гореть красному, жёлтому и зелёному сигналам соответственно (в секундах). Эти значения будем задавать константой в программе.

```
const
  Periods: array[1..3] of integer = (10, 2, 15);
  PYellow = 700; // миллисекунд горит жёлтый, 300 - чёрный

var
  NCell: byte; // 1..3 - номер цвета: красный, жёлтый, зелёный
  T: byte; // время, отсчитываемое таймером
  Step: (Up, Down); // После жёлтого идти вниз или вверх
  TYellow: integer; // время жёлтого сигнала
```

Начальные настройки программы делаем в событии Form1.OnCreate:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Shape2.Brush.Color:= clBlack;
  Shape3.Brush.Color:= clBlack;
  Label1.Color:= clBlack;
  Label1.Caption:= '';
  T:=0;
  NCell:=1;
  Step:= Down;
  Timer1.Enabled:= true;
end;
```

Основная часть работы программы – это обработчик срабатывания таймера Timer1:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  inc(T);
  case NCell of
    1: if T >= Periods[1] then begin
      T:=0;
      NCell:= 2;
      Step:= Down;
      Shape1.Brush.Color:= clBlack;
      Shape2.Brush.Color:= clYellow;
      TYellow:= 0;
      Timer2.Enabled:= true;
    end;
    2: if T > Periods[2] then begin
      T:=0;
      if Step = Down then begin
        NCell:= 3;
        Shape2.Brush.Color:= clBlack;
        Shape3.Brush.Color:= clGreen;
        Label1.Caption:= IntToStr(Periods[3]);
        Label1.Color:= clGreen;
      end
      else begin
        NCell:= 1;
        Shape2.Brush.Color:= clBlack;
        Shape1.Brush.Color:= clRed;
      end
    end
  end
```

```

    else begin
        Shape2.Brush.Color:= clYellow;
        TYellow:= 0;
        Timer2.Enabled:= true;
    end;
3: if T >= Periods[3] then begin
    T:=0;
    NCell:= 2;
    Step:= Up;
    Shape3.Brush.Color:= clBlack;
    Shape2.Brush.Color:= clYellow;
    Label1.Caption:= '';
    Label1.Color:= clBlack;
    TYellow:= 0;
    Timer2.Enabled:= true;
end
else
    Label1.Caption:= IntToStr(Periods[3]-T);
end;
end;

```

Обработчик срабатывания таймера жёлтого сигнала:

```

procedure TForm1.Timer2Timer(Sender: TObject);
begin
    inc(TYellow, Timer2.Interval);
    if TYellow >= PYellow then begin
        Shape2.Brush.Color:= clBlack;
        Timer2.Enabled:= false;
    end;
end;

```

Попробуйте самостоятельно добавить визуальный отсчёт секунд для красного сигнала.