

РАЗРАБОТКА ПРОГРАММЫ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ НА ЯЗЫКЕ DELPHI

Колесникова С.А.

г. Тверь, Тверской лицей, 11 класс

Руководитель: Наумова А.И., г. Тверь, Тверской лицей, учитель информатики высшей категории

В данной работе представлен материал по разработке проекта «Программа компьютерного тестирования».

Практической целью данной работы является разработка программы тестирования полученных знаний на языке визуального программирования Delphi.

При выполнении этого задания из трёх предложенных вариантов ответа необходимо выбрать один верный. Выбор осуществляется при помощи переключателей.

Для выполнения поставленной цели необходимо решить следующие задачи:

Для подготовки текстового файла подобрать практический материал по теме: «Визуальный язык программирования Delphi».

Разработать проект: «Программа компьютерного тестирования», выполнив последовательно следующие шаги:

- создать текстовый файл;
- создать графический интерфейс;
- создать событийные процедуры;
- компилировать проект в приложение и запустить его на выполнение.

Таким образом, последовательно представлен материал по проектированию: разработка формальной и компьютерной модели с использованием языка программирования Delphi и приведены конкретные примеры компьютерного эксперимента.

Тема достаточно актуальна и представляет практический интерес для учащихся в профильных классах при проверке полученных знаний.

Основная часть.

Разработка программы компьютерного тестирования на языке Delphi

1. Описание формальной модели проекта

Проект позволяет контролировать полученные знания учащихся с помощью компьютерного тестирования.

Основные характеристики проекта:

- проект обеспечивает работу с тестом произвольной длины (нет ограничений на количество вопросов и ответов в тесте);
- вопрос может сопровождать иллюстрацией;

- для каждого вопроса может быть до 3-х возможных вариантов ответов со своей оценкой в баллах;

- результат тестирования может быть отнесен к одному из 4-х уровней, например, «отлично», «хорошо», «удовлетворительно» или «плохо»;

- тест представляет собой текстовый файл, созданный в текстовом редакторе Блокнот;

- программа инвариантна к различным тестам (имя файла теста указывается как параметр при вызове программы);

- в программе не обеспечена возможность возврата к предыдущему вопросу. Если вопрос предложен, то на него должен быть дан ответ [3].

2. Компьютерная модель проекта

2.1. Создать текстовый файл

Тест – последовательность вопросов и соответствующих им вариантов ответов, должен находиться в текстовом файле. Количество вопросов теста не ограничено. Программа тестирования должна получать имя файла теста из командной строки. Ниже приведен пример теста. Первая строка – это название теста. За ней следует описание 4-х уровней оценки. Для каждого уровня задается сообщение и количество правильных ответов, необходимых для его достижения. Далее следуют вопросы. Каждый вопрос представляет собой текст вопроса, за которым следуют три варианта ответа. После каждого ответа указывается количество баллов, добавляемое к общей оценке, в случае выбора этого варианта ответа (правильный ответ к общей сумме добавляет 1, за выбор неправильного – 0).

Тест представляет собой последовательность вопросов, на которые испытуемый должен ответить путем выбора правильного ответа из нескольких предложенных вариантов.

Файл теста состоит из 3 разделов:

- раздел заголовка;
- раздел оценок;
- раздел вопросов

Файл теста начинается заголовком, который содержит общую информацию о тесте,

например, о его назначении. Заголовок может состоять из нескольких строк.

За заголовком следует раздел оценок, в котором приводятся названия оценочных уровней и количество баллов, необходимое для достижения этих уровней. Название уровня должно располагаться в одной строке. За разделом оценок следует раздел вопросов теста. Каждый вопрос начинается текстом вопроса, за которым может следовать имя файла иллюстрации, размещаемое в отдельной строке. После вопроса следуют альтернативные ответы. Текст альтернативного ответа может занимать несколько строк. В строке, следующей за текстом ответа, располагается оценка (количество баллов) за выбор этого ответа [3].

Пример текстового файла/

Программирование на языке Delphi
 Вы правильно ответили на все вопросы.
 Оценка – ОТЛИЧНО!
 10
 На некоторые вопросы Вы ответили неверно. Оценка – ХОРОШО.
 8
 По количеству правильных ответов оценка – УДОВЛЕТВОРИТЕЛЬНО.
 6
 Вы плохо подготовились к испытанию.
 Оценка – ПЛОХО!
 5
 Назначение библиотечного модуля StdCtrls
 Работа с математическими функциями
 0
 Дополнительные свойства командной кнопки
 1
 Работа в графическом режиме
 0
 Назначение объекта Button
 Кнопка для компиляции программы
 0
 Кнопка для запуска программы
 0
 Командная кнопка
 1
 Назначение объекта Label
 Для вывода текста в текстовое окно
 0
 Для вывода текста в графическое окно
 0
 Для вывода текста на форму
 1
 Назначение объекта RadioButton
 Переключатель
 1
 Список
 0
 Метка

0
 Графический интерфейс проекта – это Форма, на которой размещены управляющие элементы
 1
 Окно Инспектора объектов
 0
 Панель управляющих элементов
 0
 Что преобразует функция StrToInt()
 Строковый тип в целое число
 1
 Строковый тип в число с плавающей точкой
 0
 Символьный тип в число с плавающей точкой
 0
 Для чего используется процедура ShowMessage()?
 Для ввода данных
 0
 Для вывода данных
 0
 Для вывода окна с сообщением и командной кнопкой ОК
 1
 В каком текстовом редакторе необходимо создать тест?
 WordPad
 0
 Блокнот
 1
 MS Word
 0
 Как создать исполняемый файл?
 Выполнить команды: Проект – Компилировать (Project – Compile)
 1
 Выполнить команды: Файл – Сохранить Как...(File – Save As...)
 0
 Выполнить команды: Файл – Сохранить Проект Как...(File – Save Project As...)
 0
 Как запустить исполняемый файл?
 Дважды щёлкнуть по исполняемому файлу
 0
 Дважды щёлкнуть по Ярлыку исполняемого файла
 1
 Войти в приложение Delphi и выполнить команды: Выполнить – Выполнить (Run – Run)
 0
 Сохранить текстовый файл с именем INFORM в отдельную папку.

2.2. Создать графический интерфейс

Поместить на форму (вкладка Standart):

• командную кнопку Button1 (ОК) для подтверждения выбора альтернативного ответа и перехода к следующему вопросу;

• надпись Label1 (буква А) для вывода текста вопроса, начальной информации о тесте и результатов тестирования;

• три надписи Label2, Label3, Label4 для вывода текста альтернативных ответов;

• три переключателя RadioButton1, RadioButton2, RadioButton3 (круг) для выбора варианта ответа.

В диалоговом окне Инспектор объектов форме Form1 свойству Caption дать значение – Проверка знаний; командной кнопке Button1 свойству Caption дать значение – Дальше.

Чтобы задать значения шрифта, необходимо выделить объект, дважды щёлкнуть по сложному свойству Font со знаком «плюс», потом щёлкнуть на командную кнопку с тремя точками. В появившемся диалогом окне для Label1: шрифт – MS Sans Serif, начертание – полужирный, размер – 12 (рис. 1); для Label2 – Label4: начертание – обычный, размер – 12; для Button1: шрифт – MS Sans Serif, начертание – обычный, размер – 12; для всех меток свойству AutoSize дать значение False, свойствам ParentFont и WordWrap – True.

Окончательный вид формы разрабатываемого проекта представлен на (рис. 2) [1, 3, 4].

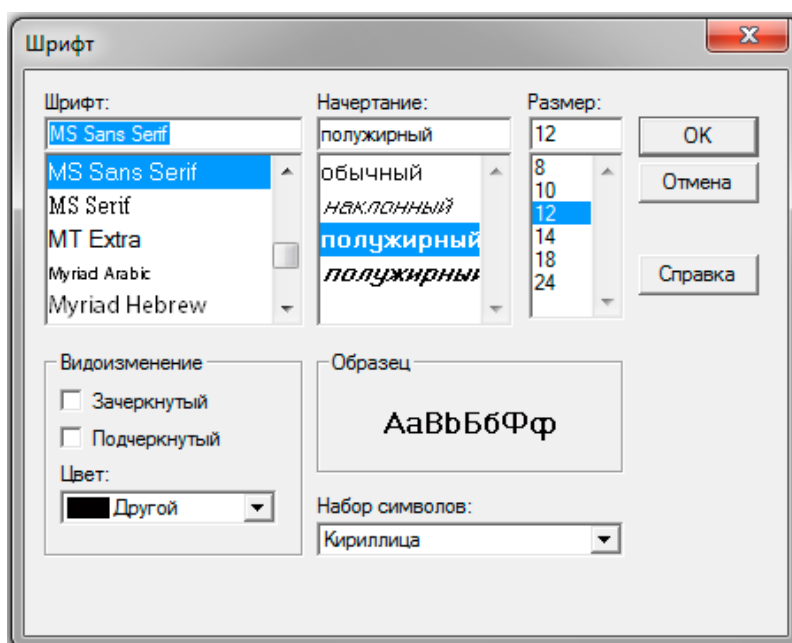


Рис. 1. Для метки Label1 выбрать шрифт и задать его параметры

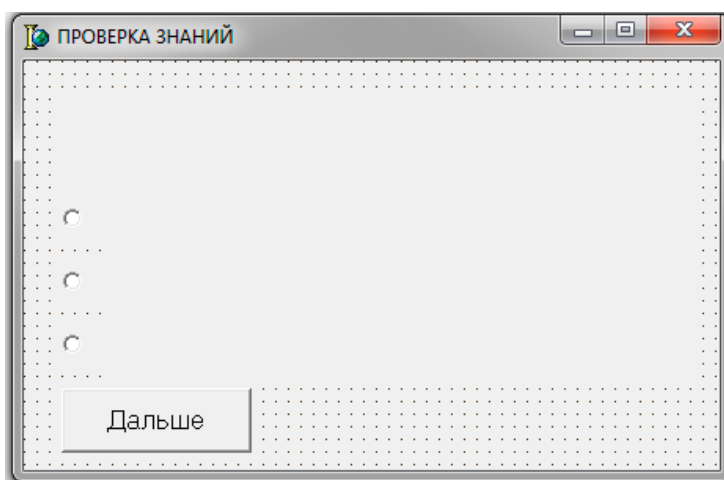


Рис. 2. Форма разрабатываемого проекта

2.3. Создать событийные процедуры

После создания формы в окно редактора кода следует поместить описание глобальных переменных программы и процедур общего назначения. Для использования дополнительных свойств командной кнопки Button1 в раздел интерфейса (interface) после слова uses в перечень библиотечных модулей добавить модуль *StdCtrls*. Затем можно приступить к созданию программного кода с описанием процедур обработки событий:

```
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure   RadioButton1Click(Sender:
TObject);
procedure   RadioButton2Click(Sender:
TObject);
procedure   RadioButton3Click(Sender:
TObject);
```

Программный код проекта в Приложении 1.

Сохранить проект в отдельную папку вместе с текстовым файлом INFORM. Для сохранения проекта из меню Файл (File) выбрать команду Сохранить проект как... (Save Project As...). При первом сохранении откроется диалоговое окно Сохранить модуль (Save Unit1 As). В поле Имя

файла ввести значение TEST01 и щёлкнуть по кнопке Сохранить. После сохранения файла модуля проекта открывается диалоговое окно Сохранить проект (Save Project As). В поле Имя файла следует ввести имя проекта TEST [1, 2].

3. Компьютерный эксперимент

Запустить проект на выполнение можно как через систему объектно-ориентированного программирования Delphi, так и с помощью исполняемого файла с расширением .exe.

Запуск проекта из приложения Delphi

После запуска Delphi из меню Файл (File) выбрать команду Открыть (Open). В открывшемся окне выбрать путь к папке с проектом, в которой находится выполняемый и текстовый файлы. Щёлкните по имени файла TEST и кнопке Открыть (Open). На экране высветится форма с управляющими элементами и программный код.

При запуске программного кода из Delphi имя файла inform.txt надо ввести в поле диалогового окна, которое доступно при выборе в меню Выполнить (Run) пункта команды Параметры (Parameters) <OK> (рис. 3).

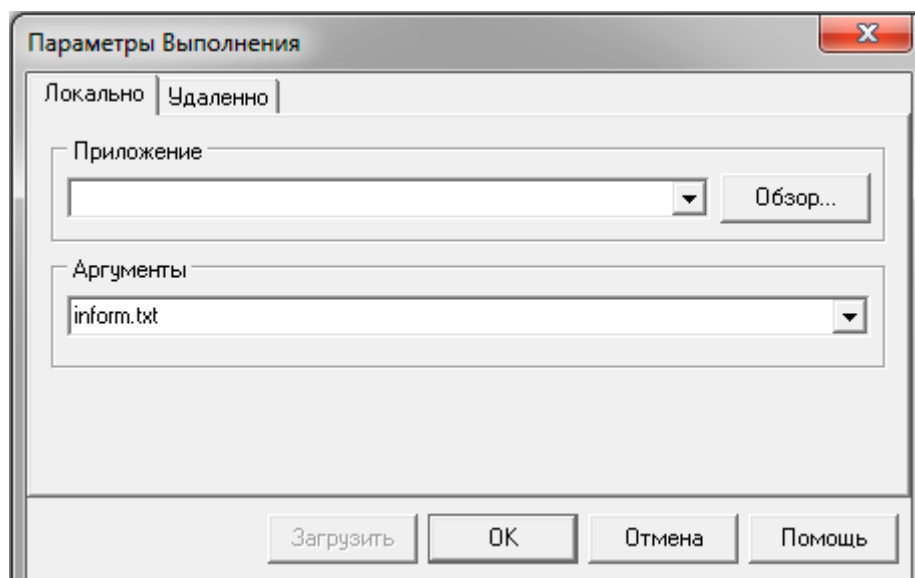


Рис. 3. Ввод имени текстового файла

Для выполнения проекта необходимо выбрать команду Выполнить (Run) в пункте меню Выполнить (Run).

Запуск проекта с помощью исполняемого файла TEST.exe

1. Папку с проектом и текстовым файлом разместить на диске D:
2. На рабочем столе исполняемому файлу TEST.exe создать Ярлык
3. Щёлкнуть правой кнопкой по Ярлыку и войти в Свойства
4. В строке Объект через пробел рядом с исполняемым файлом TEST.exe в двойных кавычках написать имя текстового фай-

ла INFORM.txt и нажать на кнопку <OK>. Вся строка будет выглядеть следующим образом: D:\Delphi\TEST.exe «INFORM.txt» (рис. 4) [1, 3].

5. Для запуска программы дважды щёлкнуть по ярлыку.

После запуска программы одним из описанных выше способов на экране появится форма с конкретным вопросом (вопрос 1) и тремя вариантами ответа. Название формы Form1 – Проверка знаний (рис. 2) меняется на Программирование на языке Delphi. Командная кнопка Button1 (Дальше) недоступна (рис. 5).

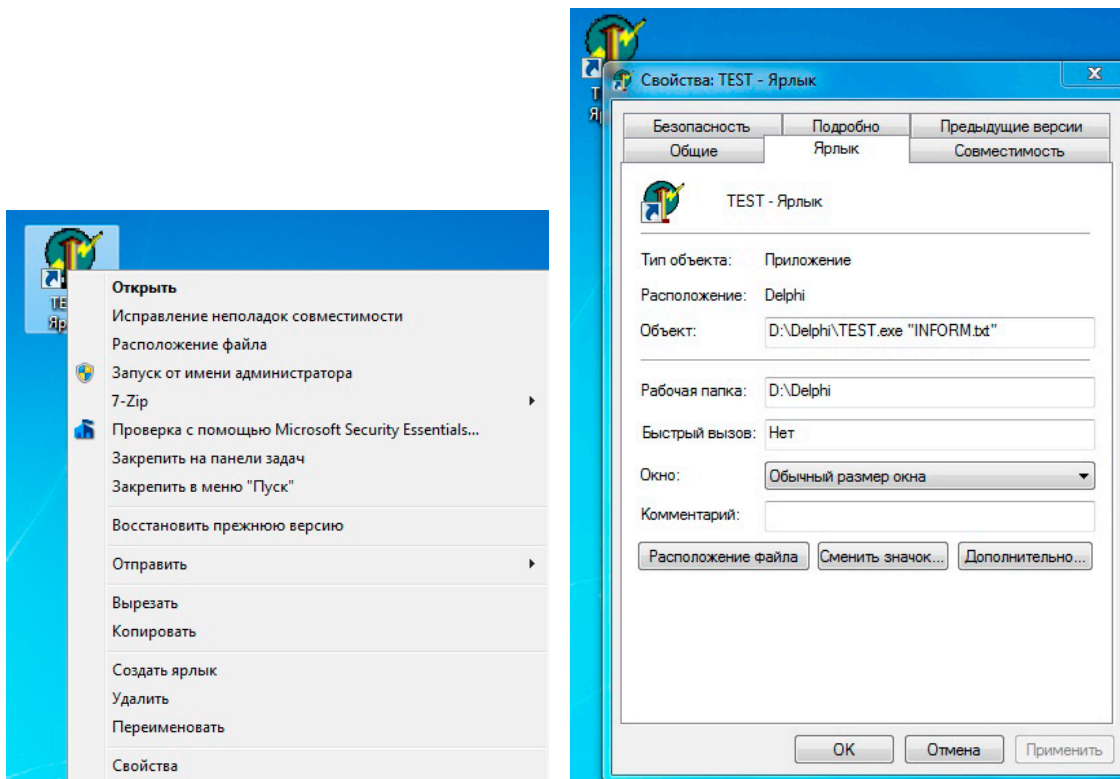


Рис. 4. Путь исполняемого файла в Ярлыке приложения (.exe)

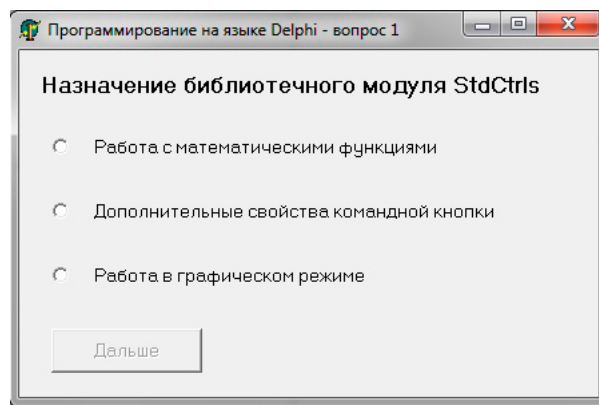


Рис. 5. Первый шаг выполнения программы

6. Приступить к прохождению тестирования.

Файл открывается с помощью инструкции открытия файла для чтения. После успешного открытия файла вызывается процедура, которая считывает информацию из файла и выводит её присваиванием прочитанного текста свойству Caption поля метки Label1. Затем вызывается процедура, которая считывает из файла теста информацию об уровнях оценки. Эта процедура заполняет соответствующие массивы.

Для начала тестирования и открытия командной кнопки Button1 (Дальше) выбрать один из вариантов ответа (рис. 6).

Для обработки события переключателей в программе используется общая процедура, получающая объект, на котором произошло событие. Сравнивая полученное значение с именами объектов-переключателей,

процедура присваивает значение глобальной переменной для увеличения набранной суммы баллов. Кроме того эта процедура делает доступной кнопку Button1, которая после вывода очередного вопроса вновь недоступна.

Таким образом, командную кнопку Button1 в программе можно использовать для начала тестирования, перехода к следующему вопросу и для завершения программы [1, 3].

Далее испытуемый последовательно отвечает на все вопросы, которые находятся в текстовом файле INFORM.txt. После выбора ответа на последний вопрос при нажатии на кнопку Дальше, на форму выводится сообщение о результатах тестирования в зависимости от набранных баллов (рис. 7 – рис. 10) и командная кнопка Button1 меняет своё название – Завершить.

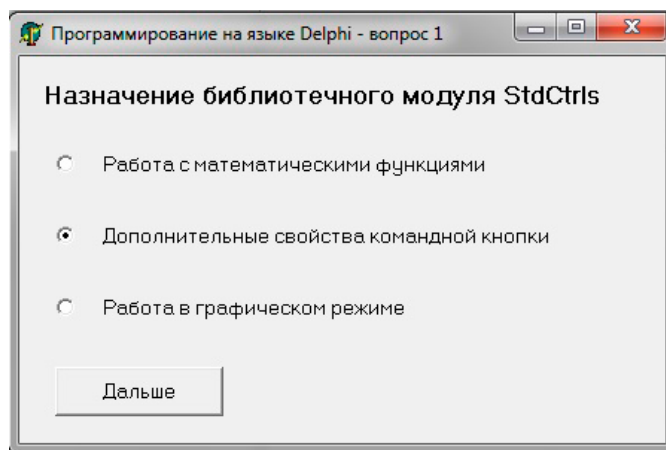


Рис. 6. Открытие командной кнопки

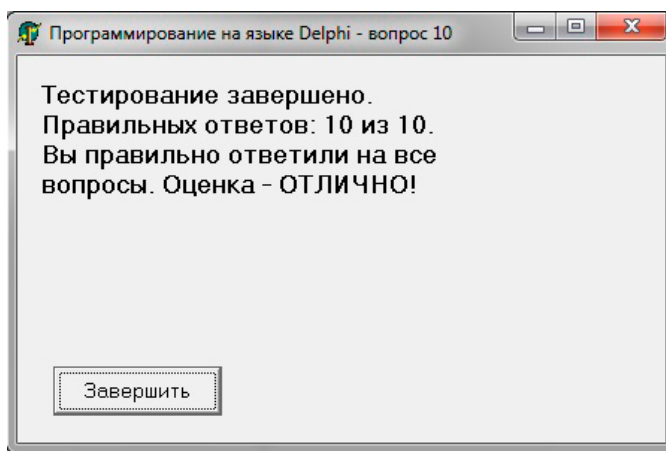


Рис. 7. Количество набранных баллов соответствует оценке – ОТЛИЧНО

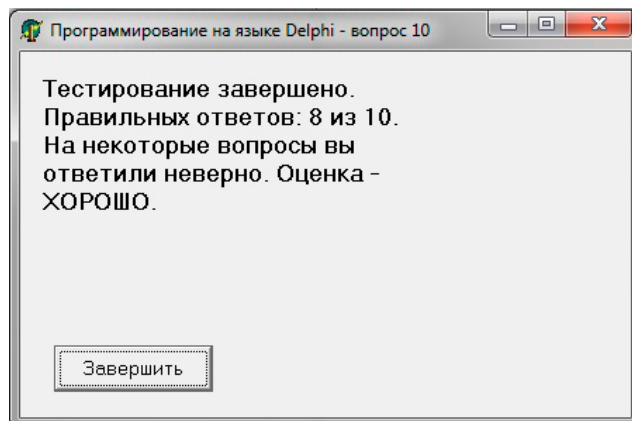


Рис. 8. Количество набранных баллов соответствует оценке – ХОРОШО

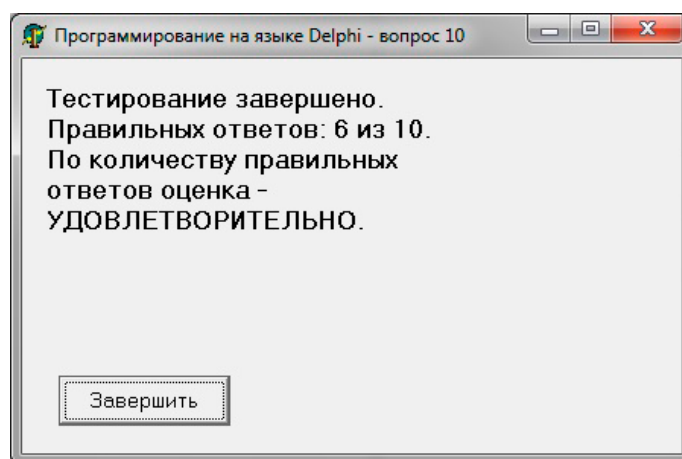


Рис. 9. Количество набранных баллов соответствует оценке – УДОВЛЕТВОРИТЕЛЬНО

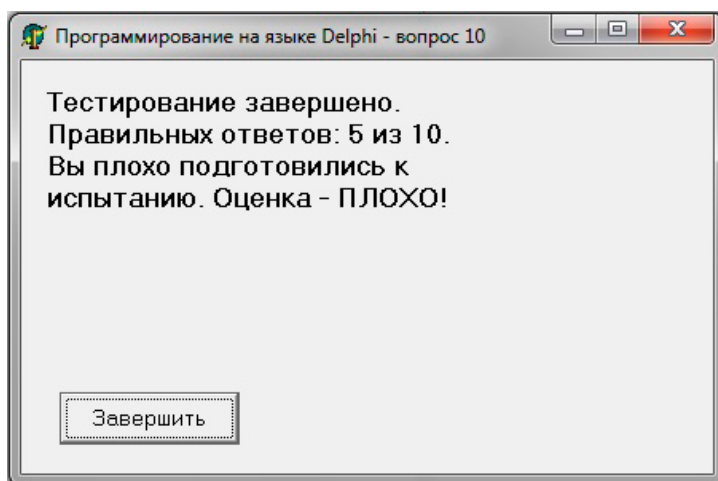


Рис. 10. Количество набранных баллов соответствует оценке – ПЛОХО

В конце тестирования щёлкнуть по командной кнопке Завершить.

Заключение

Данная программа предназначена для обобщения и систематизации имеющихся знаний у учащихся по программированию на языке Delphi. Специальная литература, а также стандартное оборудование кабинета информатики являются базой для проведения контроля, который является одной из основных форм проверки и оценки знаний. Такая форма контроля за уровнем достижений в рамках предлагаемого курса строится на том основании, что выпускники должны быть включены в них субъектно. Для этого разработаны альтернативные программы по различным типам заданий.

При желании представленную разработку можно использовать по другим темам и предметам, предварительно подготовив соответствующие текстовые файлы.

Ожидаемый результат:

В итоге тестирования учащиеся самостоятельно получают возможность

Знать:

основные темы, входящие в учебный курс по программированию на языке Delphi;

Уметь:

разрабатывать проекты.

У педагогов появляется возможность проанализировать полученные результаты.

Такие работы проводились в 11–м физико-математическом классе Тверского лицея. Качество выполненных работ достаточно высокое.

Список литературы

1. Система объектно-ориентированного программирования Delphi.
2. Кульгин Н. Delphi в задачах и примерах: Сборник задач. – СПб., 2008.
3. Кульгин Н. Учебник по программированию в среде Delphi для студентов и школьников старших классов. – СПб., 2001.
4. Угринович Н., Элективный курс: учебное пособие для учащихся старших классов информационно-технологического, физико-математического и естественно-научного профилей. – М., 2007.

Приложение 1

Программный код проекта

```
unit TEST01;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Button1: TButton;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
RadioButton3: TRadioButton;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure RadioButton1Click(Sender: TObject);
procedure RadioButton2Click(Sender: TObject);
procedure RadioButton3Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
var
f: TextFile; // файл теста (вопросы и варианты ответов)
title: string; // название теста
```



```

nq: integer; // количество вопросов в тесте
right: integer; // количество правильных ответов
// количество правильных ответов, необходимых для достижения уровня
level: array[1..10] of integer;
mes: array[1..4] of string; // сообщение об оценке
buf: string;
// читает из файла вопрос, варианты ответа и выводит их в поля формы
function NextQw : boolean;
begin
if not EOF(f) then
begin
nq:= nq + 1; // счетчик общего количества вопросов
Form1.Caption := Title + ' – вопрос ' + IntToStr(nq);
// прочитать и вывести вопрос
Readln(f,buf);
Form1.Label1.Caption := buf;
// прочитать и вывести варианты ответов
// 1-й вариант
Readln(f,buf); // прочитать 1-й вариант ответа
Form1.Label2.Caption := buf;
Readln(f,buf); // оценка за выбор этого ответа (1 – правильно, 0 – нет)
Form1.RadioButton1.Tag := StrToInt(buf);
// 2-й вариант
Readln(f,buf);
Form1.Label3.Caption := buf;
Readln(f,buf);
Form1.RadioButton2.Tag := StrToInt(buf);
// 3-й вариант
Readln(f,buf);
Form1.Label4.Caption := buf;
Readln(f,buf);
Form1.RadioButton3.Tag := StrToInt(buf);
// кнопка «Дальше» не доступна, пока не выбран один из вариантов ответа
Form1.Button1.Enabled := False;
// ни один из переключателей не выбран
Form1.RadioButton1.Checked := False; Form1.RadioButton2.Checked := False;
Form1.RadioButton3.Checked := False; NextQw := TRUE;
end
else NextQw := FALSE;
end;
// событие FormCreate возникает в момент создания формы
procedure TForm1.FormCreate(Sender: TObject);
var
i: integer;
fname: string;
begin
// При запуске программного кода из Delphi имя файла inform.txt надо
// ввести в поле диалогового окна, которое доступно при выборе в меню
// Run (Выполнить) пункта команды Parameters (Параметры) <OK>
fname := ParamStr(1); // взять имя файла теста из командной строки
if fname = '' then
begin
ShowMessage('В командной строке запуска программы' + #13+
'надо указать файл теста. ');
Application.Terminate; // завершить программу
end;
AssignFile(f,fname);
// в процессе открытия файла возможны ошибки
try
Reset(f); // эта инструкция может вызвать ошибку
except

```

```

on EInOutError do
begin
ShowMessage('Ошибка обращения к файлу теста: ' + fname);
Application.Terminate; // завершить программу
end;
end;
// файл теста успешно открыт
// прочитать название теста – первая строка файла
Readln(f,buf);
title := buf;
// прочитать оценки и комментарии
for i:=1 to 4 do
begin
Readln(f,buf);
mes[i] := buf;
Readln(f,buf);
level[i] := StrToInt(buf);
end;
right := 0; // правильных ответов
nq := 0; // всего вопросов
NextQW; // прочитать и вывести первый вопрос
end;
// щелчок на кнопке «Дальше»
procedure TForm1.Button1Click(Sender: TObject);
var
buf: string;
i: integer;
begin
if Button1.Caption = 'Завершить' then Close;
// добавим оценку за выбранный вариант ответа
// оценка находится в свойстве Button.Tag
// Button.Tag = 1 – ответ правильный, 0 – нет
if RadioButton1.Checked then right := right + RadioButton1.Tag;
if RadioButton2.Checked then right := right + RadioButton2.Tag;
if RadioButton3.Checked then right := right + RadioButton3.Tag;
// вывести следующий вопрос
// NextQW читает и выводит вопрос
// NextQw = FALSE, если в файле теста вопросов больше нет
if not NextQW then
begin
// здесь значение NextQw = FALSE
Button1.Caption := 'Завершить';
// скрыть переключатели и поля меток
RadioButton1.Visible := False; RadioButton2.Visible := False;
RadioButton3.Visible := False; Label2.Visible := False;
Label3.Visible := False; Label4.Visible := False;
buf := 'Тестирование завершено.' + #13 +
'Правильных ответов: ' + IntToStr(right) +
' из ' + IntToStr(nq) + '.' + #13;
// выставить оценку
// right – кол-во правильных ответов
i:=1; // номер уровня
while (right < level[i]) and (i < 4) do
inc(i);
buf := buf + mes[i];
Label1.AutoSize := TRUE;
Label1.Caption := buf;
end;
end;
// щелчок на переключателе выбора первого варианта ответа
procedure TForm1.RadioButton1Click(Sender: TObject);

```

```
begin
Button1.Enabled := True;           // кнопка Далее теперь доступна
end;
procedure TForm1.RadioButton2Click(Sender: TObject);
begin
Button1.Enabled := True;
end;
procedure TForm1.RadioButton3Click(Sender: TObject);
begin
Button1.Enabled := True;
end;
end.
```