

Цикл презентаций «ООП на Delphi» посвящен объектно – ориентированному программированию с использованием одной из самых распространенных систем быстрой разработки приложений – Delphi

Используя данный учебный курс, можно самостоятельно овладеть основами объектно – ориентированного программирования на Delphi. Для расширения Ваших знаний к курсу приложен ряд учебных пособий и справочников по Delphi

Цикл содержит 13 презентаций:

ООП на Delphi – 1: Знакомство с системой программирования Borland Delphi. Объекты (компоненты) и их свойства и методы

ООП на Delphi – 2: Первая программа на Delphi, сохранение и компиляция

ООП на Delphi – 3: Программное изменение свойств объектов

ООП на Delphi – 4: Условия в Delphi. Создание простого теста

ООП на Delphi – 5: Элементы ввода и вывода информации

ООП на Delphi – 6: Заставка программы и элемент таймер

ООП на Delphi – 7: Программируем свою игрушку

ООП на Delphi – 8: Меню программы, диалоги

ООП на Delphi – 9: Создаем свой текстовый редактор

ООП на Delphi – 10: Базы данных на Delphi

ООП на Delphi – 11: Калькулятор на Delphi

ООП на Delphi – 12: Создаем обучающе - тестирующую систему

ООП на Delphi – 13: Графика на Delphi

Delphi использует язык программирования Объект Паскаль, поэтому лучше сначала изучить обычный Паскаль и поработать в ТурбоПаскале, а затем и переходить к Delphi – перейти будет очень просто, т.к синтаксис языка остается неизменным.

Изучение ООП на Delphi желательно проводить в старших профильных классах – количество часов, отводимое на информатику там вполне достаточно для освоения основ ООП на Delphi

Объектно –
ориентированное
программирование на

Borland®

DELPHI - 4

DELPHI - 4

На этом уроке:

Мы научимся создавать и проверять условия, пользоваться компонентами **Radio Button** и **Checkbox** и составим простейшую тестирующую программу

Вопросы:

1. Компоненты Delphi **Radio Button** и **Checkbox**
2. Создание простейшей тестирующей программы

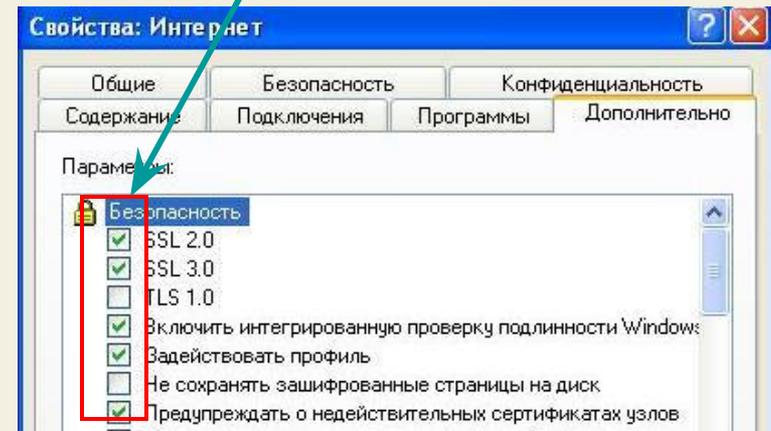
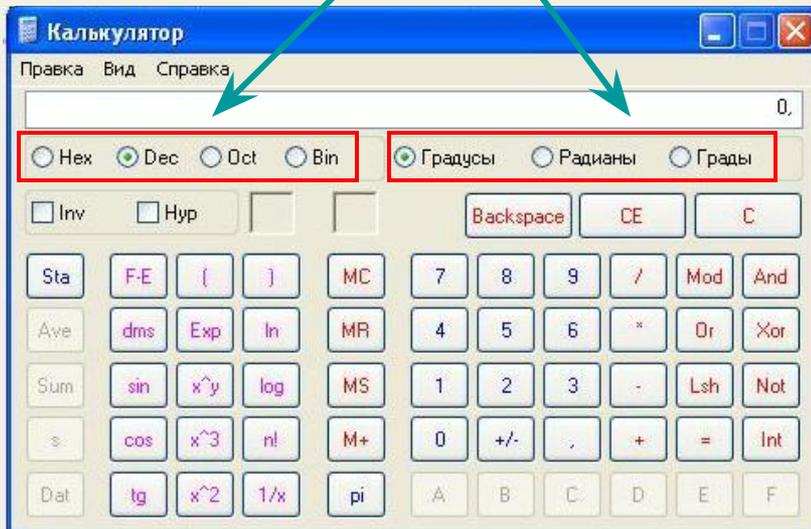
1. Компоненты Delphi Radio Button и Checkbox

Компоненты **Radio Button** и **Check Box** позволяют сделать выбор из нескольких условий, причем **Radio Button** позволяет сделать единственный выбор из многих условий, а **Check Box** – множественный выбор

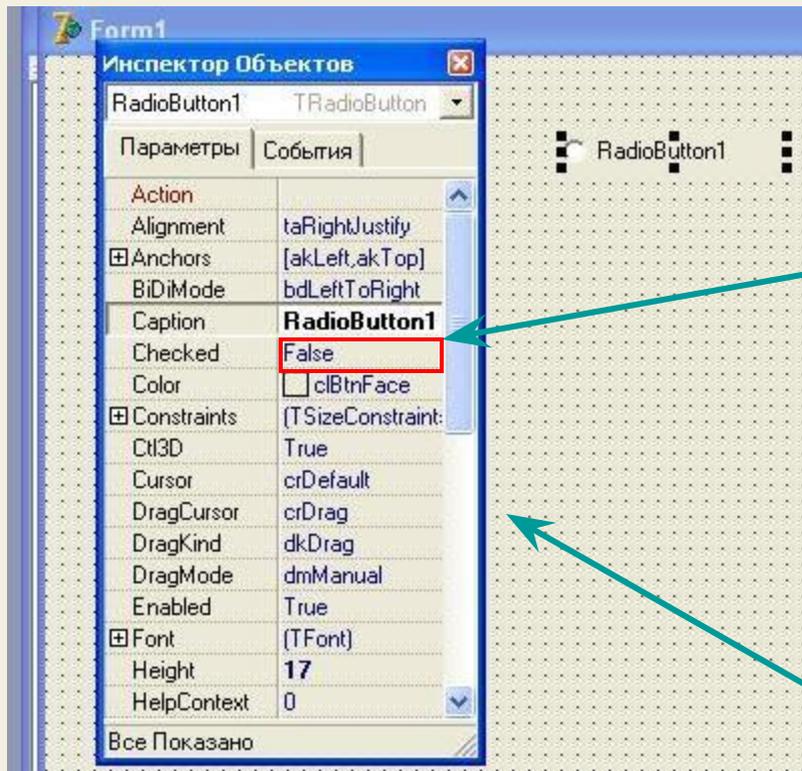
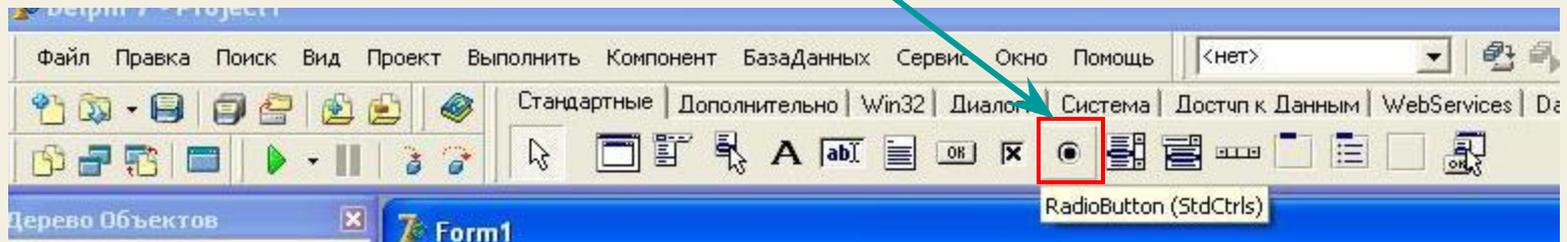
С этими компонентами мы постоянно встречаемся, работая в операционной системе Windows:

Radio Button - ы для выбора одной из нескольких систем исчисления, а также одной из мер измерения угла

Check Box - ы для выбора сразу нескольких условий из многих



Компонент Radio Button



Radio Button

может иметь только 2 значения:

False (ложно)

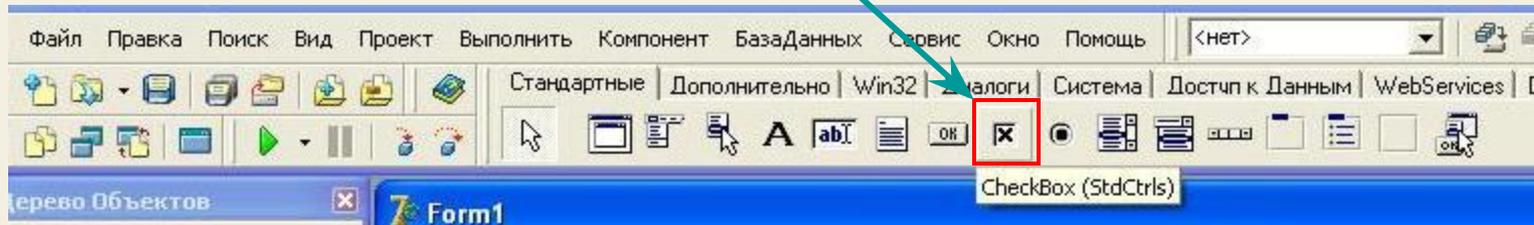
True (истинно)

Кроме этого у радиокнопки есть множество других свойств: можно изменить надпись на кнопке, ее цвет, положение, доступность ...



Посмотрите внимательно набор свойств компонента Radio Button в инспекторе объектов

Компонент Check Box

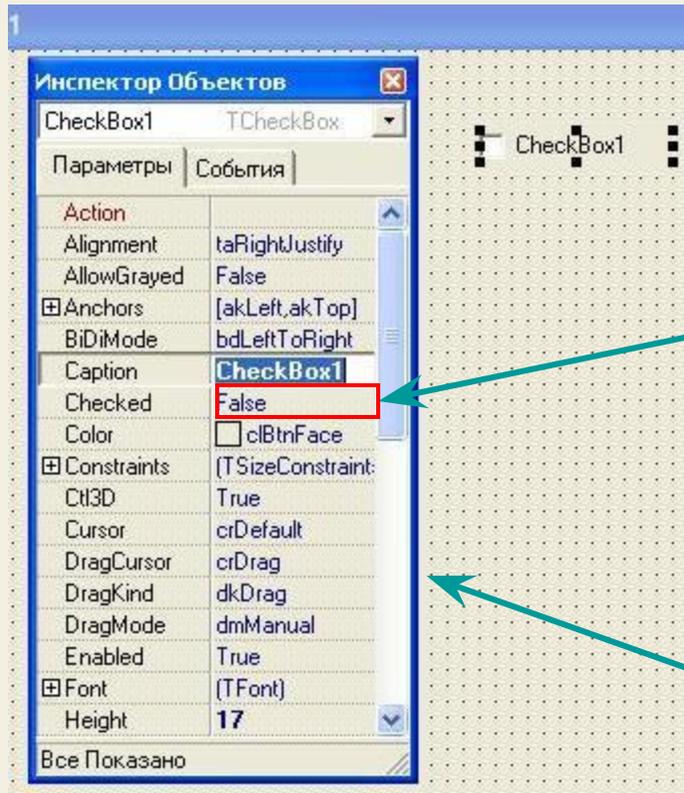


Check Box

может иметь только 2 значения:

False (ложно)

True (истинно)



Кроме этого у **Check Box**-а есть множество других свойств: можно изменить надпись, цвет, положение, доступность ...

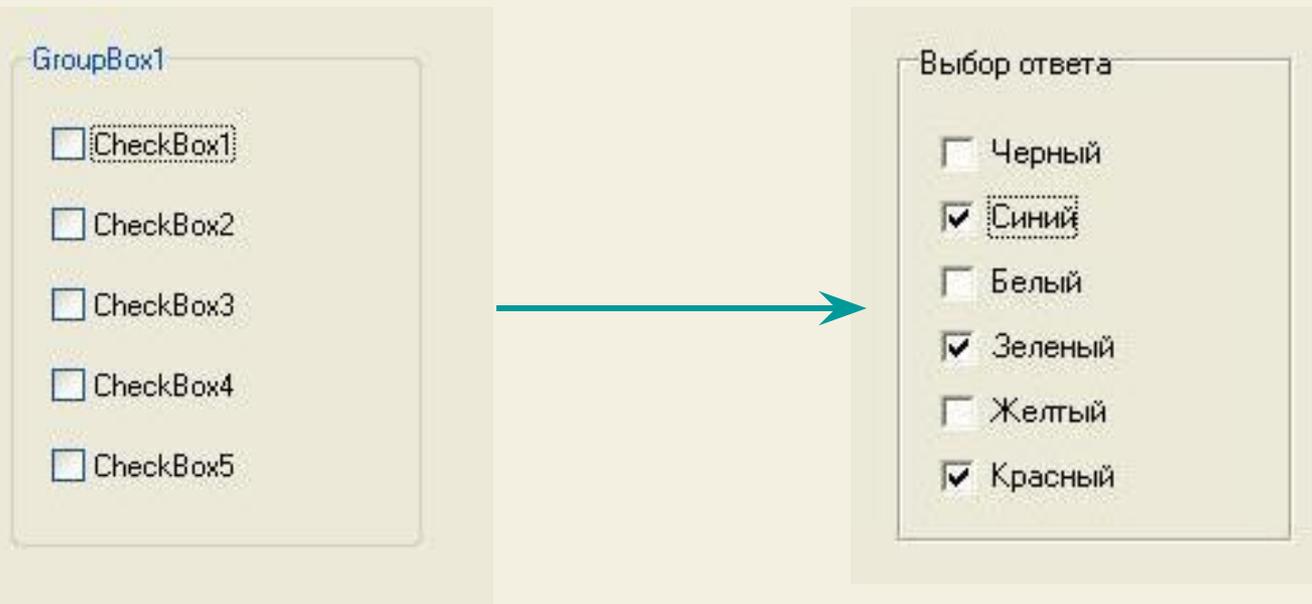


Посмотрите внимательно набор свойств компонента Check Box в инспекторе объектов

Для объединения **Radio Button** и **Check Box** при размещении на форме удобно и красиво применять компоненты **Group Box** и **Radio Group**

Использование **Group Box**:

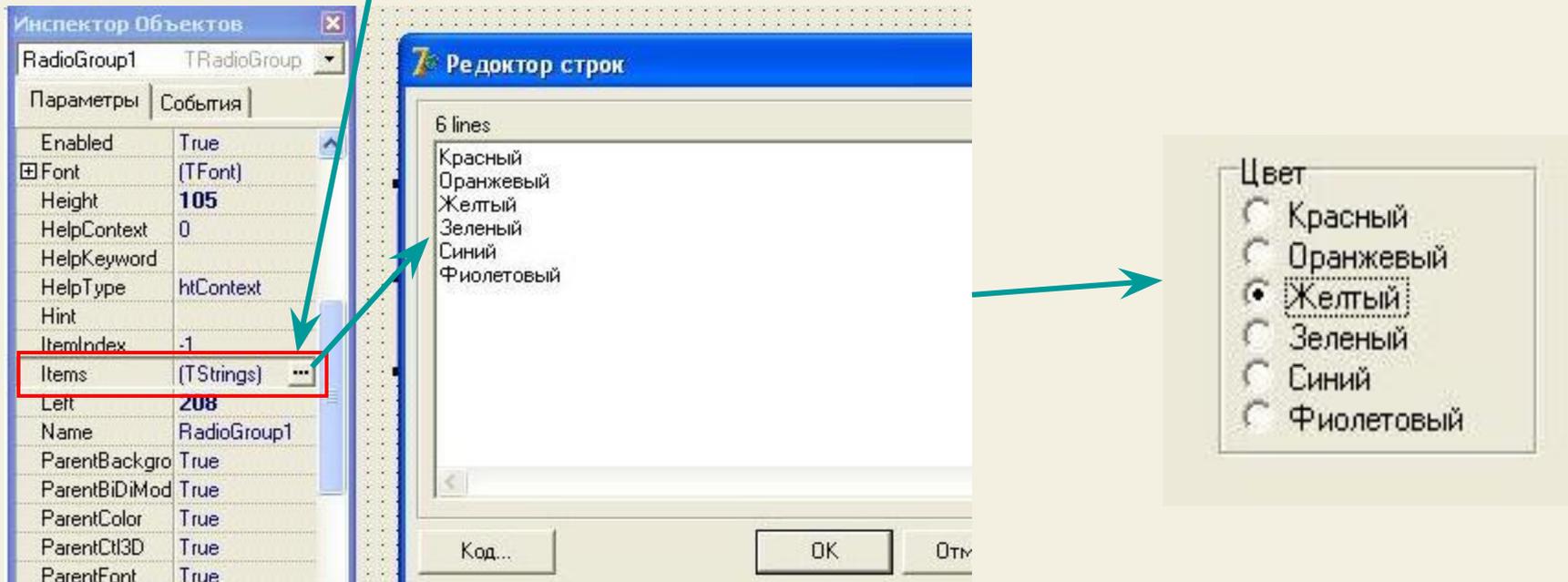
- Помещаем компонент **Group Box** на форму
- Помещаем на него необходимое количество компонент (например **Check Box**, а можно и других)
- В свойстве **Caption** изменяем надписи



В результате мы получаем группу независимых переключателей, где значение True (или False) может иметь сразу несколько переключателей – т.е. мы можем осуществить **множественный выбор**

Использование Radio Group:

- Помещаем компонент **Radio Group** на форму
- Раскрываем свойство **Items** этого компонента и заполняем строки для будущих зависимых переключателей, нажимая Enter после каждой строки
- В свойстве **Caption** компонента изменяем заголовок



В результате мы получаем группу зависимых переключателей, где значение True может иметь только один из всех переключателей, т.е. мы можем осуществить **единственный выбор**

2. Создание простого теста

А сейчас давайте попробуем, используя компоненты **Radio Button и **Check Box** создать простой тест с минимальными требованиями к программе (обойдемся пока приложением из одной формы, без регистрации, отдельных файлов тестов, картинок и пр.) – сделаем это по шагам**

ШАГ 1

Конечно, сначала нужно составить тест на бумаге. Для примера возьмем тест из 5 вопросов, причем первые три вопроса с единственным выбором, а последние два – с множественным:

с единственным выбором

1	Верно ли, что вся информация в компьютере хранится в двоичном коде	Только числовая
		Только числовая и текстовая
		Любая информация
		Вся информация за исключением видео
2	Наименьшая единица информации называется	байт
		бит
		бод
		бит/с
3	Элементарной базой первых компьютеров являлись	микросхемы
		дискретные полупроводниковые элементы
		радиолампы
		транзисторы

с множественным выбором

4	Назовите устройства вывода информации	Монитор
		мышь
		клавиатура
		принтер
5	Назовите системы программирования	Delphi
		Visual Basic
		Microsoft
		Turbo Pascal

ШАГ 2

Сейчас давайте определим требования к программе :

Для первой тестирующей программы мы ограничимся только следующими функциями:

- Подсчет числа верных ответов
- Подсчет % верных ответов
- Вывод результата после окончания теста
- Блокировка возврата к предыдущим вопросам
- Использование 4 вариантов выбора ответа
- Использование заданий как с единственным, так и множественным выбором ответа
- После каждого ответа программа должна оповещать нас о правильности нашего выбора

И на этом функциональность программы пока ограничим

ШАГ 3

Следующим шагом будет разработка внешнего вида приложения и определение компонент, которых мы будем использовать для ввода и вывода информации

The screenshot shows a Windows Forms application window titled "Form1" with a standard XP-style title bar. The form is laid out on a dotted grid background and contains five grouped boxes and one edit field:

- GroupBox1:** Contains Label1, a RadioGroup1 with four radio buttons, Label6, and Button1.
- GroupBox2:** Contains Label2, a RadioGroup2 with four radio buttons, Label7, and Button2.
- GroupBox3:** Contains Label3, a RadioGroup3 with four radio buttons, Label8, and Button3.
- GroupBox4:** Contains Label4, four checkboxes (CheckBox1 to CheckBox4), Label9, and Button4.
- GroupBox5:** Contains Label5, four checkboxes (CheckBox5 to CheckBox8), Label10, and Button5.
- Edit1:** A text input field located below GroupBox5, with Button6 positioned below it.

Размещаем компонент Group Box для каждого вопроса свой

Размещаем Label для вывода текста вопроса

Размещаем Radio Group для выбора на 4 варианта ответов

ШАГ 3

Следующим шагом будет разработка внешнего вида приложения и определение компонент, которых мы будем использовать для ввода и вывода информации

The screenshot shows a Windows Forms application window titled "Form1" with a standard XP-style title bar. The form is laid out on a dotted grid background and contains five distinct groups of controls:

- GroupBox1:** Contains a "Label1", a "RadioGroup1" with five radio buttons, a "Label6" (highlighted with a red box), and a "Button1" (also highlighted with a red box).
- GroupBox2:** Contains a "Label2", a "RadioGroup2" with five radio buttons, a "Label7", and a "Button2".
- GroupBox3:** Contains a "Label3", a "RadioGroup3" with five radio buttons, a "Label8", and a "Button3".
- GroupBox4:** Contains a "Label4", four checkboxes labeled "CheckBox1" through "CheckBox4", a "Label9", and a "Button4".
- GroupBox5:** Contains a "Label5", four checkboxes labeled "CheckBox5" through "CheckBox8", a "Label10", and a "Button5".

At the bottom center of the form, there is a large empty rectangular area and a "Button6".

Здесь помещаем Label для визуального оформления (верно/ошибно) и ответный

Составим кнопку для проверки выбранного ответа

ШАГ 3

Следующим шагом будет разработка внешнего вида приложения и определение компонент, которых мы будем использовать для ввода и вывода информации

The screenshot shows a Windows Forms application window titled "Form1" with a standard Windows XP-style title bar. The form is designed with a dotted background and contains several controls:

- GroupBox1:** Contains "Label1", a "RadioGroup1" with five radio buttons, "Label6", and "Button1".
- GroupBox2:** Contains "Label2", a "RadioGroup2" with five radio buttons, "Label7", and "Button2".
- GroupBox3:** Contains "Label3", a "RadioGroup3" with five radio buttons, "Label8", and "Button3".
- GroupBox4:** Contains "Label4", four checkboxes ("CheckBox1" through "CheckBox4"), "Label9", and "Button4".
- GroupBox5:** Contains "Label5", four checkboxes ("CheckBox5" through "CheckBox8"), "Label10", and "Button5".
- Text Area:** A large empty rectangular area located below GroupBox5.
- Button6:** A button located at the bottom right of the form.

A red rectangular border highlights GroupBox2, GroupBox3, and GroupBox4. A speech bubble on the right side of the form contains the text: "Аналогично оформляем 2 и 3 вопроса".

Аналогично
оформляем
2 и 3 вопроса

ШАГ 3

Следующим шагом будет разработка внешнего вида приложения и определение компонент, которых мы будем использовать для ввода и вывода информации

Для 4 и 5 вопросов (множественный выбор) вместо радиокнопок вставляем по 4 Check Box - а

Размещаем компонент Memo для вывода результатов теста

И, наконец, кнопку для завершения работы с программой

ШАГ 4

Сделаем соответствующие надписи на компонентах формы (и не забудем разместить манифест XP)

Тест по информатике

Вопрос 1
Верно ли, что информация в компьютере хранится в двоичном коде

Варианты ответов

- Только числовая
- Только числовая и текстовая
- Любая информация
- Вся за исключением видео

проверить

Вопрос 2
Наименьшая единица информации называется

Варианты ответов

- байт
- бит
- бод
- бит/с

проверить

Вопрос 3
Элементарной базой первых компьютеров являлись

Варианты ответов

- Микросхемы
- Дискретные полупроводники
- Радиолампы
- Транзисторы

проверить

Вопрос 4
Назовите устройства вывода информации

- Монитор
- Мышь
- Клавиатура
- Принтер

проверить

Вопрос 5
Назовите системы программирования

- Delphi
- Visual Basic
- Microsoft
- Turbo Pascal

проверить

Memo1

Выход

В результате
мы получили
примерно
такую форму,
на которой
есть вопросы,
варианты
ответов, Memo
для вывода
результатов,
кнопки
проверки и
выхода

ШАГ 5

Приложение оформлено. Начнем программирование событий. И самое первое событие, которое возникает каждый раз при запуске программы – создание формы (**On Create**)

Что должно происходить при запуске программы?

Тест по информатике

Вопрос 1
Верно ли, что информация в компьютере хранится в двоичном коде

Варианты ответов

- Только числовая
- Только числовая и текстовая
- Любая информация
- Вся за исключением видео

проверить

Label6

Вопрос 2
Наименьшая единица информации называется

Варианты ответов

- байт
- бит
- бод
- бит/с

проверить

Label7

Вопрос 3
Элементарной базой первых компьютеров являлись

Варианты ответов

- Микросхемы
- Дискретные полупроводники
- Радиолампы
- Транзисторы

проверить

Label8

Вопрос 4
Назовите устройства вывода информации

- Монитор
- Мышь
- Клавиатура
- Принтер

проверить

Label9

Вопрос 5
Назовите системы программирования

- Delphi
- Visual Basic
- Microsoft
- Turbo Pascal

проверить

Label10

Memo1

ВЫХОД

1. Метки (**Label 6,7,8,9,10**) должны быть невидимы
2. **Мемо** тоже должен быть невидим и появляться только после прохождения теста
3. Счетчик **k** верно выполненных вопросов должен обнуляться

ШАГ 5

Приложение оформлено. Начнем программирование событий. И самое первое событие, которое возникает каждый раз при запуске программы – создание формы (**On Create**)

Сделаем двойной щелчок по форме и в редакторе кода запишем процедуру создания формы (**On Create**)

```
var
  Form1: TForm1;
  k: integer;
implementation
  {$R *.dfm}

  procedure TForm1.FormCreate(Sender: TObject);
  begin
    Memo1.Visible:=false;
    Memo1.Text:='';
    Label6.Visible:=false;
    Label7.Visible:=false;
    Label8.Visible:=false;
    Label9.Visible:=false;
    Label10.Visible:=false;
    k:=0;
  end;
```

Объявляем переменную **k**, которая будет являться счетчиком числа верных ответов

Делаем невидимыми **Label** – **Label** и **Memo**

Счетчику **k** присваиваем ноль

ШАГ 6

Сейчас напишем отклик на событие нажатия на кнопку «**ПРОВЕРИТЬ**»

Сделаем двойной щелчок по кнопке «**ПРОВЕРИТЬ**» в первом вопросе и запишем соответствующий код.

```

Unit1
procedure TForm1.Button1Click(Sender: TObject);
begin
  Button1.Enabled:=false;
  if radiogroup1.ItemIndex=2 then
  begin
    k:=k+1;
    label6.Font.Color:=rgb(0,150,0);
    label6.Visible:=true;
    label6.Caption:='ВЕРНО !!';
  end
  else
  begin
    label6.Font.Color:=rgb(150,0,0);
    label6.Visible:=true;
    label6.Caption:='НЕВЕРНО !!';
  end;
end;
end;

```

Давайте его разберем

Кнопку «**ПРОВЕРИТЬ**» в первом вопросе делаем недоступной, после того, как она нажата. Это исключает возможность возврата к первому вопросу и выбора другого ответа

Проверяем условие: соответствует ли наш выбор правильному, т.е. выбран ли переключатель с индексом 2



Заметьте ! Индексы переключателей в Delphi имеют нумерацию, начинающуюся с нуля, поэтому верный ответ у нас обозначен индексом 2, хотя на самом деле это третий вариант ответов (посмотрите на тест в бумажном варианте)

ШАГ 6

Сейчас напишем отклик на событие нажатия на кнопку «**ПРОВЕРИТЬ**»

Сделаем двойной щелчок по кнопке «**ПРОВЕРИТЬ**» в первом вопросе и запишем соответствующий код.

```

Unit1
procedure TForm1.Button1Click(Sender: TObject);
begin
  Button1.Enabled:=false;
  if radiogroup1.ItemIndex=2 then
  begin
    k:=k+1;
    label6.Font.Color:=rgb(0,150,0);
    label6.Visible:=true;
    label6.Caption:='ВЕРНО !!';
  end
  else
  begin
    label6.Font.Color:=rgb(150,0,0);
    label6.Visible:=true;
    label6.Caption:='НЕВЕРНО !!';
  end;
end;
end;

```

Давайте его разберем

Если условие выполняется (выбран верный ответ), то значение счетчика верных ответов увеличиваем на единицу. - Метка (Label6), где выводится: - Метка (Label6), где выводится: и инициализируем ее в панели пишем в ней надпись **ВЕРНО** зеленым цветом

Если условие не выполняется, то счетчик не увеличиваем, а метку 6 делаем видимой и выводим надпись **НЕВЕРНО**, принимаем красным цветом



Очевидно, что для кнопок проверки 2 и 3 вопросов код будет абсолютно таким же за исключением номеров Label-ов для вывода ВЕРНО/НЕВЕРНО, поэтому Вы можете написать его самостоятельно (не забывайте, что в редакторе можно использовать копирование кода – это гораздо быстрее)

ШАГ 7

Следующим шагом опишем процедуры проверки ответа в вопросах 4 и 5 – там код будет немного другой ввиду множественного выбора и других используемых компонент

Сделаем двойной щелчок по кнопке «**ПРОВЕРИТЬ**» в 4 вопросе и запишем соответствующий код.

```

procedure TForm1.Button4Click(Sender: TObject);
begin
    Button4.Enabled:=false;
    if (checkbox1.Checked=true) and (checkbox4.Checked=true)
    and (checkbox2.Checked=false) and (checkbox3.Checked=false) then
    begin
        k:=k+1;
        label9.Font.Color:=rgb(0,150,0);
        label9.Visible:=true;
        label9.Caption:='ВЕРНО !!'
    end
    else
    begin
        label9.Font.Color:=rgb(150,0,0);
        label9.Visible:=true;
        label9.Caption:='НЕВЕРНО !!'
    end;
end;

```

Проверяем выбранные
ответы : ответ будет
верным только тогда, когда
первый и четвертый боксы
будут выбраны (True), а
второй и третий не выбраны
(false)

В этом случае- значение
счетчика верных ответов
увеличиваем на единицу
- Метку 9 (label9) делаем
видимой и пишем в ней
надпись **ВЕРНО** зеленым
цветом

Если условие не выполняется, то счетчик не увеличиваем,
метку 9 делаем видимой и выводим надпись **НЕВЕРНО** красным
цветом

ШАГ 8

Хотя **вопрос 5** тоже с множественным выбором, то код обработки этой кнопки аналогичен кнопке 4, но давайте добавим в нее дополнительные функции: после нажатия этой кнопки должны выводиться результаты теста в **Мемо**

```
procedure TForm1.Button5Click(Sender: TObject);
```

```
begin
```

```
  Button5.Enabled:=false;
```

```
  if (checkbox5.Checked=true) and (checkbox6.Checked=true)
```

```
  and (checkbox8.Checked=true) and (checkbox7.Checked=false) then
```

```
begin
```

```
k:=k+1;
```

```
label10.Font.Color:=rgb(0,150,0);
```

```
label10.Visible:=true;
```

```
label10.Caption:='ВЕРНО !!'
```

```
end
```

```
else
```

```
begin
```

```
label10.Font.Color:=rgb(150,0,0);
```

```
label10.Visible:=true;
```

```
label10.Caption:='НЕВЕРНО !!';
```

```
end;
```

```
memo1.Visible:=true;
```

```
memo1.Text:='Всего вопросов - 5'+#13+#10+'Количество верных ответов - '
```

```
+inttostr(k)+#13+#10+'Процент выполнения теста - '+floattostr(k/5*100)+' %';
```

Эта часть аналогична
кнопке 4 вопроса и
объяснений здесь не
требуется

Делаем видимым
компонент **Мемо** для
вывода результатов

Выводим результаты теста

ШАГ 8

Давайте посмотрим, как выводится в **Мемо** результат теста

```
memo1.Text:='Всего вопросов - 5'+#13+#10+'Количество верных ответов - '+
+inttostr(k)+#13+#10+'Процент выполнения теста - '+floattostr(k/5*100)+' %';
```

Всего вопросов - 5

В апострофах (одинарных кавычках), как и в Turbo Паскале, выводится строка символов:

Количество верных ответов

Опять вывод строки

начала следующей строки

следующего элемента начнется с

Это значит, что вывод

соответственно

(Enter) и в ее начало

курсор на следующую строку

зарезервированы для переноса

символы с номерами 13 и 10

кодировки символов - ASCII -

Давайте вспомним таблицу

БАЗОВАЯ ТАБЛИЦА ASCII

3	4	5	6	7	8	9	A	B	C	D	E	F
▼	◆	◆	◆	◆	◆	○						
3	4	5	6	7	8	9	10	11	12	13	14	15
!			-		↑	↓	→	←		↔	▲	▼
19	20	21	22	23	24	25	26	27	28	29	30	31
#	\$	%	&	'	()	*	+	,	-	.	/
35	36	37	38	39	40	41	42	43	44	45	46	47
3	4	5	6	7	8	9	:	;	<	=	>	?
51	52	53	54	55	56	57	58	59	60	61	62	63

ШАГ 8

Давайте посмотрим, как выводится в **Мемо** результат теста

```
memo1.Text:='Всего вопросов - 5'+#13+#10+'Количество верных ответов - '
+inttostr(k)+#13+#10+'Процент выполнения теста - '+floattostr(k/5*100)+' %';
```

Опять переводим курсор на начало строки

подсчитываем процент выполнения и переводим его в строковый тип (из вещественного)

Выводим символ %

Выводим значение счетчика k, естественно, переводя его в строковый тип (с этим мы уже знакомы) (inttostr)

Выводим текст Процент выполнения теста

И вот результат, выведенный в Мемо

```
Всего вопросов - 5
Количество верных ответов - 3
Процент выполнения теста - 60 %
```

ШАГ 9

Последняя кнопка – ВЫХОД

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
close  
end;
```

закрывается

ШАГ 10

Сохраняем и компилируем программу (см. предыдущие уроки)

А сейчас попробуем запустить наш тест

Запустить ->

На этом наш урок закончен.

Конечно, наша тестирующая программа получилась довольно примитивная, у нее отсутствуют многие необходимые функции. По мере изучения Delphi мы составим более серьезную и удобную тестирующую оболочку, а пока самое главное, что наша программа работает и мы научились использовать компоненты для проверки условий

ИТОГИ УРОКА:

На этом уроке мы научились создавать и проверять условия, пользоваться компонентами `Radio Button` и `Checkbox` и составили простейшую тестирующую программу

НА СЛЕДУЮЩЕМ УРОКЕ:

ООП на Delphi – 5:

Мы научимся использовать элементы ввода и вывода информации и составим программу расчета корней квадратного уравнения, а также познакомимся с обработкой исключительных ситуаций

Домнин Константин Михайлович

E – mail: kdomnin@list.ru

2006 год.