

Гомельский государственный университет  
имени Франциска Скорины

## **Выбор**

Составил:

Ассистент кафедры общей физики  
Соколов С.И.

Гомель, 2015

## Инструкция if

Инструкция if позволяет выбрать один из двух возможных вариантов развития программы. Выбор осуществляется в зависимости от выполнения условия. В общем виде инструкция if записывается так:

**if** условие **then**

**begin**

        // здесь инструкции, которые надо выполнить,  
        // если условие истинно.

**end**

**else**

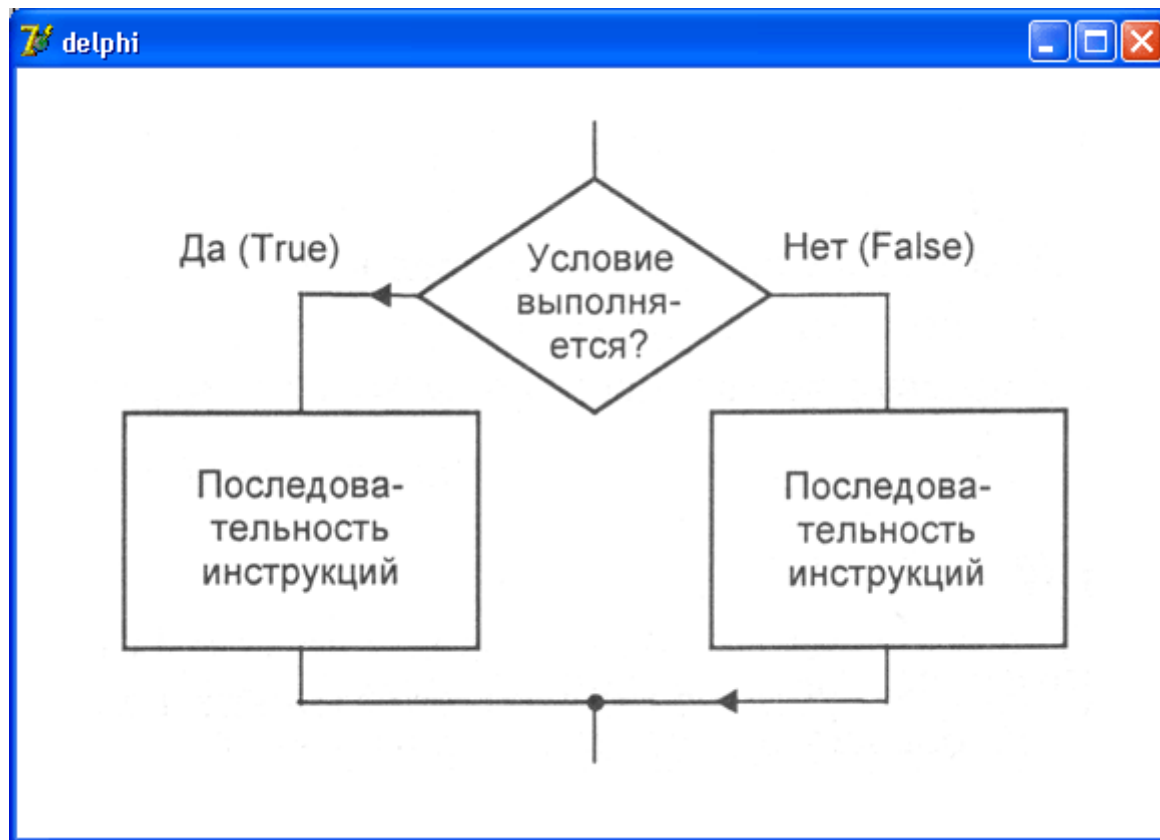
**begin**

        // здесь инструкции, которые надо выполнить, // если  
условие ложно.

**end;**

Выполняется инструкция `if` следующим образом:

1. Вычисляется значение условия (условие — выражение логического типа, значение которого может быть равно `True` или `False`).
2. Если условие истинно (значение выражения условие равно `True`), то выполняются инструкции, следующие за словом `then` (между `begin` и `end`). На этом выполнение операции `if` заканчивается, то есть инструкции, следующие за `else`, не будут выполнены.  
Если условие ложно (значение выражения условие равно `False`), то выполняются инструкции, следующие за словом `else` (между `begin` и `end`).



Например, если переменная  $t$  обозначает тип соединения сопротивлений в электрической цепи ( $t=1$  соответствует последовательному соединению,  $t=2$  — параллельному), а  $r_1$  и  $r_2$  — величины сопротивлений, то приведенная ниже инструкция `if` осуществляет выбор формулы, по которой будет выполнен расчет.

```
if t=1 then  
    begin  
        z:=r1+r2;  
    end  
else  
    begin  
        z:=(r1+r2)/(r1*r2);  
    end;
```

Если в инструкции if между begin и end находится только одна инструкция, то слова begin и end можно не писать.

Например, инструкцию

```
if otv=3  
  then  
  begin  
    prav:=prav+1 ;  
  end  
  else  
  begin  
    ShowMessage('Ошибка!');  
  end;
```

можно переписать так:

```
if otv=3 then  
  prav:=prav+1  
else  
  ShowMessage('Ошибка!') ;
```

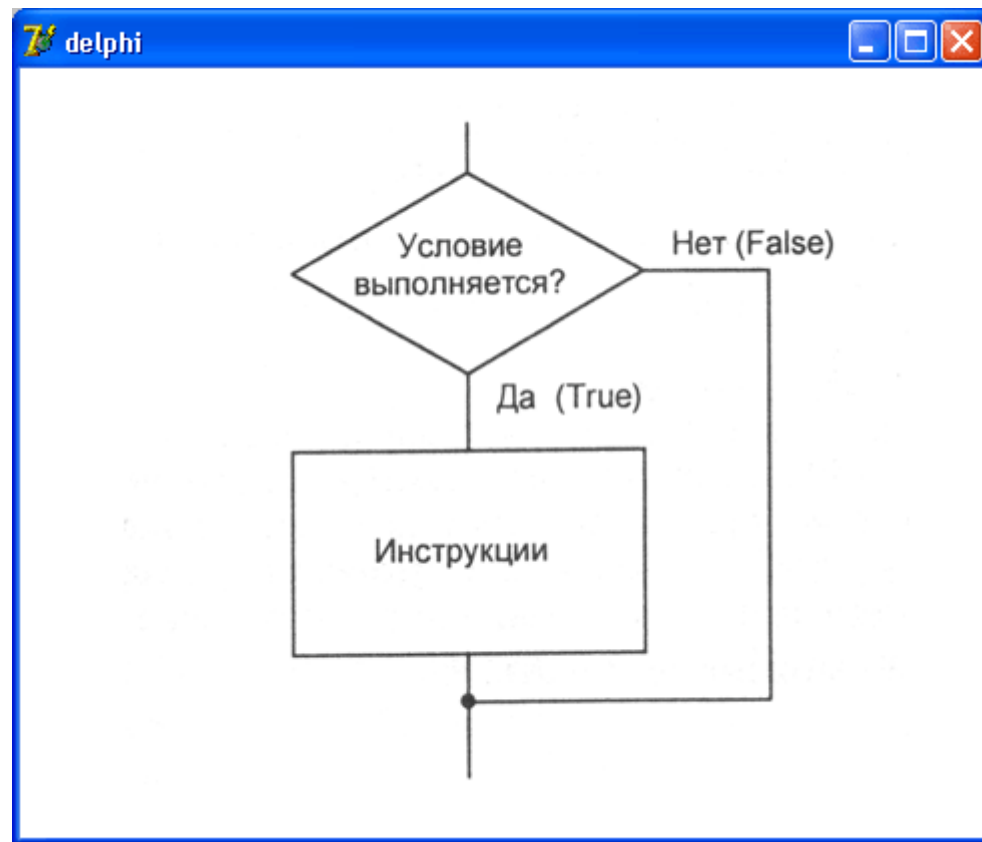
Если какое-либо действие должно быть выполнено только при выполнении определенного условия и пропущено, если это условие не выполняется, то инструкция if может быть записана так:

**if** условие **then**

**begin**

{ инструкции, которые надо выполнить, если условие выполняется, истинно }

**end**



## Пример

В качестве примера использования инструкции `if` рассмотрим программу вычисления стоимости междугороднего телефонного разговора.

Как известно, стоимость междугороднего разговора по телефону в выходные дни ниже, чем в обычные. Программа, запрашивает длительность разговора и день недели, а затем вычисляет стоимость разговора. Если день недели — суббота или воскресенье, то стоимость уменьшается на величину скидки. Цена минуты разговора и величина скидки задаются в тексте программы как константы.

Стоимость разговора

Длительность (мин.)

Edit1

Номер дня недели

Edit2

Label3

Вычислить



<b>Компонент</b>	<b>Назначение</b>
Edit1	Для ввода длительности разговора в минутах
Edit2	Для ввода номера дня недели 2
Label1, Label2	Для вывода пояснительного текста о назначении полей ввода
Label3	Для вывода результата вычисления — стоимости разговора
Button1	Для активизации процедуры вычисления стоимости разговора

<b>Свойство</b>	<b>Значение</b>
Form1.Caption	Стоимость разговора
Edit1.Text	
Edit2.Text	
Label1.Caption	Длительность (мин.)
Label2.Caption	Номер дня недели
Label3.Caption	
Button1.Caption	Вычислить

Часто в программе необходимо реализовать выбор более чем из двух вариантов. Например, известно, что для каждого человека существует оптимальное значение веса, которое может быть вычислено по формуле:

$\text{Рост(см)} - 100.$

Реальный вес может отличаться от оптимального: вес может быть меньше оптимального, равняться ему или превышать оптимальное значение.

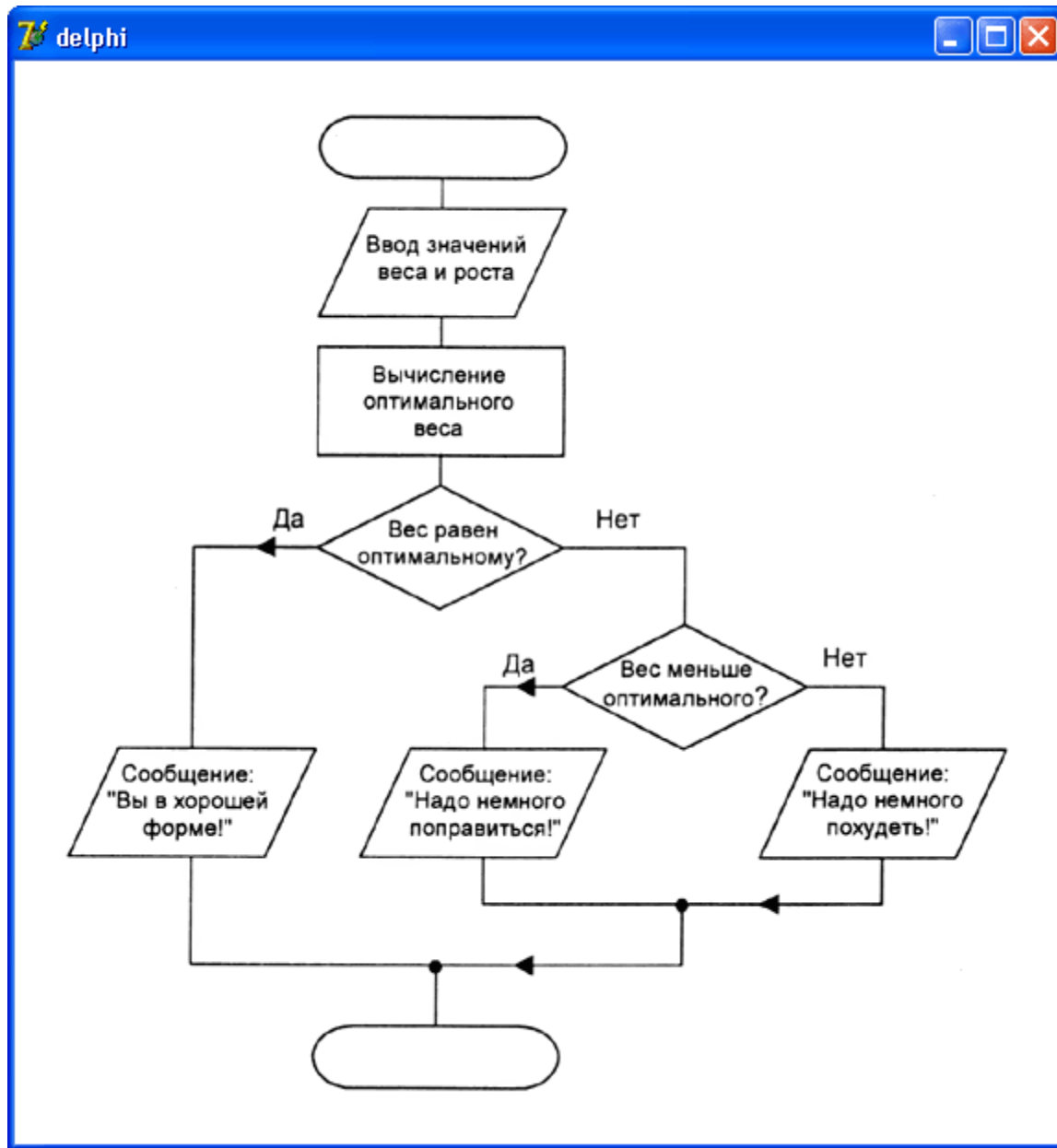
Следующая программа запрашивает вес и рост, вычисляет оптимальное значение, сравнивает его с реальным весом и выводит соответствующее сообщение.

7 Контроль веса

Ваш вес (кг)

Ваш рост (см.)

Вычислить



## Инструкция case

В предыдущем примере, в программе контроля веса, множественный выбор был реализован при помощи вложенных одна в другую инструкций if. Такой подход не всегда удобен, особенно в том случае, если количество вариантов хода программы велико.

В языке Delphi есть инструкция case, которая позволяет эффективно реализовать множественный выбор. В общем виде она записывается следующим образом:

**case** Селектор of  
список1:

```
begin  
  { инструкции 1 }  
end;
```

список2:

```
begin  
  { инструкции 2 }  
end;
```

....

список N:

```
begin  
  { инструкции N }  
end;
```

**else**

```
begin  
  { инструкции ) }  
end;
```

**end;**

где:

.Селектор — выражение, значение которого определяет дальнейший ход выполнения программы (т. е. последовательность инструкций, которая будет выполнена);

.Список N — список констант. Если константы представляют собой диапазон чисел, то вместо списка можно указать первую и последнюю константу диапазона, разделив их двумя точками. Например, список 1, 2, 3, 4, 5, 6 может быть заменен диапазоном 1..6.

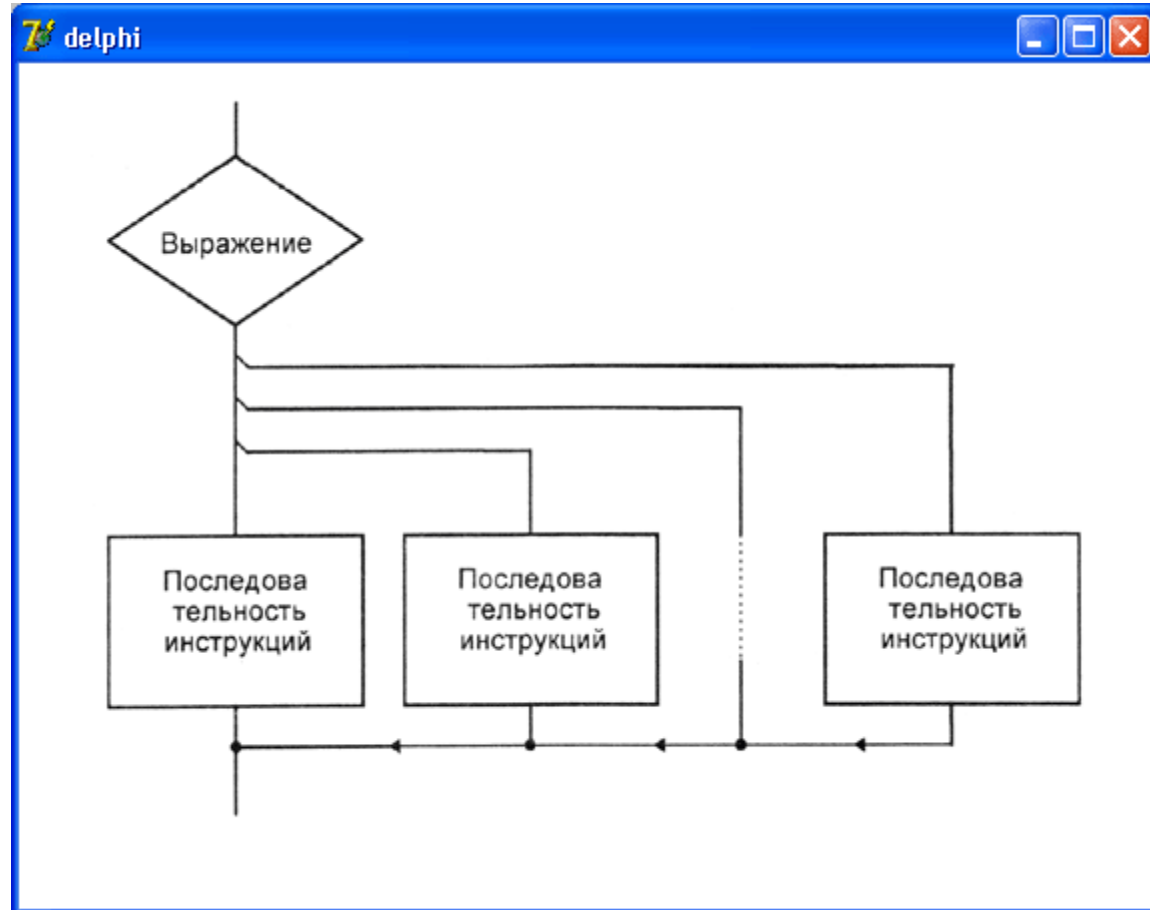
Выполняется инструкция case следующим образом:

1. Сначала вычисляется значение выражения-селектора.

2. Значение выражения-селектора последовательно сравнивается с константами из списков констант.

3. Если значение выражения совпадает с константой из списка, то выполняется соответствующая этому списку группа инструкций. На этом выполнение инструкции case завершается.

4. Если значение выражения-селектора не совпадает ни с одной константой из всех списков, то выполняется последовательность инструкций, следующая за else.





**case** n\_day **of**

1,2,3,4,5: day:='Рабочий день. ' ;

6: day:='Суббота!';

7: day:='Воскресенье!';

**end;**

**case** n\_day **of**

1..5: day:='Рабочий день.';

6: day:='Суббота!';

7: day:='Воскресенье!';

**end;**

**case** n\_day **of**

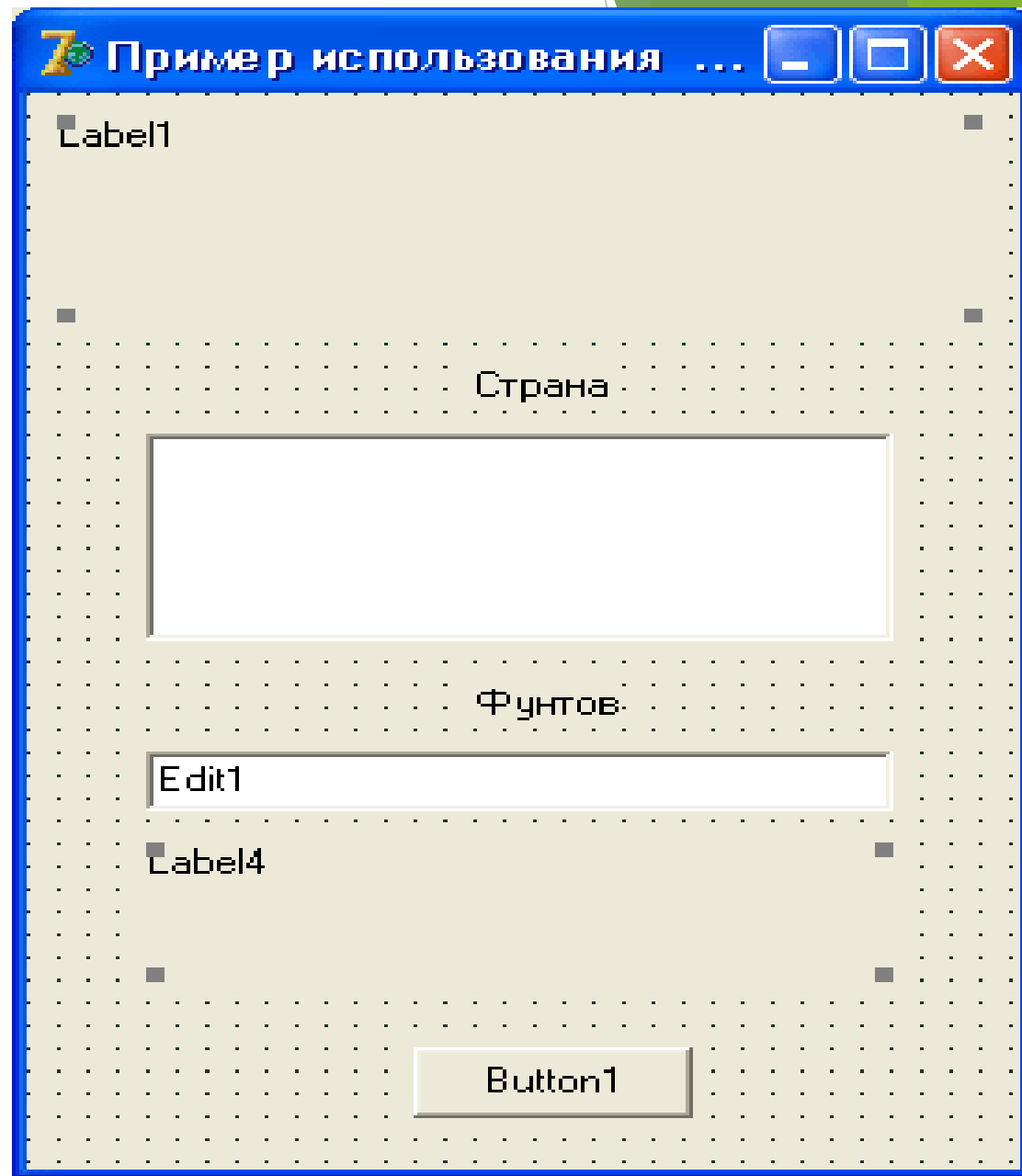
6: day:='Суббота!';

7: day:='Воскресенье!';

**else** day:='Рабочий  
день.';

**end;**

В качестве примера использования инструкции case рассмотрим программу, которая пересчитывает вес из фунтов в килограммы. Программа учитывает, что в разных странах фунт "весит" по-разному. Например, в России фунт равен 409,5 граммов, в Англии – 453,592 грамма, а в Германии, Дании и Исландии фунт весит 500 граммов.



## Свойства компонента ListBox

<b>Свойство</b>	<b>Определяет</b>
Name	Имя компонента. В программе используется для доступа к свойствам компонента
Items	Элементы списка
Itemindex	Номер выбранного элемента списка. Номер первого элемента списка равен нулю
Left	Расстояние от левой границы списка до левой границы формы
Top	Расстояние от верхней границы списка до верхней границы формы
Height	Высоту поля списка
Width	Ширину поля списка
Font	Шрифт, используемый для отображения элементов списка
Parent-Font	Признак наследования свойств шрифта родительской формы

<b>Компонент</b>	<b>Назначение</b>
ListBox1	Для выбора страны, для которой надо выполнить пересчет
Edit1	Для ввода веса в фунтах
Label1, Label2, Label3	Для вывода пояснительного текста о назначении полей ввода
Label4	Для вывода результата пересчета
Button1	Для активизации процедуры пересчета веса из фунтов в килограммы

<b>Свойство</b>	<b>Значение</b>
Form1.Caption	Пример использования case
Edit1.Text	
Label1.Caption	Выберите страну, введите количество фунтов и щелкните на кнопке Вычислить
Label2.Caption	Страна
Label3.Caption	Фунтов
Button1.Caption	Вычислить