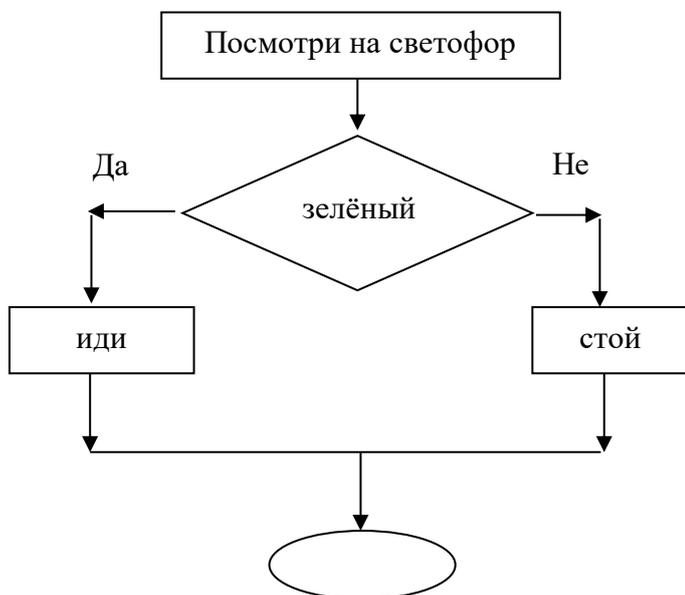


Урок 12. Оператор ветвления (альтернатива).

*Программист, ложась спать, ставит возле кровати 2 стакана:
один с водой, если захочет пить, а другой пустой, если не захочет.*

Жизнь полна неожиданностей и условностей. Каждый день приходится решать тысячи мелких задач, делая выбор. Реальная задача – переход улицы, на которой работает светофор.

1. Вот как выглядит эта задача на языке блок-схем:



2. На естественном языке такая конструкция выглядит так:
Если горит зелёный, **то** иди, **иначе** стой.

Или, в общем виде:
Если <условие>
то <ветвь «Да»>
иначе <ветвь «Нет»>

3. На языке программирования этот оператор звучит следующим образом:

```
IF <условие>  
THEN <ветвь «Да»>  
ELSE <ветвь «Нет»>  
IF...THEN...ELSE...
```

А поскольку в любой ветви может быть не один, а несколько шагов, то их необходимо оформить в виде отдельной подпрограммы (записать между **begin** и **end**), т.е.:

```
4a. IF <условие>  
THEN  
  Begin  
    <ветвь «Да»>  
  End  
ELSE  
  Begin  
    <ветвь «нет»>  
  End
```

```
4b. Если горит зелёный  
То  
  Начинай  
    иди  
  Конец  
  Иначе  
    Начинай  
      стой  
    Конец
```

1 задание: составьте блок-схему анекдота:

Председатель колхоза пришел на ферму и спрашивает коров:

– Ну, что будем сдавать, молоко или мясо?

Примечание: Вопрос нужно задавать так, чтобы на него можно было бы ответить «Да» или «нет»

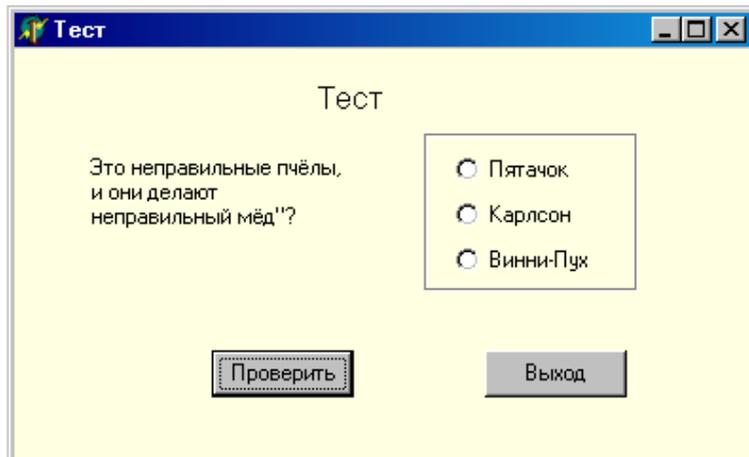
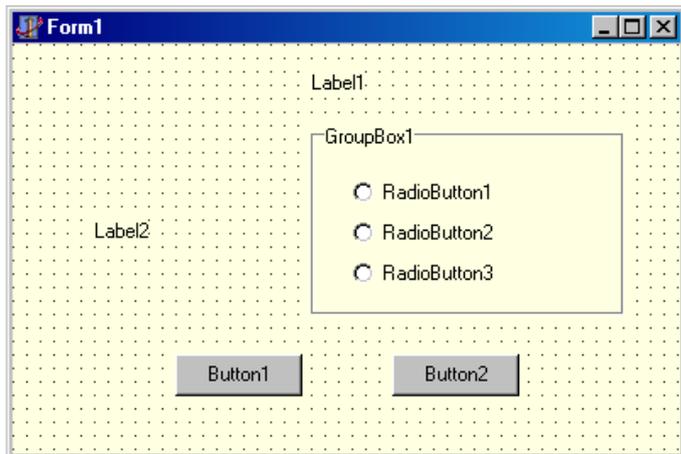
2 задание: составьте блок-схему следующей ситуации:

Если есть вода, то нужно почистить зубы, если воды нет, то нужно сходить в магазин и купить жевательную резинку.

А что же в Delphi?

Оказывается, ни одна серьезная программа не обходится без этого оператора. А как с ним работать, мы разберемся на примере **задания 14а**: создадим тест.

1. Запустите **Delphi** и сделайте так:



2. Обратите внимание, слова в компонентах **Label** располагаются не в одной, а в нескольких строчках. Прежде, чем набирать текст, свойство **WordWrap** сделайте равным **True**.

Начинаем программирование.

Как должна работать программа? Пользователь щёлкает ответ, затем нажимает кнопку **Проверить**. Программа проверяет. Если выбор правильный - выводим **«Верно»** и отключаем кнопку **Проверить**, если неверно - выводим **Ошибка** и даем возможность продолжить тест.

Значит, программировать необходимо кнопку **Проверить**.

3. Дважды щелкните на кнопке **Проверить**. Открылось окно редактора кода.
4. Пишем код:

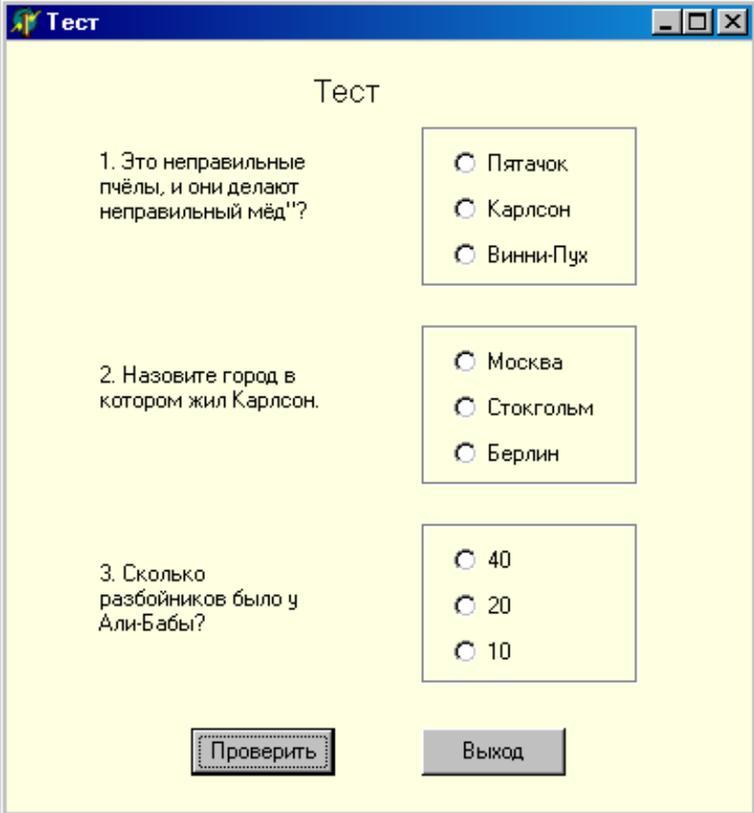
Код	Пояснения
<pre>procedure TForm1.Button1Click(Sender: TObject); begin if <i>radioButton3.Checked</i> then begin <i>label1.Caption:='Верно';</i> <i>button1.Enabled:=False;</i> end else begin <i>label1.Caption:=' Ошибка ';</i> end; end;</pre>	<p>Процедура Щелчка кнопки Проверить. Начало процедуры Если <i>выбрана 3-я радиокнопка</i> То начинай <i>Вывести надпись «Верно»</i> <i>Отключить кнопку Проверка</i> конец Иначе начинай <i>Вывести надпись «Ошибка»</i> конец; Конец процедуры;</p>

Разберите внимательно этот код, этот оператор используется часто.

Для этого оператора есть правило: если между **begin... end** стоит один оператор, то строки **begin... end** можно опустить. В нашем случае, по ветви «ДА» - 2 оператора, по ветви «НЕТ» - 1 оператор, следовательно, во втором случае **begin...end** можно убрать. И тогда код будет таким:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    if radioButton3.Checked
    then
        Begin
            label1.Caption:='Верно';
            button1.Enabled:=False;
        end
    else
        label1.Caption:=' Ошибка';
end;
```

Задание 146. Сделайте тест, состоящий из трех вопросов, например, такой:



Тест

1. Это неправильные пчёлы, и они делают неправильный мёд"?

Пятачок
 Карлсон
 Винни-Пух

2. Назовите город в котором жил Карлсон.

Москва
 Стокгольм
 Берлин

3. Сколько разбойников было у Али-Бабы?

40
 20
 10

Проверить Выход

Самостоятельно. Усовершенствуйте программу.

1. Сделайте так, чтобы она выделяла цветом неверный ответ.
2. Сделайте так, чтобы программа считала количество верных ответов.