

# События Delphi

**Тема урока:**

# События

**Цель урока:**

Сформулировать понятия

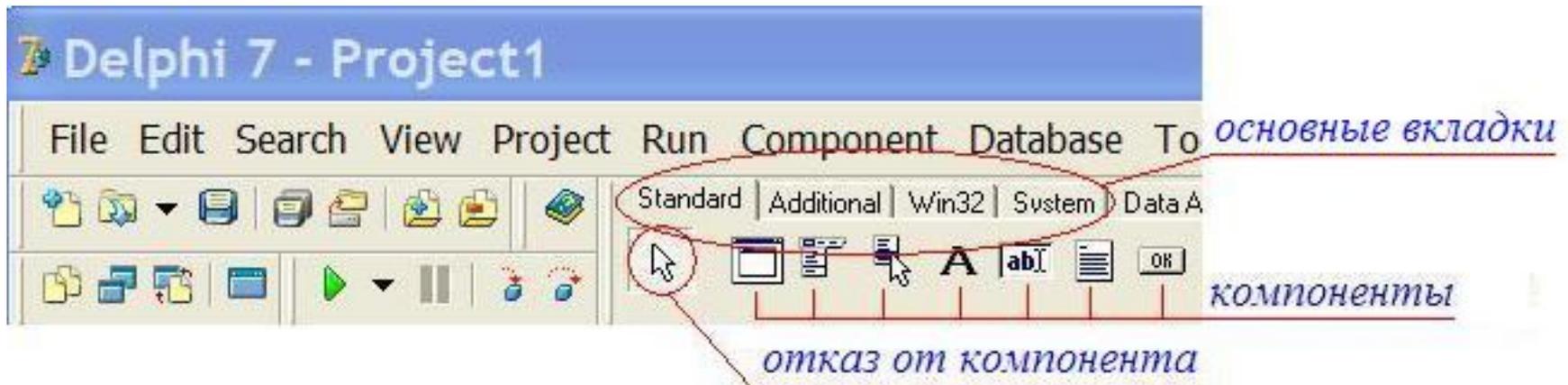
«Событие», «Обработчик события»



# Повторение

Компоненты Delphi располагаются на соответствующих вкладках палитры компонентов «Главного окна». Все основные компоненты Delphi находятся на первых четырёх вкладках:

- **Standard,**
- **Additional,**
- **Win32,**
- **System.**



Выбираем на панели **Component Palette** компонент, помещаем его на форму и компонент становится объектом, затем настраиваем его свойства с помощью инспектора объектов (т.е. меняем его свойства), создаем программный код для него, в результате чего объект превращается в элемент управления.



# Компоненты

## визуальные

компоненты, которые мы видим в окне работающего приложения

базовый класс **TControl**

## невизуальные

не видны в процессе работы приложения

### ОКОННЫЕ

*Оконный элемент управления – специализированное окно, предназначенное для решения конкретной задачи:*

- командные кнопки,
- поля редактирования,
- полосы прокрутки

базовый класс **TWinControl** – прямой потомок класса **TControl**

### НЕОКОННЫЕ

*Неоконные элементы управления не могут получать фокус и быть контейнерами для других элементов. Пример - Графические окна*

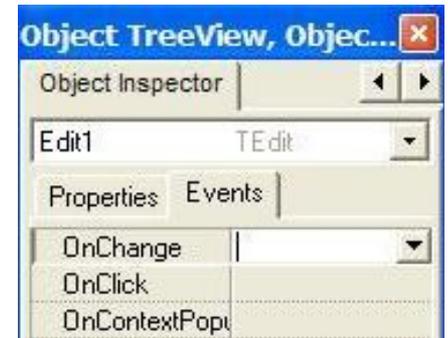
базовый класс **TGraphicControl**



# События

Визуальные компоненты способны генерировать и обрабатывать **события**. К наиболее общим событиям относятся:

- Выбор управляющего элемента;
- Перемещение указателя мыши;
- Нажатие клавиша клавиатуры;
- Получение и потеря элементом фокуса;
- Перемещение объектов



Для реализации события пишется **процедура-обработчик события**.

## Пример.

Процедура обработки события «щелчок мыши по кнопке Button1»

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    Button1.Caption:='Выход';
```

```
end;
```

# Практическая работа

**Напишем** маленькую программу, которая будет по нажатию **кнопки** отображать на форме то, что введено в **строке ввода**.



- Расположим на форме компонент **Edit (строка ввода)**. Свойство **Text** очистим.
- Далее - компонент **Button (кнопка)**. В свойстве **Caption** напишем, например, «отобразить».
- Затем - компонент **Label (метка)**. В свойстве **Font** для наглядности сделайте размер побольше, например 24, выберите шрифт и цвет. Свойство **Caption** также очистите.
- **Щёлкните дважды** по созданной кнопке. Откроется **окно модуля** с заготовкой обработчика события нажатия на кнопку.
- В обработчике пишем: **Label1** и ставим точку. После этого система **Delphi** предлагает нам выбрать из списка нужное свойство или метод. Введите "C", и строка "Caption" станет выделенной. **Delphi** пытается угадать, что мы хотим ввести. Дальше писать не обязательно, достаточно нажать **Enter**.



В результате получим: **Label1.Caption**.

Как мы видим, точкой разделяются в Delphi объект и его свойства.

Указанием на объект служит его имя (свойство **Name**).

Имя компонента можно посмотреть в Инспекторе Объектов, или наведя мышку на компонент и дождавшись появления подсказки.

Присваиваем свойству **Caption** объекта **Label1** значение оператором присваивания **:=**

**Label1.Caption:=...**

Далее пишем то, что отобразится в метке - свойство **Text** объекта **Edit1**:  
**Edit1.Text** и оператор **;**. Получилось

**Label1.Caption := Edit1.Text;**

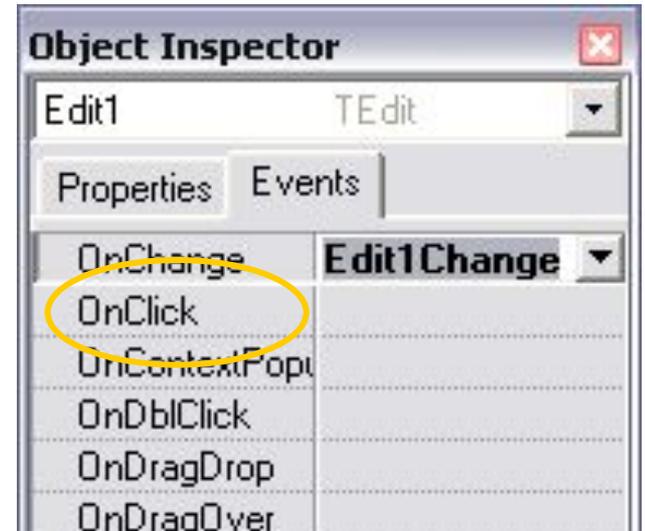
между **begin** и **end**.

Сохраняем в папке P3 и жмём F9.

```
Unit1  
Constants  
procedure TForm1.Edit1Change(Sender: TObject);  
begin  
    label1.Caption:=Edit1.Text;  
end;
```



Изменим нашу программу так, чтобы текст на форме появлялся прямо в момент его ввода. В момент ввода у **Edit1** меняется свойство **Text** - в нём появляется новая буква. Воспользуемся событием **onChange** которое и происходит в этот момент.

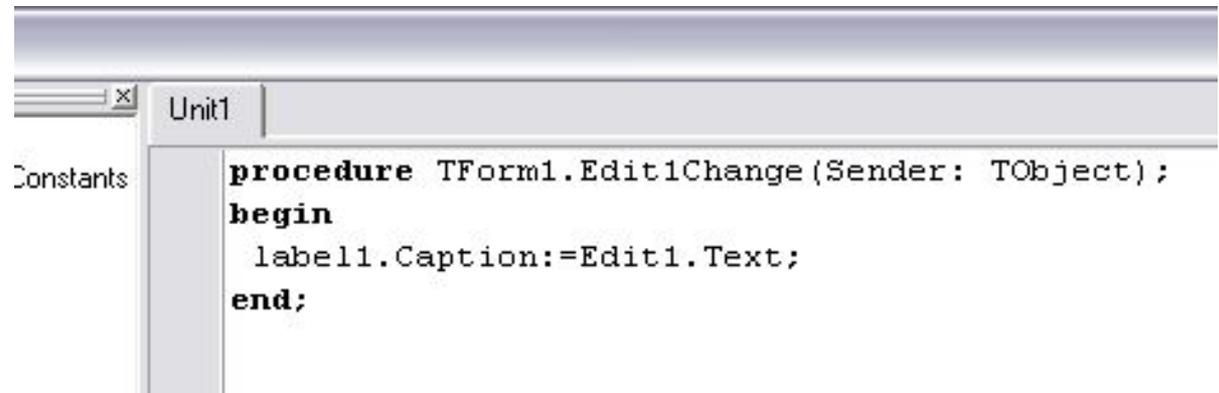


1. Щёлкнем дважды по **onChange**, и оказываемся внутри только что созданного обработчика.
2. Присваиваем свойству **Caption** компонента **Label1** значение свойства **Text** компонента **Edit1** :

**Label1.Caption := Edit1.Text;**

Или просто копируем из обработчика нажатия кнопки.

3. Запускаем программу.



Теперь рассмотрим событие **onClick**. Это событие возникает при щелчке мышкой на компоненте. Воспользуемся им, чтобы очищать строку ввода от уже введенных символов.

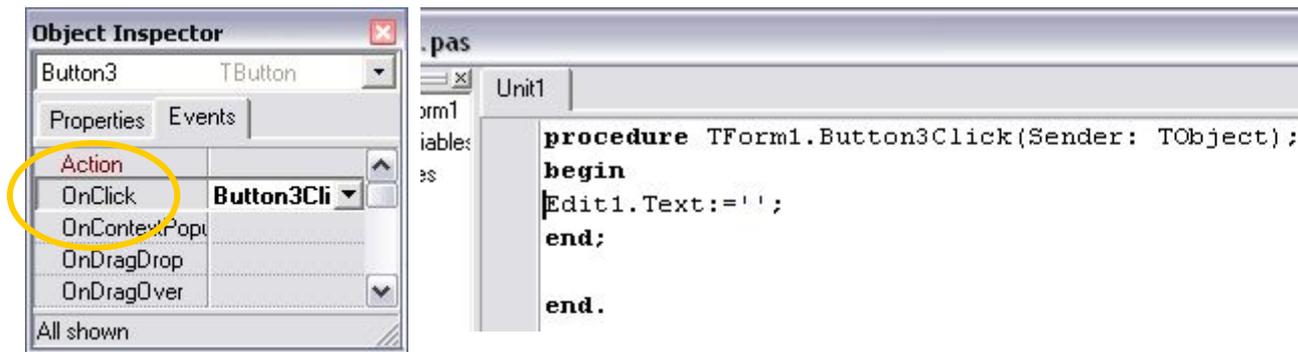
В обработчике нужно присвоить свойству **Text** значение пустой строки. Строка в **Delphi** образуется заключением текста в **одинарные** кавычки :

'Так в Delphi образуется строка'

Значит, пустая строка - это кавычки без текста: "".

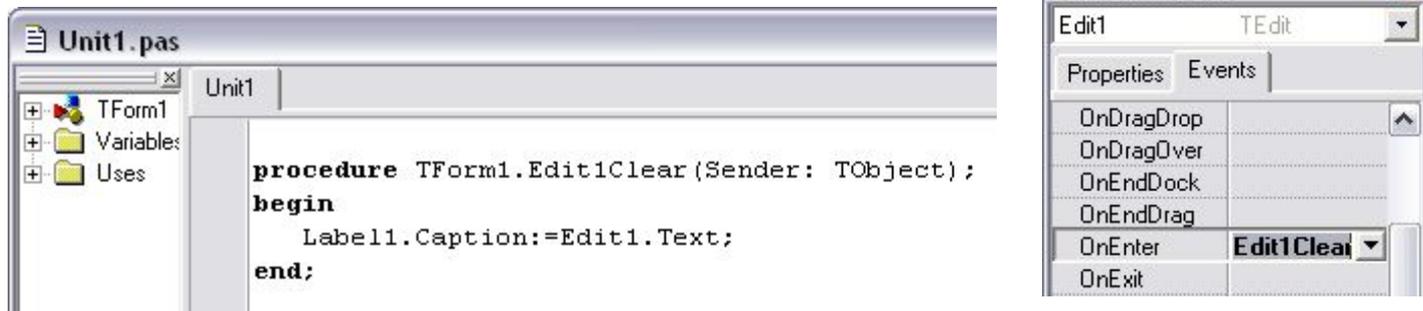
Не должно быть даже пробела, иначе он окажется в строке ввода:

**Edit1.Text:='';**



Есть другой способ, воспользоваться специально предназначенным для этого **методом компонента** Edit, который так и называется **Clear**

**Edit1.Clear;**



Другие важные события, которые есть почти у каждого визуального компонента:

**onExit** - возникает, когда компонент теряет фокус ввода;

**onDbClick** - возникает при двойном щелчке мышкой по компоненту;

**onKeyDown** - когда при нажатии на кнопку на клавиатуре она оказалась в нажатом положении;

**onKeyUp** - когда при отпускании клавиатурной кнопки она оказалась в верхнем положении;

**onKeyPress** - возникает при нажатии на клавиатурную кнопку. От событий **onKeyDown** и **onKeyUp** оно отличается типом используемого параметра **Key** ;

**onMouseDown** - когда при нажатии кнопки мышки она оказалась в нижнем положении;

**onMouseUp** - когда при отпускании кнопки мышки она оказалась в верхнем положении;

**onMouseMove** - возникает при перемещении указателя мышки над компонентом.