

## Иерархия классов: наследование

### Создание собственного класса



Синтаксис создания класса наследника на языке *Object Pascal* следующий (описание новых классов осуществляется в разделе *Type*):

```
имя_создаваемого = class(имя_родителя)
    private
        { описание собственных членов класса }
    protected
        { описание защищенных членов класса }
    public
        { описание общих членов класса }
end;
```

## Иерархия классов: наследование

### Создание собственного класса



Здесь *имя создаваемого* класса является любым еще не описанным идентификатором, а *имя родителя* – имя уже существующего (уже описанного) класса – например, класса библиотеки VCL.

Новый класс будет содержать как новые методы, так и переопределенные методы родительского класса. Под переопределенными подразумеваются методы, которые имеют то же имя, что и у родителя, но выполняют иные, чем у родителя, действия.

Методы, которые переопределены в дочернем классе должны быть отмечены ключевым словом **override**. Данный механизм называется **перегрузкой** методов.

## Создание собственного класса



Создадим новый проект в Delphi. Определим два класса кнопок. Исходя из структуры VCL, они должны быть наследниками **TButton**. В разделе **interface** после описания запишем:

```
TMyButton = class(TButton)
  private
    procedure DoClick(Sender : TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
  public
    constructor Create(cOwner : TComponent); override;
    procedure Click; override;
end;
```

## Создание собственного класса



В разделе **interface** главного модуля опишем реализацию методов и конструктора.

```
constructor TMyButton.Create(cOwner : TComponent);  
begin  
  inherited Create(cOwner); Name:='B1';  
  Caption:='Hello!'; OnMouseDown:=@DoClick;  
  Top:=10; Left:=10; Height:=30; Width:=110;  
end;
```

## Создание собственного класса



```
procedure TMyButton.DoClick(Sender : TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
    MessageBox(Form1.Handle,PChar('Its MyButton hello!'),  
PChar('TMyButton'),MB_OK or MB_ICONINFORMATION);  
end;
```

```
procedure TMyButton.Click;  
begin  
    Height:=Height+10;  
end;
```

## Создание собственного класса



Добавим еще один класс, так же наследник от **TButton**. В разделе `interface`, сразу за описание класса **TMyButton** добавим.

```
TMyButton2 = class(TButton)
private
  procedure DoClick(Sender : TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
public
  constructor Create(Owner : TComponent); override;
  procedure Click; override;
end;
```

## Создание собственного класса



В разделе **interface** главного модуля опишем реализацию методов и конструктора.

```
constructor TMyButton2.Create(Owner : TComponent);  
begin  
    inherited Create(Owner); Name:='B2';  
    Caption:='World!'; OnMouseDown:=DoClick;  
    Top:=10; Left:=130; Height:=30; Width:=110;  
end;
```

## Создание собственного класса



```
procedure TMyButton2.DoClick(Sender : TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
    MessageBox(Form1.Handle,  
PChar('Its MyButton2 Hello!'),  
PChar('TMyButton2'),MB_OK or MB_ICONWARNING);  
end;
```

```
procedure TMyButton2.Click;  
begin  
    Width:=Width+10;  
end;
```



## Код создания объектов



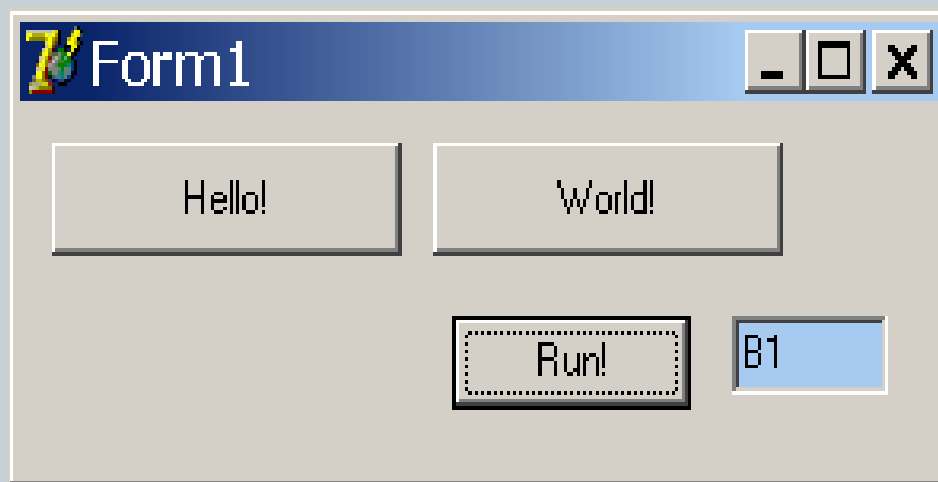
Создание компонент поместим в метод **OnCreate** формы, что позволит создать эти компоненты до вывода формы на экран.

```
procedure TForm1.FormCreate(Sender: TObject);  
var  
    Button : TButton;  
begin  
    Button := TMyButton.Create(Form1);  
    Button.Parent := Form1;  
  
    Button := TMyButton2.Create(Form1);  
    Button.Parent := Form1;  
end;
```

## Код создания объектов

Поместим на главную форму проекта кнопку **TButton** и поле ввода **TEdit**. Главная форма проекта в конструкторе форм будет выглядеть, как показано на рисунке.

Так как у нас помимо статических компонентов Button1 и



Edit1 создаются два динамических компонента, после запуска приложения внешний вид формы несколько измениться – на ней появятся еще две кнопки – динамически созданные нами наши объекты-наследники TButton.

## Код создания объектов



Опишем обработчик события OnClick созданной статически кнопки Button1.

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    Button : TButton;  
begin  
    Button:=TButton(Form1.FindComponent(Edit1.Text));  
    if Button<>nil then Button.Click;  
end;
```