

# ПОДПРОГРАММЫ: ПРОЦЕДУРЫ, ФУНКЦИИ

---

# ОБЪЯВЛЕНИЕ

---

- Процедуры и функции объявляются в разделе **InterFace**, а описание (текст) подпрограммы создается в разделе реализаций **Implementation**.
- Если объявление включить в описание класса формы, то в заголовке подпрограммы в разделе реализаций имя подпрограммы указывается с именем класса формы.

# ОБЪЯВЛЕНИЕ ПРОЦЕДУРЫ В КЛАССЕ ФОРМЫ

---

Unit1

Interface

Uses ...

Type

TForm1= class(Tform)

Button1:Tbutton;

.

.

.

Procedure <имя>[(<формальные параметры>)];

Private ...

Public ...

End;



# ОПИСАНИЕ ПРОЦЕДУР

---

Implementation

{ \$R \*.dfm }

Procedure <имя формы>.<имя процедуры>

[( <формальные параметры> )];

Const <раздел описания констант>;

Var <раздел описания переменных>;

Begin

... <тело процедуры – раздел  
операторов процедуры>;

End;

Примеры 22\1 Процедуры\Project1.dpr

(Показать объявление и текст процедур Kv, Kv1, Kv2)

## ОБЪЯВЛЕНИЕ ПРОЦЕДУРЫ ВНЕ КЛАССА ФОРМЫ

---

- В заголовке подпрограммы в разделе реализаций имя подпрограммы указывается без имени класса формы. С именами объектов, включенных в класс формы, нужно использовать имя формы.

Unit1

Interface

---

Uses ...

Type

TForm1= class(Tform)

Button1:Tbutton;

·  
·  
·

Private ...

Public ...

End;

Procedure <имя>[(<формальные параметры>)];

Var .... //объявление процедуры – до var или

после



# ОПИСАНИЕ ПРОЦЕДУР

---

Implementation

{ \$R \*.dfm }

Procedure <имя процедуры> [( <формальные  
параметры> )];

Const

Var

Begin

...

End;

Примеры 22\1 Процедуры\Project1.dpr

(Показать объявление и текст процедуры Step1)

- 
- Для параметров в объявлении процедур и в заголовке текста процедуры указываются типы параметров.



# ВЫЗОВ ПРОЦЕДУР

---

- Вызов процедур выполняется в обработчиках событий объектов, размещенных на формах или из других процедур:
- Если процедура объявлена в классе формы, то при вызове на этой же форме имя формы можно не указывать, на других формах – указывать:
  - <имя процедуры>[(<фактические параметры>)];
  - <имя формы>. <имя процедуры>[(<фактические параметры>)];

# ПАРАМЕТРЫ

---

- При вызове процедуры значения фактических параметров передаются формальным.
- Процедура может не иметь параметров, использовать общие переменные.

- Параметры могут передаваться:
  - по имени – параметры переменные, перед ними указывается **Var**, после выполнения процедуры переменная – фактический параметр будет иметь новое значение – значение формального параметра
  - по значению – перед ними не указывается **Var**, после выхода из процедуры значение фактического параметра остается таким же, как до вызова процедуры.
- Примеры 22\1  
Процедуры\Project1.dpr
- (Показать текст процедуры Step1, Step – изменение параметров-в процедурах, вызов, значения фактических



# ВЫЗОВ ПРОЦЕДУР – ОБРАБОТЧИКОВ СОБЫТИЙ ОБЪЕКТОВ

---

- **<Имя объекта>.<Событие>(<Имя объекта>)**
- Пример:  
**Button1.OnClick(Button1);** - вызов процедуры, выполняющейся при событии **OnClick** для объекта **Button1** – Procedure TForm1. Button1Click(...)
- **Примеры 22\1 Процедуры\Project1.dpr**
- (Кнопка Расчеты и на 2 форме)

# ФУНКЦИИ

- Объявление и описание, как процедур: объявляются в разделе **InterFace**, описание - в разделе реализаций **Implementation**.
- Также как для процедур, если объявление включить в описание класса формы, то в заголовке функции в разделе реализаций имя функции указывается с именем класса формы.
- Если объявление вне класса формы, то в заголовке имя функции - без имени класса формы. С именами объектов, включенных в класс формы, используется имя формы.

# ФУНКЦИИ

- Объявление и описание, функций такое же, как процедур: объявляются в разделе **InterFace** в классе формы или вне класса формы, описание - в разделе реализаций **Implementation**.
- Но функция – это подпрограмма, возвращающая значение конкретного типа. Тип функции указывается в объявлении и в заголовке функции.



## ПРИМЕР:

- Объявление в классе формы:
- `Function Step(x, p: integer; var m: integer):integer;`
- Описание:
- `Function TForm1.Step(x, p: integer; var m: integer): integer;`
- `var i: integer;`
- `begin`
- `m:=1;`
- `for i:=1 to p do`
- `m:=m*x;`
- `Step:=m;`
- `end;`

# ВЫЗОВ ФУНКЦИЙ

- Выполняется в обработчиках событий объектов, размещенных на формах, или из других функций и процедур, используя имя функции в командах.
- Значение, возвращаемое функцией присваивается в коде функции имени функции.
- Как для процедур, если функция объявлена в классе формы, то при вызове на этой же форме имя формы можно не указывать, на других формах – указывать:
  - `A:= Step(k, n, v);`
  - `Label5.caption:=form1.step(c, s, d);`

[Примеры 22\2 Функции\Project1.dpr](#)