

# Введение в объектно-ориентированное программирование

# Понятия класса, экземпляра класса, объекта

В основе ООП лежат понятия класса (class), сочетающего в себе как данные, так и действия над ними, и его физической реализации — объекта.

**Класс** является своеобразным типом и объявляется в разделе объявления типов. Он не включает в себя не только поля данных, но также и подпрограммы для обработки этих данных, именуемые методами, а также так называемые свойства, сочетающие в себе характеристики полей и методов.

**Объект** же представляет собой переменную соответствующего класса и задается в разделе объявления переменных.

**Экземпляр**ом класса называется объект, созданный в соответствии с объявлением класса.

# Понятия класса, экземпляра класса, объекта

ООП характеризуется тремя основными свойствами:

1. инкапсуляцией (encapsulation),
2. наследованием (inheritance)
3. полиморфизмом (polymorphism).

Введение нового типа данных потребовало пересмотреть некоторые концепции языка Паскаль:

ввести новые понятия, как, например, «инкапсуляция», «наследование», «полиморфизм» и «виртуальность», новые зарезервированные слова, новые приемы работы с компонентами этого типа.

# Понятия класса, экземпляра класса, объекта

**Инкапсуляция** означает объединение в одном классе и данных, и действий над ними.

**Наследование** позволяет создавать иерархию классов, начиная с некоторого первоначального (предка) и кончая более сложными, но включающими (наследующими) элементы предшествующих классов (потомков).

**Полиморфизм** означает, что для различных родственных классов можно задать единый образ действий (например, перемещение по экрану любой геометрической фигуры).

# Структура класса

Структура класса начинается с зарезервированного слова **class**, после которого в круглых скобках указывается непосредственный предок класса.

Если предок не указан, предполагается, что им не является класс *TObject*.

Далее обычно в виде отдельных строк записываются поля данных, методы (подпрограммы класса) и свойства.

Завершается класс зарезервированным словом **end**.

# Пример объявления элементов класса:

```
type
  TForm1 = class (TForm) {Объявление класса-потомка через класс-предок}
    Button1 : TButton;
    Button2 : TButton;
    procedure Button1Click (Sender : TObject);
    procedure Button2Click (Sender : TObject);
  private
    {Скрытые элементы}
    FNumber : Integer;
  protected
    {Защищенные элементы}
    procedure SetNumber (ANumber : Integer); virtual;
  public
    {Общедоступные элементы}
    Count : Integer;
  Published
    {Опубликованные элементы}
  property Number : Integer read FNumber write SetNumber;
end;
```



## Пример объявления элементов класса:

В основе приведенного примера – типичное для Delphi определение класса формы, генерируемое автоматически, при добавлении новой формы к проекту (первая форма создается автоматически, при создании проекта).

Описание должно размещаться в разделе *interface* описания модуля формы, после ключевого слова *type* (т.к. класс по-сути является новым типом данных).

Описание тел методов класса (действий, совершаемых при вызове того или иного метода) осуществляется в разделе *implementation* модуля

Для классов характерны следующие моменты:

- Управление доступом
- Конструкторы
- Деструктор
- Поля
- Методы (функции и процедуры)
- Скрытый указатель специального назначения Self