

**ОБЪЕКТНО –  
ОРИЕНТИРОВАННОЕ  
ПРОГРАММИРОВАНИЕ :  
ВИДИМОСТЬ АТТРИБУТОВ  
ОБЪЕКТОВ**

# Атрибут видимости

- Подобное описание понятия инкапсуляции связано с понятием области видимости идентификатора. Область видимости идентификатора (имени переменной, процедуры, функции или типа данных) - это часть программного кода, в которой возможен доступ к этому идентификатору. Область видимости идентификатора компонента, объявленного в описании класса, простирается от его объявления до конца определения класса, а также распространяется на все потомки этого класса и на все блоки реализации методов класса. Область видимости идентификатора компонента зависит от атрибута видимости раздела, в котором объявлен идентификатор.

# Атрибуты видимости

В Delphi используется пять атрибутов видимости, называемых также директивами:

- ◎ *published*
- ◎ *public*
- ◎ *protected*
- ◎ *private*
- ◎ *automated*

# Разделы объявлений

В объявлении типов классов имеются разделы объявлений.

- ⦿ частные (private)
- ⦿ общие (public)

# Private

В разделе частных (**private**) объявлений размещаются поля данных и методы, недоступные за пределами модуля, содержащего объявление данного класса. Данные, описанные в этом разделе, могут обрабатываться только путем вызова методов внутри класса, а также внутри данного модуля. За пределами класса все его частные элементы неизвестны и считаются несуществующими.

# Public

Поля данных и методы, объявленные в разделе общих (**public**) объявлений класса, доступны для всех процедур, программный код которых расположен в области видимости данного объекта. В разделе общих объявлений типа класса должны быть объявлены поля данных и методы, к которым будут иметь доступ методы объектов других модулей.

# Protected

С атрибутом видимости **protected** объявляются те методы, к которым за пределами данного модуля могут иметь доступ только те методы классов, порожденных от данного класса.

# Published

Директива **published** похожа на другие атрибуты видимости (`private`, `public`, `protected`) тем, что она может встречаться только в объявлении типа класса. Опубликованное (**published**) поле или метод может использоваться не только во время выполнения программы, но и во время её разработки. Все компоненты в палитре компонентов Delphi располагают `published`-интерфейсом, который используется в первую очередь инспектором объектов. Правила видимости для директивы `published` - те же, что и для `public`.

Различие между общими (`public`) и опубликованными (`published`) элементами состоит в том, что во время выполнения программы можно получить информацию о типах (RTTI - Run-time type information) опубликованных элементов класса. С помощью этой информации в приложении можно динамически определить и использовать поля и свойства любого, в том числе и неизвестного, типа класса.

# Automated

Директива **automated** введена только для совместимости с Delphi 2.0. Новый класс TAutoObject в Delphi 5 уже эту директиву не использует.