

В немного более общем случае синтаксис наследования выглядит следующим образом:

```
struct Derived : <modifier> Base { };
```

где <modifier> — это одно из ключевых слов `public`, `protected` или `private`. Нам на данном этапе интересует только публичное (`public`) наследование, поэтому смысл других модификаторов не столь важен. Если модификатор не указан явно, то используется модификатор по-умолчанию, и важно знать, что для структур по-умолчанию используется модификатор `public` (как раз то, что нам нужно), т.е. в примере:

```
struct Student : Person { };
```

`Student` публично (`public`) наследуется от класса `Person`. А для классов (в противовес структурам) по-умолчанию используется модификатор `private`, т.е. в примере:

```
class Student : Person { };
```

`Student` приватно (`private`) наследуется от класса `Person` (не то, чего мы хотим). Чтобы унаследовать класс `Student` от класса `Person` публично (то, что мы хотим), необходимо явно указать модификатор:

```
class Student : public Person { };
```

При этом не важно, является ли `Person` классом или структурой, важно лишь то, является ли наследник (`Student`) структурой или классом (в терминах C++). Эта тема будет подробнее раскрыта в конце этой недели.