**6.1. Понятие наследования. Модификаторы наследования**

Важнейшим свойством объектно-ориентированного программирования является *наследование*. Наследование – это способ повторного использования программного обеспечения, при котором новые производные классы (наследники) создаются на базе уже существующих базовых классов (родителей). При создании новый класс является наследником членов и методов ранее определенного базового класса. Создаваемый путем наследования класс является производным (derived class), который в свою очередь может выступать в качестве базового класса (based class) для создаваемых классов. Если имена методов производного и базового классов совпадают, то методы производного класса перегружают методы базового класса.

При описании класса в его заголовке перечисляются все классы, являющиеся для него базовыми. Возможность обращения к элементам этих классов регулируется с помощью **модификаторов наследования** private, protected и public:

class имя : [private | protected | public] базовый\_класс{тело класса};

Если базовых классов несколько, они перечисляются через запятую. Перед каждым может стоять свой модификатор наследования. По умолчанию он private.

Если задан модификатор наследования public, наследование называется *открытым*. Использование модификатора protected делает наследование*защищенным*, а модификатора private - *закрытым*.

До сих пор мы рассматривали только **спецификаторы доступа** private и public, применяемые к элементам класса. Для любого элемента класса может также использоваться спецификатор protected, который для одиночных классов, не входящих в иерархию, равносилен private. Разница между ними проявляется при наследовании. Возможные сочетания модификаторов и спецификаторов доступа приведены в таблице:

|  |  |  |
| --- | --- | --- |
| **Модификатор наследования** | **Спецификатор базового класса** | **Доступ в производном классе** |
| private | private  protected  public | нет  private  private |
| protected | private  protected  public | нет  protected  protected |
| public | private  protected  public | нет  protected  public |

Как видно из таблицы, private элементы базового класса в производном классе недоступны вне зависимости от ключа. Обращение к ним может осуществляться только через методы базового класса.

Элементы protected при наследовании с ключом private становятся в производном классе private, в остальных случаях права доступа к ним не изменяются.

Доступ к элементам public при наследовании становится соответствующим ключу доступа.

Если базовый класс наследуется с ключом private, можно выборочно сделать некоторые его элементы доступными в производном классе, объявив их в секции public производного класса с помощью операции доступа к области видимости:

class Base{...

public: void f();};

class Derived : private Base{...

public: Base::void f();};