**5.2. Дружественные функции и классы**

Иногда желательно иметь непосредственный доступ извне к скрытым полям класса, то есть расширить интерфейс класса. Для этого служат дружественные функции и дружественные классы.

**Дружественная функция**

Дружественные функции применяются для доступа к скрытым полям класса и представляют собой альтернативу методам. Метод, как правило, описывает свойство объекта, а в виде дружественных функций оформляются действия, не являющиеся свойствами класса, но концептуально входящие в его интерфейс и нуждающиеся в доступе к его скрытым полям, например, переопределенные операции вывода объектов.

Ниже перечислены правила описания и особенности дружественных функций.

* Дружественная функция объявляется *внутри класса*, к элементам которого ей нужен доступ, с ключевым словом **friend**. В качестве параметра ей должен передаваться объект или ссылка на объект класса, поскольку указатель **this** ей не передается.
* Дружественная функция может быть обычной функцией или методом другого ранее определенного класса. На нее не распространяется действие спецификаторов доступа, место размещения ее объявления в классе безразлично.
* Одна функция может быть дружественной сразу нескольким классами.

В качестве примера ниже приведено описание двух функций, дружественных классу **А**. Функция F является методом класса **В**, а функция G не принадлежит ни одному классу. Обеим функциям в качестве параметра передается ссылка на объект класса **А.**

class А; // Предварительное объявление класса

class В{

     public:

void F(A a);

     ...

};

class A{

     ...

     friend int G(A a);

     friend void B::F(A a);

     // Класс B должен быть определен ранее

};

int G(A a){…}

void B::F (A a){…}

    Использования дружественных функций нужно по возможности избегать, поскольку они нарушают принцип инкапсуляции и, таким образом, затрудняют отладку и модификацию программы.

**Дружественный класс**

Если все методы какого-либо класса должны иметь доступ к скрытым полям другого, весь класс объявляется дружественным с помощью ключевого слова **friend**. В приведенном ниже примере класс **B** объявляется дружественным классу **A**:

class A{

     ...

     friend class B;

}

class B{

     ...

     void f1();

     void f2();

}

Функции **f1** и **f2** являются дружественными по отношению к классу **A** (хотя и описаны без ключевого слова **friend**) и имеют доступ ко всем его полям. Объявление **friend** не является спецификатором доступа и не наследуется.