6.2. Перегрузка операторов в шаблонном классе

Перегрузка шаблонного оператора побитового сдвига

Перегрузка шаблонного `operator<<`

Т.к. дружественная функция не является членом класса, то объявления шаблона перед именем класса не распространяется на эту функцию, и для нее шаблон нужно объявлять отдельно.

Необходимо записывать следующее (т.к. мы "френдим" целый шаблон, а не только его конкретное применение (специализацию, англ. specialization), в случае чего было бы достаточно записать `<>` после `operator<<`):

```
template<typename T>
friend std::ostream& operator<<(std::ostream& out, const MyClass<T>& classObj);
```

На самом деле нет необходимости записывать оператор как дружественную функцию, если не требуется доступ до приватных и защищенных полей класса.

```
// before class definition ...
template <class T>
class MyClass;

// note that this "T" is unrelated to the T of MyClass !
template<typename T>
std::ostream& operator<<(std::ostream& out, const MyClass<T>& classObj);

// in class definition ...
friend std::ostream& operator<< <>(std::ostream& out, const MyClass<T>&
classObj);
```

В обоих описанных случаях мы объявляем специализации как "друзей" класса, однако в первом случае мы объявляем все специализации как "друзей" (помним, что функция не имеет отношения к классу, поэтому шаблон у нее независимый от шаблона класса, который в этом случае может и отличаться от шаблона класса), во втором же случае дружественными будут только те специализации `operator<<` , у которых совпадет T специализации оператора с T специализации класса, с которым будем "дружить".

Подробнее по теме:

MSDN Дружественные шаблоны

IBM Friends and templates