

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение
высшего профессионального образования
"Донской государственный технический университет"
ДГТУ

Кафедра "Программное обеспечение вычислительной техники и
автоматизированных систем"

Перегрузка операций

Методические указания к лабораторным работам по дисциплинам "Языки программирования", "Объектно-ориентированное программирование", "Технологии объектно-ориентированного программирования"

Ростов-на-Дону

20 г.

Составитель: к.ф.-м.н., доц. Габрельян Б.В.

УДК 512.3

Друзья класса. Перегрузка операций: методические указания – Ростов н/Д: Издательский центр ДГТУ, 20 . – с.

В методической разработке рассматриваются общие правила перегрузки операций и использования для этого дружественных функций в C++. Даны задания по выполнению лабораторной работы. Методические указания предназначены для студентов специальностей 230105 "Программное обеспечение вычислительной техники и автоматизированных систем", 010503 "Математическое обеспечение и администрирование информационных систем", 090102 "Компьютерная безопасность".

© Издательский центр ДГТУ, 20

I. Перегрузка операций в C++.

В C++ можно переопределить большинство значков операций заданных для стандартных типов для собственных классов. Например, если A это класс, а a1 и a2 объекты этого класса, то компилятор не знает, как выполнить сложение двух его объектов, т.е. значение выражения $a1 + a2$ может быть вычислено, только если для класса будет задана функция или метод, с алгоритмом такого вычисления. Такая функция (или такой метод) должна иметь строго определенное имя, состоящее из зарезервированного в C++ слова operator и значка операции (через пробел или без пробела, за исключением операций new и delete, для которых пробел обязателен). В общем случае выражение $a1 + a2$ компилятор трактует либо как вызов метода для объекта a1

```
a1.operator+( a2 ),
```

либо как вызов внешней функции с двумя аргументами a1 и a2

```
operator+(a1,a2).
```

Общие правила перегрузки операций в C++ таковы:

1. Нельзя задать новый значок операции.
2. Нельзя изменить арность (число аргументов), приоритет и ассоциативность операции.
3. Нельзя переопределить операцию для стандартного типа.
4. Нельзя переопределять операции "запятая", sizeof, ::, ., .*, ?:, typeid, throw, dynamic_cast<>, static_cast<>, const_cast<>, reinterpret_cast<>.

Некоторые другие ограничения рассмотрены ниже.

II. Перегрузка операций с помощью методов класса.

Операции =, [], (), -> могут быть перегружены только методами класса. Если перегруженная операция реализована как метод, то ее единственный (для унарных операций) или первый/левый аргумент (для бинарных операций) доступен методу через скрытый аргумент this. Например,

```
class A {  
    int a;
```

```

public:
    A(int x) { a = x; }
    int getA() { return a; }
    void setA(int x) { a = x; }
    A operator+(A& right) { /* перегруженная операция для сложения
объектов класса A */
        return A( a + right.a );
    }
};

int main() {
    A a1(1), a2(2), a3(3);
    a1.setA( 5 );
    a2.setA( 10 );
    a3 = a1 + a2; // a3 = a1.operator+(a2);
    return 0;
}

```

III. Перегрузка операций с помощью внешних функций.

Бывают такие виды операций, когда их перегрузка с помощью методов невозможна. Например, класс *A* рассмотренный выше позволяет попарно складывать свои объекты, но не позволяет добавить к целому числу объект класса. Вообще, если левый аргумент операции не является объектом нашего класса, такую операцию нельзя перегрузить с помощью метода нашего класса. Приходится использовать внешнюю функцию и, что дать ей прямой доступ к закрытым полям класса, такую функцию часто объявляют дружественной классу. Характерным примером, кроме приведенного выше, является также желание переопределить для класса операцию помещения в поток (или извлечения из потока). Левым аргументом такой операции должен быть поток, т.е. объект чужого класса *ostream*. Например,

```

#include <iostream>

using namespace std;

class A {
    friend ostream& operator<<(ostream&,A&); /* переопределенная опе-
рация помещения в поток */
    int a;
public:
    A(int x) { a = x; }
};

ostream& operator<<(ostream& o, A& x) {
    return o<<x.a;
}

int main() {
    A aObj( 10 );
    cout<<aObj<<endl; // помещение объекта в поток
    return 0;
}

```

IV. Задания.

1. Описать класс Test с защищенными числовым полем W и функцией Z, которая выводит сообщение "Это закрытая функция класса Test". Написать конструктор для инициализации объектов класса Test с одним параметром, принимающим по умолчанию значение 1. Объявить другом класса функцию fun, которая не возвращает значений и получает указатель на объект типа Test.
2. Описать на внешнем уровне функцию fun, которая выводит на экран значение параметра W и вызывает из класса Test функцию Z.

3. В функции main описать переменную класса Test (без явной инициализации) и применить к ней функцию fun.
4. Придумать и реализовать программу - пример использования двух классов A и B, в которой A друг B.
5. Создать класс Complex, представляющий понятие комплексных чисел с переопределенными операциями сложения, вычитания, помещения в стандартный поток вывода, извлечения из стандартного потока ввода.
6. Создать класс Array, представляющий понятие динамического массива (т.е. массива, размер которого может изменяться во время работы программы) с перегруженными операциями добавления нового элемента, удаления элемента с указанным индексом и индексирования.

V. Контрольные вопросы.

1. В чем отличие операции присваивания от конструктора копирования?
2. Назначение друзей класса?
3. Какие операции можно реализовать только методами класса?
4. В каких случаях операцию можно перегрузить только с помощью внешней функции?
5. Можно ли операцию вставки в поток реализовать как функцию-член класса?
6. Как переопределить постфиксную и префиксную формы операции инкремента?
7. Для каких классов удобно переопределить операцию индексирования?

Литература

1. Brian W. Kernighan, Dennis M. Ritchie. "The C programming language". Second edition – Prentice Hall. – 1988, 272 p.
2. Стэнли Б. Липпман. "Язык программирования C++. Вводный курс". – М.: Вильямс. – 2007, 896 с.

3. Г. Шилдт. "Самоучитель С++". – СПб.: БХВ-Петербург. – 2001, 688 с.

4. К. Арнольд, Дж. Гослинг, Д. Холмс. "Язык программирования Java". 3-е изд. – М.: Вильямс. – 2001, 624 с.