Операторы

Оператор – функция, имеющая
Фиксированное число аргументов
и специальный синтаксис.

$x = a + b;$

`operator = (x, operator + (a, b));`

Перегрузка операторов

Перегрузка – это определение его для типов аргументов, для которых он еще не определен.

Невозможно переопределить оператор, изменив при этом количество аргументов.

Перегрузка операторов, как методов класса

```
//complex.h
```

```
class Complex{
```

```
public:
```

```
    Complex operator + (double arg) const;
```

```
    //...
```

```
};
```

```
//complex.cpp
```

```
Complex Complex::operator + (double arg){
```

```
    return Complex (real + arg, image);
```

```
}
```

Перегрузка операторов, как внешних функций

```
//complex.h  
class Complex{  
    friend Complex operator + (const Complex&, double );  
    friend Complex operator + (double, const Complex&);  
    //...  
};
```

```
Complex operator + (const Complex& arg1, double arg2);  
Complex operator + (double arg1, const Complex& arg2);
```

```
//complex.cpp  
Complex operator + (const Complex& arg1, double arg2){  
    return Complex (arg1.real + arg2, arg1.image);  
}
```

Перегружаемые операторы

- Операторы присваивания: `=, +=, ++ ...`
- Операторы арифметические, битовые, логические:
`+, >>, &, &&, !, ==`
- Операторы разыменования: `*, [], ->`
- Операторы преобразования типов: `(type)`
- Оператор вызова функции
- Операторы: `new, delete`

Неперегружаемые операторы

- Арифметический условный оператор: `?:`
- Обращение к полю класса (структуры): `.`
- Разрешение пространства имен: `::`
- Определение размера:
`sizeof object, sizeof (type)`

Операторы присваивания

```
//complex.h
```

```
class Complex{
```

```
public:
```

```
    Complex& operator = (const Complex& arg);
```

```
};
```

```
//complex.cpp
```

```
Complex& Complex::operator = (const Complex& arg){
```

```
    real = arg.real;
```

```
    image = arg.image;
```

```
    return *this;
```

```
}
```

Операторы инкремента/декремента

```
//complex.h
class Complex{
public:
    Complex& operator ++ ();           //prefix
    Complex operator ++ (int dummy); //postfix
};

//complex.cpp
Complex& Complex::operator ++ (){
    ++real;
    return *this;
}
Complex Complex::operator ++ (int dummy){
    Complex copy (*this);
    ++real;
    return copy;
}
```

Арифметические операторы

- Аргументы операторов могут иметь разные типы (Complex + double)
- Операторы как правило константные.
- Результатом как правило является объект того же класса, что и один из аргументов, возвращаемый по значению

Операторы преобразования ТИПОВ

```
//complex.h
```

```
class Complex{
```

```
//...
```

```
    operator double () const;
```

```
//...
```

```
};
```

```
//complex.cpp
```

```
operator double () const{
```

```
    return sqrt (real*real + image*image); //abs
```

```
}
```