Программирование на языке C++
Лекция 5

Правила переопределения операторов

Александр Смаль

Переопределение арифметических и битовых операторов

```
struct String {
    String( char const * cstr ) { ... }

    String & operator+=(String const& s) {
        ...
        return *this;
    }
    //String operator+(String const& s2) const {...}
};

String operator+(String s1, String const& s2) {
    return s1 += s2;
}
```

```
    String s1("world");
    String s2 = "Hello " + s1;
```

“Правильное” переопределение операторов сравнения

```
bool operator==(String const& a, String const& b) {
    return ...
}
bool operator!=(String const& a, String const& b) {
    return !(a == b);
}
bool operator<(String const& a, String const& b) {
    return ...
}
bool operator>(String const& a, String const& b) {
    return b < a;
}
bool operator<=(String const& a, String const& b) {
    return !(b < a);
}
bool operator>=(String const& a, String const& b) {
    return !(a < b);
}
```

О чём стоит помнить

- Стандартная семантика операторов.

```
void operator+(A const & a, A const& b) {}
```

- Приоритет операторов.

```
Vector a, b, c;
c = a + a ^ b * a; //?????
```

- Хотя бы один из параметров должен быть пользовательским.

```
void operator*(double d, int i) {}
```