창의적 소프트웨어 설계

6 주차 실습 – Overloading and Friend

노인우 , inwoo13@hanyang.ac.kr
한중수 , soohan@hanyang.ac.kr

Overview

목표

- Operator Overloading
- Friend Class and Function

Operator Overloading

- Argument 에 의해 분리
  - int operator+ (int n, int m){ ... }
  - int operator+ (double n, double m){ .. }

- const 에 의해 분리
  - int operator+ (int n, int m){ ... }
  - const int operator+ const (int n, int m){ .. }

Friend Class and Function

- one-way!
- why do we need a “friend”?

Example

```
#include <stdio.h>
#include <string>

class Complex;

class Tester
{
public:
    double testfunc(Complex& c);
};
```

Example

```
struct Complex
{
public:
    Complex() : real(0.0), imag(0.0) {}
    Complex(double v) : real(v), imag(0.0) {}
    Complex(double r, double i) : real(r), imag(i) {}
    Complex(const Complex& c) : real(c.real), imag(c.imag) {}

    Complex& operator = (const Complex& c){
        real = c.real, imag = c.imag;
        return *this;
    }

    Complex operator+ () const { return *this; }
    Complex operator- () const { return Complex(-real, -imag); }
```

Example

```
 double& operator[] (int i) {
    printf("no const\n");
    return i == 0 ? real : imag;
}

const double& operator[] (int i) const {
    printf("const\n");
    return i == 0 ? real : imag;
}

void show(std::string sz_prefiex){
    printf("[%s] real: %lf, imag: %lf\n", sz_prefiex.c_str(), real, imag);
}
```

Example

```
private:
    double real, imag;

    friend Complex operator+ (const Complex& lhs, const Complex& rhs);
    friend bool operator< (const Complex& lhs, const Complex& rhs);

    friend double Tester::testfunc(Complex &c);
};
```

Example

```
double Tester::testfunc(Complex& c){
    printf("[Tester] %lf, %lf", c.real, c.imag);
    return c.real;
}


Complex operator+ (const Complex& lhs, const Complex& rhs){
    return Complex(lhs.real + rhs.real, lhs.imag + rhs.imag);
}


bool operator< (const Complex& lhs, const Complex& rhs){
    return lhs.real < rhs.real && lhs.imag < rhs.imag;
}
```

Example

```
int Test(){
    Complex a(11.0, 2.0), b(2.0, 5.0), c;
    const Complex cc(33.0, 11.0);
    Tester t;

    // overloading test
    a[n];
    cc[1];

    c = a + b;
    c.show("c = a + b");
    (a+b).show("(a+b)");

    // friend test
    t.testfunc(a);

    return 0;
}
```

Example

```
int main(){
    Test();


    return 0;
}
```

참고자료

1. friend class, http://genesis8.tistory.com/98


Q&A

Appendix #1. Static Member Function

```
#include <iostream>

class StrTest{
private:
    int iTest;
    static int m_version;

public:
    int pubTest;

    static int getVersion(){
        m_version = 10;
        return m_version;
    }

    StrTest(){
        std::cout << "constructor" << std::endl;
    }

    ~StrTest(){
        std::cout << "destructor" << std::endl;
    }
};
```

```
int StrTest::m_version;

int main(){
StrTest* st = new StrTest();

int result = StrTest::getVersion();

std::cout << "Result Version : " << result <<
std::endl;

delete st;

    return 0;
}
```