Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

(ПНИПУ)

**Факультет:** электротехнический

**Направление:** Промышленная мехатроника и робототехника

ОТЧЁТ

По лабораторной работе №3

«Классы»

Семестр 1

Выполнил:

студент группы ПРТ-21-1б

Поздняков Е.С.

Пермь 2022

**Задание №1**

**Постановка задачи**

Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: long (для рублей) и int (для копеек). Реклизовать:

- операцию сравнения !=

- вычитание копеек --(постфиксная и префиксная формы)

**Анализ**

1. Создаётся два объекта a и b класса Money. Пользователь вводит значения каждого из полей.
2. При помощи перегруженных операторов вычитания из значений копеек обоих объектов вычитается единица.
3. При помощи перегруженного оператора вывода выводятся значения полей объектов.
4. Объявляется логическая переменная f и инициализируется значением перегруженной функции сравнения a и b.

**Используемые переменные**

Money a, b – объекты класса Money;

bool f – результат выполнения перегруженной функции сравнения значений двух объектов.

**Код C++**

#include <iostream>

using namespace std;

class Money

{

long rub;

int kop;

public:

Money() : rub(0), kop(0) {}

Money(long r, int k) : rub(r), kop(k) {}

Money(const Money &t)

{

rub = t.rub; kop = t.kop;

}

~Money(){}

void SetRub(long r){rub = r;}

void SetKop(int k){kop = k;}

long GetRub(){return rub;}

int GetKop(){return kop;}

Money operator --()

{

Money t;

int temp = rub \* 100 + kop;

--temp;

t.rub = temp / 100;

t.kop = temp % 100;

return t;

}

Money operator --(int)

{

Money t;

int temp = rub \* 100 + kop;

temp--;

t.rub = temp / 100;

t.kop = temp % 100;

return t;

}

bool operator ==(const Money& t)

{

if ((rub == t.rub) && (kop == t.kop))

return true;

else

return false;

}

bool operator !=(const Money& t)

{

if ((rub != t.rub) && (kop != t.kop))

return true;

else

return false;

}

friend istream& operator >> (istream&in, Money&t);

friend ostream& operator << (ostream&out, const Money&t);

};

istream&operator>>(istream&in, Money&t)

{

cout << "Input rub: ";

in >> t.rub;

cout << "Input kop: ";

in >> t.kop;

return in;

}

ostream& operator<<(ostream&out, const Money&t)

{

return (out << t.rub << "," << t.kop);

}

int main()

{

Money a, b;

cin >> a;

cin >> b;

a = --a; b = b--;

cout << a << " " << b << endl;

bool f = (a == b);

cout << "a = " << a << endl;

cout << "b = " << b << endl;

cout << "f = ";

if (f == 0)

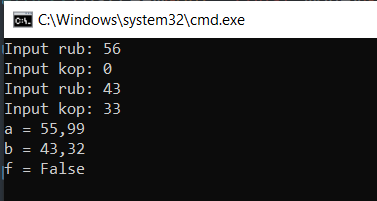
cout << "False" << endl;

else cout << "True" << endl;

return 0;

}

**Скриншоты результатов**

****

**Анализ результатов**

Пользователь вводит значения полей двух объектов, из значений вычитается 1 копейка. Полученные значения выводятся. Если значения равны, выводится true, иначе – false.

**Контрольные вопросы**

1. Для чего используются дружественные функции и классы?

Для доступа извне к скрытым полям класса.

2. Сформулировать правила описания и особенности дружественных функций.

-Дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ, с ключевым словом friend.

-В качестве параметра ей должен передаваться объект или ссылка на объект класса, поскольку указатель this ей не передаётся

-Дружественная функция может быть обычной функцией или методом другого ранее определенного класса. На нее не распространяется действие спецификаторов доступа, место размещения ее объявления в классе безразлично

-Одна функция может быть дружественной сразу нескольким классам

3. Каким образом можно перегрузить унарные операции?

- как компонентную функцию класса (тип operator знак())

-как внешнюю функцию (тип operator знак(class A))

4.Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?

0, так как неявно принимает указатель this.

5.Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?

1, надо явно передавать объект класса, над которым будет производиться операция.

6.Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?

1, левым операндом по умолчанию будет являться объект класса, правым – объект, передаваемый в качестве параметра.

7. Сколько операндов должна иметь бинарная функция-операция, определяемая вне класса?

2, так как указатель this не передается.

8.Чем отличается перегрузка префиксных и постфиксных унарных операций?

Префиксная форма – operator++(); постфиксная – operator++(int);

9.Каким образом можно перегрузить операцию присваивания?

Только как функция-компонент класса

10.Что должна возвращать операция присваивания?

Операция присваивания должна возвращать ссылку на объект, в который происходит копирование, и принимать в качестве параметра ссылку на присваиваемый объект.

11.Каким образом можно перегрузить операцию ввода-вывода?

Всегда перегружаются как внешние дружественные функции, так как левым операндом этих операций являются потоки.

12.В программе описан класс

class Student

{

…

Student& operator++();

…

};

и определен объект этого класса

Student s;

Выполняется операция ++s

Каким образом компилятор будет воспринимать вызов функции-операции?

Как вызов метода класса: s.operator++();

13. В программе описан класс

class Student

{

…

friend Student& operator++(Student&);

…

};

и определен объект этого класса

Student s;

Выполняется операция ++s

Каким образом компилятор будет воспринимать вызов функции-операции?

Как вызов глобальной функции

14. В программе описан класс

class Student

{

…

bool operator<(Student& P);

…

};

и определен объект этого класса

Student a,b;

Выполняется операция cout << a < b;

Каким образом компилятор будет воспринимать вызов функции-операции?

Если бы было записано cout << (a < b), то компилятор это воспринял бы как вызов метода класса.

15. В программе описан класс

class Student

{

…

friend bool operator >(const Person&, Person&);

…

};

и определен объект этого класса

Student a,b;

Выполняется операция cout <<a>b;

Класс Person не определен