Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

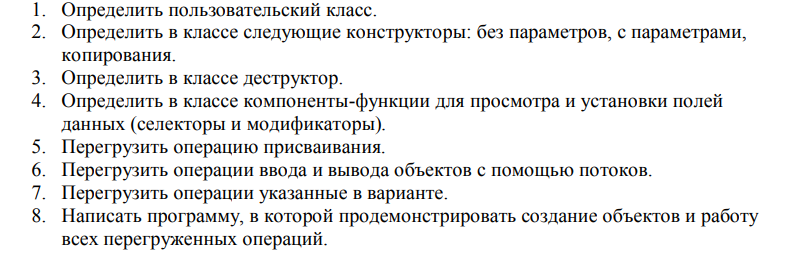
Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

 Анализ задачи:

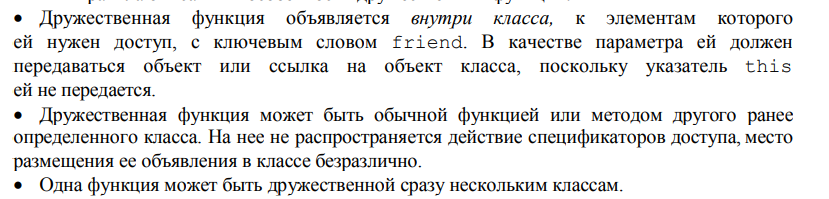
1. Класс money содержит поля long rubles и int copeyka в private секции. Доступ будем получать при помощи конструкторов и сеттеров.

2. Три конструктора – с параметрами, без них и конструктор копирования, а также установление полей с помощью сеттеров и получение значений при помощи геттеров. Реализована ф-ия show() для вывода полей нужного обьекта(в данном случае, для сеттеров, т.к. каждый конструктор имеет свой вывод). Перегрузка операторов сравнения с помощью ключевого слова «operator» и возвращаемым значением является boolево значение, если сумма у одного объекта совпадет с суммой другого, то будет получено true, иначе – false. Перегрузка оператора дикремента происходит аналогично перегрузке операторов сравнения, только в этом случае будем уменьшать количество копеек на 1 и возвращаемым значением будет объект класса.

3. В ф-ии main() происходит создание объектов, вызов конструкторов, вывод на экран булевых значений, операции префиксного(приоритетного) дикремента и постфиксного.

4. Ответы на вопросы:

1. если сумма у одного объекта совпадет с суммой другого, то будет получено true, иначе – false

2. 

3. Как метод, как внешнюю ф-ию.

4. 0.

5. 1.

6. 1. Ее операнд – вызвавший ее же объект.

7. 2 операнда типа класса.

8. префиксная приоритетнее, в нее должно(но не должно) передаваться значение.

9. Class&operator(const Class&other).

10. Объект.

11. При помощи дружественной глобальной ф-ии.

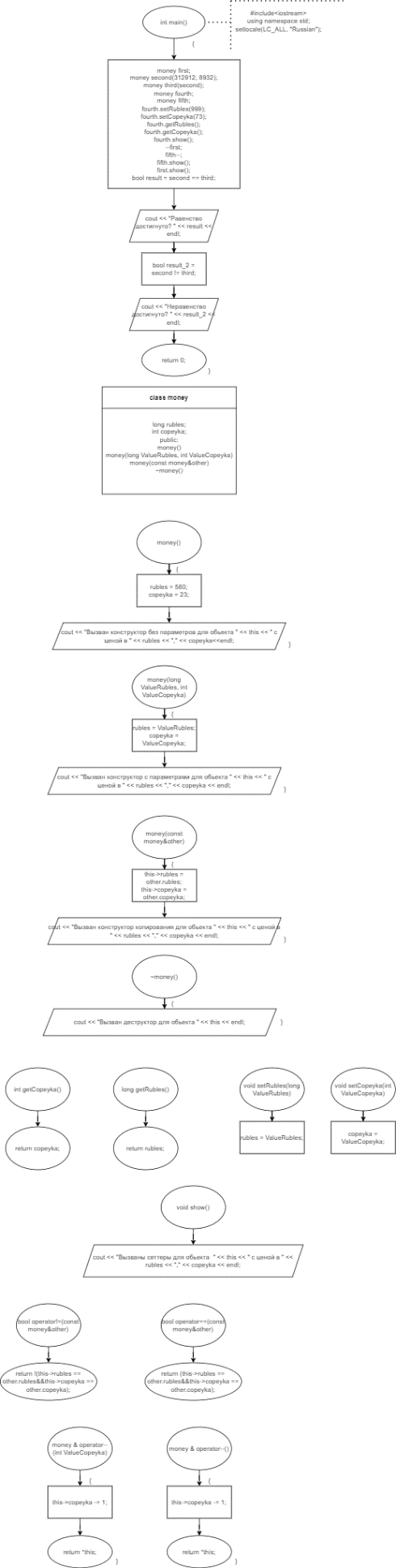
12. Префиксный инкремент.

13. Перегруженная дружественная ф-ия.

14. Если а действительно меньше b вернется 1.

15. Если а действительно больше b вернется 1.

Блок-схема:



Код программы:

#include<iostream>

using namespace std;

class money

{

long rubles;

int copeyka;

public:

money()

{

rubles = 560;

copeyka = 23;

cout << "Вызван конструктор без параметров для обьекта " << this << " с ценой в " << rubles << "," << copeyka<<endl;

}

money(long ValueRubles, int ValueCopeyka)

{

rubles = ValueRubles;

copeyka = ValueCopeyka;

cout << "Вызван конструктор с параметрами для обьекта " << this << " с ценой в " << rubles << "," << copeyka << endl;

}

money(const money&other)

{

this->rubles = other.rubles;

this->copeyka = other.copeyka;

cout << "Вызван конструктор копирования для обьекта " << this << " с ценой в " << rubles << "," << copeyka << endl;

}

~money()

{

cout << "Вызван деструктор для обьекта " << this << endl;

}

void setRubles(long ValueRubles)

{

rubles = ValueRubles;

}

long getRubles()

{

return rubles;

}

void setCopeyka(int ValueCopeyka)

{

copeyka = ValueCopeyka;

}

int getCopeyka()

{

return copeyka;

}

void show()

{

cout << "Вызваны сеттеры для обьекта " << this << " с ценой в " << rubles << "," << copeyka << endl;

}

bool operator==(const money&other)

{

return (this->rubles == other.rubles&&this->copeyka == other.copeyka);

}

bool operator!=(const money&other)

{

return !(this->rubles == other.rubles&&this->copeyka == other.copeyka);

}

money & operator--()

{

this->copeyka -= 1;

return \*this;

}

money & operator--(int ValueCopeyka)

{

this->copeyka -= 1;

return \*this;

}

};

int main()

{

setlocale(LC\_ALL, "Rus");

money first;

money second(312912, 8932);

money third(second);

money fourth;

money fifth;

fourth.setRubles(999);

fourth.setCopeyka(73);

fourth.getRubles();

fourth.getCopeyka();

fourth.show();

--first;

fifth--;

fifth.show();

first.show();

bool result = second == third;

cout << "Равенство достигнуто? " << result << endl;

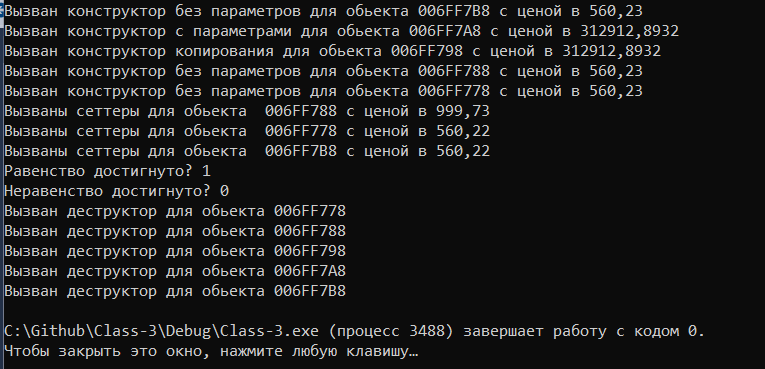
bool result\_2 = second != third;

cout << "Неравенство достигнуто? " << result\_2 << endl;

return 0;

}

Скриншоты результатов:



Анализ результатов:

Программа работает корректно. Выводятся заданные объекты, для двух из которых применяется префиксный и постфиксный дикремент, а также проверяется равенство обьектов второго и третьего.