Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Перегрузка операций

Вариант 6.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

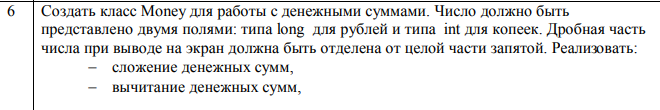
г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

1. Определить пользовательский класс.
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).
5. Перегрузить операцию присваивания.
6. Перегрузить операции ввода и вывода объектов с помощью потоков.
7. Перегрузить операции указанные в варианте.
8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.



**Код программы**

**ооп 3.cpp**

#include "Header.h"

#include <iostream>

using namespace std;

void main()

{

SperBank a;

SperBank b;

SperBank c;

cin >> a;

cin >> b;

++a;

cout << a << endl;

c = (a++) + b;

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

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

cout << "c = " << c << endl;

}

**Header.h**

**#include** <iostream>

using namespace std;

class SperBank

{

int rub, kop;

public:

SperBank() { rub = 0; kop = 0; };

SperBank(int m, int s) { rub = m; kop = s; }

SperBank(const SperBank& t) { rub = t.rub; kop = t.kop; }

~SperBank() {};

int get\_rub() { return rub; }

int get\_kop() { return kop; }

void set\_rub(int m) { rub = m; }

void set\_kop(int s) { kop = s; }

SperBank& operator=(const SperBank&);

SperBank& operator++();

SperBank operator++(int);

SperBank operator+(const SperBank&);

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

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

};

**Header.cpp**

**#include** "Header.h"

#include <iostream>

using namespace std;

SperBank& SperBank::operator=(const SperBank& t)

{

if (&t == this) return \*this;

rub = t.rub;

kop = t.kop;

return \*this;

}

SperBank& SperBank::operator++()

{

int temp = rub + kop / 100;

temp++;

rub = temp;

kop = temp / 100;

return \*this;

}

SperBank SperBank::operator ++(int)

{

int temp = rub + kop / 100;

temp++;

SperBank t(rub, kop);

rub = temp;

kop = temp / 100;

return t;

}

SperBank SperBank::operator+(const SperBank& t)

{

int temp1 = rub + kop / 100;

int temp2 = t.rub + t.kop / 100;

SperBank p;

p.rub = (temp1 + temp2);

p.kop = (temp1 + temp2) / 100;

return p;

}

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

{

cout << "rub? "; in >> t.rub;

cout << "kop? "; in >> t.kop;

return in;

}

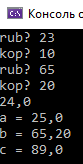
ostream& operator<<(ostream& out, const SperBank& t)

{

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

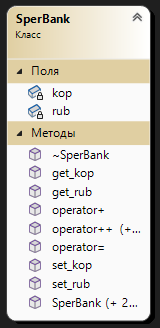
}

**Вывод программы**

****

***Рисунок 1 – вывод программы***

**UML диаграмма**

****

***Рисунок 2 – UML диаграмма***

**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.