Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.3**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Перегрузка операций

Вариант 8

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

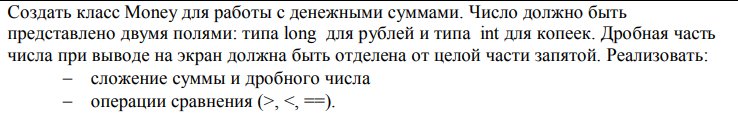
# **Цель работы**

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Изучение перегруженных функций.

# **Постановка задачи**

1. Определитель пользовательский класс.
2. Определить в классе следующие конструкторы:
3. Без параметров;
4. С параметрами;
5. Копирования;
6. Определить в классе деструктор.
7. Определить в классе компоненты-функции для просмотра и установки полей.
8. Перегрузить операции присваивания.
9. Перегрузить операции ввода и вывода объектов.
10. Перегрузить операции, указанные в варианте.
11. Написать демонстрационную программу, в которой продемонстрировать создание объектов и работу всех перегруженных функций.

(8 вариант)



# **Анализ задачи**

1. Определить какие действия предстоит выполнить:

* Разработать класс Money который хранит в себе значения рублей и копеек.
* Организовать методы для ввода данных, селекторы и модификатор, конструкторы и деструкторы, перегрузки операторов.

class Money {

long int rubles;

int penny;

public:

//Конструкторы

Money() { //без параметров

rubles = 0;

penny = 0;

}

Money(int r, int p) { //с параметрами

rubles = r;

penny = p;

}

Money(const Money& temp) { //копирования

rubles = temp.rubles;

penny = temp.penny;

}

//Деструктор

~Money() {};

//Селекторы

int get\_rubles() {

return rubles;

}

int get\_penny() {

return penny;

}

//Модификаторы

void set\_rubles(long int r) {

rubles = r;

}

void set\_penny(int p) {

penny = p;

}

//Перегруженные операции сравнения

bool operator == (const Money&);

bool operator > (const Money&) ;

bool operator < (const Money&);

//Перегруженные функции

Money& operator + (int);

Money& operator - (const Money&);

//Перегрузка функций ввода-вывода

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

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

};

* Протестировать работу перегруженных функций в функции main.

void main() {

system("chcp 1251");

Money sum1, sum2, sum3;

cout << "Сумма 1\n"; cin >> sum1; cout << "\nСумма 1: " << sum1;

cout << "\n\nСумма 2\n"; cin >> sum2; cout << "\nСумма 2: " << sum2 << "\n\nВведите сколько копеек прибавить: ";

int x, op;

cin >> x; cout << "\nК какой сумме прибавить копейки\n0 - Сумма 1\n1 - Сумма 2\n2 - К обеим суммам\nВаш выбор: "; cin >> op;

switch (op) {

case(0):

sum1 + x;

break;

case(1):

sum2 + x;

break;

case(2):

sum1 + x;

sum2 + x;

break;

default:

cout << "\nУпс ошибочка\n";

}

cout << "\nСумма 1: " << sum1 << "\n\nСумма 2: " << sum2 << endl;

if (sum1 == sum2) {

cout << "\nСуммы равны\n";

}

else if (sum1 > sum2) {

sum3 = sum1;

sum3 - sum2;

cout << "\nСумма 1 больше Суммы 2 на " << sum3 << endl;

}

else if (sum1 < sum2) {

sum3 = sum2;

sum3 - sum1;

cout << "\nСумма 1 меньше суммы 2 на " << sum3 << endl;

}

system("pause");

}

1. C какими типами данных предстояло работать:

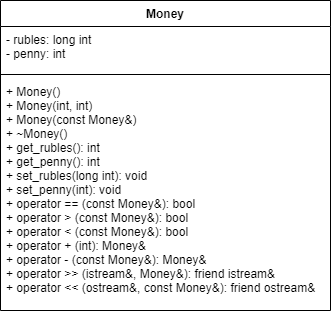
* Для хранения кол-ва рублей реализована переменная rubles типа long int.

long int rubles;

* Для хранения кол-ва копее реализована переменная penny типа int.

int penny;

# **UML диаграмма**



# **Код программы на языке C++**

Заголовочный файл Money.h

#pragma once

#include <iostream>

using namespace std;

class Money {

long int rubles;

int penny;

public:

//Конструкторы

Money() { //без параметров

rubles = 0;

penny = 0;

}

Money(int r, int p) { //с параметрами

rubles = r;

penny = p;

}

Money(const Money& temp) { //копирования

rubles = temp.rubles;

penny = temp.penny;

}

//Деструктор

~Money() {};

//Селекторы

int get\_rubles() {

return rubles;

}

int get\_penny() {

return penny;

}

//Модификаторы

void set\_rubles(long int r) {

rubles = r;

}

void set\_penny(int p) {

penny = p;

}

//Перегруженные операции сравнения

bool operator == (const Money&);

bool operator > (const Money&) ;

bool operator < (const Money&);

//Перегруженные функции

Money& operator + (int);

Money& operator - (const Money&);

//Перегрузка функций ввода-вывода

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

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

};

Обьявление класса в Money.cpp

#include <iostream>

#include "Money.h"

using namespace std;

bool Money::operator == (const Money& temp) {

return (rubles == temp.rubles && penny == temp.penny);

}

bool Money::operator > (const Money& temp) {

return (rubles \* 100 + penny > temp.rubles \* 100 + temp.penny);

}

bool Money::operator < (const Money& temp) {

return (rubles \* 100 + penny < temp.rubles \* 100 + temp.penny);

}

Money& Money::operator + (int x) {

long int temp = rubles \* 100 + penny;

temp += x;

rubles = temp / 100;

penny = temp % 100;

return \*this;

}

Money& Money::operator - (const Money& temp) {

long int sum1 = rubles \* 100 + penny;

long int sum2 = temp.rubles \* 100 + temp.penny;

sum1 -= sum2;

rubles = sum1 / 100;

penny = sum1 % 100;

return \*this;

}

istream& operator >> (istream& in, Money& temp) {

do {

cout << "Рубли: "; in >> temp.rubles;

} while (temp.rubles < 0);

do {

cout << "Копейки: "; in >> temp.penny;

} while (temp.penny>99 || temp.penny < 0);

return in;

}

ostream& operator << (ostream& out, const Money& temp) {

if (temp.penny < 10) {

return (out << temp.rubles << ",0" << temp.penny);

}

else {

return (out << temp.rubles << "," << temp.penny);

}

}

Main файл

#include "Money.h"

#include <iostream>

using namespace std;

void main() {

system("chcp 1251");

Money sum1, sum2, sum3;

cout << "Сумма 1\n"; cin >> sum1; cout << "\nСумма 1: " << sum1;

cout << "\n\nСумма 2\n"; cin >> sum2; cout << "\nСумма 2: " << sum2 << "\n\nВведите сколько копеек прибавить: ";

int x, op;

cin >> x; cout << "\nК какой сумме прибавить копейки\n0 - Сумма 1\n1 - Сумма 2\n2 - К обеим суммам\nВаш выбор: "; cin >> op;

switch (op) {

case(0):

sum1 + x;

break;

case(1):

sum2 + x;

break;

case(2):

sum1 + x;

sum2 + x;

break;

default:

cout << "\nУпс ошибочка\n";

}

cout << "\nСумма 1: " << sum1 << "\n\nСумма 2: " << sum2 << endl;

if (sum1 == sum2) {

cout << "\nСуммы равны\n";

}

else if (sum1 > sum2) {

sum3 = sum1;

sum3 - sum2;

cout << "\nСумма 1 больше Суммы 2 на " << sum3 << endl;

}

else if (sum1 < sum2) {

sum3 = sum2;

sum3 - sum1;

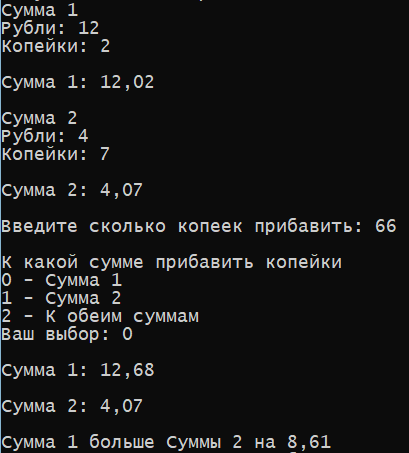
cout << "\nСумма 1 меньше суммы 2 на " << sum3 << endl;

}

system("pause");

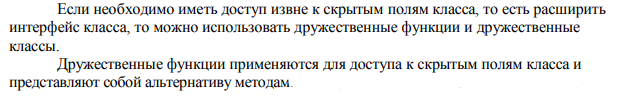
}

# **Скриншоты тестов**

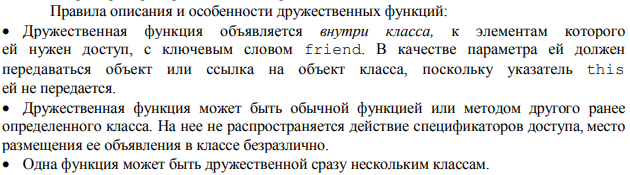


**Контрольные вопросы**





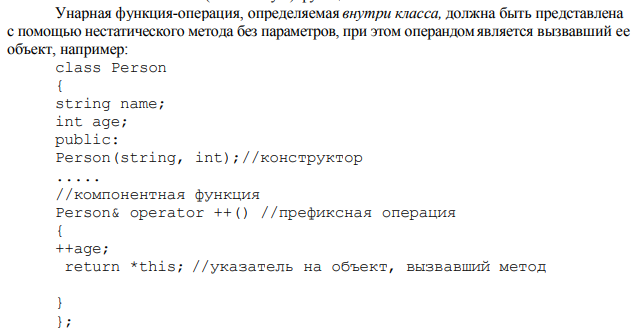




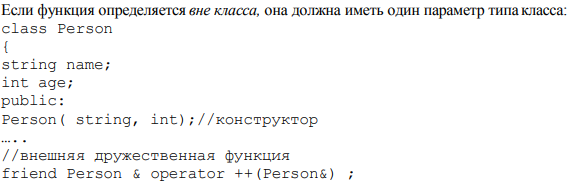


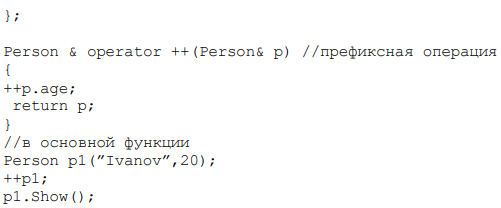




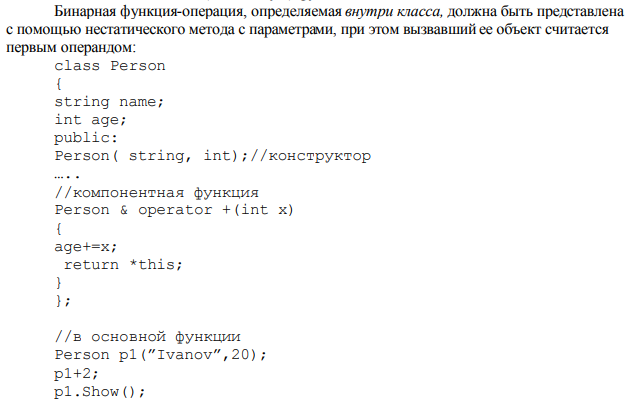




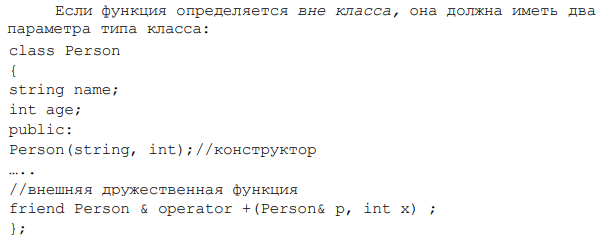








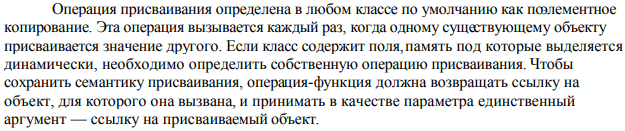










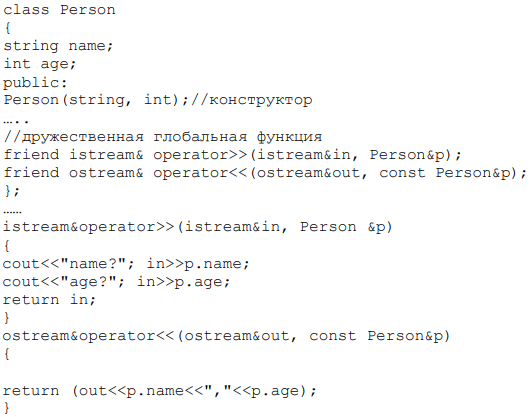


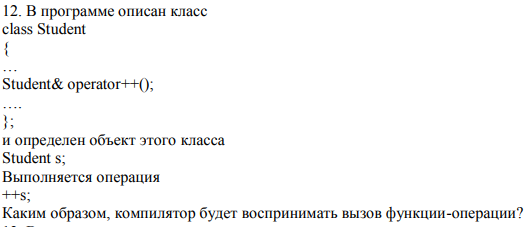


Ссылку на объект, для которого операция была вызвана.

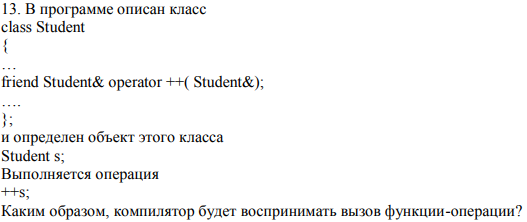




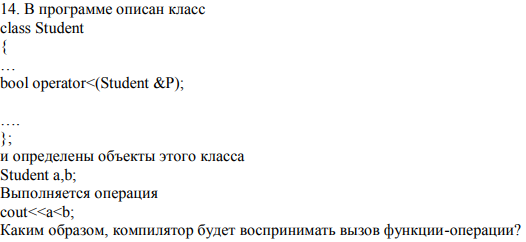




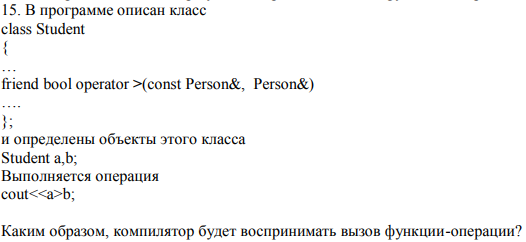
Будет осуществлён вызов функции перегрузки оператора префиксного инкремента.



Будет осуществлён вызов функции перегрузки оператора постфиксного инкремента.



Будет осуществлён вызов функции перегрузки оператора сравнения (<), затем через cout будет выведен результат сравнения (1 или 0).



Будет осуществлён вызов функции перегрузки оператора сравнения (>), затем через cout будет выведен результат сравнения (1 или 0).