

Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Дисциплина «Объектно-ориентированное программирование»

Лабораторная работа №2

Тема: Операторы, литералы

Студент:	Колесова М. И.
Группа:	М8О-206Б-18
Руководитель:	Журавлев А.А.
Оценка:	
Дата:	

1 – Задание

Создать класс Money для работы с денежными суммами. Сумма денег должна быть предоставлена двумя полями: типа unsigned long long для рублей и char для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой.

Реализовать сложение сумм, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число, и операции сравнения.

Операции сложения, вычитания, умножения, деления, сравнения (на равенство, больше и меньше) должны

быть выполнены в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа Money.

2 - Репозиторий GitHub

https://github.com/KolesovaMariya/oop_exercise_02

3 - Описание программы

Создан класс **Money**, в котором хранятся две переменные, отображающие рубли и копейки в сумме денег. Для получения этих значения написаны функции GetRub() и GetKop().

Перегружены операторы +, - (сложение и вычитание элементов суммы), операторы *, / (умножение и деление на дробное число), так же оператор / перегружен для двух элементов класса Money. Перегружены операторы сравнения <, >, ==, а так же операторы считывания со стандартного потока и вывод в стандартный поток <<, >>. Реализован пользовательский литерал _money.

Входные и выходные значения:

На вход подается целое число, которое обозначает количество пар сумм, которые будут введены, и далее вводятся два элемента класса, и одно дробное число.

Программа делает все операции, прописанные в задании, и выводит результат с пояснениями (в виде $a + b = 10$, например).

4 – Тесты

test_00.txt	test_01.txt
5 100,00 10 10 200,5 200,5 0,5 30,0 40,0 1 1,5 0,5 2 3,20 2,40 4,5	5 10 5 2 14,55 14,5 10 17,8 10 0 20,20 0 10,10 0 55,5 55,5
Сколько пар сумм вы хотите ввести? 5 Введите суммы a и b:	Сколько пар сумм вы хотите ввести? 5 Введите суммы a и b:

Введите дробное число c:

a: 100,00 b: 10,00 c: 10

$a + b = 110,00$

$a - b = 90,00$

$a / b = 10,00$

$a / c = 10,00$

$b / c = 1,00$

$a * c = 1000,00$

$b * c = 100,00$

$a > b$? Да

$a < b$? Нет

$a == b$? Нет

17,20

Введите суммы a и b:

Введите дробное число c:

a: 200,50 b: 200,50 c: 0

$a + b = 401,00$

$a - b = 0,00$

$a / b = 1,00$

a / c : Невозможно вывести результат деления на 0

b / c : Невозможно вывести результат деления на 0

$a * c = 0,00$

$b * c = 0,00$

$a > b$? Нет

$a < b$? Нет

$a == b$? Да

17,20

Введите суммы a и b:

Введите дробное число c:

a: 0,50 b: 30,00 c: 40

$a + b = 30,50$

$a - b = 0,00$

$a / b = 0,01$

$a / c = 0,01$

$b / c = 0,75$

$a * c = 20,00$

$b * c = 1200,00$

$a > b$? Нет

$a < b$? Да

$a == b$? Нет

17,20

Введите суммы a и b:

Введите дробное число c:

a: 0,00 b: 1,00 c: 1

$a + b = 1,00$

$a - b = 0,00$

$a / b = 0,00$

$a / c = 0,00$

Введите дробное число c:

a: 10,00 b: 5,00 c: 2

$a + b = 15,00$

$a - b = 5,00$

$a / b = 2,00$

$a / c = 5,00$

$b / c = 2,50$

$a * c = 20,00$

$b * c = 10,00$

$a > b$? Да

$a < b$? Нет

$a == b$? Нет

17,20

Введите суммы a и b:

Введите дробное число c:

a: 14,55 b: 0,00 c: 10

$a + b = 14,55$

$a - b = 14,55$

a / b : Невозможно вывести результат деления на нулевую сумму

$a / c = 1,45$

$b / c = 0,00$

$a * c = 145,50$

$b * c = 0,00$

$a > b$? Да

$a < b$? Нет

$a == b$? Нет

17,20

Введите суммы a и b:

Введите дробное число c:

a: 17,80 b: 10,00 c: 0

$a + b = 27,80$

$a - b = 7,80$

$a / b = 1,78$

a / c : Невозможно вывести результат деления на 0

b / c : Невозможно вывести результат деления на 0

$a * c = 0,00$

$b * c = 0,00$

$a > b$? Да

$a < b$? Нет

$a == b$? Нет

17,20

Введите суммы a и b:

Введите дробное число c:

a: 20,20 b: 0,00 c: 10

$a + b = 20,20$

<p> $b / c = 1,00$ $a * c = 0,00$ $b * c = 1,00$ $a > b?$ Нет $a < b?$ Да $a == b?$ Нет 17,20 Введите суммы а и b: Введите дробное число c: a: 0,50 b: 0,50 c: 2 $a + b = 1,00$ $a - b = 0,00$ $a / b = 1,00$ $a / c = 0,25$ $b / c = 0,25$ $a * c = 1,00$ $b * c = 1,00$ $a > b?$ Нет $a < b?$ Нет $a == b?$ Да 17,20 Process finished with exit code 0 </p>	<p> $a - b = 20,20$ a / b: Невозможно вывести результат деления на нулевую сумму $a / c = 2,02$ $b / c = 0,00$ $a * c = 202,00$ $b * c = 0,00$ $a > b?$ Да $a < b?$ Нет $a == b?$ Нет 17,20 Введите суммы а и b: Введите дробное число c: a: 0,10 b: 0,00 c: 55 $a + b = 0,10$ $a - b = 0,10$ a / b: Невозможно вывести результат деления на нулевую сумму $a / c = 0,00$ $b / c = 0,00$ $a * c = 5,50$ $b * c = 0,00$ $a > b?$ Да $a < b?$ Нет $a == b?$ Нет 17,20 Process finished with exit code 0 </p>
--	---

6 - Код программы

money.h

```

#ifndef LAB1_MONEY_H
#define LAB1_MONEY_H
#include <iostream>
class Money {
public:
    Money();
    Money(unsigned long long r, unsigned char k);

    unsigned long long GetRub();
    unsigned char GetKop();

    unsigned long long GetRub() const;
    unsigned char GetKop() const;

    friend std::ostream& operator<<(std::ostream &out, Money const &a);
    friend std::istream& operator>>(std::istream &in, Money &a);

```

```

Money operator+(const Money &b);
Money operator-(const Money &b);
Money operator*(const double &b);
Money operator/(const double &b);
Money operator/(const Money &b);

Money operator+(const Money &b) const;
Money operator-(const Money &b) const;
Money operator*(const double &b) const;
Money operator/(const double &b) const;
Money operator/(const Money &b) const;

bool operator>(const Money &b) const;
bool operator<(const Money &b) const;
bool operator==(const Money &b) const;

```

```

private:
    unsigned long long rub;
    unsigned char kop;
};
Money operator"" _money(long double);
#endif //LAB1_MONEY_H

```

money.cpp

```

#include "money.h"

Money::Money() {
    rub = 0;
    kop = 0;
}
Money::Money(unsigned long long r, unsigned char k) {
    rub = r;
    kop = k;
}

unsigned long long Money::GetRub() {
    return rub;
}
unsigned char Money::GetKop() {
    return kop;
}
unsigned long long Money::GetRub() const {
    return rub;
}
unsigned char Money::GetKop() const {
    return kop;
}

```

```

std::ostream &operator<<(std::ostream &out, Money const &a) {
    if (a.GetKop() > 9) {
        out << a.GetRub() << "," << (int)a.GetKop();
    } else {
        out << a.GetRub() << "," << "0" << (int)a.GetKop();
    }
    return out;
}

std::istream& operator>>(std::istream &in, Money &a) {
    unsigned long long new_rub = 0;
    unsigned char new_kop = 0;
    int i = 0;
    char c[20];
    in >> c;
    while ((c[i] != ',') && (c[i] != '.') && (c[i] != 0)) {
        new_rub = new_rub * 10 + (int)(c[i] - '0');
        i++;
    }
    if (c[i] != 0) {
        i += 1;
        new_kop += (c[i] - '0') * 10;
        i++;
        if (c[i] != 0) {
            new_kop += (c[i] - '0');
        }
    }
    a.rub = new_rub;
    a.kop = new_kop;
    return in;
}

bool Money::operator>(const Money &b) const {
    if (rub == b.rub) {
        return kop > b.kop;
    }
    return rub > b.rub;
}

bool Money::operator<(const Money &b) const {
    if (rub == b.rub) {
        return kop < b.kop;
    }
    return rub < b.rub;
}

bool Money::operator==(const Money &b) const {
    if (rub == b.rub) {
        return kop == b.kop;
    }
    return false;
}

Money Money::operator+(const Money &b) {
    Money tmp = {rub, kop};
    tmp.rub = rub + b.rub;
    tmp.kop = kop + b.kop;
    if (tmp.kop > 99) {
        tmp.rub += 1;
        tmp.kop -= 100;
    }
}

```

```

    }
    return tmp;
}
Money Money::operator-(const Money &b) {
    unsigned long long new_rub = rub;
    unsigned char new_kop = kop;
    if(*this < b) {
        return {0, 0};
    }
    Money tmp = {new_rub, new_kop};
    tmp.rub = new_rub - b.rub;
    if (new_kop < b.kop) {
        tmp.rub -= 1;
        new_kop += 100;
    }
    tmp.kop = new_kop - b.kop;
    return tmp;
}
Money Money::operator*(const double &b) {
    Money tmp_m;
    unsigned long long tmp = rub*100 + kop;
    double result = tmp*b;
    tmp_m.rub = (unsigned long long)result/100;
    tmp_m.kop = (unsigned char)((unsigned long long )result%100);
    return tmp_m;
}
Money Money::operator/(const double &b) {
    Money tmp_money;
    unsigned long long tmp = rub*100 + kop;
    tmp /= b;
    tmp_money.kop = tmp%100;
    tmp_money.rub = tmp/100;
    return tmp_money;
}
Money Money::operator/(const Money &b) {
    Money tmp_money;
    unsigned long long tmp_1 = rub*10000 + (int)kop*100;
    unsigned long long tmp_2 = b.rub*100 + b.kop;
    unsigned long long result = tmp_1/tmp_2;
    tmp_money.rub = result/100;
    tmp_money.kop = result%100;
    return tmp_money;
}
Money Money::operator+(const Money &b) const {
    Money tmp = {rub, kop};
    tmp.rub = rub + b.rub;
    tmp.kop = kop + b.kop;
    if (tmp.kop > 99) {
        tmp.rub += 1;
        tmp.kop -= 100;
    }
    return tmp;
}
Money Money::operator-(const Money &b) const {
    unsigned long long new_rub = rub;
    unsigned char new_kop = kop;
    if(*this < b) {
        return {0, 0};
    }

```

```

    }
    Money tmp = {new_rub, new_kop};
    tmp.rub = new_rub - b.rub;
    if (new_kop < b.kop) {
        tmp.rub -= 1;
        new_kop += 100;
    }
    tmp.kop = new_kop - b.kop;
    return tmp;
}
Money Money::operator*(const double &b) const {
    Money tmp_m;
    unsigned long long tmp = rub*100 + kop;
    double result = tmp*b;
    tmp_m.rub = (unsigned long long)result/100;
    tmp_m.kop = (unsigned char)((unsigned long long )result%100);
    return tmp_m;
}
Money Money::operator/(const double &b) const {
    Money tmp_money;
    unsigned long long tmp = rub*100 + kop;
    tmp /= b;
    tmp_money.kop = tmp%100;
    tmp_money.rub = tmp/100;
    return tmp_money;
}
Money Money::operator/(const Money &b) const {
    Money tmp_money;
    unsigned long long tmp_1 = rub*10000 + (int)kop*100;
    unsigned long long tmp_2 = b.rub*100 + b.kop;
    unsigned long long result = tmp_1/tmp_2;
    tmp_money.rub = result/100;
    tmp_money.kop = result%100;
    return tmp_money;
}
Money operator"" _money(long double a) {
    unsigned long long num = a*100;
    Money tmp = {num/100, (unsigned char)(num%100)};
    return tmp;
}

```

main.cpp

```

#include <iostream>
#include "money.h"
#include <fstream>
int main() {
    //для чтения из файла в CLion на винде
    std::ifstream in(R"(C:\Users\LENOVO\CLionProjects\OOP\lab2\test_01.txt)");
    std::cin.rdbuf(in.rdbuf());
    int n;
    std::cout << "Сколько пар сумм вы хотите ввести? " << std::endl;
    std::cin >> n;
    std::cout << n << std::endl;
    for(int i = 0; i < n; i++) {
        Money a, b;
        double c;
        std::cout << "Введите суммы a и b: " << std::endl;
    }
}

```



```

std::cin >> a >> b;
std::cout << "Введите дробное число c: " << std::endl;
std::cin >> c;
std::cout << "a: " << a << " b: " << b << " c: " << c << std::endl;
std::cout << "a + b = " << a + b << std::endl;
std::cout << "a - b = " << a - b << std::endl;
if((b.GetRub() == 0) && (b.GetKop() == 0)) {
    std::cout << "a / b: Невозможно вывести результат деления на нулевую сумму" <<
std::endl;
} else {
    std::cout << "a / b = " << a / b << std::endl;
}
if(c == 0) {
    std::cout << "a / c: Невозможно вывести результат деления на 0" << std::endl;
} else {
    std::cout << "a / c = " << a / c << std::endl;
}
if(c == 0) {
    std::cout << "b / c: Невозможно вывести результат деления на 0" << std::endl;
} else {
    std::cout << "b / c = " << b / c << std::endl;
}
std::cout << "a * c = " << a * c << std::endl;
std::cout << "b * c = " << b * c << std::endl;
std::cout << "a > b? ";
if (a > b) {
    std::cout << "Да" << std::endl;
} else {
    std::cout << "Нет" << std::endl;
}
std::cout << "a < b? ";
if (a < b) {
    std::cout << "Да" << std::endl;
} else {
    std::cout << "Нет" << std::endl;
}

std::cout << "a == b? ";
if (a == b) {
    std::cout << "Да" << std::endl;
} else {
    std::cout << "Нет" << std::endl;
}
}
Money q = 17.20_money;
std::cout << q << std::endl;
}

```

7 – Вывод

Я научилась перезагружать операторы в простых классах, а так же получила навык работы с пользовательским литералом.