МИНИСТЕРСТВО ТРАНСПОРТА РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»

Кафедра АСУ

**Отчет**

По лабораторной работе

по дисциплине   
«Языки программирования высокого уровня»

Тема: «КЛАССЫ И ОБЪЕКТЫ»

Задание № 2

Вариант № 4

Выполнил:   
 ст. гр. УВА-111:

Букин А. А.

Проверил:

Савушкин С.А.

Москва 2020 г.

**Содержание:**

[**Постановка задачи 3**](#_heading=h.30j0zll)

[**Общие требования 3**](#_heading=h.1fob9te)

[**Задание 3**](#_heading=h.3znysh7)

[**Авторское понимание и формализация задачи**](#_heading=h.2et92p0) **4**

[**Текст программы**](#_heading=h.tyjcwt) **5**

[**Результаты тестирования**](#_heading=h.3dy6vkm) **12**

[**Вывод**](#_heading=h.1t3h5sf) **13**

**Постановка задачи**

**Общие требования**

Во всех заданиях, помимо указанных в задании операций, обязательно должны быть реализованы следующие методы:

– метод инициализации init;

– ввод с клавиатуры read;

– вывод на экран display;

– преобразование в строку toString.

**Задание**

Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операция сравнения.

# Авторское понимание и формализация задачи

Мы создаем класс Money, для работы с ним. Вводим определенную сумму, для расчета количества купюр различного достоинства. Так же реализуем сложение, вычитание, деление, деление на дробное число, умножение на дробное число, операцию сравнения этой суммы. И получившийся результат выводим на экран.

# Текст программы

**Файл – lab2.cpp**

**#include <string>**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**struct Money\_s {**

**long rub; // рубли**

**unsigned char cop; // копейки**

**};**

**Money\_s init(long r, unsigned char c) {**

**if (c > 99) { // проверка значений копеек**

**cout << "Ошибочные значения, завершение программы..." << endl;**

**exit(0);**

**}**

**Money\_s mon = {r, c};**

**return (mon);**

**}**

**Money\_s toMoney\_s(double n) {**

**long rub = (long)n;**

**double cop\_d = n - (double)((long)n);**

**unsigned char cop = (unsigned char)abs(round((cop\_d) \* 100.0));**

**if (n < 0.0)**

**{**

**rub--;**

**cop = 100 - cop;**

**}**

**Money\_s mon = init(rub, cop);**

**return (mon);**

**}**

**double toDouble(Money\_s m) {**

**int cop = m.cop;**

**if (m.rub < 0) {**

**cop -= 100;**

**m.rub++;**

**}**

**return ((double) m.rub + (double) cop / 100.0);**

**}**

**Money\_s read() {**

**double sum;**

**cout << "Введите сумму: ";**

**cin >> sum;**

**return (toMoney\_s(sum));**

**}**

**string toString(Money\_s money) {**

**char buf[100];**

**int res = snprintf(buf, sizeof(buf), "%.2f", toDouble(money));**

**string str = "0.00";**

**if (res >= 0 && res < sizeof(buf))**

**str = buf;**

**size\_t pos = str.find('.');**

**str.replace(pos, 1, ",");**

**return (str);**

**}**

**void display(Money\_s money) {**

**cout << "Сумма: " << toString(money) << endl;**

**}**

**int compare(Money\_s mon1, Money\_s mon2) {**

**double m1 = toDouble(mon1);**

**double m2 = toDouble(mon2);**

**if (m1 == m2)**

**return (0);**

**return (m1 > m2 ? 1 : -1);**

**}**

**Money\_s summa(Money\_s mon1, Money\_s mon2) {**

**return (toMoney\_s(toDouble(mon1) + toDouble(mon2)));**

**}**

**Money\_s raznost(Money\_s mon1, Money\_s mon2) {**

**double m1 = round(toDouble(mon1) \* 100);**

**double m2 = round(toDouble(mon2) \* 100);**

**double m3 = round(m1 - m2) / 100;**

**return (toMoney\_s(m3));**

**}**

**double divide(Money\_s mon1, Money\_s mon2) {**

**double m1 = round(toDouble(mon1) \* 100);**

**double m2 = round(toDouble(mon2) \* 100);**

**if (round(m2) == 0.0)**

**return (0.0);**

**return (m1 / m2);**

**}**

**Money\_s multiply\_num(Money\_s m, double n) {**

**return (toMoney\_s(toDouble(m) \* n));**

**}**

**Money\_s divide\_num(Money\_s m, double n) {**

**if (n == 0.0)**

**return (toMoney\_s(0.0));**

**double res = round(toDouble(m) \* 100) / n / 100;**

**return (toMoney\_s(res));**

**}**

**void demo\_struct()**

**{**

**cout << "┍━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┑" << endl;**

**cout << "| Демонстрация структурного варианта |" << endl;**

**cout << "┕━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┙" << endl << endl;**

**Money\_s mon1 = init(2, 02);**

**Money\_s mon2 = init(3, 55);**

**cout << "mon1: "; display(mon1);**

**cout << "mon2: "; display(mon2);**

**cout << endl;**

**cout << "sum: " + toString(summa(mon1, mon2)) << endl;**

**cout << "raz: " + toString(raznost(mon1, mon2)) << endl;**

**cout << "cmp: " + to\_string(compare(mon1, mon2)) << endl;**

**cout << "div: " + to\_string(divide(mon1, mon2)) << endl;**

**cout << endl;**

**double num = 1.5;**

**cout << "num: " + to\_string(num) << endl;**

**Money\_s mon3 = read();**

**cout << "mon: " + toString(mon3) << endl;**

**cout << endl;**

**cout << "multiplication : " + toString(multiply\_num(mon3, num)) << endl;**

**cout << "division : " + toString(divide\_num(mon3, num)) << endl;**

**}**

**//----------------------------------------------------------//**

**class Money\_c {**

**private:**

**long rub; // рубли**

**unsigned char cop; // копейки**

**public:**

**Money\_c(long r, unsigned char c);**

**Money\_c(double n);**

**Money\_c();**

**string toString();**

**double toDouble();**

**void display();**

**void add(Money\_c a);**

**void substract(Money\_c s);**

**static int compare(Money\_c mon1, Money\_c mon2);**

**static double divide(Money\_c mon1, Money\_c mon2);**

**void mult\_num(double n);**

**void divide\_num(double n);**

**};**

**string Money\_c::toString() {**

**char buf[100];**

**int res = snprintf(buf, sizeof(buf), "%.2f", toDouble());**

**string str = "0.00";**

**if (res >= 0 && res < sizeof(buf))**

**str = buf;**

**size\_t pos = str.find('.');**

**str.replace(pos, 1, ",");**

**return (str);**

**}**

**double Money\_c::toDouble() {**

**int icop = cop;**

**if (rub < 0) {**

**icop -= 100;**

**rub++;**

**}**

**return ((double) rub + (double) icop / 100.0);**

**}**

**void Money\_c::display() {**

**cout << "Сумма: " << toString() << endl;**

**}**

**void Money\_c::add(Money\_c a) {**

**\*this = Money\_c(toDouble() + a.toDouble());**

**}**

**Money\_c::Money\_c(long r, unsigned char c) {**

**if (c > 99) { // проверка значений копеек**

**cout << "Ошибочные значения, завершение программы..." << endl;**

**exit(0);**

**}**

**rub = r;**

**cop = c;**

**}**

**Money\_c::Money\_c(double n) {**

**long lrub = (long)n;**

**double cop\_d = n - (double)((long)n);**

**int icop = (unsigned char)abs(round((cop\_d) \* 100.0));**

**if (n < 0.0)**

**{**

**lrub--;**

**icop = 100 - icop;**

**}**

**\*this = Money\_c(lrub, icop);**

**}**

**Money\_c::Money\_c() {**

**double sum;**

**cout << "Введите сумму: ";**

**cin >> sum;**

**Money\_c tmp = Money\_c(sum);**

**rub = tmp.rub;**

**cop = tmp.cop;**

**}**

**void Money\_c::substract(Money\_c s) {**

**Money\_c tmp = Money\_c(toDouble() - s.toDouble());**

**rub = tmp.rub;**

**cop = tmp.cop;**

**}**

**int Money\_c::compare(Money\_c mon1, Money\_c mon2) {**

**double m1 = mon1.toDouble();**

**double m2 = mon2.toDouble();**

**if (m1 == m2)**

**return (0);**

**return (m1 > m2 ? 1 : -1);**

**}**

**double Money\_c::divide(Money\_c mon1, Money\_c mon2) {**

**double m1 = round(mon1.toDouble() \* 100);**

**double m2 = round(mon2.toDouble() \* 100);**

**if (round(m2) == 0.0)**

**return (0.0);**

**return (m1 / m2);**

**}**

**void Money\_c::mult\_num(double n) {**

**Money\_c tmp = Money\_c(toDouble() \* n);**

**rub = tmp.rub;**

**cop = tmp.cop;**

**}**

**void Money\_c::divide\_num(double n) {**

**if (n == 0.0) {**

**rub = 0;**

**cop = 0;**

**return ;**

**}**

**Money\_c tmp = Money\_c(toDouble() / n);**

**rub = tmp.rub;**

**cop = tmp.cop;**

**}**

**void demo\_class()**

**{**

**cout << "┍━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┑" << endl;**

**cout << "| Демонстрация классового варианта |" << endl;**

**cout << "┕━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┙" << endl << endl;**

**Money\_c mon1 = Money\_c(2, 02);**

**Money\_c mon2 = Money\_c(3, 55);**

**cout << "mon1: "; mon1.display();**

**cout << "mon2: "; mon2.display();**

**cout << endl;**

**Money\_c sum = mon1;**

**sum.add(mon2);**

**cout << "sum: " + sum.toString() << endl;**

**Money\_c raz = mon1;**

**raz.substract(mon2);**

**cout << "raz: " + raz.toString() << endl;**

**cout << "cmp: " + to\_string(Money\_c::compare(mon1, mon2)) << endl;**

**cout << "div: " + to\_string(Money\_c::divide(mon1, mon2)) << endl;**

**cout << endl;**

**double num = 1.5;**

**Money\_c input = Money\_c(); // вводится из терминала**

**cout << "number: " + to\_string(num) << endl;**

**cout << "money : " + input.toString() << endl;**

**cout << endl;**

**Money\_c mult = input;**

**mult.mult\_num(num);**

**cout << "multiplication : " + mult.toString() << endl;**

**Money\_c div = input;**

**div.divide\_num(num);**

**cout << "division : " + div.toString() << endl;**

**}**

**int main()**

**{**

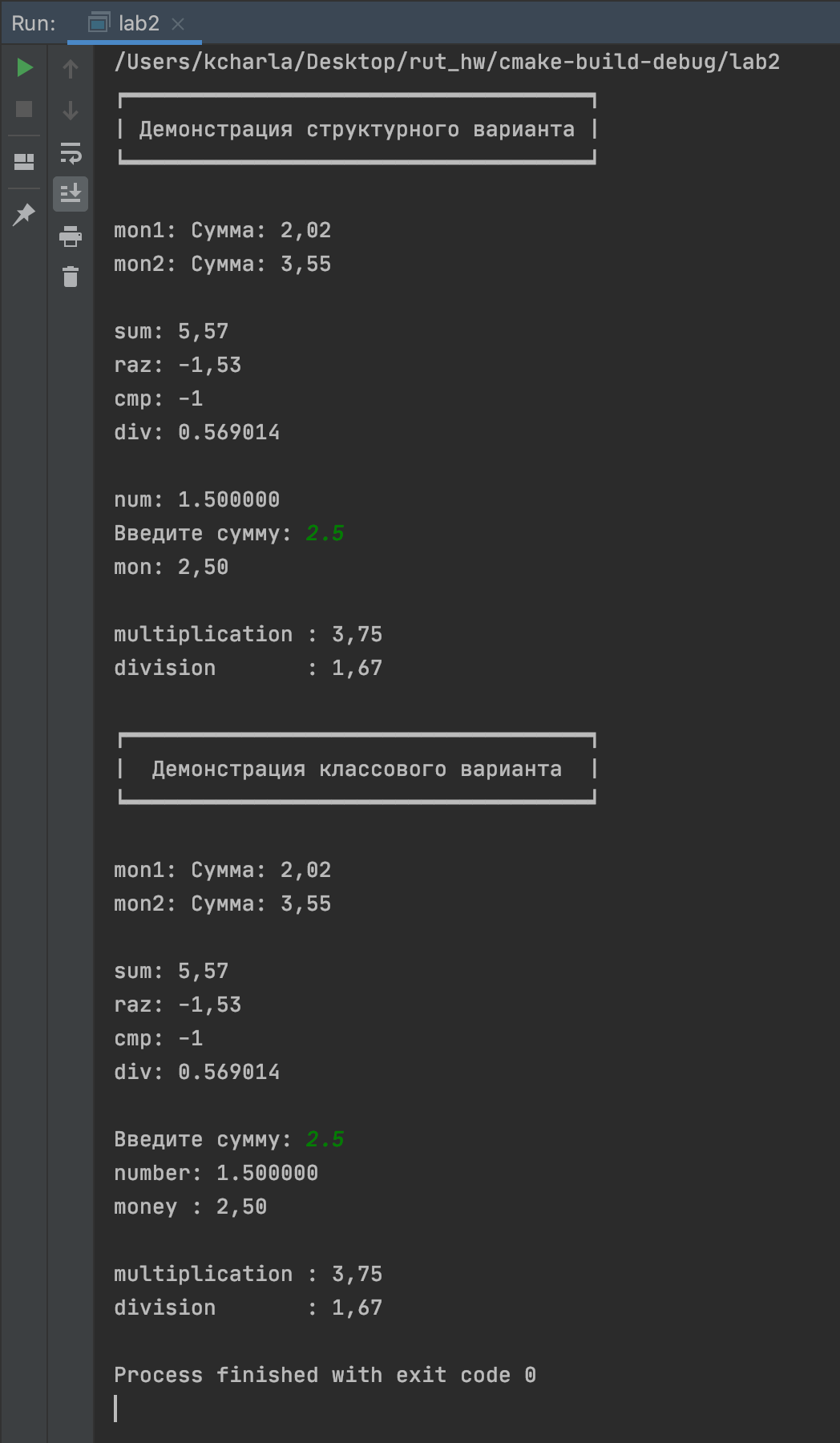
**demo\_struct();**

**cout << endl;**

**demo\_class();**

**}**

# Результаты тестирования



# Вывод

Программа выдает ожидаемые результаты и работает корректно. Поскольку полученные результаты совпали с ожидаемыми, программу можно считать правильной. Я научился создавать структуры и их объекты.