**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Кулагин Даниил

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. Постановка задачи

Создать класс Money для работы с денежными суммами. Сумма денег должна быть представлено двумя полями: типа unsigned long 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 сумм, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения.

1. Описание программы

Ссылка на GitHub: <https://github.com/AnnyWay775/oop_exercise_01>

➽Программа реализовывает работу с двумя полями класса Money. ➽ Переменная r для рублей (unsigned long long) и 2 переменные c1 и c2 соответственно для работы с копейками (unsigned char), так как unsigned char считывается только посимвольно, с помощью внутреннего представления (код ASCII: 0 = 48, 1 = 49, и т.д.).

➽ Затем данные переменные преобразуются в единую “сумму” типа double (fullSumm) с помощью функции get\_normal\_sum, для корректного вывода на экран терминала и дальнейших арифметических действий.

➽ В данной функции реализована защита от неправильного ввода копеек (например gg, вместо 89) с помощью конструкции “try-throw-catch”. Аналогичная конструкция применена и при делении на ноль в действиях “деление сумм” и “деление суммы на дробное число”.

➽ При получении отрицательных значений после вычитания сумм также будет предупреждение “Сумма денег не может быть отрицательна”.

➽ Работа с программой ведётся из консоли посредством меню команд.

➽ Более того, программа не даёт пользователю ввести и отрицательное значение и рублей (unsigned long long переходит в минус, начиная с 9223372036854775807 в целочисленном представлении).

1. Набор testcases

Три тест-файла: test\_01.txt, test\_02.txt и test\_03.txt

test\_01.txt - простой тест, без ошибок.

test\_02.txt - демонстрация “отрицательное значение рублей” с последующим выходом из терминала.

test\_03.txt - демонстрация остальных ошибок, указанных выше.

|  |  |  |
| --- | --- | --- |
| test\_01.txt | test\_02.txt | test\_03.txt |
| 345678532  32  2  86748323242  76  3  4543933234  99  4  23214223  22  5  4.3343242  1  6  231.7453244  7  56956  54  0 | -221301 | 432423424  tr  3  9492349294924  45  6  0  4  0  00  0 |

1. Результаты выполнения тестов.

**test\_01.txt**

Введите рубли:

345678532

Введите копейки:

32

345678532,32

Введите число от 1 до 6 или 0 для действий:

1. Распечатать меню

2. Сложение сумм

3. Вычитание сумм

4. Деление сумм

5. Умножение суммы на дробное число

6. Деление суммы на дробное число

7. Сравнение двух сумм

0. Выход из программы

Введите номер пункта меню >

2

Введите рубли:

86748323242

Введите копейки:

76

87094001775,08

Введите номер пункта меню >

3

Введите рубли:

4543933234

Введите копейки:

99

82550068540,09

Введите номер пункта меню >

4

Введите рубли:

23214223

Введите копейки:

22

3556,01

Введите номер пункта меню >

5

Введите множитель:

4.3343242

15412,91

Введите номер пункта меню >

1

Введите число от 1 до 6 или 0 для действий:

1. Распечатать меню

2. Сложение сумм

3. Вычитание сумм

4. Деление сумм

5. Умножение суммы на дробное число

6. Деление суммы на дробное число

7. Сравнение двух сумм

0. Выход из программы

Введите номер пункта меню >

6

Введите делитель:

231.7453244

66,51

Введите номер пункта меню >

7

Введите рубли:

56956

Введите копейки:

54

Вторая сумма больше первой

Введите номер пункта меню >

0

**test\_02.txt**

Введите рубли:

-221301

Ошибка: Сумма не может быть отрицательной

**test\_03.txt**

Введите рубли:

432423424

Введите копейки:

tr

Ошибка: Некорректный ввод

Ошибка: Некорректный ввод

432423424,00

Введите число от 1 до 6 или 0 для действий:

1. Распечатать меню

2. Сложение сумм

3. Вычитание сумм

4. Деление сумм

5. Умножение суммы на дробное число

6. Деление суммы на дробное число

7. Сравнение двух сумм

0. Выход из программы

Введите номер пункта меню >

3

Введите рубли:

9492349294924

Введите копейки:

45

Ошибка: Сумма денег не может быть отрицательна

432423424,00

Введите номер пункта меню >

6

Введите делитель:

0

Ошибка: деление на ноль

Введите номер пункта меню >

4

Введите рубли:

0

Введите копейки:

00

Ошибка: деление на ноль

432423424,00

Введите номер пункта меню >

0

1. Листинг программы

/\*

Создать класс Money для работы с денежными суммами. Сумма денег должна быть представлено двумя

полями: типа unsigned long 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 сумм, вычитание,

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

Кулагин Даниил Николаевич

М8О-201Б-18

\*/

#include <iostream>

#include <cmath>

#include <stdexcept>

using namespace std;

class Money

{

public:

unsigned long long r; //рубли типа unsigned long long

unsigned char c1, c2; //копейки представлениы двумя переменными типа unsigned char

double fullSumm; //внутреннее представление суммы для дальнейших преобразований

void get\_normal\_sum(Money a); //функция, переводящая исходные переменные в fullSumm

void Add(Money a); //функция сложения сумм

void Sub(Money a); //функция разности сумм

void Div(Money a); //функция деления сумм

int Comp(Money a); //функция сравнения сумм

};

void Money::Add(Money a) {

fullSumm += a.fullSumm;

}

void Money::Sub(Money a) {

try {

if (fullSumm<a.fullSumm) throw "Сумма денег не может быть отрицательна";

fullSumm -= a.fullSumm;

}

catch (char\* str) { cout << "Ошибка: " << str << endl; }

}

void Money::Div(Money a) {

try {

if (a.fullSumm == 0) throw "деление на ноль";

fullSumm /= a.fullSumm;

}

catch (char\* str) {

cout << "Ошибка: " << str << endl;

}

}

int Money::Comp(Money a) {

if (fullSumm > a.fullSumm) return 1; else

if (fullSumm < a.fullSumm) return 2; else

return 3;

}

void Money::get\_normal\_sum(Money a) {

double normal\_cop = 0;

fullSumm = 0;

try {

if (a.c1 == 48) normal\_cop += 0.0; else

if (a.c1 == 49) normal\_cop += 0.1; else

if (a.c1 == 50) normal\_cop += 0.2; else

if (a.c1 == 51) normal\_cop += 0.3; else

if (a.c1 == 52) normal\_cop += 0.4; else

if (a.c1 == 53) normal\_cop += 0.5; else

if (a.c1 == 54) normal\_cop += 0.6; else

if (a.c1 == 55) normal\_cop += 0.7; else

if (a.c1 == 56) normal\_cop += 0.8; else

if (a.c1 == 57) normal\_cop += 0.9; else throw "Некорректный ввод";

}

catch (char\* str) { cout << "Ошибка: " << str << endl; }

try {

if (a.c2 == 48) normal\_cop += 0.00; else

if (a.c2 == 49) normal\_cop += 0.01; else

if (a.c2 == 50) normal\_cop += 0.02; else

if (a.c2 == 51) normal\_cop += 0.03; else

if (a.c2 == 52) normal\_cop += 0.04; else

if (a.c2 == 53) normal\_cop += 0.05; else

if (a.c2 == 54) normal\_cop += 0.06; else

if (a.c2 == 55) normal\_cop += 0.07; else

if (a.c2 == 56) normal\_cop += 0.08; else

if (a.c2 == 57) normal\_cop += 0.09; else throw ("Некорректный ввод");

}

catch (char\* str) { cout << "Ошибка: " << str << endl; }

fullSumm = double(r + normal\_cop);

}

int main()

{

setlocale(LC\_ALL, "Russian");

Money a1, b1;

int indicator = 0;

try {

int indicator = 0;

cout << "Введите рубли: " << endl;

cin >> a1.r;

if (a1.r > 9223372036854775807) throw "Сумма не может быть отрицательной";

}

catch (char\* str) {

cout << "Ошибка: " << str << endl;

return 0;

}

cout << "Введите копейки: " << endl;

cin >> a1.c1;

cin >> a1.c2;

a1.get\_normal\_sum(a1);

printf("%.2f\n", a1.fullSumm);

int user\_command = 1;

for (;;) {

if (user\_command == 1) {

cout << "Введите число от 1 до 6 или 0 для действий: " << endl <<

"1. Распечатать меню" << endl <<

"2. Сложение сумм" << endl <<

"3. Вычитание сумм" << endl <<

"4. Деление сумм" << endl <<

"5. Умножение суммы на дробное число" << endl <<

"6. Деление суммы на дробное число" << endl <<

"7. Сравнение двух сумм" << endl <<

"0. Выход из программы" << endl;

}

else if (user\_command == 2) {

cout << "Введите рубли: " << endl;

cin >> b1.r;

cout << "Введите копейки: " << endl;

cin >> b1.c1;

cin >> b1.c2;

b1.get\_normal\_sum(b1);

a1.Add(b1);

printf("%.2f\n", a1.fullSumm);

}

else if (user\_command == 3) {

cout << "Введите рубли: " << endl;

cin >> b1.r;

cout << "Введите копейки: " << endl;

cin >> b1.c1;

cin >> b1.c2;

b1.get\_normal\_sum(b1);

a1.Sub(b1);

printf("%.2f\n", a1.fullSumm);

}

else if (user\_command == 4) {

cout << "Введите рубли: " << endl;

cin >> b1.r;

cout << "Введите копейки: " << endl;

cin >> b1.c1;

cin >> b1.c2;

b1.get\_normal\_sum(b1);

a1.Div(b1);

printf("%.2f\n", a1.fullSumm);

}

else if (user\_command == 5) {

float f;

cout << "Введите множитель: " << endl;

cin >> f;

a1.fullSumm \*= f;

printf("%.2f\n", a1.fullSumm);

}

else if (user\_command == 6) {

try {

double f;

cout << "Введите делитель: " << endl;

cin >> f;

if (f == 0) throw "деление на ноль";

a1.fullSumm /= f;

printf("%.2f\n", a1.fullSumm);

}

catch (char\* str) {

cout << "Ошибка: " << str << endl;

}

}

else if (user\_command == 7) {

cout << "Введите рубли: " << endl;

cin >> b1.r;

cout << "Введите копейки: " << endl;

cin >> b1.c1;

cin >> b1.c2;

b1.get\_normal\_sum(b1);

if (a1.Comp(b1) == 1) cout << "Первая сумма больше второй" << endl; else

if (a1.Comp(b1) == 2) cout << "Вторая сумма больше первой" << endl; else

cout << "Суммы равны" << endl;

}

else if (user\_command == 0)

break;

else

cout << "Нет такого пункта меню" << endl;

cout << "Введите номер пункта меню > " << endl;

cin >> user\_command;

}

return 0;

}

1. Выводы

Я научился создавать примитивные классы на языке C++ а также решать простые задачи с использованием классов.

СПИСОК ЛИТЕРАТУРЫ

1. ООП с примерами (часть 1) [электронный ресурс].

URL:<https://habr.com/ru/post/87119/>

1. Объектно-ориентированное программирование. Определение классов. [электронный ресурс].

URL:<https://metanit.com/cpp/tutorial/5.1.php>

1. function precision [электронный ресурс].

URL:<http://www.cplusplus.com/reference/ios/ios_base/precision/>

1. Классы в C++. [электронный ресурс].

URL: [http://cppstudio.com/post/439](http://cppstudio.com/post/439/)

1. Исключения в C++ (exception) [электронный ресурс].

URL:<http://cppstudio.com/post/9773/>