Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа

Дисциплина: «Объектно-ориентированное программирование»

I I семестр

Задание 1: «Простые классы»

|  |  |
| --- | --- |
| Группа: | М8О-208Б-18, №9 |
| Студент: | Игитова Александра Андреевна |
| Преподаватель: | Журавлёв Андрей Андреевич |
| Оценка: |  |
| Дата: | 30.09.2019 |

Москва, 2019

1. **Задание**

(*вариант № 9*): **Создать класс Money** для работы с денежными суммами в старой британской система. Сумма денег должна быть представлено тремя полями: типа unsigned long long для фунтов стерлингов, типа unsigned char – для шиллингов, unsigned char – для пенсов (пенни). Реализовать сложение сумм, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. 1 фунт = 20 шиллингов, 1 шиллинг = 12 пенни.

1. **Адрес репозитория на GitHub**

<https://github.com/SandraIgitova/oop_exercise_01>

1. **Код программы на С++**

main.cpp

#include <iostream>

#include <sstream>

#include <cinttypes>

#include "money.h"

int main() {

Money a;

uint32\_t x;

a.ConIn();

begin:

std::cout << "1. Вывести количетсво фунтов, шиллингов и пенсов\n2. Сложить 2 суммы\n3. Вычесть из большей суммы меньшую\n4. Разделить первую сумму на вторую\n5. Разделить первую сумму на дробное число\n6. Умножить первую сумму на дробное число\n7. Операции сравнения\n8. Показать фунты, шиллинги и пенсы\n" << std::endl;

std::cin >> x;

std::cout << std::endl;

if (x == 1) {

a.Show();

}

if (x == 2) {

Money a2;

std::cout << "Введите вторую сумму." << std::endl;

a2.ConIn();

a.Plus(&a2);

//std::cout << "\nРезультат сложения:" << std::endl;

//a.Show();

}

if (x == 3) {

Money a2;

std::cout << "Введите вторую сумму." << std::endl;

a2.ConIn();

//std::cout << "\nРезультат вычитания:" << std::endl;

if (a2.AllSumGet() > a.AllSumGet())

{

a2.Minus(&a);

//a2.Show();

}

else {

a.Minus(&a2);

//a.Show();

}

}

if (x == 4) {

Money a2;

std::cout << "Введите вторую сумму." << std::endl;

a2.ConIn();

//std::cout << "\nРезультат деления:" << std::endl;

//std::cout <<

a.Div(&a2); //<< std::endl;

}

if (x == 5) {

std::cout << "Введите число:" << std::endl;

float arg;

std::cin >> arg;

//std::cout << "\nРезультат деления:" << std::endl;

a.DivN(arg);

//a.Show();

}

if (x == 6) {

std::cout << "Введите число:" << std::endl;

float arg;

std::cin >> arg;

//std::cout << "\nРезультат умножения:" << std::endl;

a.Mult(arg);

//a.Show();

}

if (x == 7) {

Money a2;

std::cout << "Введите вторую сумму." << std::endl;

a2.ConIn();

switch (a.Compare(&a2)) {

case 0:

std::cout << "Суммы равны" << std::endl; break;

case 1:

std::cout << "Первая сумма меньше второй." << std::endl; break;

case 2:

std::cout << "Первая сумма больше второй." << std::endl;

}

}

if (x == 8) {

a.Print();

}

std::cout << " \n Нажмите Ввод для завершения программы или любую клавишу для продолжения \n";

std::cin.clear();

std::cin.ignore(32767, '\n');

if (std::cin.get() != '\n') { goto begin; };

return 0;

}

money.h

#ifndef MONEY\_H

#define MONEY\_H

#include <cinttypes>

#include <iostream>

#include "money.h"

class Money {

public:

Money();

// разделение всей суммы на фунты, шиллинги и пенсы с записью их в разные переменные

void SplitUp(uint64\_t allSum);

// Получить сумму

void Set(uint64\_t f, uint16\_t s, uint8\_t p);

// Ввести с консоли

void ConIn();

// Вывести сумму на консоль

void Show();

// сложение сумм

void Plus(Money\* a2);

// вычитание сумм

void Minus(Money\* a2);

// деление 2х сумм

double Div(Money\* a2);

// деление суммы на число

void DivN(float arg);

// умножение суммы на число

void Mult(float arg);

// сравнение

uint8\_t Compare(Money\* a2);

//выводит фунты, шиллинги и пенсы.

void Print();

//геттер для allSum

uint64\_t AllSumGet();

private:

uint64\_t funt; // количество фунтов funt

uint16\_t shil; // количество шиллингов shil

uint8\_t pen; // количество пенсов

};

#endif

money.cpp

#include "money.h"

Money::Money() {

funt = 0;

shil = 0;

pen = 0;

}

void Money::SplitUp(uint64\_t allSum) {

funt = (uint64\_t)((uint64\_t)(allSum/12)/20);

shil = (uint16\_t)((allSum - funt\*20\*12)/12);

pen = (uint8\_t)(allSum-funt\*20\*12-((uint64\_t)shil)\*12);

}

// Записать значений в объект (не обязательно из консоли)

void Money::Set(uint64\_t f, uint16\_t s, uint8\_t p) {

funt = f;

shil = s;

pen = p;

}

// Ввести значения c консоли

void Money::ConIn() {

uint64\_t f;

uint16\_t s, p;

std::cout << "Введите количество фунтов, шиллингов и пенсов:" << std::endl;

std::cin >> f;

std::cin >> s;

std::cin >> p;

p = (uint8\_t)(p);

this->Set(f, s, p);

}

// Вывести сумму

void Money::Show() {

std::cout << "Фунты: " << funt << std::endl;

std::cout << "Шиллинги: " << shil << std::endl;

std::cout << "Пенсы: " << (uint16\_t)pen << std::endl;

}

// сложение сумм

void Money::Plus(Money\* a2) {

this->SplitUp((this->AllSumGet() + a2->AllSumGet()));

}

// все действия проводяться между текущим объектом и объектом, передаваемым в качестве аргумента

// вычитание сумм

void Money::Minus(Money\* a2) {

this->SplitUp((this->AllSumGet() - a2->AllSumGet()));

}

// деление 2х сумм

double Money::Div(Money\* a2) {

return (double)(((double)(this->AllSumGet())) / ((double)(a2->AllSumGet())));

}

// деление суммы на число

void Money::DivN(float arg) {

this->SplitUp((uint64\_t)((float)(this->AllSumGet())) / arg);

}

// умножение суммы на число

void Money::Mult(float arg) {

this->SplitUp((uint64\_t)((float)(this->AllSumGet())) \* arg);

}

// сравнение

uint8\_t Money::Compare(Money\* a2)

{

uint8\_t ret;

if (a2->AllSumGet() == this->AllSumGet()) { ret = 0; }

if (a2->AllSumGet() > this->AllSumGet()) { ret = 1; }

if (a2->AllSumGet() < this->AllSumGet()) { ret = 2; }

return ret;

}

//выводит фунты, шиллинги и пенсы.

void Money::Print() {

std::cout << "Фунты: " << funt << " " << "Шиллинги: " << shil << " " << "Пенсы: " << (uint16\_t)pen << " ";

}

// allSum - Сумма в пенсах

uint64\_t Money::AllSumGet() {

return (funt \* 20 \* 12 + (uint64\_t)shil \* 12 + (uint64\_t)pen);;

}

CMakeLists.txt

cmake\_minimum\_required(VERSION 3.5)

project(oop\_exercise\_01)

add\_executable(oop\_exercise\_01 money.cpp main.cpp)

set\_target\_properties(oop\_exercise\_01 PROPERTIES

CXX\_STANDARD 14

CXX\_STANDARD\_REQUIRED ON

)

test\_01.txt

234

543

234

test\_02.txt

11

345

33

1. **Результаты выполнения тестов**

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

234

543

234

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

1

Фунты: 234

Шиллинги: 543

Пенсы: 234

Нажмите Ввод для завершения программы или любую клавишу для продолжения

\

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

234

543

234

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

5

Введите число:

123

Нажмите Ввод для завершения программы или любую клавишу для продолжения

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

234

543

234

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

3

Введите вторую сумму.

Введите количество фунтов, шиллингов и пенсов:

234

44

3

Нажмите Ввод для завершения программы или любую клавишу для продолжения

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

11

345

33

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

2

Введите вторую сумму.

Введите количество фунтов, шиллингов и пенсов:

23

455

7

Нажмите Ввод для завершения программы или любую клавишу для продолжения

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

11

345

33

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

6

Введите число:

123

Нажмите Ввод для завершения программы или любую клавишу для продолжения

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$ ./oop\_exercise\_01

Введите количество фунтов, шиллингов и пенсов:

11

345

33

1. Вывести количетсво фунтов, шиллингов и пенсов

2. Сложить 2 суммы

3. Вычесть из большей суммы меньшую

4. Разделить первую сумму на вторую

5. Разделить первую сумму на дробное число

6. Умножить первую сумму на дробное число

7. Операции сравнения

8. Показать фунты, шиллинги и пенсы

7

Введите вторую сумму.

Введите количество фунтов, шиллингов и пенсов:

32

4444

6

Первая сумма меньше второй.

Нажмите Ввод для завершения программы

sandra@sandra-VB:~/Рабочий стол/OOP/Lab1/build$

1. **Объяснение результатов работы программы**

Программа просит на вход три параметра, после чего выходит меню с действиями, которые может вызвать пользователь. Далее выводятся те входные данные, которые запрашивает программа. Происходят вычисления и выход.

1. **Вывод**

Изучила основы объектно-ориентированного программирования, методы, классы, написала простой класс Money, который реализует математические действия с деньгами типа из трех параметров. При этом вычисления происходят в разных системах счисления, а не в десятичной, что усложняет процесс. Во время выполнения задания происходили проблемы с типом данных и стандартным считыванием и вводом. Это удалось исправить путем введения некоторых корректировок, что позволило подробнее изучить типы и количество памяти, занимаемой ими. Также удалось развить собственное мышление, путем анализа задания, которое не имело достаточно точной информации для выполнения. ☺