Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа

Дисциплина: «Объектно-ориентированное программирование»

III семестр

Задание 2: «Операторы, литералы»

|  |  |
| --- | --- |
| Группа: | М8О-208Б-18, №9 |
| Студент: | Игитова Александра Андреевна |
| Преподаватель: | Журавлёв Андрей Андреевич |
| Оценка: |  |
| Дата: | 25.11.2019 |

Москва, 2019

1. **Задание**

Изучение механизмов перегрузки операторов;

Изучение механизмов работы с пользовательскими литералами;

1. **Адрес репозитория на GitHub**

https://github.com/SandraIgitova/oop\_exercise\_02

1. **Код программы на С++**

*Main.cpp*

#include <iostream>

#include <sstream>

#include <cinttypes>

#include "money.h"

int main() {

Money a;

uint32\_t x;

// a.Read(std::cin);

std::cin >> a;

Money m = 23323\_funt + 232\_shil + 12\_pen;

std::cout << m << std::endl;

while(true) {

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.Write(std::cout);

std::cout << a << std::endl;

}

if (x == 2) {

Money a2;

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

// a2.Read(std::cin);

std::cin >> a2;

a = a+a2;

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

//a.Show();

}

if (x == 3) {

Money a2;

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

std::cin >> a2;

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

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

{

a2=a2-a;

//a2.Show();

}

else {

a=a-a2;

//a.Show();

}

}

if (x == 4) {

Money a2;

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

// a2.Read(std::cin);

std::cin >> a2;

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

//std::cout <<

a/a2; //<< std::endl;

}

if (x == 5) {

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

float arg;

std::cin >> arg;

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

a = a / arg;

//a.Show();

}

if (x == 6) {

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

float arg;

std::cin >> arg;

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

a = a \* arg;

//a.Show();

}

if (x == 7) {

Money a2;

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

// a2.Read(std::cin);

std::cin >> a2;

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 -1:

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

}

}

if (x == 8) {

// a.Write(std::cout);

std::cout << a << std::endl;

}

//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; };

if (x > 8)

break;

}

return 0;

}

//Оператор чтения и вывода в поток (std::cin >> a; std::cout << a;) вместо рид и врайт.

//Пользовательский литтерал(operator"")

*Money.h*

#ifndef MONEY\_H

#define MONEY\_H

#include <cinttypes>

#include <iostream>

#include "money.h"

class Money {

public:

Money();

Money(uint64\_t f, uint32\_t s, uint8\_t p);

// 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 Read(std::istream& is);

// Вывести сумму на консоль

void Write(std::ostream& os) const;

friend std::istream& operator>>(std::istream& is, Money& m);

friend std::ostream& operator<<(std::ostream& os, const Money& m);

// сложение сумм

Money operator+(const Money& a2) const;

// вычитание сумм

Money operator-(const Money& a2) const;

// деление 2х сумм

double operator/(const Money& a2) const;

// деление суммы на число

Money operator/(float arg);

// умножение суммы на число

Money operator\*(float arg);

// сравнение

int8\_t Compare(const Money& a2) const;

//геттер для allSum

uint64\_t AllSumGet() const;

void Print();

private:

uint64\_t funt; // количество фунтов funt

uint16\_t shil; // количество шиллингов shil

uint8\_t pen; // количество пенсов

};

// Money operator ""\_m(const char\* str, size\_t size);

Money operator"" \_funt(unsigned long long int num);

Money operator"" \_shil(unsigned long long int num);

Money operator"" \_pen(unsigned long long int num);

#endif

Rectangle.cpp

#include "money.h"

Money::Money(): funt(0), shil(0), pen(0)

{

}

Money::Money(uint64\_t f, uint32\_t s, uint8\_t p): funt(f), shil(s), pen(p)

{

}

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::Read(std::istream& is) {

uint64\_t f;

uint16\_t s, p;

is >> f;

is >> s;

is >> p;

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

}

std::istream& operator>>(std::istream& is, Money& m){

uint32\_t tmp;

is >> m.funt >> m.shil >> tmp;

m.pen = tmp;

}

// Вывести сумму

void Money::Write(std::ostream& os) const {

os << funt << ' ';

os << shil << ' ';

os << (int)pen << ' ';

}

std::ostream& operator<<(std::ostream& os, const Money& m){

os << m.funt << " " << m.shil << " " << (uint32\_t)m.pen;

}

// сложение сумм

Money Money::operator+(const Money& a2) const {

Money result;

result.SplitUp(AllSumGet() + a2.AllSumGet());

return result;

}

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

// вычитание сумм

Money Money::operator-(const Money& a2) const {

Money result;

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

}

// деление 2х сумм

double Money::operator/(const Money& a2) const {

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

}

// деление суммы на число

Money Money::operator/(float arg) {

Money result;

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

}

// умножение суммы на число

Money Money::operator\*(float arg) {

Money result;

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

}

// сравнение

int8\_t Money::Compare(const Money& a2) const

{

int8\_t ret;

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

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

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

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() const {

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

}

Money operator"" \_funt(unsigned long long int num)

{

Money res(num, 0, 0);

return res;

}

Money operator"" \_shil(unsigned long long int num)

{

Money res(0, num, 0);

return res;

}

Money operator"" \_pen(unsigned long long int num)

{

Money res(0, 0, num);

return res;

}

CMakeLists.txt

cmake\_minimum\_required(VERSION 2.8) # Проверка версии CMake.

# Если версия установленой программы

# старее указаной, произайдёт аварийный выход.

project(lab2) # Название проекта

set(SOURCE\_EXE main.cpp) # Установка переменной со списком исходников для исполняемого файла

set(SOURCE\_LIB money.cpp) # Тоже самое, но для библиотеки

add\_library(money STATIC ${SOURCE\_LIB}) # Создание статической библиотеки с именем foo

add\_executable(main ${SOURCE\_EXE}) # Создает исполняемый файл с именем main

target\_link\_libraries(main money)

1. **Результаты выполнения тестов**

123

23

66

23334 13 0

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

123 23 66

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.

123

32

23

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

249 2 5

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

123

23

66

23334 13 0

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

123 23 66

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.

123

32

23

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

249 2 5

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. **Вывод**

Благодаря перегрузке логических операторов программист, который будет использовать данный класс, получает более гибкое и простое управление над экземплярами класса, а благодаря литералам пользователь в одной строке может задать параметры экземпляра и начать работу с ним.