1) Код программы:

#include <iostream>

#include <clocale>

using namespace std;

class Budget {

public:

Budget(double a, double b) {

this->a = int(a \* 100 + 0.5) / 100.0;

this->b = int(b \* 100 + 0.5) / 100.0;

}

void print\_budget() {

cout << "Собственная часть: "<< this->a << " Заёмная часть: " << this->b << '\n';

}

friend Budget operator+ (const Budget& A, const Budget& B) {

return Budget(A.a + B.a, A.b + B.b);

}

friend Budget operator- (const Budget& A, const Budget& B) {

return Budget(A.a - B.a, A.b - B.b);

}

void multiply\_own\_part(double value) {

a \*= value;

}

void multiply\_return\_part(double value) {

b \*= value;

}

void divide\_own\_part(double value) {

a /= value;

}

void divide\_return\_part(double value) {

b /= value;

}

bool operator== (const Budget& B) {

if (this->a == B.a && this->b == B.b) {

return true;

}

else {

return false;

}

}

bool operator> (const Budget& B) {

if ((this->a - this->b) > (B.a - B.b)) { // Заёмные деньги надо отдавать

return true;

}

else {

return false;

}

}

bool operator< (const Budget& B) {

if ((this->a - this->b) < (B.a - B.b)) { // Заёмные деньги надо отдавать

return true;

}

else {

return false;

}

}

private:

double a; // own part

double b; // return part

};

int main() {

setlocale(LC\_ALL, "rus");

double a1, b1, a2, b2;

cout << "Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: ";

cin >> a1 >> b1 >> a2 >> b2;

Budget a = { a1, b1 };

Budget b = { a2, b2 };

cout << "Первый бюджет: ";

a.print\_budget();

cout << "Второй бюджет: ";

b.print\_budget();

cout << "Сумма бюджетов: ";

(a + b).print\_budget();

cout << "Разность бюджетов: ";

(a - b).print\_budget();

if (a == b) {

cout << "Бюджеты равны" << '\n';

}

else {

cout << "Бюджеты не равны" << '\n';

}

if (a > b) {

cout << "Первый бюджет больше" << '\n';

}

else {

cout << "Второй бюджет больше" << '\n';

}

cout << "Умножим собственные деньги первого бюджета на 2" << '\n';

a.multiply\_own\_part(2);

a.print\_budget();

cout << "Поделим заёмные деньги первого бюджета на 5" << '\n';

a.divide\_return\_part(5);

a.print\_budget();

return 0;

}

2) <https://github.com/RomaMaster228/oop_exercise_01>

3) Testcases:

1.2 2.3 4.5 6.7

1.21 2.34 8.76 9.47

1.23242431 9.457828228 67.23467 12.6789999

76543.21 65.4 3.2 1.999

1.0 2.0 3.1 4.0

213.45 67.233 99.9999 2467.21

892779.12 8888888 1131.4

9091.3 780 80 12

1 2 3 4

4) Вывод:

Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: 1.2 2.3 4.5 6.7

Первый бюджет: Собственная часть: 1.2 Заёмная часть: 2.3

Второй бюджет: Собственная часть: 4.5 Заёмная часть: 6.7

Сумма бюджетов: Собственная часть: 5.7 Заёмная часть: 9

Разность бюджетов: Собственная часть: -3.29 Заёмная часть: -4.39

Бюджеты не равны

Первый бюджет больше

Умножим собственные деньги первого бюджета на 2

Собственная часть: 2.4 Заёмная часть: 2.3

Поделим заёмные деньги первого бюджета на 5

Собственная часть: 2.4 Заёмная часть: 0.46

Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: 1.23242431 9.457828228 67.23467 12.6789999

Первый бюджет: Собственная часть: 1.23 Заёмная часть: 9.46

Второй бюджет: Собственная часть: 67.23 Заёмная часть: 12.68

Сумма бюджетов: Собственная часть: 68.46 Заёмная часть: 22.14

Разность бюджетов: Собственная часть: -65.99 Заёмная часть: -3.21

Бюджеты не равны

Второй бюджет больше

Умножим собственные деньги первого бюджета на 2

Собственная часть: 2.46 Заёмная часть: 9.46

Поделим заёмные деньги первого бюджета на 5

Собственная часть: 2.46 Заёмная часть: 1.892

5) Программа округляет до второго знака, перенося два знака после запятой налево, умножая на 100, прибавляя 0,5 и приводя к целому типу. Потом делим на 100, чтобы перенести две цифры направо обратно. Сумма бюджетов складывает собственные деньги и заёмные деньги двух объектов. Равенство бюджетов также по двум компонентам. Больше или меньше определяется вычетом из собственных денег заёмные и сравнением двух объектов.

**Федеральное государственное автономное образовательное учреждение высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

**Московский институт электроники и математики им. А.Н.Тихонова**

 Департамент компьютерной инженерии

**Лабораторная работа № 1**

по курсу «ООП»

Тема: Простые классы на языке С++

Студент: Лисин Роман Сергеевич

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. **Постановка задачи**

Вариант 17: Создать класс Budget для работы с бюджетом. Класс состоит из двух вещественных чисел (a,b). Где a – собственная часть средств бюджета в рублях, b – заемная часть средств бюджета рублях. Оба числа должны округляться до второго знака после запятой. Реализовать арифметические операции сложения, вычитания, умножения и деления, а также операции сравнения.

1. **Описание программы**

Реализуется класс Бюджет с заемной частью (b) и собственными деньгами (a). Методы этого класса: print\_budget, operator+, operator-, operator>, operator<, operator==, multiply\_own\_part,multiply\_return\_part, divide\_own\_part, divide\_return\_part. Пользователь вводит данные для программы, и программа сама показывает свой функционал. Программа округляет до второго знака, перенося два знака после запятой налево, умножая на 100, прибавляя 0,5 и приводя к целому типу. Потом делим на 100, чтобы перенести две цифры направо обратно.  Сумма бюджетов складывает собственные деньги и заёмные деньги двух объектов. Равенство бюджетов также по двум компонентам. Больше или меньше определяется вычетом из собственных денег заёмные и сравнением двух объектов.

Файлы [main.cpp](https://github.com/RomaMaster228/oop_exercise_01/blob/main/main.cpp), [CMakeLists.txt](https://github.com/RomaMaster228/oop_exercise_01/blob/main/CMakeLists.txt), [test\_01.txt](https://github.com/RomaMaster228/oop_exercise_01/blob/main/test_01.txt), [test\_02.txt](https://github.com/RomaMaster228/oop_exercise_01/blob/main/test_02.txt) размещены в папке oop\_exercise\_01.

1. **Набор  и результаты тестов**

Тесты представляют собой входные данные для программы, которая выдает весь функционал класса.

Примеры работы:

Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: 1.2 2.3 4.5 6.7

Первый бюджет: Собственная часть: 1.2 Заёмная часть: 2.3

Второй бюджет: Собственная часть: 4.5 Заёмная часть: 6.7

Сумма бюджетов: Собственная часть: 5.7 Заёмная часть: 9

Разность бюджетов: Собственная часть: -3.29 Заёмная часть: -4.39

Бюджеты не равны

Первый бюджет больше

Умножим собственные деньги первого бюджета на 2

Собственная часть: 2.4 Заёмная часть: 2.3

Поделим заёмные деньги первого бюджета на 5

Собственная часть: 2.4 Заёмная часть: 0.46

Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: 1.23242431 9.457828228 67.23467 12.6789999

Первый бюджет: Собственная часть: 1.23 Заёмная часть: 9.46

Второй бюджет: Собственная часть: 67.23 Заёмная часть: 12.68

Сумма бюджетов: Собственная часть: 68.46 Заёмная часть: 22.14

Разность бюджетов: Собственная часть: -65.99 Заёмная часть: -3.21

Бюджеты не равны

Второй бюджет больше

Умножим собственные деньги первого бюджета на 2

Собственная часть: 2.46 Заёмная часть: 9.46

Поделим заёмные деньги первого бюджета на 5

Собственная часть: 2.46 Заёмная часть: 1.892

1. **Листинг программы**

#include <iostream>

#include <clocale>

using namespace std;

class Budget {

public:

Budget(double a, double b) {

this->a = int(a \* 100 + 0.5) / 100.0;

this->b = int(b \* 100 + 0.5) / 100.0;

}

void print\_budget() {

cout << "Собственная часть: "<< this->a << " Заёмная часть: " << this->b << '\n';

}

friend Budget operator+ (const Budget& A, const Budget& B) {

return Budget(A.a + B.a, A.b + B.b);

}

friend Budget operator- (const Budget& A, const Budget& B) {

return Budget(A.a - B.a, A.b - B.b);

}

void multiply\_own\_part(double value) {

a \*= value;

}

void multiply\_return\_part(double value) {

b \*= value;

}

void divide\_own\_part(double value) {

a /= value;

}

void divide\_return\_part(double value) {

b /= value;

}

bool operator== (const Budget& B) {

if (this->a == B.a && this->b == B.b) {

return true;

}

else {

return false;

}

}

bool operator> (const Budget& B) {

if ((this->a - this->b) > (B.a - B.b)) {  // Заёмные деньги надо отдавать

return true;

}

else {

return false;

}

}

bool operator< (const Budget& B) {

if ((this->a - this->b) < (B.a - B.b)) {  // Заёмные деньги надо отдавать

return true;

}

else {

return false;

}

}

private:

double a;  // own part

double b;  // return part

};

int main() {

setlocale(LC\_ALL, "rus");

double a1, b1, a2, b2;

cout << "Введите через пробел собственные деньги и заёмные деньги одного и второго бюджетов: ";

if (!(cin >> a1 >> b1 >> a2 >> b2)) {

cout << "Данные некорректны!" << '\n';

return -1;

}

Budget a = { a1, b1 };

Budget b = { a2, b2 };

cout << "Первый бюджет: ";

a.print\_budget();

cout << "Второй бюджет: ";

b.print\_budget();

cout << "Сумма бюджетов: ";

(a + b).print\_budget();

cout << "Разность бюджетов: ";

(a - b).print\_budget();

if (a == b) {

cout << "Бюджеты равны" << '\n';

}

else {

cout << "Бюджеты не равны" << '\n';

}

if (a > b) {

cout << "Первый бюджет больше" << '\n';

}

else {

cout << "Второй бюджет больше" << '\n';

}

cout << "Умножим собственные деньги первого бюджета на 2" << '\n';

a.multiply\_own\_part(2);

a.print\_budget();

cout << "Поделим заёмные деньги первого бюджета на 5" << '\n';

a.divide\_return\_part(5);

a.print\_budget();

return 0;

}

1. **Вывод**

Данный класс помогает планировать свой бюджет пользователю и быстро выполнять нужные операции.

**Список литературы**

1. https://ravesli.com/uroki-cpp/