

Московский Авиационный Институт

(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторной работе № 01**  
**по курсу «Объектно-ориентированное программирование»**

**Тема:**  
**«Простые классы»**

Студент:	Пшеницын А. А.
Группа:	М80-208Б-18
Преподаватель:	Журавлев А. А.
Вариант:	17
Оценка:	
Дата:	

Цель:

- Изучение системы сборки на языке C++, изучение систем контроля версии
- Изучение основ работы с классами в C++

### Задание (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 в рублях. Оба числа должны округляться до второго знака после запятой. Реализовать арифметические операции сложения, вычитания, умножения и деления, а также операции сравнения.

### Код программы на языке C++

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>

typedef struct Budget Budget;

struct Budget{
private:
    double a; //собственная часть
    double b; //заемная часть
public:
    Budget();
    Budget(double ch1, double ch2);
    void read(std::istream& is);
    void print(std::ostream& os);
    Budget add(const Budget bud) const;
    Budget sub(const Budget bud) const;
    Budget mult(const Budget bud) const;
    Budget div(const Budget bud) const;
    bool com(const Budget bud) const;
    bool no(const Budget bud) const;
};

Budget::Budget(){
    a = 0;
    b = 0;
}

Budget::Budget(double ch1, double ch2){
    a = round(ch1 * 100) / 100;
    b = round(ch2 * 100) / 100;
}

void Budget::read(std::istream& is){
    double ch1, ch2;
    is >> ch1;
    is >> ch2;
```

```

    a = round(ch1 * 100) / 100;
    b = round(ch2 * 100) / 100;
}

void Budget::print(std::ostream& os){
    os << "Main part of the budget: " << a << '\n' << "Borrowed part of the budget: " << b << '\n' <<
    '\n';
}

Budget Budget::add(const Budget bud) const{
    Budget res{};
    res.a = a + bud.a;
    res.b = b + bud.b;
    return res;
}

Budget Budget::sub(const Budget bud) const{
    Budget res{};
    if(a > bud.a){
        res.a = a - bud.a;
    } else {
        res.a = bud.a - a;
    }
    if(b > bud.b){
        res.b = b - bud.b;
    } else {
        res.b = bud.b - b;
    }
    return res;
}

Budget Budget::mult(const Budget bud) const{
    Budget res{};
    res.a = round(a * bud.a * 100) / 100;
    res.b = round(b * bud.b * 100) / 100;
    return res;
}

Budget Budget::div(const Budget bud) const{
    Budget res{};
    res.a = round(a / bud.a * 100) / 100;
    res.b = round(b / bud.b * 100) / 100;
    return res;
}

bool Budget::com(const Budget bud) const{
    return bud.a == a && bud.b == b;
}

bool Budget::no(const Budget bud) const{
    return (bud.a != a || bud.b != b);
}

```

```

int main(int argc, char *argv[]){
    if (argc != 2) {
        std::cout << "Error";
        return 111;
    }
    std::ifstream fr;
    fr.open(argv[1]);
    if (!fr){
        std::cout << "Error";
        return 100;
    }
    Budget my_bud{};
    my_bud.read(fr);
    std::cout << "Object1" << '\n';
    my_bud.print(std::cout);
    Budget his_bud{};
    his_bud.read(fr);
    std::cout << "Object2" << '\n';
    his_bud.print(std::cout);
    bool tmp1 = my_bud.com(his_bud);
    if(tmp1){
        std::cout << "Object1 == Object2 - Yes" << '\n' << '\n';
    } else{
        std::cout << "Object1 == Object2 - No" << '\n' << '\n';
    }
    bool tmp2 = my_bud.no(his_bud);
    if(tmp2){
        std::cout << "Object1 != Object2 - Yes" << '\n' << '\n';
    } else{
        std::cout << "Object1 != Object2 - No" << '\n' << '\n';
    }
    std::cout << "Addition" << '\n';
    my_bud.add(his_bud).print(std::cout);
    std::cout << "Subtraction" << '\n';
    my_bud.sub(his_bud).print(std::cout);
    std::cout << "Division" << '\n';
    my_bud.div(his_bud).print(std::cout);
    std::cout << "Multiplication" << '\n';
    my_bud.mult(his_bud).print(std::cout);
    fr.close();
    return 0;
}

```

### Набор тестов

Тест	Результат выполнения
1 1	Main part of the budget: 1
1 1	Borrowed part of the budget: 1
	Main part of the budget: 1

	<p>Borrowed part of the budget: 1</p> <p>Object1 == Object2 - Yes Object1 != Object2 - No</p> <p>Main part of the budget: 2 Borrowed part of the budget: 2</p> <p>Main part of the budget: 0 Borrowed part of the budget: 0</p> <p>Main part of the budget: 1 Borrowed part of the budget: 1</p> <p>Main part of the budget: 1 Borrowed part of the budget: 1</p>
100.111    20.3359 5.01    4.999	<p>Main part of the budget: 100.11 Borrowed part of the budget: 20.34</p> <p>Main part of the budget: 5.01 Borrowed part of the budget: 5</p> <p>Object1 == Object2 - No Object1 != Object2 - Yes</p> <p>Main part of the budget: 105.12 Borrowed part of the budget: 25.34</p> <p>Main part of the budget: 95.1 Borrowed part of the budget: 15.34</p> <p>Main part of the budget: 19.98 Borrowed part of the budget: 4.07</p> <p>Main part of the budget: 501.55 Borrowed part of the budget: 101.7</p>
55    9 5    3	<p>Main part of the budget: 55 Borrowed part of the budget: 9</p> <p>Main part of the budget: 5 Borrowed part of the budget: 3</p> <p>Object1 == Object2 - No Object1 != Object2 - Yes</p> <p>Main part of the budget: 60 Borrowed part of the budget: 12</p> <p>Main part of the budget: 50 Borrowed part of the budget: 6</p> <p>Main part of the budget: 11 Borrowed part of the budget: 3</p> <p>Main part of the budget: 275 Borrowed part of the budget: 27</p>

## Ссылка на репозиторий на GitHub

[https://github.com/AlexPshen/oop\\_exercise\\_01.git](https://github.com/AlexPshen/oop_exercise_01.git)

### Объяснение результатов работы программы

Первый и второй объекты в данной программе получают данные из файлов test\_?.txt. Эти данные округляются до двух знаков после запятой. Вывод результата работы программы осуществляется в стандартный поток вывода.

Данная программа выводит следующие данные:

1. Данные первого объекта класса Budget
2. Данные второго объекта класса Budget
3. Результат сравнения двух данных классов, проверяющее равенство этих классов (функция `bool com(const Budget bud, std::ostream& os) const`)
4. Результат сравнения двух данных классов, проверяющее неравенство этих классов (функция `bool no(const Budget bud, std::ostream& os) const`)
5. Результат суммы 2 объектов
6. Результат вычитания по модулю 2 объектов
7. Результат деления первого объекта на второй
8. Результат умножения 2 объектов

### Вывод

В данной лабораторной работе я получил опыт реализации простых классов, познакомился с системой сборки Cmake, поработал с GitHub. Я создал класс моего варианта задания, реализовал для него арифметические операции сложения, вычитания, умножения, деления, а также операции сравнения.