**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Курносов Максим Юрьевич

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

1. **Постановка задачи**

Создать класс Budget для работы с бюджетом. класс состоит из двух вещественных чисел (a,b). где а - 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 деления, а также операции сравнения в виде перегрузки операторов. необходимо реализовать пользовательский литерал для работы с константами типа budget.

1. **Описание программы**

Программа в файле main.cpp с помощью дружественных функций и методов класса из файла Budget.cpp , где и описан весь класс

Budget ,и перегружены все заданые операции , а так же реализованы операции сравнения, работает с входными данными из файлов test\_01.txt test\_02.txt test\_03.txt и выводит результаты в стандартный вывод.

В результате выполнения программы выводятся значения соответствующие выполнению всех перегруженных операций и показ значений, поступивших на ввод программы в формате , описанном в Budget.cpp.

1. **Набор тестов**

Все тесты содержат 4 значения , хранятся через пробел.

Первое и третье значения: первый аргумент класса Budget.

Второе и четвёртое значения: второй аргумент класса Budget.

test\_01.txt:

15.02 23.061 7776.0055 880.051561

test\_02.txt:

67.003 2793.856 678.465 88780.221

test\_03.txt:

144.2 234.01 1176.12 8540.151

1. **Результаты выполнения тестов**

**For file test\_01.txt:**

a {

15.02 - собств. часть

23.06 - заёмн. часть

}

b {

7776.01 - собств. часть

880.05 - заёмн. часть

}

a + b:

7791.03 - собств. часть

903.11 - заёмн. часть

a - b:

-7760.99 - собств. часть

-856.99 - заёмн. часть

a \* b:

116795.60 - собств. часть

20294.87 - заёмн. часть

a / b:

0.00 - собств. часть

0.03 - заёмн. часть

a > b is false

a >= b is false

a < b is true

a <= b is true

a == b is false

a != b is true

701.72 - собств. часть

74225.12 - заёмн. часть

For file test\_02.txt:

a {

67.00 - собств. часть

2793.86 - заёмн. часть

}

b {

678.47 - собств. часть

88780.22 - заёмн. часть

}

a + b:

745.47 - собств. часть

91574.08 - заёмн. часть

a - b:

-611.46 - собств. часть

-85986.36 - заёмн. часть

a \* b:

45459.19 - собств. часть

248039136.00 - заёмн. часть

a / b:

0.10 - собств. часть

0.03 - заёмн. часть

a > b is false

a >= b is false

a < b is true

a <= b is true

a == b is false

a != b is true

701.72 - собств. часть

74225.12 - заёмн. часть

For file test\_03.txt:

a {

144.20 - собств. часть

234.01 - заёмн. часть

}

b {

1176.12 - собств. часть

8540.15 - заёмн. часть

}

a + b:

1320.32 - собств. часть

8774.16 - заёмн. часть

a - b:

-1031.92 - собств. часть

-8306.14 - заёмн. часть

a \* b:

169596.50 - собств. часть

1998480.75 - заёмн. часть

a / b:

0.12 - собств. часть

0.03 - заёмн. часть

a > b is false

a >= b is false

a < b is true

a <= b is true

a == b is false

a != b is true

701.72 - собств. часть

74225.12 - заёмн. часть

1. **Листинг программы**

**For file Budget.cpp:**

#include <iostream>

#include <cstdio>

using namespace std;

/\*В этом файле описан классс Budget и его методы\*/

class Budget

{

public:

float a; //собственная часть средств бюджета в рублях

float b; //заёмная часть бюджета в рублях

Budget(float x, float y)

{

this->a = x;

this->b = y;

}

~Budget()

{

this->a = 0;

this->b = 0;

}

void Show(char \*x)

{

printf("%s {\n", x);

printf("%.2f - собств. часть\n %.2f - заёмн. часть\n", this->a, this->b);

printf("}\n\n");

}

void Show()

{

printf("%.2f - собств. часть\n%.2f - заёмн. часть\n\n", this->a, this->b);

}

// ////////////////// операторы

Budget operator+(Budget x)

{

Budget Bu((this->a + x.a), (this->b + x.b));

return Bu;

}

Budget operator-(Budget x)

{

Budget Bu((this->a - x.a), (this->b - x.b));

return Bu;

}

Budget operator\*(Budget x)

{

Budget Bu((this->a \* x.a), (this->b \* x.b));

return Bu;

}

Budget operator/(Budget x)

{

Budget Bu((this->a / x.a), (this->b / x.b));

return Bu;

}

// ////////////////// сравнение

bool operator>(Budget x)

{

if (this->a > x.a)

{

return true;

}

return false;

}

bool operator<(Budget x)

{

if (this->a < x.a)

{

return true;

}

return false;

}

bool operator>=(Budget x)

{

if (this->a >= x.a)

{

return true;

}

return false;

}

bool operator<=(Budget x)

{

if (this->a <= x.a)

{

return true;

}

return false;

}

bool operator==(Budget x)

{

if (this->a == x.a)

{

return true;

}

return false;

}

bool operator!=(Budget x)

{

if (this->a != x.a)

{

return true;

}

return false;

}

};

**For file main.cpp:**

#include <iostream>

#include <fstream>

#include "Budget.cpp"

using namespace std;

Budget null(-1.0f, -1.0f);

Budget zero(0.0f, 0.0f);

Budget one(10.0f, 10.0f);

Budget two(701.7159f, 74225.114446f);

Budget operator"" \_bg(const char \*s, size\_t st)

{

if (s == "zero")

{

return zero;

}

if (s == "one")

{

return one;

}

if (s == "two")

{

return two;

}

return null;

}

int main(int argc, char \*argv[])

{

for (int i = 1; i < argc; i++)

{

cout << "For file " << argv[i] << ":\n" << endl;

ifstream fin;

fin.open(argv[i]);

float x, y, d, f;

fin >> x >> y >> d >> f;

fin.close();

Budget a(x, y), b(d, f);

a.Show("a");

b.Show("b");

Budget c = a + b;

cout << "a + b: " << endl;

;

c.Show();

c = a - b;

cout << "a - b: " << endl;

;

c.Show();

c = a \* b;

cout << "a \* b: " << endl;

;

c.Show();

c = a / b;

cout << "a / b: " << endl;

;

c.Show();

cout << "a > b is ";

if (a > b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

cout << "a >= b is ";

if (a >= b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

cout << "a < b is ";

if (a < b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

cout << "a <= b is ";

if (a <= b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

cout << "a == b is ";

if (a == b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

cout << "a != b is ";

if (a != b)

{

cout << "true\n"

<< endl;

}

else

{

cout << "false\n"

<< endl;

}

Budget lit = "two"\_bg;

lit.Show();

}

}

**For file see.cpp:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main(int argc, char \*argv[])

{

string line;

ofstream fout;

fout.open(".see.output");

for (int i = 1; i < argc; i++)

{

fout << "For file " << argv[i] << ":\n" << endl;

ifstream fin;

fin.open(argv[i]);

if (fin.is\_open())

{

while (! fin.eof() )

{

getline(fin,line);

fout << line << endl;

}

printf("\n\n");

fin.close();

}

fin.close();

}

fout.close();

}

**For file CMakeLists.txt:**

all: oop\_exercise\_02

oop\_exercise\_02: main.o Budget.o

g++ main.o Budget.o -o oop\_exercise\_02

main.o: main.cpp

g++ -c main.cpp

Budget.o: Budget.cpp

g++ -c Budget.cpp

clean:

rm -rf \*.o oop\_exercise\_02

1. **Вывод**

В процессе выполнения лабораторной работы были закреплены навыки создания классов, работы с классами, перегрузкой операторов , а также приобретены новые навыки в области работы с перегрузкой операторов .

1. **Источники информации:**

1.Бьярне Страуструп. Программирование: принципы и практика использования C++, исправленное издание = Programming: Principles and Practice Using C++. — М.: «Вильямс», 2011. — С. 1248. — ISBN 978-5-8459-1705-8.

2.Канал #SimpleCode YouTube <https://m.youtube.com/channel/UCtLKO1Cb2GVNrbU7Fi0pM0w>

3. Статья на сайте Хабр автор: [Tibr](https://habr.com/ru/users/Tibr/)7 ноября 2011 в 01:02 https://habr.com/ru/post/132014/