**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Гребенков Дмитрий Игоревич

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

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

В классе Budget используются две переменные типа double, которые представляют собственную и заемную часть бюджета. Добавлены функции, реализующие ввод и вывод информации об объекте класса, а также функции, выполняющие операции согласно варианту задания.

Взаимодействие с пользователем осуществляется при помощи меню.

1. Наборы тестов, результаты работы программы

Первый набор (демонстрация работы основных функций):

0. Print menu.

1. Enter the data of the first budget.

2. Enter the data of the second budget.

3. Print the data of the first and second budgets.

4. Addition of the first and second budgets.

5. Subtract the second budget from the first.

6. Multiply the first budget by number.

7. Divide first budget by number.

8. Compare the first budget with the second.

9. Quit the program.

Choose option:

1

Enter own part of the budget: 4.449

Enter borrowed part of the budget: 3.313

Choose option:

2

Enter own part of the budget: 5

Enter borrowed part of the budget: 1

Choose option:

3

(4.45, 3.31).

(5, 1).

Choose option:

4

(9.45, 4.31).

Choose option:

5

Error: budget cannot be negative.

Choose option:

2

Enter own part of the budget: 2

Enter borrowed part of the budget: 1.31

Choose option:

3

(4.45, 3.31).

(2, 1.31).

Choose option:

5

(2.45, 2).

Choose option:

1

Enter own part of the budget: 2

Enter borrowed part of the budget: 3

Choose option:

6

Enter multiplier: 2

(4, 6).

Choose option:

7

Enter divider: 2

(1, 1.5).

Choose option:

7

Enter divider: 0

Error: division by zero.

Choose option:

1

Enter own part of the budget: 2

Enter borrowed part of the budget: 3

Choose option:

2

Enter own part of the budget: 2

Enter borrowed part of the budget: 2

Choose option:

8

Budgets are not equal.

Choose option:

1

Enter own part of the budget: 0

Enter borrowed part of the budget: 2

Choose option:

2

Enter own part of the budget: 0

Enter borrowed part of the budget: 2

Choose option:

8

Budgets are equal.

Choose option:

9

Второй набор (некорректные данные):

0. Print menu.

1. Enter the data of the first budget.

2. Enter the data of the second budget.

3. Print the data of the first and second budgets.

4. Addition of the first and second budgets.

5. Subtract the second budget from the first.

6. Multiply the first budget by number.

7. Divide first budget by number.

8. Compare the first budget with the second.

9. Quit the program.

Choose option:

1

Enter own part of the budget: -5

Error: budget cannot be negative.

Choose option:

2

Enter own part of the budget: 8

Enter borrowed part of the budget: -5.9

Error: budget cannot be negative.

Choose option:

3

Error: no first budget data entered.

Error: no second budget data entered.

Choose option:

4

Error: no first budget data entered.

Error: no second budget data entered.

Choose option:

5

Error: no first budget data entered.

Error: no second budget data entered.

Choose option:

6

Error: no first budget data entered.

Choose option:

7

Error: no first budget data entered.

Choose option:

8

Error: no first budget data entered.

Error: no second budget data entered.

Choose option:

0

0. Print menu.

1. Enter the data of the first budget.

2. Enter the data of the second budget.

3. Print the data of the first and second budgets.

4. Addition of the first and second budgets.

5. Subtract the second budget from the first.

6. Multiply the first budget by number.

7. Divide first budget by number.

8. Compare the first budget with the second.

9. Quit the program.

Choose option:

1

Enter own part of the budget: 0

Enter borrowed part of the budget: 1

Choose option:

1

Enter own part of the budget: 0

Enter borrowed part of the budget: 1

Choose option:

2

Enter own part of the budget: 2

Enter borrowed part of the budget: 2

Choose option:

3

(0, 1).

(2, 2).

Choose option:

7

Enter divider: 0

Error: division by zero.

Choose option:

5

Error: budget cannot be negative.

Choose option:

9

1. Листинг программы

/\*

Лабораторная работа № 1

Гребенков Дмитрий Игоревич

М8О-207Б-19

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.

\*/

#include <iostream>

#include <cmath>

class Budget {

public:

double a = 0;

double b = 0;

void ent (double x, double y) {

a = (round (x \* 100)) / 100;

b = (round (y \* 100)) / 100;

}

void output () {

std::cout << "(" << a << ", " << b << "). " << std::endl;

}

};

void plus (Budget first, Budget second, Budget \* third) {

third->a = first.a + second.a;

third->b = first.b + second.b;

}

int minus (Budget first, Budget second, Budget \* third) {

if ((first.a >= second.a) && (first.b >= second.b)) {

third->a = first.a - second.a;

third->b = first.b - second.b;

return 0;

}

else {

std::cout << "Error: budget cannot be negative." << std::endl;

return 1;

}

}

void multiply (Budget first, double c, Budget \* third) {

third->a = round ((c \* first.a) \* 100) / 100;

third->b = round ((c \* first.b) \* 100) / 100;

}

int divide (Budget first, double c, Budget \* third) {

if (c != 0) {

third->a = round ((first.a / c) \* 100) / 100;

third->b = round ((first.b / c) \* 100) / 100;

return 0;

}

else {

std::cout << "Error: division by zero." << std::endl;

return 1;

}

}

void compare (Budget first, Budget second) {

if ((first.a == second.a) && (first.b == second.b)) {

std::cout << "Budgets are equal." << std::endl;

}

else {

std::cout << "Budgets are not equal." << std::endl;

}

}

int main () {

int point = 0;

double x;

double y;

double c;

int first\_flag = 0;

int second\_flag = 0;

int exf = 0;

Budget first;

Budget second;

Budget third;

std::cout << "0. Print menu." << std::endl;

std::cout << "1. Enter the data of the first budget." << std::endl;

std::cout << "2. Enter the data of the second budget." << std::endl;

std::cout << "3. Print the data of the first and second budgets." << std::endl;

std::cout << "4. Addition of the first and second budgets." << std::endl;

std::cout << "5. Subtract the second budget from the first." << std::endl;

std::cout << "6. Multiply the first budget by number." << std::endl;

std::cout << "7. Divide first budget by number." << std::endl;

std::cout << "8. Compare the first budget with the second." << std::endl;

std::cout << "9. Quit the program." << std::endl;

while (point != 9) {

std::cout << "Choose option:" << std::endl;

std::cin >> point;

switch (point) {

case 0:

std::cout << "0. Print menu." << std::endl;

std::cout << "1. Enter the data of the first budget." << std::endl;

std::cout << "2. Enter the data of the second budget." << std::endl;

std::cout << "3. Print the data of the first and second budgets." << std::endl;

std::cout << "4. Addition of the first and second budgets." << std::endl;

std::cout << "5. Subtract the second budget from the first." << std::endl;

std::cout << "6. Multiply the first budget by number." << std::endl;

std::cout << "7. Divide first budget by number." << std::endl;

std::cout << "8. Compare the first budget with the second." << std::endl;

std::cout << "9. Quit the program." << std::endl;

break;

case 1: {

std::cout << "Enter own part of the budget: ";

std::cin >> x;

if (x < 0) {

std::cout << "Error: budget cannot be negative." << std::endl;

break;

}

std::cout << "Enter borrowed part of the budget: ";

std::cin >> y;

if (y < 0) {

std::cout << "Error: budget cannot be negative." << std::endl;

break;

}

first\_flag = 1;

first.ent (x, y);

break;

}

case 2: {

std::cout << "Enter own part of the budget: ";

std::cin >> x;

if (x < 0) {

std::cout << "Error: budget cannot be negative." << std::endl;

break;

}

std::cout << "Enter borrowed part of the budget: ";

std::cin >> y;

if (y < 0) {

std::cout << "Error: budget cannot be negative." << std::endl;

break;

}

second\_flag = 1;

second.ent (x, y);

break;

}

case 3: {

if (first\_flag == 1) {

first.output ();

}

else {

std::cout << "Error: no first budget data entered." << std::endl;

}

if (second\_flag == 1) {

second.output ();

}

else {

std::cout << "Error: no second budget data entered." << std::endl;

}

break;

}

case 4: {

if (first\_flag != 1) {

std::cout << "Error: no first budget data entered." << std::endl;

}

if (second\_flag != 1) {

std::cout << "Error: no second budget data entered." << std::endl;

}

if ((first\_flag != 1) || (second\_flag != 1)) {

break;

}

else {

plus (first, second, &third);

third.output ();

}

break;

}

case 5: {

if (first\_flag != 1) {

std::cout << "Error: no first budget data entered." << std::endl;

}

if (second\_flag != 1) {

std::cout << "Error: no second budget data entered." << std::endl;

}

if ((first\_flag != 1) || (second\_flag != 1)) {

break;

}

else {

exf = minus (first, second, &third);

if (exf == 0) {

third.output ();

}

}

break;

}

case 6: {

if (first\_flag != 1) {

std::cout << "Error: no first budget data entered." << std::endl;

}

if ((first\_flag != 1) || (second\_flag != 1)) {

break;

}

else {

std::cout << "Enter multiplier: ";

std::cin >> c;

multiply (first, c, &third);

third.output ();

}

break;

}

case 7: {

if (first\_flag != 1) {

std::cout << "Error: no first budget data entered." << std::endl;

}

if ((first\_flag != 1) || (second\_flag != 1)) {

break;

}

else {

std::cout << "Enter divider: ";

std::cin >> c;

exf = divide (first, c, &third);

if (exf == 0) {

third.output ();

}

}

break;

}

case 8: {

if (first\_flag != 1) {

std::cout << "Error: no first budget data entered." << std::endl;

}

if (second\_flag != 1) {

std::cout << "Error: no second budget data entered." << std::endl;

}

if ((first\_flag != 1) || (second\_flag != 1)) {

break;

}

else {

compare (first, second);

}

break;

}

case 9: {

break;

}

default: {

std::cout << "Error: incorrect input." << std::endl;

break;

}

}

}

return 0;

}

1. Вывод

Получены базовые навыки работы с классами в C++.

Литература:

1. C++ | Объектно-Ориентированное программирование [Электронный ресурс] URL: <https://metanit.com/cpp/tutorial/5.1.php> (дата обращения 30.09.2020)