ЛАБОРАТОРНАЯ РАБОТА №6

“Программирование с использованием объявления пользовательских классов, вызова методов”

**Цель работы:** сформировать знания и умения по объявлению и использованию классов в ЯП Си++, приобрести практические навыки создания простейших программ на языке Си++ с использованием классов.

**Задание 2:**

#include<iostream>

using namespace std;

class fraction {

int m; //числитель

int n; //знаменатель

public: void input();// ввод дроби

void output();// вывод дроби

void mult(int a);// умножение дроби на целое число

};

//Программируем методы класса fraction

void fraction::input() {// ввод дроби

cout << "Chislitel':"; cin >> m; cout << endl;

cout << "Znamenatel'"; cin >> n; cout << endl;

}

void fraction::output() {// вывод дроби

cout << m << "/" << n;

}

void fraction::mult(int a) {// умножение дроби на число

m = m \* a;

}

//Программируем алгоритм решения основной задачи

int main() {

fraction A; //дробь

int k; // число на которое нужно умножить дробь

// ввод дроби

cout << "Vvedite drob': " << endl;

/\*Чтобы ввести дробь А, нужно у объекта A вызвать метод input. \*/

A.input();

//умножение дроби на целое число

cout << "Vvedite chislo na kotoroe nuzno ymnozit' drob'";

cin >> k;

/\*Чтобы умножить дробь А на число k нужно у объекта A вызвать \*/

A.mult(k);//вызываем метод mult для дроби A

// вывод дроби

A.output();//вызываем метод output для дроби A

return 0;

}

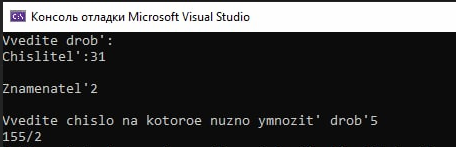


Рисунок 1 – выполнение кода примера 1

**Задание 3:**

# include<iostream>

using namespace std;

class fraction {

int m; //числитель

int n; //знаменатель

public: void input();// ввод дроби

void output();// вывод дроби

void mult(int a);// умножение дроби на целое число

void set(int a, int b);// изменение значений дроби

void check();

};

void fraction::check() {

if (m > n) {

cout << "Drob' nepravilnaya" << endl;;

}

else { cout << "Pravilnaya"<<endl; }

}

void fraction::input() {

cout << "vvedite drob" << endl;

cout << "Chislitel':"; cin >> m;

cout << "Znamenatel':"; cin >> n;

}void fraction::output() {

cout << m << "/" << n<<endl;

}

void fraction::set(int a, int b) {

//присваивание значений числителю и знаменателю дроби

m = a; n = b;

}

void fraction::mult(int a) {

m = m \* a;

cout << m << "/" << n;

}

int main() {

fraction A; //дробь

A.set(1, 2);//присваиваем дроби значение 1/2

A.output();// выводим дробь

fraction B[2]; //массив дробей

for (int i = 0; i < 2; i++) { B[i].input(); }// ввод массива дробей

for (int i = 0; i < 2; i++) { B[i].output(); B[i].check(); }// вывод массива дробей

//Доступ к объектам можно получить через указатель.В этом случае к членам объекта обращаются с

//помощью оператора->.

// объявляем указатель типа fraction и присваиваем ему адрес дроби A

fraction\* p = &A;

p->input();// ввод дроби

p->output();// вывод дроби

return 0;

}

//В Си++ можно создавать и использовать массивы объектов.

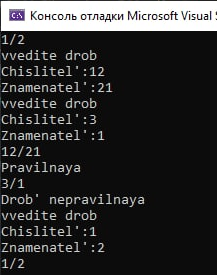


Рисунок 2 – выполнение кода примера 2 и 3 задания

**Задание 4:**

/\*4.Создать класс комната .Свойства: длина, ширина, высота. Методы : ввод, редактирование, вывод,

подсчет площади, периметра, площади стен.С использованием данного класса реализуйте алгоритм

определения количества катушек обоев, необходимых для оклейки стен.\*/

#include <iostream>

class Room

{

int dlina;

int wieght;

int hieght;

public:

void Input();

void Redakt(int d, int w, int h) {

dlina = d; wieght = w; hieght = h;

}

void Output();

int Ploshd();

int Perimetr();

int PloshdSten();

int Oboi();

};

void Room::Input()

{

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

std::cin >> dlina;

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

std::cin >> wieght;

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

std::cin >> hieght;

std::cout << std::endl;

std::cout << std::endl;

}

void Room::Output()

{

std::cout << "Длина = " << dlina << std::endl;

std::cout << "Ширина = " << wieght << std::endl;

std::cout << "Высота = " << hieght << std::endl;

std::cout << std::endl;

std::cout << std::endl;

std::cout << std::endl;

}

int Room::Ploshd()

{

int S = dlina \* wieght;

std::cout << "Площадь комнаты = " << S << std::endl;

return S;

}

int Room::PloshdSten()

{

int S1 = ((dlina \* 2) + (wieght \* 2)) \* hieght;

std::cout << "Площадь стен в комнате = " << S1 << std::endl;

return S1;

}

int Room::Perimetr()

{

int S2 = (dlina + wieght) \* 2;

std::cout << "Периметр комнаты = " << S2 << std::endl;

return S2;

}

int Room::Oboi()

{

int S3=(((dlina \* 2) + (wieght \* 2)) \* hieght)/25;

std::cout << "Рулоны = " << S3 << std::endl;

return S3;

}

int main()

{

Room room;

// std::cout << "Ваши расходы";

room.Input();

room.Output();

room.Redakt(20, 15, 10);

room.Output();

room.Ploshd();

room.Perimetr();

room.PloshdSten();

room.Oboi();

return 0;

}

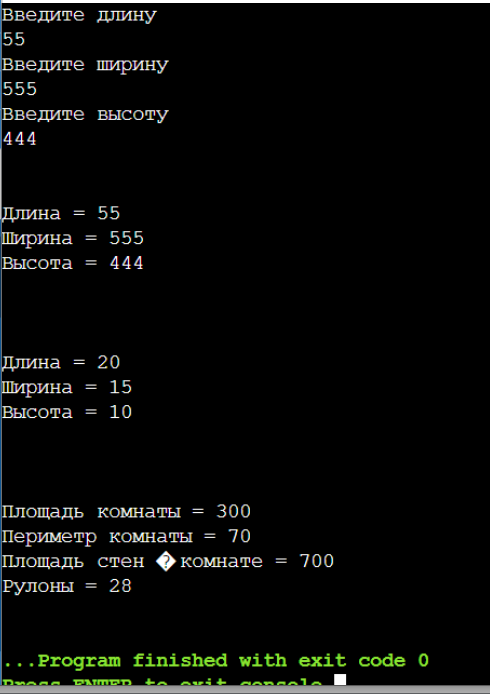


Рисунок 3 – выполнение кода 4 задания

**Контрольные вопросы:**

1. Класс – это структурированный тип данных, включающий в себе в качестве элементов типизированные данные и функции, применяемые по отношению к этим данным.

2. имя\_переменной-объекта.имя\_функции-метода;//вызов метода

3. Спецификаторы доступа определяют откуда будут видны члены класса

4. Инкапсуляция – механизм объединения данных и методов, позволяющих манипулировать этими данными, и защиты и того и другого от внешнего вмешательства или неправильного использования. Инкапсуляция означает, что в качестве единицы целого рассматривается объединение некоторой группы данных и некоторой группы функций.

Наследование – это процесс, посредством которого один объект может наследовать свойства другого объекта и добавлять к ним черты, характерные только для него. Наследование позволяет строить иерархию объектов, переходя от более общего к частному, уточняя и конкретизируя объект.

Полиморфизм – это свойство, которое позволяет одно и тоже имя использовать для решения нескольких технически разных задач. Т. е. имя определяет класс (область) действий, которые в зависимости от типа данных могут существенно отличаться. Например, можно определить три типа переменных: целые, комплексные числа и векторы. Для каждого из них можно определить функцию

**Вывод:** сформировала знания и умения по объявлению и использованию классов в ЯП Си++, приобрести практические навыки создания простейших программ на языке Си++ с использованием классов.