

## Проверка работы программы.

В данной программе я реализовала класс *Квадрат* или как он называется “*Squareclasses*”. В данном файле описана проверка всех методов и необходимых пунктов из задания лабораторной работы №2.

В данном классе описаны методы (Помимо сеттеров и геттеров): Периметр и площадь. Создадим переменную типа нашего класса и зададим ей значение координат такие, чтобы квадрат существовал. После проверим наши методы.

```
Squareclasses sqr;
//Задаём квадрат со стороной a=3
sqr.setXY(0,0);
sqr.setX1Y1(3,0);
sqr.setX2Y2(3,3);
sqr.setX3Y3(0,3);
//Теперь проверим функции нахождения площади и периметра
try {
    cout<<"perimeter="<<sqr.perimeter()<<endl;
    cout<<"area="<<sqr.area()<<endl;
}
catch (Except e) {
    cout<<"Error:Square doesn't exist"<<endl;
}
```

Далее создадим 3 вида массивов, обычный массив из объектов, динамический массив из объектов и массив из указателей на объекты.

```
//Теперь создадим массив из объектов этого класса
Squareclasses sqr1[2]; //Создали массив из двух объектов
sqr1[0]=sqr; //Чтобы не писать заново, возьмём значения из kv
//Возьмём площадь ещё раз, чтобы проверить работают ли методы класса в массиве
try{
    cout<<"area="<<sqr1[0].area(); //Всё выводится значит методы работают
}
catch(Except e)
{
    cout<<"Error:Square doesn't exist"<<endl;
}
//Теперь выделим память под объект динамически и ещё раз создадим массив
Squareclasses* sqr2=new Squareclasses[2];
sqr2[0].setXY(1,1);
sqr2[0].setX1Y1(-5,1);
sqr2[0].setX2Y2(-5,-5);
sqr2[0].setX3Y3(1,5);
//Методы работают с динамически созданным объектом
//Теперь создадим массив из указателей на объект
Squareclasses** sqr3=new Squareclasses*[2];
for(unsigned i=0;i<2;i++){
    sqr3[i]=new Squareclasses;
    sqr3[i]->setXY(rand()%100,rand()%100);
    sqr3[i]->setX1Y1(rand()%100,rand()%100);
    sqr3[i]->setX2Y2(rand()%100,rand()%100);
    sqr3[i]->setX3Y3(rand()%100,rand()%100);
    cout<<"("<<sqr3[i]->getX()<<","<<sqr3[i]->getY()<<")"<<endl;
    cout<<"("<<sqr3[i]->getX1()<<","<<sqr3[i]->getY1()<<")"<<endl;
    cout<<"("<<sqr3[i]->getX2()<<","<<sqr3[i]->getY2()<<")"<<endl;
    cout<<"("<<sqr3[i]->getX3()<<","<<sqr3[i]->getY3()<<")"<<endl;
    cout<<endl;
}
```

Теперь проверим ввод объектов класса в файл и вывод их же из файла. Для примера введём и выведем одну из точек квадрата.

```
//Теперь запишем в файл наш объект одну из координат sqr2[0] для примера
ofstream f;
f.open("test.txt");
f<<sqr2[0].getX()<<endl;
f<<sqr2[0].getY()<<endl;
f.close();
//Теперь для объекта sqr2[1] возьмём данные sqr2[0] из файла
ifstream f1;
float k;
f1.open("test.txt");
f1>>k;
sqr2[1].setXY(k,0);
f1>>k;
sqr2[1].setXY(sqr2[1].getX(),k);
//Для проверки выведем данный объект
cout<<"("<<sqr2[1].getX()<<","<<sqr2[1].getY()<<")"<<endl;
```

Также напомним автотест класса:

testsquare.h

```
#ifndef TESTSQUARE_H
#define TESTSQUARE_H
#include "../squareclasses.h"

void test_squareclasses_setXY(); //Проверяет сеттер первой точки
void test_squareclasses_setX1Y1(); //Проверяет сеттер второй точки
void test_squareclasses_setX2Y2(); //Проверяет сеттер третьей точки
void test_squareclasses_setX3Y3(); //Проверяет сеттер четвёртой точки
void test_squareclasses_perimeter(); //Проверяет метод, вычисляющий периметр
void test_squareclasses_area(); //Проверяет метод, вычисляющий площадь
void test_squareclasses_all(); //Выполняет все проверки выше

#endif // TESTSQUARE_H
```

testsquare.cpp

```
#include<iostream>
#include<cassert>
#include "testsquare.h"

void test_squareclasses_setXY() {
    Squareclasses sqr;
    sqr.setXY(5,4);
    assert((sqr.getX()==5) and (sqr.getY()==4));
}

void test_squareclasses_setX1Y1() {
    Squareclasses sqr;
    sqr.setX1Y1(3,4);
    assert((sqr.getX1()==3) and (sqr.getY1()==4));
}

void test_squareclasses_setX2Y2() {
    Squareclasses sqr;
    sqr.setX2Y2(3,6);
    assert((sqr.getX2()==3) and (sqr.getY2()==6));
}
```

```

void test_squareclasses_setX3Y3() {
    Squareclasses sqr;
    sqr.setX3Y3(5, 6);
    assert((sqr.getX3() == 5) and (sqr.getY3() == 6));
}

void test_squareclasses_perimeter() {
    Squareclasses sqr;
    sqr.setXY(5, 4);
    sqr.setX1Y1(3, 4);
    sqr.setX2Y2(3, 6);
    sqr.setX3Y3(5, 6);
    assert(sqr.perimeter() == 4);
}

void test_squareclasses_area() {
    Squareclasses sqr;
    sqr.setXY(5, 4);
    sqr.setX1Y1(3, 4);
    sqr.setX2Y2(3, 6);
    sqr.setX3Y3(5, 6);
    assert(sqr.area() == 4);
}

void test_squareclasses_all() {
    void test_squareclasses_setXY();
    void test_squareclasses_setX1Y1();
    void test_squareclasses_setX2Y2();
    void test_squareclasses_setX3Y3();
    void test_squareclasses_perimeter();
    void test_squareclasses_area();
    std::cout << "All is okay :)" << std::endl;
}

```

Теперь запустим его в main.cpp введя команду `test_squareclasses_area()` ;

Вот что показывает консоль:

```

All is okay :)
perimeter=12
area=9
area=9(67,67)
(0,34)
(24,69)
(58,78)

(64,64)
(45,5)
(27,81)
(91,61)

(1,1)

```

Всё работает отлично :)