№1

#include <string>

#include <math.h>

#include <iostream>

using namespace std;

class point {

public:

int x1, y1, x2, y2;

float rast() {

float r;

r = sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2));

return r;

}

};

int main() {

point p;

int a, b, c, d;

cin » a » b » c » d;

p.x1 = a;

p.x2 = b;

p.y1 = c;

p.y2 = d;

cout « p.rast();

}

№2

#include <string>

#include <math.h>

#include <iostream>

using namespace std;

class point {

public:

int x1, x2, y1, y2;

float root;

public:

point() {

x1 = 0;

x2 = 0;

y1 = 0;

y2 = 0;

root = 0;

}

public:

int\* getx() {

int arr[2];

arr[0] = x1;

arr[1] = x2;

return arr;

}

int\* gety() {

int arr[2];

arr[0] = y1;

arr[1] = y2;

return arr;

}

void Setx(int a, int b) {

this->x1 = a;

this->x2 = b;

}

void Sety(int a, int b) {

this->y1 = a;

this->y2 = b;

}

private:

float Distance() {

root = sqrt(pow(x2 - x1, 2) + pow((y2 - y1), 2));

}

public:

int getdistance() {

Distance();

return root;

}

~point()

};

int main() {

point c;

int x1, x2, y1, y2;

cin»x1»x2»y1»y2;

c.Setx(x1, x2);

c.Sety(y1, y2);

cout«c.getdistance();

return 0;

}

№3

#include <iostream>

#include <string>

using namespace std;

class triangle {

public:

int a;

int b;

int c;

void SetCoor(int x, int y, int z) {

a = x;

b = y;

c = z;

}

void print() {

cout«a«endl;

cout«b«endl;

cout«c«endl;

}

string GetIsTriangle() {

string s = "";

if (isTriangle())

s = "это треугольник";

else

s = "это не треугольник";

return s;

}

string typeTriangle() {

if (isTriangle()) {

if ((a == 60) && (b == 60) && (c == 60)) {

return "равносторонний треугольник";

}

else if ((a == b) || (a == c) || (b == c)) {

return "равнобедренный треугольник";

}

else {

return "неравносторонний треугольник";

}

}

else {

cout«"это не треугольник";

}

}

private:

bool isTriangle() {

if ((a + b + c) == 180) {

return true;

}

else {

return false;

}

}

};

int main() {

triangle t;

int a, b, c;

cout«"Введите значения"«endl;

cin»a»b»c;

t.SetCoor(a, b, c);

t.print();

cout«t.GetIsTriangle()«endl;

cout«t.typeTriangle()«endl;

return 0;

}