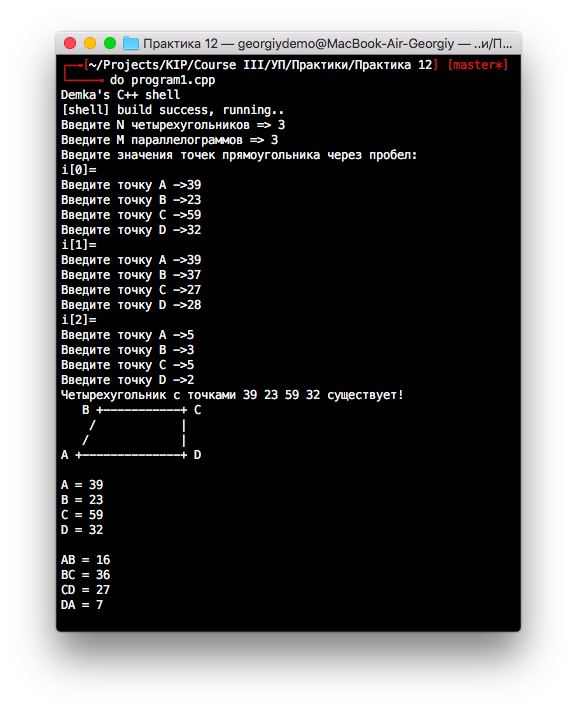
Задание №1

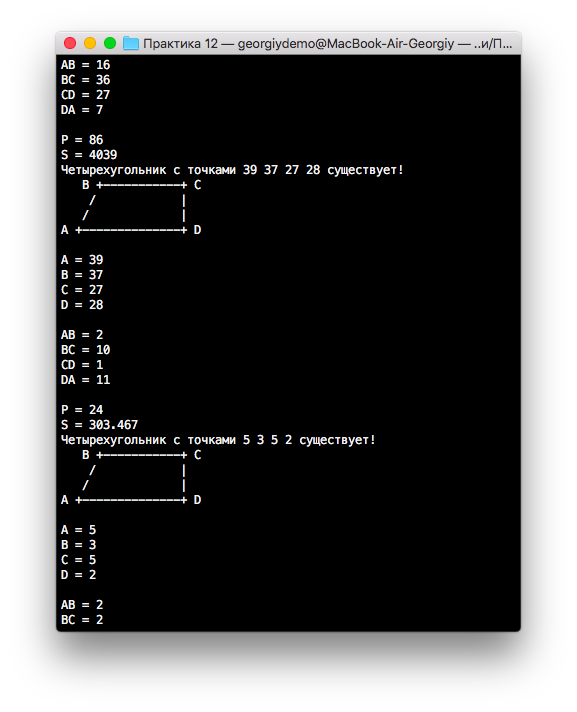
Условие:

Создать класс четырехугольник, члены класса – координаты 4-хточек. Предусмотреть в классе методы проверки существования четырехугольника вычисления и вывода сведений о фигуре – длины сторон, периметр, площадь. Создать производный класс –параллелограмм, предусмотреть в классе проверку, является ли фигура параллелограммом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M параллелограммов, найти среднюю площадь N четырехугольников и параллелограммы наименьшей и наибольшей площади.

Исходный код программы:#include **<iostream>**#include **<cmath>  
using namespace** std;  
**bool** sys = **false**;  
**class** QuadrangleClass {  
  
**public**:  
 **double** s[4];  
 **double** AB, BC, CD, DA;  
};  
  
**class** DefaultClass : **public** QuadrangleClass {  
  
**public**:  
 DefaultClass(**double** s1, **double** s2, **double** s3, **double** s4)  
 {  
 s[0] = s1;  
 s[1] = s2;  
 s[2] = s3;  
 s[3] = s4;  
 AB = abs(s1 - s2);  
 BC = abs(s2 - s3);  
 CD = abs(s3 - s4);  
 DA = abs(s4 - s1);  
 }  
  
 **int** sychestv()  
 {  
 **if** ((AB < (BC + CD + DA)) && (BC < (AB + CD + DA)) && (CD < (AB + BC + DA)) && (DA < (AB + CD + DA)))  
 **return** 1;  
 **return** 0;  
 }  
 **double** perimetr()  
 {  
 **double** p = AB + BC + CD + DA;  
 cout << **"P = "** << p << **"\n"**;  
 **return** p;  
 }  
 **double** ploschad()  
 {  
  
 **double** p = AB + BC + CD + DA;  
 **double** s = sqrt((p - AB) \* (p - BC) \* (p - CD) \* (p - DA));  
 cout << **"S = "** << s << **"\n"**;  
 **return** s;  
 }  
  
 **void** storona()  
 {  
 cout << **" B +-----------+ C\n"**;  
 cout << **" / | \n"**;  
 cout << **" / | \n"**;  
 cout << **"A +--------------+ D\n"**;  
 cout << **"\nA = "** << s[0] << **"\nB = "** << s[1] << **"\nC = "** << s[2] << **"\nD = "** << s[3] << **"\n"**;  
 cout << **"\nAB = "** << AB << **"\n"**;  
 cout << **"BC = "** << BC << **"\n"**;  
 cout << **"CD = "** << CD << **"\n"**;  
 cout << **"DA = "** << DA << **"\n\n"**;  
 }  
};  
  
**class** ParallelClass : **public** QuadrangleClass {  
**public**:  
 ParallelClass(**double** s11, **double** s21, **double** s31, **double** s41)  
 {  
 s[0] = s11;  
 s[1] = s21;  
 s[2] = s31;  
 s[3] = s41;  
 AB = abs(s11 - s21);  
 BC = abs(s21 - s31);  
 CD = abs(s31 - s41);  
 DA = abs(s41 - s11);  
 }  
  
 **int** sychestv()  
 {  
 **if** ((AB == CD) && (BC == DA))  
 **return** 1;  
  
 **return** 0;  
 }  
};  
  
**int** main()  
{  
  
 **int** n, m, o;  
 cout << **"Введите N четырехугольников => "**;  
 cin >> n;  
 cout << **"Введите M параллелограммов => "**;  
 cin >> m;  
  
 **bool** k = **false**, k1 = **false**;  
 **double** \*a, \*b, \*c, \*d, \*a1, \*b1, \*c1, \*d1, \*pl;  
  
 a = **new double**[n];  
 b = **new double**[n];  
 c = **new double**[n];  
 d = **new double**[n];  
 pl = **new double**[n];  
  
 a1 = **new double**[m];  
 b1 = **new double**[m];  
 c1 = **new double**[m];  
 d1 = **new double**[m];  
  
 cout << **"Введите значения точек прямоугольника через пробел:\n"**;  
 **for** (**int** i = 0; i < n; i++) {  
 cout << **"i["** << i << **"]=\n"**;  
 cout << **"Введите точку A ->"**;  
 cin >> a[i];  
 cout << **"Введите точку B ->"**;  
 cin >> b[i];  
 cout << **"Введите точку C ->"**;  
 cin >> c[i];  
 cout << **"Введите точку D ->"**;  
 cin >> d[i];  
 }  
  
 **for** (**int** i = 0; i < n; i++) {  
 DefaultClass ob1(a[i], b[i], c[i], d[i]);  
 **if** (ob1.sychestv() == 1) {  
 cout << **"Четырехугольник с точками "** << a[i] << **" "** << b[i] << **" "** << c[i] << **" "** << d[i] << **" существует!\n"**;  
 ob1.storona();  
 ob1.perimetr();  
 pl[i] = ob1.ploschad();  
 *//pl[i] = ob1.ploschad();* k = **true**;  
 }  
 **else** {  
 cout << **"Четырехугольника с точками "** << a[i] << **" "** << b[i] << **" "** << c[i] << **" "** << d[i] << **" не существует :c\n"**;  
 pl[i] = 0;  
 }  
 }  
 *//Средняя площадь* **double** pl\_sum = 0;  
 **for** (**int** i = 0; i < n; i++)  
 pl\_sum += pl[i];  
 **double** pl\_sred = pl\_sum / n;  
 cout << **"Средняя площадь: "** << pl\_sred << **"\n"**;  
  
 cout << **"Введите значения точек параллелограмма через пробел:"** << endl;  
 **for** (**int** i = 0; i < m; i++) {  
 cout << **"i["** << i << **"]=\n"**;  
 cout << **"Введите точку A ->"**;  
 cin >> a1[i];  
 cout << **"Введите точку B ->"**;  
 cin >> b1[i];  
 cout << **"Введите точку C ->"**;  
 cin >> c1[i];  
 cout << **"Введите точку D ->"**;  
 cin >> d1[i];  
 }  
  
 **for** (**int** i = 0; i < m; i++) {  
 ParallelClass ob1(a1[i], b1[i], c1[i], d1[i]);  
 **if** (ob1.sychestv() == 1) {  
 cout << **"Параллелограмм с точками "** << a1[i] << **" "** << b1[i] << **" "** << c1[i] << **" "** << d1[i] << **" существует!\n"**;  
 k1 = **true**;  
 }  
 **else** {  
 cout << **"Параллелограмма с точками "** << a1[i] << **" "** << b1[i] << **" "** << c1[i] << **" "** << d1[i] << **" не существует :c\n"**;  
 }  
 }  
 **double** maxpl = pl[0];  
 **double** minpl = maxpl;  
 **int** ind\_max = 0;  
 **int** ind\_min = ind\_max;  
  
 **for** (**int** i = 0; i < n; i++) {  
 **if** (pl[i] > maxpl) {  
 maxpl = pl[i];  
 ind\_max = i;  
 }  
 }  
  
 **for** (**int** i = 0; i < n; i++) {  
 **if** (pl[i] < minpl) {  
 minpl = pl[i];  
 ind\_min = i;  
 }  
 }  
  
 **if** (k1 == **true**) {  
 cout << **"S макс = "** << maxpl << **" при точках "** << a[ind\_max] << **" "** << b[ind\_max] << **" "** << c[ind\_max] << d[ind\_max] << **"\n"**;  
 cout << **"S мин = "** << minpl << **" при точках "** << a[ind\_min] << **" "** << b[ind\_min] << **" "** << c[ind\_min] << d[ind\_min] << **"\n"**;  
 }  
 **else** cout << **"Нет ни одного прямоугольника\n"**;  
  
 **return** 0;  
}

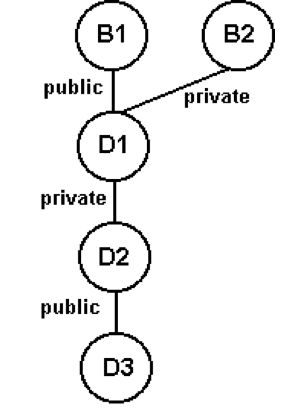
Скриншоты программы:





Задание №2

Условие:



Исходный код программы:

#include **<iostream>  
using namespace** std;  
**class** B2 {  
 **int** e;  
  
**public**:  
 B2(**int** x, **int** y){};  
 B2(**int** x) {   
 e = x; }  
 **void** show\_B2() { cout << **"B2= "** << e << **"\n"**; }  
};  
  
**class** B1 {  
 **int** a;  
  
**public**:  
 B1(**int** x, **int** y, **int** i, **int** j){};  
 B1(**int** x) { a = x; }  
 **void** show\_B1() { cout << **"B1= "** << a << **"\n"**; }  
};  
  
**class** D1 : **private** B2, **public** B1 {  
 **int** b;  
  
**public**:  
 D1(**int** x, **int** y,**int** i,**int** u) : B2(x), B1(i) { b = x; };  
  
 **void** show\_D1()  
 {  
 cout << **"D1= "** << b << **"\n"**;  
 show\_B1();  
 show\_B2();  
 }  
};  
  
**class** D2 : **public** D1 {  
 **int** c;  
**public**:  
 D2(**int** x, **int** y,**int** i,**int** u) : D1(x, y, i,u) { c = i; }  
 **void** show\_D2() { cout << **"D2= "** << c << **"\n"**; show\_D1(); }  
};  
  
  
**class** D3 : **public** D2 {  
 **int** d;  
**public**:  
 D3(**int** x, **int** y,**int** i,**int** u) : D2(x, y, i,u) { d = i; }  
 **void** show\_D3() { cout << **"D3= "** << d << **"\n"**; show\_D2(); }  
};  
  
**int** main()  
{  
 setlocale(LC\_ALL, **"rus"**);  
  
 D3 temp(100, 200, 300, 400);  
 cout << **"\nПолученная иерархия: \n"**;  
 temp.show\_D3();  
  
 **return** 0;  
}

Скриншоты программы:

