**נספח**

 **להלן תוכנית המבקשת משוואה מדרגה 3 ומוצאת את השורשים הממשיים שלה:**

/\* cubic\_eq2.java - find real roots of a cubic equation \*/

import java.util.\*;

class ComplexNumber

{

 double realp;

 double imagp;

}

public class cubic\_eq2

{

 static final double EPS=0.0000000000001;

 static final double PI=3.1415926536;

 static int solve\_quadratic(double a, double b,

 double c,

 ComplexNumber c1,

 ComplexNumber c2)

 {

 double delta, delta\_root, xv, qv;

 if ( a == 0)

 if ( b == 0)

 return 0;

 else

 {

 c1.realp = -c/b;

 c1.imagp = 0;

 return 1;

 } /\* else \*/

 xv = -b/(2\*a); /\* Take care of square quad \*/

 qv = a\*xv\*xv + b\*xv + c;

 if(Math.abs(qv) < EPS)

 {

 c1.realp = c2.realp = xv;

 c1.imagp = c2.imagp = 0.0;

 return 2;

 } /\* if \*/

 delta = b\*b - 4\*a\*c;

 if ( delta < 0.0 )

 {

 c1.realp = c2.realp = -b/(2\*a); /\* real part only \*/

 delta\_root = Math.sqrt(-delta);

 c1.imagp = - delta\_root/(2.0\*a);

 c2.imagp = delta\_root/(2.0\*a);

 return 0;

 } /\* if \*/

 else

 {

 delta\_root = Math.sqrt(delta);

 c1.realp = (-b - delta\_root)/(2.0\*a);

 c2.realp = (-b + delta\_root)/(2.0\*a);

 c1.imagp = 0;

 c2.imagp = 0;

 return 2;

 } /\* else \*/

 } /\* solve\_quadratic \*/

 static void divide\_cubic(double a[], double x)

 {

 a[1] = a[1] + a[0]\*x;

 a[2] = a[2] + a[1]\*x;

 } /\* divide\_cubic \*/

 static void cubic\_root(double x, double y, ComplexNumber c )

 {

 double radius, theta, temp;

 if (Math.abs(x) < EPS && Math.abs(y) < EPS)

 {

 c.realp = c.imagp = 0.0;

 return;

 } /\* if \*/

 radius = Math.sqrt(x\*x + y\*y);

 theta = (Math.asin(Math.abs(y/radius)));

 temp = Math.exp(Math.log(radius)/3.0);

 if ((x < 0)&&(y>=0))

 theta = 3\*PI - theta;

 else

 if ((x<0)&&(y<=0))

 theta = 3\*PI + theta;

 else

 if ((x>=0)&&(y<0))

 theta=4\*PI-theta;

 c.realp = temp\*Math.cos(theta/3.0);

 c.imagp = temp\*Math.sin(theta/3.0);

 } /\* cubic\_root \*/

 static int solve\_cubic(double a1, double a2,

 double a3, double a4,

 ComplexNumber cn1,

 ComplexNumber cn2,

 ComplexNumber cn3)

 {

 double p, q;

 double a, b, c;

 double u1, u2, wr,wi;

 double x, t, ur, ui, vr, vi;

 double b2, c2;

 double i1, i2;

 double abc[] = new double [3];

 ComplexNumber u = new ComplexNumber();

 ComplexNumber v = new ComplexNumber();

 int n;

 if ( a1 == 0.0)

 {

 n = solve\_quadratic(a2, a3, a4, cn1, cn2);

 return n;

 } /\* if \*/

 a = a2/a1;

 b = a3/a1;

 c = a4/a1;

 p = b - a\*a/3.0;

 q = c + (2.0\*a\*a\*a - 9.0\*a\*b)/27.0 ;

 if ( (Math.abs(p) < EPS ) && (Math.abs(q) < EPS))

 {

 cn1.realp = -a/3;

 cn2.realp = -a/3;

 cn3.realp = -a/3;

 cn1.imagp = 0;

 cn2.imagp = 0;

 cn3.imagp = 0;

 return 3;

 } /\* if \*/

 if (Math.abs(p) < EPS) /\* q != 0 \*/

 {

 cubic\_root(-q,0, u);

 x = u.realp - (a/3.0);

 cn1.realp = x;

 cn1.imagp = 0;

 abc[0] = 1.0;

 abc[1] = a;

 abc[2] = b;

 divide\_cubic(abc, x);

 n = solve\_quadratic(1.0, abc[1], abc[2], cn2, cn3);

 return n+1;

 } /\* if \*/

 solve\_quadratic(1.0, -q, -((p\*p\*p)/27.0), u, v);

 if (u.realp > v.realp)

 {

 wr = u.realp;

 wi = u.imagp;

 } /\* if \*/

 else

 {

 wr = v.realp;

 wi = v.imagp;

 } /\* else \*/

 cubic\_root(wr, wi, u);

 cubic\_root(-q + wr, wi, v);

 t = v.realp - u.realp;

 x = t - (a/3.0);

 cn1.realp = x;

 cn1.imagp = 0;

 abc[0] = 1.0;

 abc[1] = a;

 abc[2] = b;

 divide\_cubic(abc, x);

 n = solve\_quadratic(1.0, abc[1], abc[2], cn2, cn3);

 return n+1;

 } /\* solve\_cubic \*/

 static public void main(String args[])

 {

 double r,i;

 double a, b, c, d;

 ComplexNumber c1 = new ComplexNumber();

 ComplexNumber c2 = new ComplexNumber();

 ComplexNumber c3 = new ComplexNumber();

 int no\_of\_solutions;

 Scanner in = new Scanner(System.in);

 System.out.println("Enter 4 coefs, first must be =/= 0:\n");

 a = in.nextDouble();

 b = in.nextDouble();

 c = in.nextDouble();

 d = in.nextDouble();

 no\_of\_solutions =

 solve\_cubic(a, b, c, d, c1, c2, c3);

 if (no\_of\_solutions == 1)

 {

 System.out.println("Single real solution is: " + c1.realp);

 System.out.print("Complex solution: " + c2.realp);

 if ( c2.imagp < 0)

 System.out.println(" " + c2.imagp + "i");

 else

 System.out.println(" + " + c2.imagp + "i" );

 System.out.print("Complex solution: " + c3.realp);

 if ( c3.imagp < 0)

 System.out.println(" " + c3.imagp + "i");

 else

 System.out.println(" + " + c3.imagp + "i" );

 } /\* if \*/

 else /\* 3 real solutions \*/

 System.out.println("Real Solutions:" + c1.realp + " " +

 c2.realp + " " + c3.realp);

 } /\* main \*/

} // cubic\_eq2

פלט ריצה:

Enter 4 coefs, first must be =/= 0:

1 3 -3 1

Single real solution is: -3.8473221018630728

Complex solution: 0.4236610509315364 -0.2836060010268824i

Complex solution: 0.4236610509315364 + 0.2836060010268824i