KRZYSZTOF SOKÓŁ-SZOŁTYSEK
PROGRAM 3

Program oblicza dwie największe wartości własne dla zadanej w poleceniu macierzy używając metody potęgowej. Poprawne wyniki to 4 i 3, jednak program uzyskuje zadowalającą dokładność(zmienne typu double)

PRZYKŁADOWY INPUT

- brak, macierz z zadania w kodzie

OUTPUT

```
wspolczynnik normalizacji1.6069428333146204E60
Wektor własny
1.0
1.0
1.0
1.0
1.0
1.0
Lambda : 4.0000001192092896
nowa iteracja

wspolczynnik normalizacji1.0307581133579662E49
Wektor własny
-0.9999999999999998
-1.2593393132797063E-16
-2.5186786265594126E-16
-2.5186786265594126E-16
-1.2593393132797063E-16
1.0
Lambda : 3.0000000894069627
```

LISTING

```
import java.util.Arrays;

public class Main {

    public static void main(String[] args) {
```

```
 double[][] A={{19, 13, 10, 10, 13, -17},
                {13, 13, 10, 10, -11, 13},
                {10, 10, 10, -2, 10, 10},
                {10, 10, -2, 10, 10, 10},
                {13, -11, 10, 10, 13, 13},
                {-17, 13, 10, 10, 13, 19}};

 double[]v0={1,1,1,1,1,1};
 double b =1/12f;
// System.out.println(+b);
 for(int i=0;i<6;i++)
      for(int j=0;j<6;j++)
      {
           A[i][j]=A[i][j]*b;
          // System.out.println(+A[i][j]);
      }
 //System.out.println(A.length);
 PM returnValues = new PM();
 double reduction=returnValues.start(A,v0);
 //zmodyfikowac A
// System.out.println();
// System.out.println();
// System.out.println("red0"+reduction);
 for(int i=0;i<6;i++)
      for(int j=0;j<6;j++)
      {
           if (i==5)
           {
                A[i][j]=A[i][j]+4*reduction;
           }
           else {
                A[i][j] = A[i][j] - reduction;

           }
           //System.out.println(+A[i][j]);
      }
 System.out.println("nowa iteracja");
 PM returnSecond= new PM();
 returnSecond.start(A,v0);
 }
```

```
}



class PM {


public double start(double mat[][], double v0[]){

      //System.out.println(+mat[0][0]);
      //System.out.println();
      //pomnozyc macierz przez v kilkanascie razy
      double[] temp = new double[6];
      for(int w=0;w<100;w++) {
            for (int i = 0; i < 6; i++)
                  for (int j = 0; j < 6; j++) {
                              temp[i] += mat[i][j] * v0[j];
                  }

            v0=temp.clone();
            for(int k=0;k<v0.length;k++)
            {
                // System.out.print(" "+v0[k] );

            }
           // System.out.println();
            Arrays.fill(temp,0);
      }
      System.out.println();
      for(int j=0;j<6;j++)
      {
          // System.out.print(" "+v0[j]);
          // System.out.println();
      }
     // double a=Math.round(v0[5]);

    /*   for(int i=0;i<6;i++)
            //for(int j=0;j<6;j++)
            {
                  System.out.print(" "+temp[i]);
                  System.out.println();
            }*/
      //podzielic wynik przez dolna wartosc
```

```
   double[] vnorm =new double[6];
   double wn=v0[5];
   //double wn=1/v0[5];
   System.out.println("wspolczynnik normalizacji"+wn);
   System.out.println("Wektor własny");
   for(int i=0;i<6;i++)
   {

         vnorm[i]=v0[i]/v0[5];
         //vnorm[i]=v0[i]*wn;
         System.out.println(+vnorm[i]);
   }
   double[] Ax= new double[6];
   for (int i = 0; i < 6; i++){
         for (int j = 0; j < 6; j++) {
               Ax[i] += mat[i][j] * vnorm[j];


         }
     //   System.out.println(Ax[i]);

   }


//   System.out.println();
  /* double[] Axx= new double[6];
   for (int i = 0; i < 6; i++)
          {
               Axx[i] += Ax[i] * vnorm[i];
               System.out.println(Axx[i]);
         }*/
//   System.out.println();
   double[] xx= new double[6];
   for (int i = 0; i < 6; i++)
   {
         xx[i] += vnorm[i] * vnorm[i];
     //   System.out.println(xx[i]);
   }


   double VAxx=0;
   for (int i = 0; i < 6; i++)
   {
         VAxx += Ax[i] * vnorm[i];
```

```
            // System.out.println(+VAxx);
        }

        double Vxx=0;
        for (int i = 0; i < 6; i++)
        {
             Vxx += vnorm[i] * vnorm[i];
            // System.out.println(+Vxx);
        }
        double Lambda =VAxx/Vxx;
        double WNsq=wn*wn;
       // System.out.println(v0[5]);
        double reduction=Lambda;
       // System.out.println(reduction);
        System.out.println("Lambda : "+Lambda);
/*
        double[] VectorT ={1,1,1,1,1,-4};
        for (int i = 0; i < 6; i++)
        {
              += vnorm[i] * vnorm[i];
             System.out.println(+);
        }*/

    return reduction;
    }}
```