

Struktura e të dhënave

Provimi periodik 2/2, Forma: A

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

1. Shkruani metodën vijuese:

```
/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][0]*m2[0][j]+m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)
```

Për pyetjet 2-5 shqyrtoni klasën vijuese.

```
import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { int numCandidates = 4;
    int[] votes = new int[numCandidates];
    boolean processing = true;
    while ( processing )
    { int v = new Integer(JOptionPane.showInputDialog
      ("Votoni për (0,1,2,3):")).intValue();
      if ( v == -1 )
      { processing = false; }
      else if ( v >= 0 && v < numCandidates )
      { votes[v] = votes[v] + 1; }
      else { JOptionPane.showMessageDialog(null,
        "Gabim në votim: " + v);
      }
    }
    for ( int i = 0; i < numCandidates; i = i + 1 )
    { System.out.println("Kandidati " + i + " ka "
      + votes[i] + " vota"); }
  }
}
```

2. Modifikoni class `VoteCount` ashtu që aplikacioni së pari të kërkojë numrin e kandidatëve në zgjedhjet. Pasi që të është dhënë numri, jepen votat në mënyrën e zakonshme.
3. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të kërkojë emrat e kandidatëve. Pasi që të jenë rradhitur emrat, jepen votat në mënyrën e zakonshme, dhe afishohen rezultatet me emrin dhe votat për secilin kandidat.
4. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të afishojë emrat dhe numrin e votave për secilin kandidat, si dhe emrin e kandidatit fitues. (Në qoftë se ka më tepër kandidatë me numër maksimal votash, atëherë të afishohet emri i njërit nga këta kandidatë.)

5. Çfarë kryhet në klasën vijuese?

```
import javax.swing.*;
import java.awt.*;
public class VoteGraph extends JPanel
{ private int[] degrees = new int[4];
  private Color[] color = {Color.red, Color.blue, Color.yellow, Color.green};
  public VoteGraph(int[] votes)
  { int numCandidates = votes.length;
    if ( numCandidates < 3 )
    { System.out.println("Gabim: numër i pamjaftueshëm kandidatësh");
      System.exit(0);
    }
    else
    { int others = 0;
      for ( int i = 3; i < numCandidates; i = i + 1 )
      { others = others + votes[i]; }
      int total = votes[0] + votes[1] + votes[2] + others;
      for ( int i = 0; i != 3; i = i + 1 )
      { degrees[i] = votes[i] * 360 / total; }
      degrees[3] = others * 360 / total;
    }
    JFrame f = new JFrame();
    f.getContentPane().add(this);
    f.setSize(300, 200);
    f.setTitle("Paraqitja grafike e votave");
    f.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { int sumDegrees = 0;
    for ( int i = 0; i != 4; i = i + 1 )
    { g.setColor(color[i]);
      g.fillArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      g.setColor(Color.black);
      g.drawArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      sumDegrees = sumDegrees + degrees[i];
    }
  }
}
```

6. Krijoni një varg `d` prej 50 numrash `double` të tillë që vlera e `d[i]` të jetë mbetja gjatë pjesëtimit të plotë të gjashtëfishit të numrit `i` me numrin 15.

7. Shkruani metodën vijuese:

```
/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)
```

8. Shkruani një aplikacion, `TestMax`, i cili zbaton metodën nga detyra paraprake për të gjetur elementin maksimal të vargut nga detyra 6.

9. Shkruani metodën vijuese:

```

/** Kthen matricën e transponuar e një matrice të dhënë
 *   (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).
 * @param m - matrica e dhënë
 * @return matrica e transponuar */
public double[] [] transpond(double[] [] m)

```

10. Shkruani metodën vijuese:

```

/** Shumëzon skalarisht dy vektorë (vargje).
 * @param v1 - një varg
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.
 * @return prodhimi skalar i dy vargjeve:
 *   numri real  $v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+...$  */
public double multiply (double[] v1, double[] v2)

```

Çelësi i provimit A

1. Shkruani metodën vijuese:

```
/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][0]*m2[0][j]+m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)
```

Përgjegjja:

```
/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+m1[i][3]*m2[3][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)
{ int rows = m1.length;
  int cols = m2[0].length;
  double[][] answer = new double[rows][cols];
  if ( m1[0].length != m2.length )
  { System.out.println("Gabim: matrica jokompatible"); }
  else
  { for ( int i = 0; i < rows; i++ )
    for ( int j = 0; j < cols; j++ )
    { double sum = 0.0;
      for ( int k = 0; k < m2.length; k++ )
      { sum += m1[i][k] * m2[k][j]; }
      answer[i][j] = sum;
    }
  }
  return answer;
}
```

Për pyetjet 2-5 shqyrtoni klasën vijuese.

```
import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { int numCandidates = 4;
    int[] votes = new int[numCandidates];
    boolean processing = true;
```

```

while ( processing )
{ int v = new Integer(JOptionPane.showInputDialog
    ("Votoni për (0,1,2,3):")).intValue();
  if ( v == -1 )
  { processing = false; }
  else if ( v >= 0 && v < numCandidates )
  { votes[v] = votes[v] + 1; }
  else { JOptionPane.showMessageDialog(null,
    "Gabim në votim: " + v);
    }
}
for ( int i = 0; i < numCandidates; i = i + 1 )
{ System.out.println("Kandidati " + i + " ka "
  + votes[i] + " vota"); }
}
}

```

2. Modifikoni class `VoteCount` ashtu që aplikacioni së pari të kërkojë numrin e kandidatëve në zgjedhjet. Pasi që të është dhënë numri, jepen votat në mënyrën e zakonshme.

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektoralë.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { int numCandidates =
    new Integer(JOptionPane.showInputDialog(
      "Numri i kandidatëve: ")).intValue();
    if ( numCandidates <= 0 )
    { JOptionPane.showMessageDialog(null,
      "Gabim: " + numCandidates);
      }
    else
    { int[] votes = new int[numCandidates];
      // ... sikur më parë
    }
  }
}

```

3. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të kërkojë emrat e kandidatëve. Pasi që të jenë rradhitur emrat, jepen votat në mënyrën e zakonshme, dhe afishohen rezultatet me emrin dhe votat për secilin kandidat.

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektoralë.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { // ... sikur më parë

```

```

    if ( numCandidates <= 0 )
    { // ... sikur më parë
    }
    else
    { String[] names = new String[numCandidates];
      for ( int i = 0; i < numCandidates; i++ )
      { names[i] =
        JOptionPane.showInputDialog("Emri i kandidatit " + i + ": "); }
      int[] votes = new int[numCandidates];
      // ... sikur më parë
      for ( int i = 0; i < numCandidates; i++ )
      { System.out.println("Kandidati " + names[i] + " ka "
        + votes[i] + " vota"); }
    }
  }
}

```

4. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të afishojë emrat dhe numrin e votave për secilin kandidat, si dhe emrin e kandidatit fitues. (Në qoftë se ka më tepër kandidatë me numër maksimal votash, atëherë të afishohet emri i njërit nga këta kandidatë.)

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { // ... sikur më parë
    if ( numCandidates <= 0 )
    { // ... sikur më parë
    }
    else
    { // ... sikur më parë
      int maxVotesIndex = 0;
      for ( int i = 1; i < numCandidates; i++ )
      { if ( votes[i] > votes[maxVotesIndex] )
        { maxVotesIndex = i; }
      }
      System.out.println("Fitues është kandidati "
        + names[maxVotesIndex]);
    }
  }
}

```

5. Çfarë kryhet në klasën vijuese?

```

import javax.swing.*;
import java.awt.*;
public class VoteGraph extends JPanel
{ private int[] degrees = new int[4];
  private Color[] color = {Color.red, Color.blue, Color.yellow, Color.green};
  public VoteGraph(int[] votes)
  { int numCandidates = votes.length;

```

```

    if ( numCandidates < 3 )
    { System.out.println("Gabim: numër i pamjaftueshëm kandidatësh");
      System.exit(0);
    }
    else
    { int others = 0;
      for ( int i = 3; i < numCandidates; i = i + 1 )
      { others = others + votes[i]; }
      int total = votes[0] + votes[1] + votes[2] + others;
      for ( int i = 0; i != 3; i = i + 1 )
      { degrees[i] = votes[i] * 360 / total; }
      degrees[3] = others * 360 / total;
    }
    JFrame f = new JFrame();
    f.getContentPane().add(this);
    f.setSize(300, 200);
    f.setTitle("Paraqitja grafike e votave");
    f.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { int sumDegrees = 0;
    for ( int i = 0; i != 4; i = i + 1 )
    { g.setColor(color[i]);
      g.fillArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      g.setColor(Color.black);
      g.drawArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      sumDegrees = sumDegrees + degrees[i];
    }
  }
}

```

Përgjegjja: Në një kornizë do të afishohet një grafik rrethor („pie chart“) i rezultateve të arritura, në formën vijuese: sipërfaqja e shoqëruar rezultatit të parë ngjyroset në të kuqe, ajo e rezultatit të dytë në të kaltër, e të tretit në të verdhë, kurse pjesa e mbetur në të gjelbërt.

6. Krijoni një varg `d` prej 50 numrash `double` të tillë që vlera e `d[i]` të jetë mbetja gjatë pjesëtimit të plotë të gjashtëfishit të numrit `i` me numrin 15.

Përgjegjja:

```

double[] d = new double[50];
for ( int i = 0; i < d.length; i++ )
{ d[i] = 6 * i % 15; }

```

7. Shkruani metodën vijuese:

```

/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)

```

Përgjegjja:

```

/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)
{ double max = 0.0;
  if ( v.length == 0 )
    { System.out.println("Gabim: vargu bosh"); }
  else
    { max = v[0];
      for ( int i = 1; i != v.length; i++ )

        { // invariantë: max përmban vlerën maksimale të r[0]...r[i-1]
          if ( v[i] > max )
            { max = v[i]; }
        }
    }
  return max;
}

```

8. Shkruani një aplikacion, `TestMax`, i cili zbaton metodën nga detyra paraprake për të gjetur elementin maksimal të vargut nga detyra 6.

Përgjegjja:

```

public class Array
{
  /** Kthen elementin më të madh të vargut të dhënë
   * të numrave thyesorë.
   * @param v - vargu i dhënë
   * @return elementi maksimal (më i madh) */
  public double maxElement(double v)
  { // ... sikur më parë
  }
}

public class TestMax
{
  public static void main(String args)
  {
    double[] d = new double[50];
    for ( int i = 0; i < d.length; i++ )
      { d[i] = 6 * i % 15; }
    Array calc = new Array();
    System.out.println(calc.maxElement(d));
  }
}

```

9. Shkruani metodën vijuese:

```

/** Kthen matricën e transponuar e një matrice të dhënë
 * (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).
 * @param m - matrica e dhënë
 * @return matrica e transponuar */
public double[][] transpond(double[][] m)

```


Përgjegjja:

```
/** Kthen matricën e transponuar e një matrice të dhënë
 * (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).
 * @param m - matrica e dhënë
 * @return matrica e transponuar */
public double[][] transpond(double[][] m)
{ int rows = m.length;
  int cols = m[0].length;
  double[][] answer = new double[cols][rows];
  for ( int i = 0; i < rows; i++ )
    for ( int j = 0; j < cols; j++ )
      { answer[j][i] = m[i][j]; }
  return answer;
}
```

10. Shkruani metodën vijuese:

```
/** Shumëzon skalarisht dy vektorë (vargje).
 * @param v1 - një varg
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.
 * @return prodhimi skalar i dy vargjeve:
 * numri real v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+... */
public double multiply (double[] v1, double[] v2)
```

Përgjegjja:

```
/** Shumëzon skalarisht dy vektorë (vargje)
 * @param v1 - një varg
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.
 * @return prodhimi skalar i dy vargjeve:
 * numri real v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+... */
public double multiply (double[] v1, double[] v2)
{ double answer = 0.0;
  if ( v1.length != v2.length )
    { System.out.println("Gabim: vargje jokompatible"); }
  else
    { for ( int i = 0; i < v1.length; i++ )
      { answer += v1[i] * v2[i]; }
    }
  return answer;
}
```

Struktura e të dhënave

Provimi periodik 2/2, Forma: B

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

1. Krijoni një varg `d` prej 100 numrash `double` të tillë që vlera e `d[i]` të jetë mbetja gjatë pjesëtimit të plotë të gjashtëfishit të numrit `i` me numrin 15.

2. Shkruani metodën vijuese:

```
/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)
```

3. Shkruani një aplikacion, `TestMax`, i cili zbaton metodën nga detyra paraprahe për të gjetur elementin maksimal të vargut nga detyra 1.

Për pyetjet 4-7 shqyrtoni klasën vijuese.

```
import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { int numCandidates = 4;
    int[] votes = new int[numCandidates];
    boolean processing = true;
    while ( processing )
    { int v = new Integer(JOptionPane.showInputDialog
      ("Votoni për (0,1,2,3):")).intValue();
      if ( v == -1 )
      { processing = false; }
      else if ( v >= 0 && v < numCandidates )
      { votes[v] = votes[v] + 1; }
      else { JOptionPane.showMessageDialog(null,
        "Gabim në votim: " + v);
      }
    }
    for ( int i = 0; i < numCandidates; i = i + 1 )
    { System.out.println("Kandidati " + i + " ka "
      + votes[i] + " vota"); }
  }
}
```

4. Modifikoni `class VoteCount` ashtu që aplikacioni së pari të kërkojë numrin e kandidatëve në zgjedhjet. Pasi që të është dhënë numri, jepen votat në mënyrën e zakonshme.

5. Modifikoni zgjidhjen e detyrës paraprahe ashtu që aplikacioni të kërkojë emrat e kandidatëve. Pasi që të jenë rradhitur emrat, jepen votat në mënyrën e zakonshme, dhe afishohen rezultatet me emrin dhe votat për secilin kandidat.

6. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të afishojë emrat dhe numrin e votave për secilin kandidat, si dhe emrin e kandidatit fitues. (Në qoftë se ka më tepër kandidatë me numër maksimal votash, atëherë të afishohet emri i njërit nga këta kandidatë.)
7. Çfarë kryhet në klasën vijuese?

```
import javax.swing.*;
import java.awt.*;
public class VoteGraph extends JPanel
{ private int[] degrees = new int[4];
  private Color[] color = {Color.red, Color.blue, Color.yellow, Color.green};
  public VoteGraph(int[] votes)
  { int numCandidates = votes.length;
    if ( numCandidates < 3 )
    { System.out.println("Gabim: numër i pamjaftueshëm kandidatësh");
      System.exit(0);
    }
    else
    { int others = 0;
      for ( int i = 3; i < numCandidates; i = i + 1 )
      { others = others + votes[i]; }
      int total = votes[0] + votes[1] + votes[2] + others;
      for ( int i = 0; i != 3; i = i + 1 )
      { degrees[i] = votes[i] * 360 / total; }
      degrees[3] = others * 360 / total;
    }
    JFrame f = new JFrame();
    f.getContentPane().add(this);
    f.setSize(300, 200);
    f.setTitle("Paraqitja grafike e votave");
    f.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { int sumDegrees = 0;
    for ( int i = 0; i != 4; i = i + 1 )
    { g.setColor(color[i]);
      g.fillArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      g.setColor(Color.black);
      g.drawArc(20, 20, 100, 100, sumDegrees, degrees[i]);
      sumDegrees = sumDegrees + degrees[i];
    }
  }
}
```

8. Shkruani metodën vijuese:

```
/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][0]*m2[0][j]+m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)
```

9. Shkruani metodën vijuese:

```
/** Shumëzon skalarisht dy vektorë (vargje).  
 * @param v1 - një varg  
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.  
 * @return prodhimi skalar i dy vargjeve:  
 * numri real v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+... */  
public double multiply (double[] v1, double[] v2)
```

10. Shkruani metodën vijuese:

```
/** Kthen matricën e transponuar e një matrice të dhënë  
 * (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).  
 * @param m - matrica e dhënë  
 * @return matrica e transponuar */  
public double[][] transpond(double[][] m)
```

Çelësi i provimit B

1. Krijoni një varg `d` prej 100 numrash `double` të tillë që vlera e `d[i]` të jetë mbetja gjatë pjesëtimit të plotë të gjashtëfishit të numrit `i` me numrin 15.

Përgjegjja:

```
double[] d = new double[100];
for ( int i = 0; i < d.length; i++ )
{ d[i] = 6 * i % 15; }
```

2. Shkruani metodën vijuese:

```
/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)
```

Përgjegjja:

```
/** Kthen elementin më të madh të vargut të dhënë
 * të numrave thyesorë.
 * @param v - vargu i dhënë
 * @return elementi maksimal (më i madh) */
public double maxElement(double[] v)
{ double max = 0.0;
  if ( v.length == 0 )
  { System.out.println("Gabim: vargu bosh"); }
  else
  { max = v[0];
    for ( int i = 1; i != v.length; i++ )

    { // invariantë: max përmban vlerën maksimale të r[0]...r[i-1]
      if ( v[i] > max )
      { max = v[i]; }
    }
  }
  return max;
}
```

3. Shkruani një aplikacion, `TestMax`, i cili zbaton metodën nga detyra paraprake për të gjetur elementin maksimal të vargut nga detyra 1.

Përgjegjja:

```
public class Array
{
  /** Kthen elementin më të madh të vargut të dhënë
   * të numrave thyesorë.
   * @param v - vargu i dhënë
   * @return elementi maksimal (më i madh) */
  public double maxElement(double v)
```

```

    { // ... sikur më parë
    }
}

public class TestMax
{
    public static void main(String args)
    {
        double[] d = new double[100];
        for ( int i = 0; i < d.length; i++ )
            { d[i] = 6 * i % 15; }
        Array calc = new Array();
        System.out.println(calc.maxElement(d));
    }
}

```

Për pyetjet 4-7 shqyrtoni klasën vijuese.

```

import javax.swing.*;
/** Numëron votat për kandidatët elektoralë.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { int numCandidates = 4;
    int[] votes = new int[numCandidates];
    boolean processing = true;
    while ( processing )
    { int v = new Integer(JOptionPane.showInputDialog
      ("Votoni për (0,1,2,3):")).intValue();
      if ( v == -1 )
        { processing = false; }
      else if ( v >= 0 && v < numCandidates )
        { votes[v] = votes[v] + 1; }
      else { JOptionPane.showMessageDialog(null,
        "Gabim në votim: " + v);
      }
    }
    for ( int i = 0; i < numCandidates; i = i + 1 )
    { System.out.println("Kandidati " + i + " ka "
      + votes[i] + " vota"); }
  }
}

```

4. Modifikoni class `VoteCount` ashtu që aplikacioni së pari të kërkojë numrin e kandidatëve në zgjedhjet. Pasi që të është dhënë numri, jepen votat në mënyrën e zakonshme.

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektoralë.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)

```

```

    { int numCandidates =
      new Integer(JOptionPane.showInputDialog(
        "Numri i kandidatëve: ")).intValue();
      if ( numCandidates <= 0 )
        { JOptionPane.showMessageDialog(null,
          "Gabim: " + numCandidates);
        }
      else
        { int[] votes = new int[numCandidates];
          // ... sikur më parë
        }
      }
    }
  }
}

```

5. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të kërkojë emrat e kandidatëve. Pasi që të jenë rradhitur emrat, jepen votat në mënyrën e zakonshme, dhe afishohen rezultatet me emrin dhe votat për secilin kandidat.

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount
{ public static void main(String[] args)
  { // ... sikur më parë
    if ( numCandidates <= 0 )
      { // ... sikur më parë
      }
    else
      { String[] names = new String[numCandidates];
        for ( int i = 0; i < numCandidates; i++ )
          { names[i] =
            JOptionPane.showInputDialog("Emri i kandidatit " + i + ": "); }
        int[] votes = new int[numCandidates];
        // ... sikur më parë
        for ( int i = 0; i < numCandidates; i++ )
          { System.out.println("Kandidati " + names[i] + " ka "
            + votes[i] + " vota"); }
        }
      }
    }
}

```

6. Modifikoni zgjidhjen e detyrës paraprake ashtu që aplikacioni të afishojë emrat dhe numrin e votave për secilin kandidat, si dhe emrin e kandidatit fitues. (Në qoftë se ka më tepër kandidatë me numër maksimal votash, atëherë të afishohet emri i njërit nga këta kandidatë.)

Përgjegjja:

```

import javax.swing.*;
/** Numëron votat për kandidatët elektorale.
 * input: një varg votash, i terminuar nga -1
 * output: lista e rezultateve të votave për kandidatë */
public class VoteCount

```

```

{ public static void main(String[] args)
  { // ... sikur më parë
    if ( numCandidates <= 0 )
      { // ... sikur më parë
        }
      else
        { // ... sikur më parë
          int maxVotesIndex = 0;
          for ( int i = 1; i < numCandidates; i++ )
            { if ( votes[i] > votes[maxVotesIndex] )
              { maxVotesIndex = i; }
            }
          System.out.println("Fitues është kandidati "
            + names[maxVotesIndex]);
        }
      }
}

```

7. Çfarë kryhet në klasën vijuese?

```

import javax.swing.*;
import java.awt.*;
public class VoteGraph extends JPanel
{ private int[] degrees = new int[4];
  private Color[] color = {Color.red, Color.blue, Color.yellow, Color.green};
  public VoteGraph(int[] votes)
  { int numCandidates = votes.length;
    if ( numCandidates < 3 )
      { System.out.println("Gabim: numër i pamjaftueshëm kandidatësh");
        System.exit(0);
      }
    else
      { int others = 0;
        for ( int i = 3; i < numCandidates; i = i + 1 )
          { others = others + votes[i]; }
        int total = votes[0] + votes[1] + votes[2] + others;
        for ( int i = 0; i != 3; i = i + 1 )
          { degrees[i] = votes[i] * 360 / total; }
        degrees[3] = others * 360 / total;
      }
    JFrame f = new JFrame();
    f.getContentPane().add(this);
    f.setSize(300, 200);
    f.setTitle("Paraqitja grafike e votave");
    f.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { int sumDegrees = 0;
    for ( int i = 0; i != 4; i = i + 1 )
      { g.setColor(color[i]);
        g.fillArc(20, 20, 100, 100, sumDegrees, degrees[i]);
        g.setColor(Color.black);
        g.drawArc(20, 20, 100, 100, sumDegrees, degrees[i]);
        sumDegrees = sumDegrees + degrees[i];
      }
  }
}

```



```

    }
  }
}

```

Përgjegjja: Në një kornizë do të afishohet një grafik rrethor („pie chart“) i rezultateve të arritura, në formën vijuese: sipërfaqja e shoqëruar rezultatit të parë ngjyrosët në të kuqe, ajo e rezultatit të dytë në të kaltër, e të tretit në të verdhë, kurse pjesa e mbetur në të gjelbërt.

8. Shkruani metodën vijuese:

```

/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][0]*m2[0][j]+m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)

```

Përgjegjja:

```

/** Shumëzon dy matrica të dhëna.
 * @param m1 - matrica e parë
 * @param m2 - matrica e dytë.
 * Numri i rreshtave të matricës së dytë është i barabartë
 * me numrin e shtyllave të matricës së parë.
 * @return matrica prodhim i matricës m1 me matricën m2:
 * elementi (i,j) i prodhimit është i barabartë me
 * m1[i][1]*m2[1][j]+m1[i][2]*m2[2][j]+m1[i][3]*m2[3][j]+... */
public double[][] multiply(double[][] m1, double[][] m2)
{ int rows = m1.length;
  int cols = m2[0].length;
  double[][] answer = new double[rows][cols];
  if ( m1[0].length != m2.length )
  { System.out.println("Gabim: matrica jokompatibile"); }
  else
  { for ( int i = 0; i < rows; i++ )
    for ( int j = 0; j < cols; j++ )
    { double sum = 0.0;
      for ( int k = 0; k < m2.length; k++ )
        { sum += m1[i][k] * m2[k][j]; }
      answer[i][j] = sum;
    }
  }
  return answer;
}

```

9. Shkruani metodën vijuese:

```

/** Shumëzon skalarisht dy vektorë (vargje).
 * @param v1 - një varg
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.
 * @return prodhimi skalar i dy vargjeve:
 * numri real v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+... */
public double multiply (double[] v1, double[] v2)

```

Përgjegjja:

```
/** Shumëzon skalarisht dy vektorë (vargje)
 * @param v1 - një varg
 * @param v2 - një varg. Gjatësia e vargjeve duhet të jetë e njëjtë.
 * @return prodhimi skalar i dy vargjeve:
 * numri real v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2]+... */
public double multiply (double[] v1, double[] v2)
{ double answer = 0.0;
  if ( v1.length != v2.length )
    { System.out.println("Gabim: vargje jokompatible"); }
  else
    { for ( int i = 0; i < v1.length; i++ )
      { answer += v1[i] * v2[i]; }
    }
  return answer;
}
```

10. Shkruani metodën vijuese:

```
/** Kthen matricën e transponuar e një matrice të dhënë
 * (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).
 * @param m - matrica e dhënë
 * @return matrica e transponuar */
public double[][] transpond(double[][] m)
```

Përgjegjja:

```
/** Kthen matricën e transponuar e një matrice të dhënë
 * (d.m.th. matricën e cila për rreshta ka shtyllat e matricës së dhënë).
 * @param m - matrica e dhënë
 * @return matrica e transponuar */
public double[][] transpond(double[][] m)
{ int rows = m.length;
  int cols = m[0].length;
  double[][] answer = new double[cols][rows];
  for ( int i = 0; i < rows; i++ )
    for ( int j = 0; j < cols; j++ )
      { answer[j][i] = m[i][j]; }
  return answer;
}
```