

Struktura e të dhënave

Provimi periodik 1/2, Forma: **A**

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

1. Çfarë afishon aplikacioni vijues?

```
import javax.swing.*;
import java.awt.*;
public class RecursiveSquareWriter extends JPanel
{ private double scale = 0.8;
  private int width = 300;
  private int height = 300;
  private int squareSize = 200;

  public RecursiveSquareWriter()
  { JFrame frame = new JFrame();
    frame.getContentPane().add(this);
    frame.setTitle("Vizatuesi rekursiv i katrorëve");
    frame.setSize(width, height);
    frame.setBackground(Color.white);
    frame.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { paintSquare(width, height, squareSize, g); }

  private void paintSquare(int right, int bottom,
                          int size, Graphics g)
  { int backgroundSize = (int)(size * scale);
    if ( backgroundSize > 0 )
    { paintSquare((int)(right * scale), (int)(bottom * scale),
                  backgroundSize, g);
    }
    g.setColor(Color.black);
    g.drawRect(0, 0, right, bottom);
  }

  public static void main(String[] args)
  { new RecursiveSquareWriter(); }
}
```

2. Shkruani një aplikacion i cili lejon shfrytëzuesin të llogarisë gjatësitë e sado stringjesh joboshe të cilat shfrytëzuesi i jep në mënyrë interaktive. Programi ndalon kur të merrë string bosh për hyrje.

Për pyetjet 3–6 shqyrtoni metodën vijuese.

```
public int mean(int low, int up)
{ while ( low < up )
  { low++;
    up--;
  }
  return low ;
}
```

3. Shkruani një aplikacion, `TestMean`, i cili përmban metodën `mean`, kurse metoda inicuese `main` e tij kënaq specifikacionin vijues:

```
/** Teston metodën mean */
public class TestMean {
    public int mean(int low, int up)
    { // ... njësoj sikur më parë
    }
    /** Invokon
     * mean(1,1), mean(1,2), ..., mean(1, 199 )
     * dhe afishon rezultatet.
     * @param args - argumentët programorë */
    public static void main(String args)
    }
}
```

4. Çfarë rezultati kthen invokimi `mean(7, 11)`?
5. Shkruani një gjurmë ekzekutimi për invokimin `mean(7, 10)`.
6. Çfarë rezultati kthen invokimi `mean(100, 201)`?
7. Duke shfrytëzuar formulën rekursive

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad (k = 1, 2, \dots; n = k + 1, k + 2, \dots)$$

për llogaritjen e kombinacioneve të klasës k nga n elemente (d.m.th. numrit të nënbashkësive k elementëshe të bashkësisë n elementëshe), shkruani metodën rekursive

```
/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)
```

8. Shpjegoni çfarë bën metoda vijuese `main`?

```
public static void main(String[] args)
{ double total = 0.0;
  boolean processing = true;
  while ( processing )
  { int num = new Integer(JOptionPane.showInputDialog
    ("Jepni një numër të plotë jonegativ:")).intValue();
    if ( num < 0 )
    { processing = false; }
    else { total += Math.sqrt(num); }
  }
  JOptionPane.showMessageDialog(null, "Shuma: " + total);
}
```

Për pyetjet 9–10 shqyrtoni metodën vijuese.

```
public int findRightmostChar(char c, String s)
{ boolean found = false;
  int index = s.length() - 1;
  while ( !found && index >= 0 )
  { if (s.charAt(index) == c)
```

```

    { found = true; }
    else { index--; }
}
if ( !found ) { index = -1; }
return index;
}

```

9. Çfarë afishon aplikacioni

```

public class SearchString
{ public int findRightmostChar(char c, String s)
  { // ... njësoj sikur më parë
  }

  public static void main (String args)
  { SearchString searcher = new SearchString();
    String str = "Tungjatjeta!";
    System.out.println(searcher.findRightmostChar('j' , str));
    System.out.println(searcher.findRightmostChar('x', str));
  }
}

```

10. Modifikoni metodën `findRightmostChar` ashtu që të alokojë paraqitjen e parë nga e majta të karakterit në string.

Çelësi i provimit A

1. Çfarë afishon aplikacioni vijues?

```
import javax.swing.*;
import java.awt.*;
public class RecursiveSquareWriter extends JPanel
{ private double scale = 0.8;
  private int width = 300;
  private int height = 300;
  private int squareSize = 200;

  public RecursiveSquareWriter()
  { JFrame frame = new JFrame();
    frame.getContentPane().add(this);
    frame.setTitle("Vizatuesi rekursiv i katrorëve");
    frame.setSize(width, height);
    frame.setBackground(Color.white);
    frame.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { paintSquare(width, height, squareSize, g); }

  private void paintSquare(int right, int bottom,
                          int size, Graphics g)
  { int backgroundSize = (int)(size * scale);
    if ( backgroundSize > 0 )
    { paintSquare((int)(right * scale), (int)(bottom * scale),
                  backgroundSize, g);
    }
    g.setColor(Color.black);
    g.drawRect(0, 0, right, bottom);
  }

  public static void main(String[] args)
  { new RecursiveSquareWriter(); }
}
```

Përgjegjja: Afishon një katror me brinjë 200 piksel, brenda të cilit ndodhet një katror tjetër i madhësisë 0.8 madhësi të të parit, brenda të cilit ndodhet një katror tjetër i madhësisë 0.8 madhësi të të dytit, ..., derisa katrorët të mos tkurren në madhësinë 0.

2. Shkruani një aplikacion i cili lejon shfrytëzuesin të llogarisë gjatësitë e sado stringjesh joboshe të cilat shfrytëzuesi i jep në mënyrë interaktive. Programi ndalon kur të merrë string bosh për hyrje.

Përgjegjja:

```
import javax.swing.*;
public class StringLength
{ public static void main(String[] args)
  { boolean processing = true;
    while ( processing )
```

```

        { String input =
            JOptionPane.showInputDialog("Jepni një string jobosh:");
            if ( input.equals("") )
                { processing = false; }
            else
                { JOptionPane.showMessageDialog
                    (null, "Gjatësia e stringut: " + input.length());
                }
            }
        }
    }
}

```

Për pyetjet 3–6 shqyrtoni metodën vijuese.

```

public int mean(int low, int up)
{ while ( low < up )
  { low++;
    up--;
  }
return low ;
}

```

3. Shkruani një aplikacion, `TestMean`, i cili përmban metodën `mean`, kurse metoda inicuese `main` e tij kënaq specifikacionin vijues:

```

/** Teston metodën mean */
public class TestMean {
    public int mean(int low, int up)
    { // ... njësoj sikur më parë
    }
    /** Invokon
     * mean(1,1), mean(1,2), ..., mean(1, 199 )
     * dhe afishon rezultatet.
     * @param args - argumentët programorë */
    public static void main(String args)
    }
}

```

Përgjegjja:

```

/** Teston metodën mean */
public class TestMean {
    public int mean(int low, int up)
    { // ... njësoj sikur më parë
    }
    /** Invokon
     * mean(1,1), mean(1,2), ..., mean(1, 199 )
     * dhe afishon rezultatet.
     * @param args - argumentët programorë */
    public static void main(String args)
    { TestMean tm = new TestMean();
      for ( int i = 1; i < 100 ; i++ )
        { System.out.println(tm.mean(1, i)); }
    }
}

```

4. Çfarë rezultati kthen invokimi `mean(7, 11)`?

Përgjegjja: 9.

5. Shkruani një gjurmë ekzekutimi për invokimin `mean(7, 10)`.

Përgjegjja:

```
int low == 7
int up == 10
> while ( low < up )
{ low++;
  up--;
}
```

```
int low == 7
int up == 10
while ( true )
{ > low++;
  up--;
}
```

```
int low == 8
int up == 9
while ( true )
{ low++;
  up--;
> }
```

```
int low == 8
int up == 9
> while ( low < up )
{ low++;
  up--;
}
```

```
int low == 9
int up == 8
while ( true )
{ low++;
  up--;
> }
```

```
int low == 9
int up == 8
> while ( low < up )
{ low++;
  up--;
}
```

```

int low == 9
int up == 8
while ( false )
{ low++;
  up--;
}
return low ;

```

6. Çfarë rezultati kthen invokimi `mean(100, 201)`?

Përgjegjja: 151

7. Duke shfrytëzuar formulën rekursive

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad (k = 1, 2, \dots; n = k + 1, k + 2, \dots)$$

për llogaritjen e kombinacioneve të klasës k nga n elemente (d.m.th. numrit të nënbashkësive k elementëshe të bashkësisë n elementëshe), shkruani metodën rekursive

```

/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)

```

Përgjegjja:

```

/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)
{ long answer = -1;
  if ( k >= 0 && n >= k )
  { if ( k == 0 || n == k )
    { answer = 1; }
    else
    { answer = binomial(n - 1, k) + binomial(n - 1, k - 1); }
  }
  return answer;
}

```

8. Shpjegoni çfarë bën metoda vijuese `main`?

```

public static void main(String[] args)
{ double total = 0.0;
  boolean processing = true;
  while ( processing )
  { int num = new Integer(JOptionPane.showInputDialog
    ("Jepni një numër të plotë jonegativ:")).intValue();
    if ( num < 0 )
    { processing = false; }
    else { total += Math.sqrt(num); }
  }
  JOptionPane.showMessageDialog(null, "Shuma: " + total);
}

```

Përgjegjja: Llogarit shumën e rrënjëve katrore të sado numrave pozitivë, të cilat shfrytëzuesi i jep në mënyrë interaktive. Aplikacioni ndalon kur të merrë numër negativ për hyrje.

Për pyetjet 9–10 shqyrtoni metodën vijuese.

```
public int findRightmostChar(char c, String s)
{ boolean found = false;
  int index = s.length() - 1;
  while ( !found && index >= 0 )
  { if (s.charAt(index) == c)
    { found = true; }
    else { index--; }
  }
  if ( !found ) { index = -1; }
  return index;
}
```

9. Çfarë afishon aplikacioni

```
public class SearchString
{ public int findRightmostChar(char c, String s)
  { // ... njësoj sikur më parë
  }

  public static void main (String args)
  { SearchString searcher = new SearchString();
    String str = "Tungjatjeta!";
    System.out.println(searcher.findRightmostChar('j' , str));
    System.out.println(searcher.findRightmostChar('x', str));
  }
}
```

Përgjegjja: 7

-1

10. Modifikoni metodën `findRightmostChar` ashtu që të alokojë paraqitjen e parë nga e majta të karakterit në string.

Përgjegjja:

```
/** Alokon paraqitjen e parë nga e majta të karakterit në string
 * @param c - karakteri për t'u gjetur
 * @param s - stringu për t'u kërkuar
 * @return indeksi i paraqitjes;
 * -1 në qoftë se c nuk paraqitet në s */
public int findChar(char c, String s)
{ boolean found = false;
  int index = 0;
  while ( !found && index < s.length() )
  { // për found == false c nuk është asnjë nga karakterët 0...(index-1)
    // për found == true c është s.charAt(index)
    if (s.charAt(index) == c)
    { found = true; }
    else { index = index + 1; }
  }
}
```



```
if ( !found ) { index = -1; }  
return index;  
}
```

Struktura e të dhënave

Provimi periodik 1/2, Forma: B

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

Për pyetjet 1–4 shqyrtoni metodën vijuese.

```
public int mean(int low, int up)
{ while ( low < up )
  { low++;
    up--;
  }
  return up ;
}
```

1. Shkruani një aplikacion, `TestMean`, i cili përmban metodën `mean`, kurse metoda inicuese `main` e tij kënaq specifikacionin vijues:

```
/** Teston metodën mean */
public class TestMean {
  public int mean(int low, int up)
  { // ... njësoj sikur më parë
  }
  /** Invokon
   * mean(1,1), mean(1,2), ..., mean(1, 199 )
   * dhe afishon rezultatet.
   * @param args - argumentët programorë */
  public static void main(String args)
  }
}
```

2. Çfarë rezultati kthen invokimi `mean(17, 21)`?
3. Shkruani një gjurmë ekzekutimi për invokimin `mean(7, 10)`.
4. Çfarë rezultati kthen invokimi `mean(200, 301)`?
5. Çfarë afishon aplikacioni vijues?

```
import javax.swing.*;
import java.awt.*;
public class RecursiveSquareWriter extends JPanel
{ private double scale = 0.8;
  private int width = 300;
  private int height = 300;
  private int squareSize = 200;

  public RecursiveSquareWriter()
  { JFrame frame = new JFrame();
    frame.getContentPane().add(this);
    frame.setTitle("Vizatuesi rekursiv i katrorëve");
    frame.setSize(width, height);
    frame.setBackground(Color.white);
    frame.setVisible(true);
  }
}
```

```

public void paintComponent(Graphics g)
{ paintSquare(width, height, squareSize, g); }

private void paintSquare(int right, int bottom,
                        int size, Graphics g)
{ int backgroundSize = (int)(size * scale);
  if ( backgroundSize > 0 )
  { paintSquare((int)(right * scale), (int)(bottom * scale),
                backgroundSize, g);
  }
  g.setColor(Color.black);
  g.drawRect(0, 0, right, bottom);
}

public static void main(String[] args)
{ new RecursiveSquareWriter(); }
}

```

Për pyetjet 6–7 shqyrtoni metodën vijuese.

```

public int findRightmostChar(char c, String s)
{ boolean found = false;
  int index = s.length() - 1;
  while ( !found && index >= 0 )
  { if (s.charAt(index) == c)
    { found = true; }
    else { index--; }
  }
  if ( !found ) { index = -1; }
  return index;
}

```

6. Çfarë afishon aplikacioni

```

public class SearchString
{ public int findRightmostChar(char c, String s)
  { // ... njësoj sikur më parë
  }

  public static void main (String args)
  { SearchString searcher = new SearchString();
    String str = "Tung! Tung!";
    System.out.println(searcher.findRightmostChar('g' , str));
    System.out.println(searcher.findRightmostChar('x', str));
  }
}

```

7. Modifikoni metodën `findRightmostChar` ashtu që të alokojë paraqitjen e parë nga e majta të karakterit në string.
8. Shkruani një aplikacion i cili lejon shfrytëzuesin të llogarisë gjatësitë e sado stringjesh joboshe të cilat shfrytëzuesi i jep në mënyrë interaktive. Programi ndalon kur të merrë string bosh për hyrje.
9. Shpjegoni çfarë bën metoda vijuese `main`?

```

public static void main(String[] args)
{ double total = 0.0;
  boolean processing = true;
  while ( processing )
  { int num = new Integer(JOptionPane.showInputDialog
    ("Jepni një numër të plotë jonegativ:")).intValue();
    if ( num < 0 )
    { processing = false; }
    else { total += Math.sqrt(num); }
  }
  JOptionPane.showMessageDialog(null, "Shuma: " + total);
}

```

10. Duke shfrytëzuar formulën rekursive

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad (k = 1, 2, \dots; n = k+1, k+2, \dots)$$

për llogaritjen e kombinacioneve të klasës k nga n elemente (d.m.th. numrit të nënbashkësive k elementëshe të bashkësisë n elementëshe), shkruani metodën rekursive

```

/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)

```

Çelësi i provimit B

Për pyetjet 1–4 shqyrtoni metodën vijuese.

```
public int mean(int low, int up)
{ while ( low < up )
  { low++;
    up--;
  }
  return up ;
}
```

1. Shkruani një aplikacion, `TestMean`, i cili përmban metodën `mean`, kurse metoda inicuese `main` e tij kënaq specifikacionin vijues:

```
/** Teston metodën mean */
public class TestMean {
  public int mean(int low, int up)
  { // ... njësoj sikur më parë
  }
  /** Invokon
   * mean(1,1), mean(1,2), ..., mean(1, 199 )
   * dhe afishon rezultatet.
   * @param args - argumentët programorë */
  public static void main(String args)
  }
```

Përgjegjja:

```
/** Teston metodën mean */
public class TestMean {
  public int mean(int low, int up)
  { // ... njësoj sikur më parë
  }
  /** Invokon
   * mean(1,1), mean(1,2), ..., mean(1, 199 )
   * dhe afishon rezultatet.
   * @param args - argumentët programorë */
  public static void main(String args)
  { TestMean tm = new TestMean();
    for ( int i = 1; i < 200 ; i++ )
    { System.out.println(tm.mean(1, i)); }
  }
}
```

2. Çfarë rezultati kthen invokimi `mean(17, 21)`?

Përgjegjja: 19.

3. Shkruani një gjurmë ekzekutimi për invokimin `mean(7, 10)`.

Përgjegjja:

```
int low == 7
int up == 10
> while ( low < up )
{ low++;
  up--;
}
```

```
int low == 7
int up == 10
while ( true )
{ > low++;
  up--;
}
```

```
int low == 8
int up == 9
while ( true )
{ low++;
  up--;
}>
```

```
int low == 8
int up == 9
> while ( low < up )
{ low++;
  up--;
}
```

```
int low == 9
int up == 8
while ( true )
{ low++;
  up--;
}>
```

```
int low == 9
int up == 8
> while ( low < up )
{ low++;
  up--;
}
```

```
int low == 9
int up == 8
while ( false )
{ low++;
  up--;
}> return up ;
```

4. Çfarë rezultati kthen invokimi `mean(200, 301)`?

Përgjegjja: 250

5. Çfarë afishon aplikacioni vijues?

```
import javax.swing.*;
import java.awt.*;
public class RecursiveSquareWriter extends JPanel
{ private double scale = 0.8;
  private int width = 300;
  private int height = 300;
  private int squareSize = 200;

  public RecursiveSquareWriter()
  { JFrame frame = new JFrame();
    frame.getContentPane().add(this);
    frame.setTitle("Vizatuesi rekursiv i katrorëve");
    frame.setSize(width, height);
    frame.setBackground(Color.white);
    frame.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { paintSquare(width, height, squareSize, g); }

  private void paintSquare(int right, int bottom,
                           int size, Graphics g)
  { int backgroundSize = (int)(size * scale);
    if ( backgroundSize > 0 )
    { paintSquare((int)(right * scale), (int)(bottom * scale),
                  backgroundSize, g);
    }
    g.setColor(Color.black);
    g.drawRect(0, 0, right, bottom);
  }

  public static void main(String[] args)
  { new RecursiveSquareWriter(); }
}
```

Përgjegjja: Afishon një katror me brinjë 200 piksel, brenda të cilit ndodhet një katror tjetër i madhësisë 0.8 madhësi të të parit, brenda të cilit ndodhet një katror tjetër i madhësisë 0.8 madhësi të të dytit, ..., derisa katrorët të mos tkurren në madhësinë 0.

Për pyetjet 6–7 shqyrtoni metodën vijuese.

```
public int findRightmostChar(char c, String s)
{ boolean found = false;
  int index = s.length() - 1;
  while ( !found && index >= 0 )
  { if (s.charAt(index) == c)
    { found = true; }
    else { index--; }
  }
  if ( !found ) { index = -1; }
  return index;
}
```

6. Çfarë afishon aplikacioni

```

public class SearchString
{ public int findRightmostChar(char c, String s)
  { // ... njësoj sikur më parë
  }

  public static void main (String args)
  { SearchString searcher = new SearchString();
    String str = "Tung! Tung!" ;
    System.out.println(searcher.findRightmostChar('g' , str));
    System.out.println(searcher.findRightmostChar('x', str));
  }
}

```

Përgjegjja: 9

-1

7. Modifikoni metodën findRightmostChar ashtu që të alokojë paraqitjen e parë nga e majta të karakterit në string.

Përgjegjja:

```

/** Alokon paraqitjen e parë nga e majta të karakterit në string
 * @param c - karakteri për t'u gjetur
 * @param s - stringu për t'u kërkuar
 * @return indeksi i paraqitjes;
 * -1 në qoftë se c nuk paraqitet në s */
public int findChar(char c, String s)
{ boolean found = false;
  int index = 0;
  while ( !found && index < s.length() )
  { // për found == false c nuk është asnjë nga karakterët 0...(index-1)
    // për found == true c është s.charAt(index)
    if (s.charAt(index) == c)
    { found = true; }
    else { index = index + 1; }
  }
  if ( !found ) { index = -1; }
  return index;
}

```

8. Shkruani një aplikacion i cili lejon shfrytëzuesin të llogarisë gjatësitë e sado stringjesh joboshe të cilat shfrytëzuesi i jep në mënyrë interaktive. Programi ndalon kur të merrë string bosh për hyrje.

Përgjegjja:

```

import javax.swing.*;
public class StringLength
{ public static void main(String[] args)
  { boolean processing = true;
    while ( processing )
    { String input =
      JOptionPane.showInputDialog("Jepni një string jobosh:");
      if ( input.equals("") )
      { processing = false; }
      else

```



```

        { JOptionPane.showMessageDialog
          (null, "Gjatësia e stringut: " + input.length());
        }
      }
    }
  }
}

```

9. Shpjegoni çfarë bën metoda vijuese main?

```

public static void main(String[] args)
{ double total = 0.0;
  boolean processing = true;
  while ( processing )
  { int num = new Integer(JOptionPane.showInputDialog
    ("Jepni një numër të plotë jonegativ:")).intValue();
    if ( num < 0 )
    { processing = false; }
    else { total += Math.sqrt(num); }
  }
  JOptionPane.showMessageDialog(null, "Shuma: " + total);
}

```

Përgjegjja: Llogarit shumën e rrënjëve katrore të sado numrave pozitivë, të cilat shfrytëzuesi i jep në mënyrë interaktive. Aplikacioni ndalon kur të merrë numër negativ për hyrje.

10. Duke shfrytëzuar formulën rekursive

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad (k = 1, 2, \dots; n = k+1, k+2, \dots)$$

për llogaritjen e kombinacioneve të klasës k nga n elemente (d.m.th. numrit të nënbashkësive k elementëshe të bashkësisë n elementëshe), shkruani metodën rekursive

```

/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)

```

Përgjegjja:

```

/** Llogarit kombinacionet e klases se dhene me numer te dhene elementesh.
 * @param n - numri i elementeve
 * @param k - klasa
 * @return numri i kombinacioneve */
public long binomial(int n, int k)
{ long answer = -1;
  if ( k >= 0 && n >= k )
  { if ( k == 0 || n == k )
    { answer = 1; }
    else
    { answer = binomial(n - 1, k) + binomial(n - 1, k - 1); }
  }
  return answer;
}

```