

Struktura e të dhënave

Provimi përfundimtar, Forma: A

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

1. Krijoni një matricë numrash realë me 12 rreshta dhe 12 shtylla dhe inicializojeni ashtu që elementi në reshtin i dhe shtyllën j të përmbajë herësin $\frac{i}{j}$ ($i, j = 1, 2, \dots, 12$). Afishoni pastaj tabelën e pjesëtimit deri në 10 ashtu që herësat të paraqiten me dy shifra pas pikës decimale. (**Udhëzim:** Keni kujdes në operacionin e pjesëtimit me operandë numra të plotë.)
2. Shpjegoni çfarë bën metoda vijuese `main`?

```
public static void main(String[] args)
{
    boolean processing = true;
    while ( processing )
    { int num = new Integer(JOptionPane.showInputDialog
        ("Jepni një numër të plotë (zero për fund):"));intValue();
      if ( num == 0 )
        { processing = false; }
      else
        { JOptionPane.showMessageDialog
            (null, "Vlera reciproke: " + 1.0 / num); }
    }
}
```

3. Shkruani metodën vijuese:

```
/** Kthen elementin më të vogël të vargut të dhënë.
 * @param v - vargu i dhënë
 * @return elementi minimal (më i vogël) */
public int minElement(int[] v)
```

4. Krijoni një aplikacion i cili së pari kërkon nga shfrytëzuesi një vlerë të plotë pozitive n , pastaj krijon një varg prej n numrash të plotë, dhe kërkon nga shfrytëzuesi futjen interaktive të të gjitha vlerave të elementeve të vargut. Më në fund, aplikacioni llogarit dhe afishon numrin e elementeve pozitive të vargut. (P.sh., në qoftë se shfrytëzuesi ka futur vlerën 4 për n dhe pastaj vlerat vijuese për katër elementet: 2, -1, 0, 5, aplikacioni duhet të afishojë se numri i elementeve pozitive të vargut është 2.)

Për pyetjet 5–7 shqyrtoni metodën vijuese.

```
public long factorial(int n)
{ long answer = 0;
  if ( n < 0 || n > 20 )
    { answer = -1; }
  else
    { int i = 1;
      while ( i < n)
        { i++;
          answer *= i;
        }
    }
  return answer;
}
```

5. Çfarë rezultati kthen invokimi `factorial(5)`?
6. Modifikoni metodën `factorial` ashtu që të kthejë faktorialin e një numri të plotë nga intervali 0 deri 20, të dhënë si argument.
7. Shfrytëzoni metodën `factorial` nga detyra paraprake për të implementuar metodën vijuese.

```

/** Llogarit kosinusin e argumentit sipas formulës së Taylor-it:
 *   cos(x)=1-(x^2/2!)+(x^4/4!)-(x^6/6!)+...-(x^20/20!)
 * @param x - vlera, në radian, e argumentit
 * @return vlera e kosinuset e llogaritur sipas formulës */
public double cos(double x)

```

(Udhëzim: Shfrytëzoni `Math.pow(a,b)` për të llogaritur a^b .)

8. Shkruani metodën vijuese për llogaritjen e shumës së vlerave reciproke:

```

/** Llogarit shumën 1/1+1/2+...+1/n
 * @param n - kufiri i sipërm; pozitiv
 * @return shuma për n pozitiv;
 *   -1 për n jopozitiv */
public double sumReciprocals(int n)

```

9. Çfarë do të afishojë urdhëri `while` vijues?

```

int i = 10;
while ( i > 0 )
{ System.out.println(i);
  i = i - 3;
}

```

10. Krijoni një varg prej 50 numrash `double` të tillë që vlera e `d[i]` të jetë i^2 .

Struktura e të dhënave

Provimi përfundimtar, Forma: B

Emri: _____

Nr. indeksit: _____

Drejtimi: _____

Data: _____

1. Shpjegoni çfarë bën metoda vijuese main?

```
public static void main(String[] args)
{
    boolean processing = true;
    while ( processing )
    { int num = new Integer(JOptionPane.showInputDialog
        ("Jepni një numër të plotë (zero për fund):"));
      if ( num == 0 )
      { processing = false; }
      else
      { JOptionPane.showMessageDialog
          (null, "Vlera reciproke: " + 1.0 / num); }
    }
}
```

Për pyetjet 2-4 shqyrtoni metodën vijuese.

```
public long factorial(int n)
{ long answer = 0;
  if ( n < 0 || n > 20 )
  { answer = -1; }
  else
  { int i = 1;
    while ( i < n)
    { i++;
      answer *= i;
    }
  }
  return answer;
}
```

2. Çfarë rezultati kthen invokimi factorial(7)?
3. Modifikoni metodën factorial ashtu që të kthejë faktorialin e një numri të plotë nga intervali 0 deri 20, të dhënë si argument.
4. Shfrytëzoni metodën factorial nga detyra paraprake për të implementuar metodën vijuese.

```
/** Llogarit kosinusin e argumentit sipas formulës së Taylor-it:
 *   cos(x)=1-(x^2/2!)+(x^4/4!)-(x^6/6!)+...-(x^20/20!)
 * @param x - vlera, në radian, e argumentit
 * @return vlera e kosinuit e llogaritur sipas formulës */
public double cos(double x)
```

(Udhëzim: Shfrytëzoni Math.pow(a,b) për të llogaritur a^b .)

5. Krijoni një aplikacion i cili së pari kërkon nga shfrytëzuesi një vlerë të plotë pozitive n , pastaj krijon një varg prej n numrash të plotë, dhe kërkon nga shfrytëzuesi futjen interaktive të të gjitha vlerave të elementeve të vargut. Më në fund, aplikacioni llogarit dhe afishon numrin e elementeve negative të vargut. (P.sh., në qoftë se shfrytëzuesi ka futur vlerën 4 për n dhe pastaj vlerat vijuese për katër elementet: 2, -1, 0, -5, aplikacioni duhet të afishojë se numri i elementeve negative të vargut është 2.)

6. Shkruani metodën vijuese:

```
/** Kthen elementin më të vogël të vargut të dhënë.  
 * @param v - vargu i dhënë  
 * @return elementi minimal (më i vogël) */  
public int minElement(int[] v)
```

7. Shkruani metodën vijuese për llogaritjen e shumës së vlerave reciproke:

```
/** Llogarit shumën  $1/1+1/2+\dots+1/n$   
 * @param n - kufiri i sipërm; pozitiv  
 * @return shuma për  $n$  pozitiv;  
 * -1 për  $n$  jopozitiv */  
public double sumReciprocals(int n)
```

8. Krijoni një matricë numrash realë me 12 rreshta dhe 12 shtylla dhe inicializojeni ashtu që elementi në reshtin i dhe shtyllën j të përmbajë herësin $\frac{i}{j}$ ($i, j = 1, 2, \dots, 12$). Afishoni pastaj tabelën e pjesëtimit deri në 10 ashtu që herësat të paraqiten me dy shifra pas pikës decimale. (**Udhëzim:** Keni kujdes në operacionin e pjesëtimit me operandë numra të plotë.)

9. Çfarë do të afishojë urdhëri `while` vijues?

```
int j = 10;  
int i = -3;  
while ( j > i )  
{  
    j = j - 2;  
    System.out.println(i + " " + j);  
    i = i + 2;  
}
```

10. Krijoni një varg prej 100 numrash `double` të tillë që vlera e `d[i]` të jetë i^2 .