GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2023 מועד א'
סמל השאלון: 97105
שם התלמיד/ה: _____

מבנה נתונים ותכנות מונחה עצמים

הנדסת תוכנה · הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: **בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 14) בכל מבחן 10 שאלות. **חלק א' – 48 נקודות** שאלות 1-4: יש לענות על **שלוש** שאלות בלבד. ערך כל שאלה 16 נקודות. **חלק ב' – 36 נקודות** שאלות 5-8: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 18 נקודות. **חלק ג' – 16 נקודות** שאלות 9-10: יש לענות על **אחת** בלבד. ערך שאלה 16 נקודות. **בסך הכול: 100 נקודות.**

ג. חומר עזר מותר בשימוש: 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית). 2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר. **אין לצרף ספרים או חוברות עם פתרונות.**

ד. הוראות מיוחדות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות. 4. יש לכתוב את התשובות במחברת הבחינה **בעט בלבד**, בכתב יד ברור. 5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה. 6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק. 7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד. 8. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.** 9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לכם להמשיך בפתרון השאלה, נמקו את בחירתכם.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

1. ציוני תלמידים שמורים בשני תורים: תור ראשון מכיל ציונים ותור שני מכיל עבור כל אחד מהתלמידים את מספר הבחינות שהשלים.

לדוגמה:

תור ציונים **marks**: (באיור הראשון לשאלה)

תור בחינות **tests**: (באיור השני לשאלה)

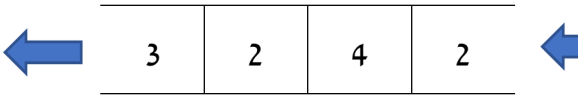
תלמיד ראשון השלים שלוש בחינות וקיבל את הציונים הבאים: 80, 90, 100.

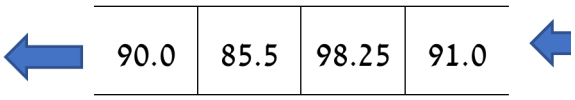
תלמיד שני השלים שתי בחינות וקיבל את הציונים הבאים: 75, 96.

תלמיד שלישי השלים ארבע בחינות וקיבל את הציונים הבאים: 100, 100, 97, 96.

תלמיד רביעי השלים שתי בחינות וקיבל את הציונים הבאים: 94, 88.







א. כתבו פעולה בשם averageQueue המקבלת שני תורים ומחזירה תור חדש של ממוצעי ציונים של כל התלמידים יחד.

עבור שני התורים שבדוגמה הפעולה תחזיר תור חדש המכיל את ממוצע הציונים של כל אחד מבין ארבעת התלמידים (התור באיור לסעיף).

כותרת הפעולה:

Java:

```
public static Queue<Double> averageQueue(Queue<Integer> marks,
                                         Queue<Integer> tests)
```

C#:

```
public static Queue<double> AverageQueue(Queue<int> marks,
                                         Queue<int> tests)
```

(12 נקודות)

ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

(4 נקודות)

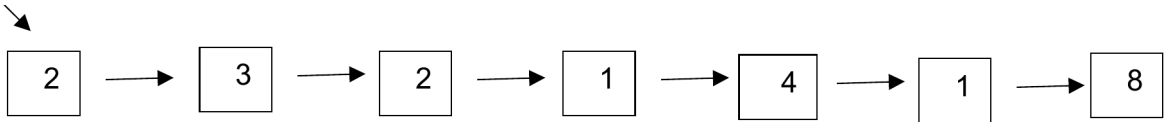
נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

2. אפשר לייצג מספר שלם חיובי גדול מאוד כשרשרת חוליות כאשר כל חוליה מכילה ספרה אחת של המספר.

החוליה **האחרונה** מכילה ספרת האחדות.

לדוגמה:

המספר 2321418 (במילים: "שני מיליון שלוש מאות עשרים ואחד אלף ארבע מאות ושמונה עשרה"), מיוצג על ידי שרשרת החוליות הבאה:



א. כתבו פעולה חיצונית בשם numDigits המקבלת הפניה לחוליה ראשונה של שרשרת חוליות המייצגת מספר שלם וחיובי. הפעולה תחזיר כמות ספרות של מספר.

כותרת הפעולה:

Java:

```
public static int numDigits(Node<Integer> n)
```

C#:

```
public static int NumDigits(Node<int> n)
```

(5 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. כתבו פעולה חיצונית בשם compare המקבלת הפניות לשתי שרשראות חוליות n1 ו-n2 המייצגות שני מספרים שלמים חיוביים.

אם המספר המיוצג בשרשרת הראשונה גדול מהמספר המיוצג בשרשרת השנייה, הפעולה תחזיר 1.

אם המספר המיוצג בשרשרת הראשונה קטן מהמספר המיוצג בשרשרת השנייה, הפעולה תחזיר 2.

אם שני מספרים שווים, הפעולה תחזיר 0.

כותרת הפעולה:

Java:

```
public static int compare(Node<Integer> n1, Node<Integer> n2)
```

C#:

```
public static int Compare(Node<int> n1, Node<int> n2)
```

(8 נקודות)

ג. מהי סיבוכיות הפעולה compare מהסעיף הקודם? הסבירו את תשובתכם.

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

3. נתונות שתי המחלקות Fruit ו-Apple.

Java:

```
public class Fruit
{
 protected int weight;
 public Fruit (int val)  { weight = val; }
 public int getWeight()  { return weight; }
}
public class Apple extends Fruit
 {
 private String color;
 public Apple (int val, String col)
   {
     super(val);
     color=col;
   }
 public boolean validWeight()
   {
    return weight > 80 && weight < 140;
   }
}
```

C#:

```
public class Fruit
{
  protected int weight;
  public Fruit (int val)   { weight = val; }
  public int GetWeight()   { return weight; }
}
public class Apple : Fruit
 {
  private string color;
  public Apple (int val, string col):base(val)
   {
     color=col;
   }
  public bool ValidWeight()
   {
     return weight > 80 && weight < 140;
   }
}
```

א. **לפניכם ארבעה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון ונמקו את קביעתכם.**

1. **המחלקה Fruit יורשת את הפעולה validWeight() מהמחלקה Apple.**
2. **המחלקה Apple יורשת את כל התכונות ואת כל הפעולות של המחלקה Fruit.**
3. **המחלקה Apple יכולה לגשת ישירות לתכונה weight של המחלקה Fruit.**
4. **המחלקה Fruit יכולה לגשת לתכונה color של המחלקה Apple.**
5. **למחלקה Apple אין פעולה toString()**

(ב- ValidWeight: C#() ו-ToString())

(5 נקודות)

ב. לפניכם קטע קוד מהפעולה הראשית (main) של המחלקה TestFruit:

Java:

```
Fruit    first = new Apple (100, "RED");
Fruit    second = new Fruit (90);
Apple    third = new Apple(100, "RED");
```

C#:

```
Fruit    first = new Apple (100, "RED");
Fruit    second = new Fruit (90);
Apple    third = new Apple(100, "RED");
```

בעבור כל אחת מההוראות שלפניכם קבעו אם היא תקינה או אינה תקינה.

אם היא אינה תקינה, כתבו אם זו שגיאת ריצה או שגיאת הידור (קומפילציה).

Java:

```
1. boolean b = first. validWeight();
2. boolean b = second. validWeight();
3. boolean b = ((Apple)first). validWeight();
4. boolean b = ((Apple)second). validWeight();
5. boolean b = ((Apple)first).color.equals(third.color);
```

C#:

```
1. bool b = first. ValidWeight();
2. bool b = second. ValidWeight();
3. bool b = ((Apple)first). ValidWeight();
4. bool b = ((Apple)second). ValidWeight();
5. bool b = ((Apple)first).color.Equals(third.color);
```

(5 נקודות)

ג. כתבו פעולה המקבלת מערך עצמים מטיפוס Object. הפעולה תדפיס כמה עצמים הם מטיפוס Apple, כמה עצמים מטיפוס Fruit ואינו מטיפוס Apple וכמה עצמים הם לא מטיפוס Fruit.

(6 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

4. נתונות שלוש המחלקות הבאות:

Java:

```
public class A
{
  protected int x;
  public A(){
    this.x = 5;
  }

  public A(int x) {
    this.x = x;
  }

  public A(int x, int y) {
    this(x + y);
  }
  public int method (int i)  {
      return x + i;
  }
  public String toString() {
    return this.x + " ˙ ";
  }
}
public class B extends A
{
  private int y ;
  public B (int x) {
    super(x, x);
    this.y = x;
  }

  public B(int x, int y) {
    this(x ˙ y);
  }

  public B(String s) {
    System.out.println (s + "% " + this.y);
  }

  public String toString() {
    return this.x + " " + this.y ;
  }
}
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
public class Test
{
  public static void main(String[] args)
  {
    A a1 = new B(4, 2);
    System.out.println(a1);

    A a2 = new A(4, 2);
    System.out.println(a2);

    B a3 = new B(4, 2);
    System.out.println(a3);

    A b1 = new A();
    System.out.println(b1);

    A c1 = new B("something");
    System.out.println(c1);
  }
}
```

C#:

```
public class A
{
  protected int x;
  public A(){
    this.x = 5;
  }

  public A(int x) {
    this.x = x;
  }

  public A(int x, int y): this(x +y) {

  }
  public int Method (int i)  {
       return x + i;
  }
  public override string ToString  () {
    return this.x + " ' ";
  }
}
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
public class B : A
{
  private int y;
  public B (int x): base(x, x) {
    this.y = x;
  }

  public B(int x, int y): this(x * y){
  }

  public B(string s) {
    Console.WriteLine (s + "% " + this.y);
  }

  public override string ToString () {
    return this.x + " " + this.y ;
  }
}
```

```
public class Test
{
  public static void Main(string[] args)
  {
    A a1 = new B(4, 2);
    Console.WriteLine(a1);

    A a2 = new A(4, 2);
    Console.WriteLine(a2);

    B a3 = new B(4, 2);
    Console.WriteLine(a3);

    A b1 = new A();
    Console.WriteLine(b1);

    A c1 = new B("something");
    Console.WriteLine(c1);
  }
}
```

א. עקבו אחרי ביצוע פעולה הראשית main ורשמו מה יהיה הפלט. יש להראות את העצמים שנוצרו ואת ערכי התכונות שלהם.

(10 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. לכל אחת מהפעולות הבאות קבעו אם אפשר **להוסיף** אותה במחלקה A, וציינו את האפשרות המתאימה מבין האפשרויות הבאות:

- שגיאת קומפילציה.
- קוד תקין, דריסה (overriding).
- קוד תקין, העמסה (overloading).

Java:

```
1. public int method () { return 1; }
2. public double method (int i){ return i; }
3. public void method (double i){ System.out.print(i); }
```

C#:

```
1. public virtual int Method () { return 1; }
2. public virtual double Method (int i){ return i; }
3. public virtual void Method (double i){ Console.Write(i); }
```

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ג. אין קשר לסעיף ב'.

לכל אחת מהפעולות הבאות קבעו אם אפשר **להגדיר** אותה במחלקה B וציינו את האפשרות המתאימה מבין האפשרויות הבאות:

- שגיאת קומפילציה.
- קוד תקין, דריסה (overriding).
- קוד תקין, העמסה (overloading).

Java:

```
1. public int method () { return 10; }
2. public int method (int i){ return i*10; }
3. public void method (int i){ System.out.print(10*i); }
```

C#:

```
1. public override int Method () { return 10; }
2. public override int Method (int i){ return i*10; }
3. public override void Method (int i){ Console.Write(10*i); }
```

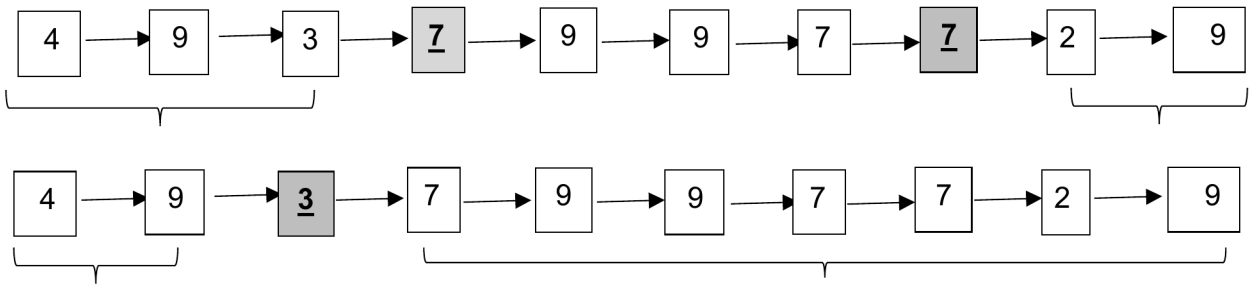
(3 נקודות)

5. נתונה שרשרת חוליות המכילה מספרים שלמים. כל ערך בשרשרת יכול להופיע כמה פעמים.

נגדיר "**מרחק**" של ערך כלשהו בשרשרת כסכום המרחקים הקטנים ביותר של מופע של אותו מספר מקצוות השרשרת.

דוגמה 1: ה"מרחק" של מספר 7 בשרשרת הבאה הינו 5 (3 מתחילת השרשרת ו- 2 עד לסוף השרשרת).

דוגמה 2: ה"מרחק" של מספר 3 בשרשרת הבאה הינו 9 (2 מתחילת השרשרת ו-7 עד לסוף השרשרת)



נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

א. כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת חוליות ומספר num ומחזירה את ה"מרחק" של המספר num. אם num לא נמצא בשרשרת הפעולה תחזיר 1-.

כותרת הפעולה:

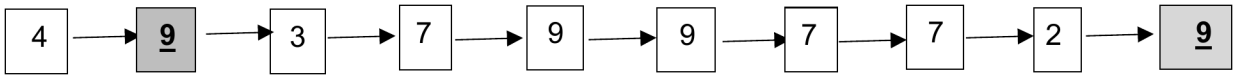
Java:

```
public static int distance(Node<Integer> lst, int num)
```

C#:

```
public static int Distance(Node<int> lst, int num)
```

(7 נקודות)



ב. כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת חוליות ומחזירה את **המספר** בעל ה"מרחק" הקטן ביותר.

עבור השרשרת הבאה הפעולה תחזיר 9 (ה"מרחק" של הערך 9 הוא הקטן ביותר).

כותרת הפעולה:

Java:

```
public static int minDistanceValue(Node<Integer> lst)
```

C#:

```
public static int MinDistanceValue(Node<int> lst)
```

(7 נקודות)

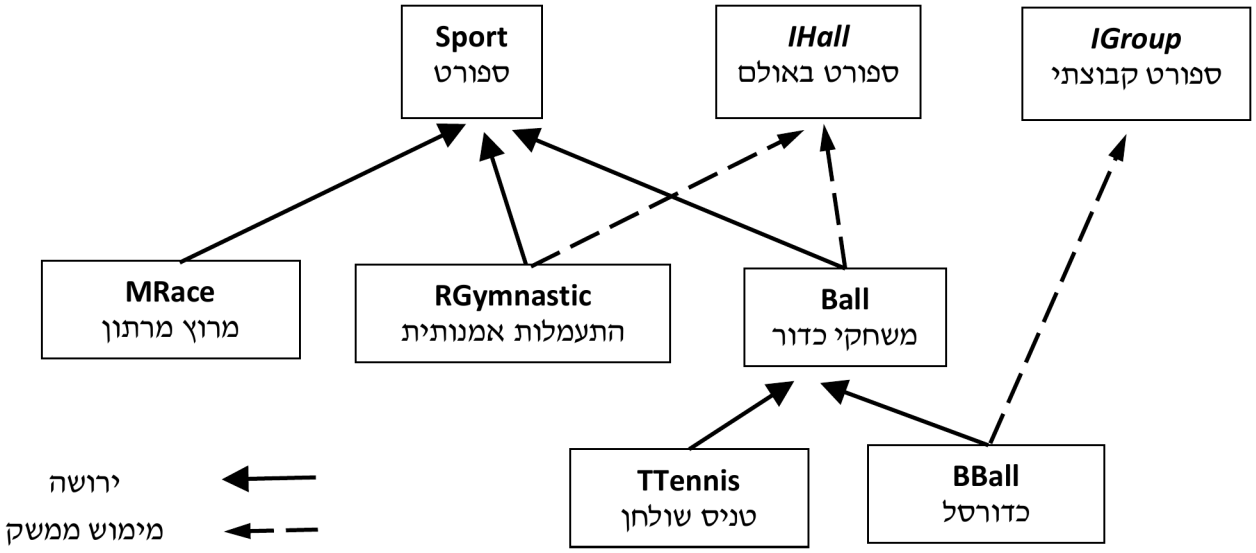
ג. מהן סיבוכיות של הפעולות Distance ו-MinDistanceValue מסעיפים א' ו- ב'? הסבירו את תשובתכם.

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

6. נתונה היררכיית המחלקות הבאה הכוללת שש מחלקות: Sport, MRace, RGymnastic, Ball, TTennis, BBall ושני ממשקים: IHall ו-IGroup.

בכל מחלקה מוגדרת פעולה בונה ללא פרמטרים.



א. כתבו עבור כל מחלקה, את כותרת המחלקה.

(3 נקודות)

ב. עבור כל אחת מהפקודות הבאות קבעו אם היא נכונה או אינה נכונה. **נמקו את קביעתכם.**

```
1. Sport s1 = new BBall();
2. Sport s2 = new MRace();
3. TTennis  t1 = new Ball();
4. BBall b1 = new IGroup();
5. IHall i1 = new Ball();
6. IHall i2 = new TTennis();
```

(6 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ג. כתבו פעולה המקבלת מערך עצמים מסוג Sport. הפעולה תחזיר מספר עצמים המייצגים ענף ספורט קבוצתי שמתקיים באולם.

כותרת הפעולה:

Java:

```
public static int count (Sport[] s)
```

C#:

```
public static int Count (Sport[] s)
```

שימו לב: יש לכתוב את הפעולה כך שהיא תעבוד נכון גם כאשר יתווספו לפרויקט מחלקות נוספות, כמו VBall (כדורעף), HBall (כדוריד) וכד'

(6 נקודות)

ד. רוצים להוסיף לפרויקט מחלקה חדשה המתארת כדורגל. האם אפשר להוסיף את המחלקה FBall תוך שמירה על היררכיה קיימת או שיש לשנות את ההיררכיה? ציירו תרשים של היררכיית המחלקות החדשה (כולל את המחלקה FBall).

(3 נקודות)

7. לפניכם הפעולות הבאות:

Java:

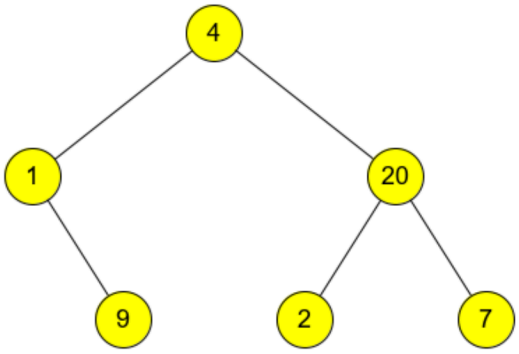
```
public static String what(BinNode<Integer> t)
{
   String output = "(";
   if (t.getLeft()!= null)
      output += what(t.getLeft());
   output += t.getValue();
   if (t.getRight()!= null)
      output += what (t.getRight());
   output += ")";
   return output;
}
public static String why(BinNode<Integer> t)
{
   String output = "(";
   output += t.getValue();
   if (t.getLeft()!= null)
      output += why(t.getLeft());
   if (t.getRight()!= null)
      output += why (t.getRight());
   output += ")";
   return output;
}
```

C#:

```
public static string  What(BinNode<int> t)
{
   string  output = "(";
   if (t.GetLeft()!= null)
      output += What(t.GetLeft());
   output += t.GetValue();
   if (t.GetRight()!= null)
      output += What (t.GetRight());
   output += ")";
   return output;
}
public static string  Why(BinNode<int> t)
{
   string  output = "(";
   output += t.GetValue();
   if (t.GetLeft()!= null)
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
    output += Why(t.GetLeft());
  if (t.GetRight()!= null)
    output += Why (t.GetRight());
  output += ")";
  return output;
}
```



א. נתון העץ הבינרי bt הבא:

מה תהיה תוצאת זימון הפעולה what(bt)? **חובה להראות את המעקב!**

(7 נקודות)

ב. לפעולה what הועברה כפרמטר הפניה לשורש של עץ בינרי. המחרוזת שהוחזרה מהשיטה היא:

`((((30)) 20 ((50)) 5)) 40 ((10))`

ציירו את העץ הבינרי שהועבר כפרמטר פעולה.

(7 נקודות)

ג. האם קיים עץ בינארי bt בעל שלושה צמתים לפחות שעבורו תוצאת זימון הפעולה what(bt) תהיה זהה לתוצאת זימון הפעולה why(bt)? אם כן - ציירו את העץ, אם לא - הסבירו מדוע.

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

8. נתונות שלוש המחלקות הבאות: One, Two, Tester

Java:

```
public class One
{
  protected int x;

  public One(){
    this.x = 1;
  }

  public One(int x) {
    this.x = x;
  }

  public void f(int x) {
    this.x += x;
  }

  public String toString() {
    return "x=" + this.x;
  }
} // end of One

public class Two extends One
{
  public Two() {
    super(3);
  }

  public Two(int x) {
    super.f(x);
    f(x);
  }

  public void f(int x)
  {
    this.x -= x;
    super.f(x);
  }
}// end of Two
```

```
public class Tester
{
  public static void print(One[] arr)
  {
```

```
    for(int i=0; i < arr.length; i++)
    {
      System.out.println(i+" "+arr[i]);
    }
  }

  public static void doF(One[] arr, int num)
  {
    for(int i=0; i < arr.length; i++)
    {
      arr[i].f(num);
    }
  }

  public static void what(One[] arr)
  {
    int k=0;
    for(int i = 0; i < arr.length; i++)
    {
      if(arr[i] instanceof Two)
        k++;
    }
    k = arr.length - k;
    System.out.println("k = "+ k);
  }

  public static void main(String [] args)
  {
    One[] arr = new One[6];
    arr[0]= new One();
    arr[1]= new One(3);
    arr[2] = new Two();
    arr[3] = new Two(2);
    arr[4] = new Two(5);
    arr[5] = arr[1];
    print(arr);
    doF(arr, 2);
    print(arr);
    what(arr);
  }
}
```

C#:

```
public class One
{
   protected int x;

   public One(){
      this.x = 1;
   }

   public One(int x) {
      this.x = x;
   }

   public virtual void F(int x) {
      this.x += x;
   }

   public override string ToString () {
      return "x=" + this.x;
   }
} // end of One

public class Two : One
{
   public Two(): base(3) {

   }

   public Two(int x) {
      base.F(x);
      F(x);
   }

   public override void F(int x)
   {
      this.x -= x;
      base.F(x);
   }
}// end of Two
```

```
public class Tester
{
   public static void Print(One[] arr)
   {
      for(int i=0; i < arr.Length; i++)
      {
         Console.WriteLine(i+" "+arr[i]);
```

```
    }
  }

  public static void DoF(One[] arr, int num)
  {
    for(int i=0; i < arr.Length; i++)
    {
      arr[i].F(num);
    }
  }

  public static void What(One[] arr)
  {
    int k=0;
    for(int i = 0; i < arr.Length; i++)
    {
      if(arr[i] is Two)
        k++;
    }
    k = arr.Length - k;
    Console.WriteLine("k = "+ k);
  }

  public static void Main(string [] args)
  {
    One[] arr = new One[6];
    arr[0]= new One();
    arr[1]= new One(3);
    arr[2] = new Two();
    arr[3] = new Two(2);
    arr[4] = new Two(5);
    arr[5] = arr[1];
    Print(arr);
    DoF(arr, 2);
    Print(arr);
    What(arr);
  }
}
```

א. עקבו אחרי ביצוע הפעולה הראשית ורשמו מה יהיה הפלט של הפעולה.
יש להציג את כל העצמים שנוצרו ואת השינויים בערכי התכונות שלהם במהלך הביצוע.
(15 נקודות)

ב. מה מבצעת הפעולה what?
(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

9. לפניכם מספר פעולות המוגדרות על המחסנית:

- `public static int numOf(Stack<Integer> m)`

הפעולה מחזירה מספר איברים במחסנית m.

- `public static int randChoice (Stack<Integer> m)`

הפעולה מחזירה איבר שנבחר באקראי מתוך מחסנית m.

- `public static Stack<Integer> createSmall(Stack<Integer> m, int a)`

הפעולה מקבלת מחסנית m המכילה מספרים שלמים וערך a ומחזירה מחסנית חדשה המכילה את כל המספרים מתוך m הקטנים מ- a.

- `public static Stack<Integer> createLarge(Stack<Integer> m, int a)`

הפעולה מקבלת מחסנית m המכילה מספרים שלמים וערך a ומחזירה מחסנית חדשה המכילה את כל המספרים מתוך m הגדולים מ- a.

- `public static void eraseSmalls(Stack<Integer> m, int k)`

הפעולה מוחקת ממחסנית m את k האיברים **הקטנים ביותר**. **הנחה**: במחסנית יש לפחות k איברים.

- `public static void eraseLarges(Stack<Integer> m, int k)`

הפעולה מוחקת ממחסנית m את k האיברים **הגדולים ביותר**. **הנחה**: במחסנית יש לפחות k איברים

כותרות הפעולות ב-C#:

```
public static int NumOf(Stack<int> m)
public static int RandChoice (Stack<int> m)
public static Stack<int> CreateSmall(Stack<int> m, int a)
public static Stack<int> CreateLarge(Stack<int> m, int a)
public static void EraseSmalls(Stack<int> m, int k)
public static void EraseLarges(Stack<int> m, int k)
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il



43	-3	8	9	12	-9	7	3	1

א. נתונה פעולה רקורסיבית what המשתמשת בפעולות הנ"ל. הפעולה מקבלת מחסנית s, המכילה מספר אי זוגי של מספרים שלמים, שכולם שונים זה מזה.

Java:

```
public static int what(Stack<Integer> s)
{
  int a = randChoice(s);
  Stack<Integer> m1 = createSmall(s,a);
  int k1 = numOf(m1);
  Stack<Integer> m2 = createLarge(s,a);
  int k2 = numOf(m2);
  if(k1 > k2)
  {
    eraseSmalls(m1, k2+1);
    return what(m1);
  }
  if(k2 > k1)
  {
    eraseLarges(m2, k1+1);
    return what(m2);
  }
  return a;
}
```

C#:

```
public static int What(Stack<int> s)
{
  int a = RandChoice(s);
  Stack<int> m1 = CreateSmall(s,a);
  int k1 = NumOf(m1);
  Stack<int> m2 = CreateLarge(s,a);
  int k2 = NumOf(m2);
  if(k1 > k2)
  {
    EraseSmalls(m1, k2+1);
    return What(m1);
  }
  if(k2 > k1)
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
  {
    EraseLarges(m2, k1+1);
    return What(m2);
  }
  return a;
}
```

עקבו אחרי זימון הפעולה what(s) עבור מחסנית הבאה ורשמו מה תהיה תוצאת הזימון.

יש להראות ערכי משתנים ותכנים של מחסניות בכל זימון הרקורסיבי.

(7 נקודות)

ב. מה מבצעת הפעולה what עבור מחסנית המכילה מספר אי זוגי של מספרים שלמים, שכולם שונים זה מזה?

(3 נקודות)

ג. ממשו את הפעולה createSmall.

(3 נקודות)

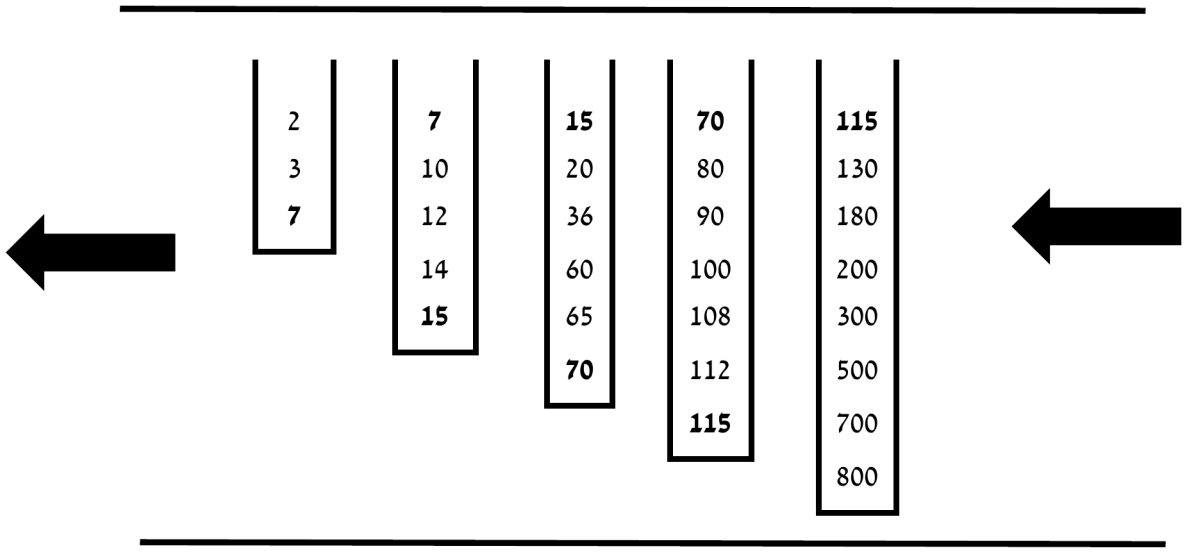
ד. ממשו את הפעולה randChoice.

(3 נקודות)

10. תור מחסניות נקרא **תור מושלם** אם הוא עונה על שלושת התנאים הבאים:

- הערכים בכל מחסנית הם סדרה עולה (הערך הקטן ביותר נמצא בראש המחסנית).
- האורכים של המחסניות (מספר האיברים בכל מחסנית) הם סדרה עולה (המחסנית הקצרה ביותר נמצאת בראש התור).
- איבר הנמצא בראש כל מחסנית שווה לאיבר הנמצא בתחתית המחסנית הקודמת לה בתור.

לדוגמה: תור מחסניות הבא הוא **תור מושלם**:



א. כתבו פעולה המקבלת תור מחסניות ובודקת אם הוא תור מושלם. אם כן - הפעולה תחזיר true, ואם לא - הפעולה תחזיר false.

כותרת הפעולה:

Java:

```
public static boolean isPerfect(Queue <Stack<Integer>> q)
```

C#:

```
public static bool IsPerfect(Queue <Stack<int>> q)
```

הערה: אם כתבתם פעולות עזר חובה לתעד אותן (לציין פרמטרים שהפעולה מקבלת, הנחות, ותיאור של מה הפעולה מבצעת). כמו כן, יש לציין את הסיבוכיות של כל פעולת עזר שכתבתם.

(12 נקודות)

ב. מהי סיבוכיות הפעולה isPerfect?

(4 נקודות)

כל מבחני מה"ט עם פתרונות מלאים: www.guyai.co.il/mahat