GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2024 מועד ב'
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 17) בכל מבחן עשר שאלות. **חלק א' – 48 נקודות** שאלות 1-5: יש לענות על **שלוש** שאלות בלבד. ערך כל שאלה 16 נקודות. **חלק ב' – 36 נקודות** שאלות 6-10: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 18 נקודות. **חלק ג' – 16 נקודות** שאלות 11-12: יש לענות על שאלה **אחת** בלבד. ערך השאלה 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. **חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!**

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

בהצלחה!

השאלות

1. **א.** מחסנית של מספרים שלמים נקראת "מחסנית שוויון" אם היא עונה על שלושת התנאים הבאים:

- במחסנית יש רק ערכים החיוביים.
- מספר האיברים הזוגיים שווה למספר האיברים האי-זוגיים.
- סכום האיברים הזוגיים שווה לסכום האיברים האי-זוגיים.

כתבו פעולה המקבלת מחסנית של מספרים שלמים ובודקת אם היא "מחסנית שוויון".

יש לשמור על תוכן של המחסנית אחרי ביצוע הפעולה. כותרת הפעולה:

```
public static boolean isParityStack(Stack<Integer> st)
```

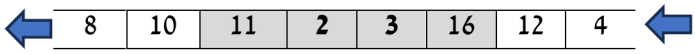
```
public static bool IsParityStack(Stack<int> st)
```

(12 נקודות)

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

(4 נקודות)

2.



א. נתון התור q של מספרים שלמים. כל איבר בתור מופיע פעם אחת (אין חזרות). המרחק בין שני איברים בתור הוא כמות האיברים הנמצאים ביניהם.

כתבו פעולה המקבלת תור q ושני מספרים נוספים x ו- y. הפעולה **תחשב ותחזיר** את המרחק בין שני איברי התור שערכם x ו- y.

לדוגמה:

עבור התור q הבא ופרמטרים $x = 10, y = 12$ הפעולה תחזיר 4 כי בין הערכים 10 ו- 12 יש בתור ארבעה איברים:

עבור התור q ופרמטרים $x = 12, y = 10$ הפעולה גם תחזיר 4 כי בין הערכים 10 ו- 12 ישנם בתור ארבעה איברים.

אם בתור אין איבר שערכו x או אין איבר שערכו y, הפעולה תחזיר ערך -1 .

כותרת הפעולה:

```
public static int distance(Queue<Integer> q, int x, int y)
```

```
public static int Distance(Queue<int> q, int x, int y)
```

(12 נקודות)

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

(4 נקודות)

3. א. שרשרת נקראת "שרשרת חזרות" אם כל ערך שבה מופיע **בדיוק פעמיים**. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת תווים. הפעולה תבדוק אם השרשרת היא "שרשרת חזרות". אם כן - הפעולה תחזיר ערך true, ולא - הפעולה תחזיר ערך false.

כותרת הפעולה:

```
public static boolean repeatList(Node<Character> chain)
```

```
public static bool RepeatList(Node<char> chain)
```

(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

4. ברשות הדואר החליטו למחשב את המערך לקבלת דברי הדואר. לצורך כך הוגדרו שלוש מחלקות שונות:

MailItem, Letter ו- Package.

המחלקה Letter מייצגת מכתב. בעבור כל מכתב יש לשמור את הנתונים הבאים:

- שם שולח המכתב, מטיפוס מחרוזת, String (ב-C#: string).
- מען המכתב (כתובת), מטיפוס מחרוזת, String (ב-C#: string).
- דואר רשום מטיפוס בוליאני, boolean (ב-C#: bool): true - עבור דואר רשום, false - עבור דואר רגיל.
- גודל המכתב, טיפוס תווי, char ('S' - עבור מכתב קטן, 'B' - עבור מכתב בגודל סטנדרטי, 'L' - עבור מכתב גדול).
- מחיר בסיסי למשלוח מכתב בגודל סטנדרטי, מטיפוס מספר ממשי, double

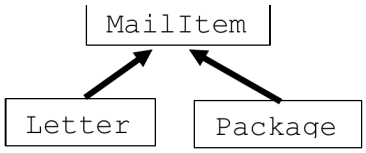
מחיר משלוח מכתב תלוי בגודלו ובסוגו: בעבור מכתב בדואר רשום יש להוסיף 10 ₪, עבור מכתב קטן יש הנחה של שני שקלים, עבור מכתב גדול יש תוספת של 3 ₪

המחלקה Package מייצגת חבילה. בעבור כל מכתב יש לשמור את הנתונים הבאים:

- שם שולח החבילה, מטיפוס מחרוזת, String (ב-C#: string).
- מען המכתב (כתובת), מטיפוס מחרוזת, String (ב-C#: string).
- משקל החבילה בקילוגרמים מטיפוס מספר שלם, int.
- האם שביר?: true - אם תוכן החבילה שביר, false - אם לא שביר.
- מחיר בסיסי למשלוח חבילה מטיפוס מספר ממשי, double.

מחיר משלוח חבילה תלוי בגודל החבילה וסוגה: בעבור משקל פחות מ- 5 קילוגרמים יש לשלם מחיר בסיסי. מעל 5 קילוגרמים יש לשלם תוספת של 3 ₪ לכל קילוגרם נוסף. אם התוכן שביר יש להוסיף עוד 15 ₪.

המחלקה MailItem היא מחלקת האב למחלקות Letter ו-Package (המחלקות Letter ו- Package יורשות מהמחלקה MailItem) בהתאם לעקרונות תכנות מונחה עצמים:



א. כתבו כותרות ותכונות של כל אחת מהמחלקות MailItem, Letter ו- Package

(6 נקודות)

ב. כתבו פעולה

```
public double getRealPrice()
```

```
public double GetRealPrice()
```

הפעולה תחשב ותחזיר מחיר סופי של משלוח. יש לציין באיזה מחלקה/מחלקות צריך להוסיף את הפעולה בהתאם לעקרונות תכנות מונחה עצמים.

(4 נקודות)

ג. כתבו פעולה

```
public static String bigSender(MailItem[] args)
```

```
public static string BigSender(MailItem[] args)
```

הפעולה מקבלת מערך של משלוחים ומחזירה את שם השולח של החבילה היקרה ביותר (חבילה שמחיר המשלוח הוא הגבוה ביותר). אם אין אף חבילה במערך, הפעולה תחזיר ערך null. אפשר להניח שיש רק משלוח אחד כזה. אפשר להניח שקיימות פעולות set/get (ב-:C# Set/Get) בכל מחלקה.

(6 נקודות)

5.

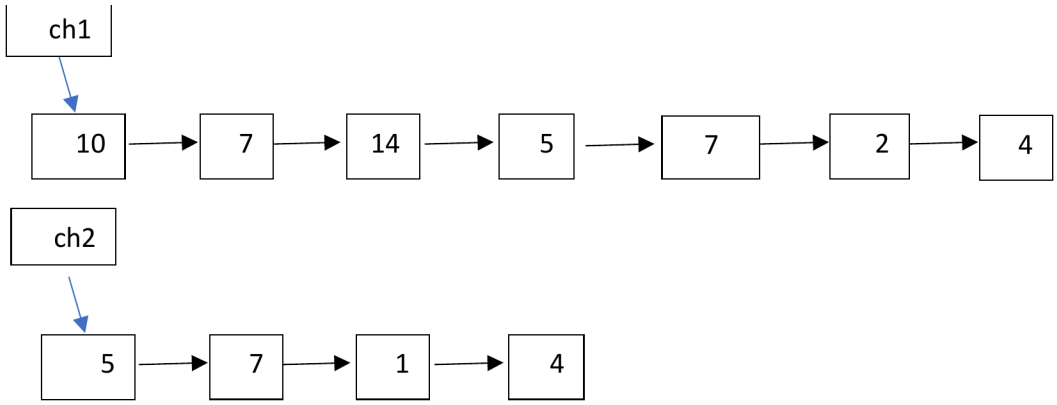
א. נתונה הפעולה what הבאה, מה מבצעת הפעולה?

```
public static int what(Node<Integer> ch){
    int c = 0;
    while(ch != null)
    {
      c++;
      ch = ch.getNext();
    }
    return c;
  }
```

```
public static int What(Node<int> ch){
    int c = 0;
    while(ch != null)
    {
      c++;
      ch = ch.GetNext();
    }
    return c;
  }
```

(1 נקודות)

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



ב. נתונה הפעולה why המשתמשת בפעולה what, מה תחזיר הפעולה why(ch1, ch2) עבור שתי שרשרות החוליות הבאות: **יש להראות מעקב אחרי ביצוע הפעולה why.**

אין צורך במעקב אחרי what.

```
public static boolean why(Node<Integer> ch1, Node<Integer> ch2)
{
    int len1 = what(ch1);
    int len2 = what(ch2);
    for(int i = 0; i < len1-len2; i++)
    {
      ch1 = ch1.getNext();
    }
    while(ch1!=null)
    {
      if(ch1.getValue() != ch2.getValue())
        return false;
      ch1 = ch1.getNext();
      ch2 = ch2.getNext();
    }
    return true;
}
```

```
public static bool Why(Node<int> ch1, Node<int> ch2) {
    int len1 = What(ch1);
    int len2 = What(ch2);
    for(int i = 0; i < len1-len2; i++)
    {
      ch1 = ch1.GetNext();
    }
    while(ch1!=null)
    {
      if(ch1.GetValue() != ch2.GetValue())
        return false;
      ch1 = ch1.GetNext();
      ch2 = ch2.GetNext();
```


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

```
    }
    return true;
}
```

(4 נקודות)

ג. תנו דוגמה של שרשרת ch3 באורך של **ארבע** חוליות, באופן שבו זימון הפעולה why(ch1, ch3) יחזיר ערך שונה מזה שהוחזר בסעיף ב'.

(4 נקודות)

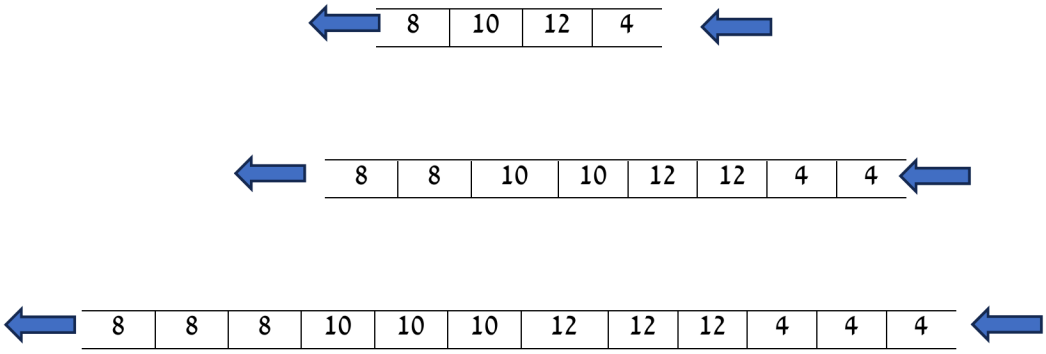
ד. תנו דוגמה של שרשרת ch4 באורך של **ארבע** חוליות, באופן שבו זימון הפעולה why(ch4, ch1) יחזיר ערך true. אם אי אפשר ליצור שרשרת כזאת, הסבירו למה.

(4 נקודות)

ה. מה מבצעות הפעולות what(ch) ו- why(ch1,ch2) באופן כללי?

(3 נקודות)

6.



א. כתבו פעולה void one(Queue<Integer> q, int k) (ב-C#: void One(Queue<int> q, int k)) המקבלת תור של מספרים שלמים q ומספר שלם וחיובי k. הפעולה "תכפיל" את התור באופן הבא:

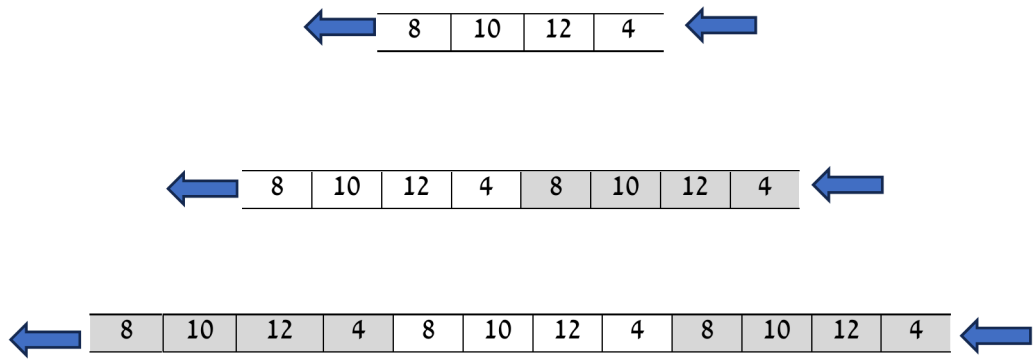
אם לפני זימון הפעולה התור q היה כך: 8, 10, 12, 4

אחרי זימון הפעולה one(q, 2) התור q יהיה כך: 8, 8, 10, 10, 12, 12, 4, 4

אחרי זימון הפעולה one(q, 3) התור q יהיה כך: 8, 8, 8, 10, 10, 10, 12, 12, 12, 4, 4, 4

(ראש התור משמאל; שלושת התורים מוצגים באיור לסעיף א'.)

(7 נקודות)



ב. כתבו פעולה void two(Queue<Integer> q, int k) (ב-C#: void Two(Queue<int> q, int k)) המקבלת תור של מספרים שלמים q ומספר שלם וחיובי k. הפעולה "תכפיל" את תור בצורה הבאה:

אם לפני זימון הפעולה התור q היה: 8, 10, 12, 4

אז אחרי ביצוע זימון הפעולה two(q, 2) התור q יהיה: 8, 10, 12, 4, 8, 10, 12, 4

אז אחרי ביצוע זימון הפעולה two(q, 3) התור q יהיה:

8, 10, 12, 4, 8, 10, 12, 4, 8, 10, 12, 4

(ראש התור משמאל; שלושת התורים מוצגים באיור לסעיף ב'.)

(7 נקודות)

ג. מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'-ב'? **הסבירו את תשובתכם.**

(4 נקודות)

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

7. נתונה מחלקה המתארת אבן דומינו (Domino):

```
public class Domino
{
 private int left;  // ערך מספרי צד שמאלי בין 0 ל-6 (כולל)
 private int right;  // ערך מספרי צד ימני בין 0 ל-6 (כולל)

 public Domino(int left, int right)
 {
  this.left = left;
  this.right = right;
 }
}
```


```
public class Domino
{
 private int left;  // ערך מספרי צד שמאלי בין 0 ל-6 (כולל)
 private int right;  // ערך מספרי צד ימני בין 0 ל-6 (כולל)

 public Domino(int left, int right)
 {
  this.left = left;
  this.right = right;
 }
}
```


במחלקה הוגדרו פעולות set/get (ב-C#: Set/Get)

א. כתבו במחלקה Domino את הפעולה rotate() (ב-C#: Rotate()). הפעולה מסובבת את האבן הנוכחית

כלומר: מחליפה בין הערכים left ו- right.

(4 נקודות)

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

נתונה המחלקה Deck המתארת רשימת אבני דומינו:

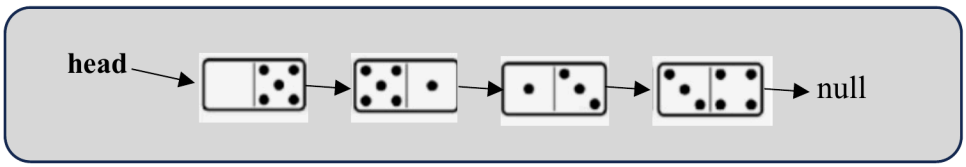
```
public class Deck
{
 private Node<Domino> head;
 public Deck()
 {
  this.head = null;
 }
}
```


```
public class Deck
{
 private Node<Domino> head;
 public Deck()
 {
  this.head = null;
 }
}
```


רשימת אבני דומינו נקראת "מסודרת היטב" אם ערך של מספרי צד ימני של כל אבן (פרט לאבן האחרונה ברשימה) זהה לערך של מספרי צד שמאלי של האבן הבאה.

לדוגמה: הרשימה הבאה מוגדרת "מסודרת היטב" (באיור לשאלה: האבנים $(0,5)$, $(5,1)$, $(1,3)$, $(3,4)$).

רשימה ריקה או רשימה שיש בה רק אבן אחת, היא רשימה "מסודרת היטב".



ב. כתבו במחלקה Deck פעולה בוודקת שרשימה היא "מסודרת היטב". אם כן - הפעולה תחזיר ערך true, ולא - תחזיר ערך false.

(7 נקודות)

ג. כתבו במחלקה Deck פעולה המקבלת אבן דומינו ובודקת אם אפשר להוסיף אותה לרשימה "מסודרת היטב" כך שהיא תישאר להיות "מסודרת היטב". אם כן - הפעולה תוסיף את האבן ותחזיר ערך true, ולא - תחזיר ערך false.

לדוגמה:

אפשר להוסיף לרשימה שבדוגמה בסעיף א' את האבנים הבאות $(0,4)$, $(1,1)$, $(4,6)$.

אי אפשר להוסיף את האבנים הבאות: $(3,2)$, $(5,6)$, $(2,2)$.

(7 נקודות)

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

8. **הגדרה:** עץ "**מסודר**" הוא עץ בינארי של מספרים שלמים העונה על הכללים הבאים:

- לכל צומת פרט לעלים יש ערך זוגי.
- הערכים של כל העלים הם מספרים אי-זוגיים.
- הערך של צומת הבן גדול מהערך של צומת האב.

א. ציירו עץ "מסודר" הכולל לפחות שישה צמתים.

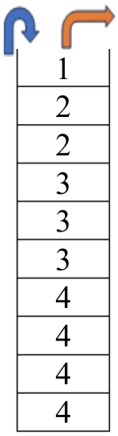
(4 נקודות)

ב. כתבו פעולה המקבלת הפניה לשורש של עץ בינרי של מספרים שלמים. הפעולה תבדוק אם העץ הוא "עץ מסודר". אם כן - הפעולה תחזיר ערך true,
ולא - הפעולה תחזיר ערך false.

(14 נקודות)

9. מחסנית "פירמידה n" היא מחסנית של מספרים שלמים וחיוביים שהמספר הראשון (מספר שנימצא בראש המחסנית) הוא 1 וכל מספר נוסף מופיע בו ברצף, מספר פעמים השווה לערכו, עד N. כלומר, 1 יופיע פעם אחת, לאחריו 2 יופיע שתי פעמיים, מספר 3 מופיע שלוש פעמים וכו'.

לדוגמה: המחסנית "פירמידה 4" (מהראש): $1, 2, 2, 3, 3, 3, 4, 4, 4, 4$



א. כתבו פעולה המקבלת מספר שלם וחיובי **n** ומחזירה מחסנית "פירמידה n".

(7 נקודות)

ב. כתבו פעולה המקבלת מחסנית של מספרים שלמים. הפעולה תבדוק אם המחסנית היא "פירמידה n". אם כן - הפעולה תחזיר ערך true,
ולא - הפעולה תחזיר ערך false.

(7 נקודות)

ג. מהן סיבוכיות הפעולות שכתבתם בסעיפים א' ו- ב'? **הסבירו את תשובתכם.**

(4 נקודות)

10. נתונות המחלקות A, B, C הבאות:

```
public class A
{
   private static int countA = 0;
   protected int x;
   public A()
    {
      countA++;
      this.x = 0;
   }
   public A(int x)
    {
      countA++;
      this.x = x;
   }
   public String toString()
    {
      return "A:"+this.x;
    }
   public int doIt(int z)
    {
      return this.x * z;
    }
   public static int getCount(){ return countA; }
}// end of class A

public class B extends A
{
   protected int y;
   private static int countB = 0;
   public B(int y)
   {
      this.y = y;
      countB++;
   }
   public B(int x, int y)
   {
      super(x);
      countB++;
      this.y = y;
   }
   public String toString()
   {
     return "B:"+super.toString()+":"+this.y;
   }
```


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

```
  public static int getCount(){return countB;}
  public int doIt(int z)
  {
    return super.doIt(z) + this.y;
   }
} //end of class B
public class C extends B
{
  private A myA;
  private static int countC = 0;
  public C(int x, int y)
  {
    super(x, y);
    this.myA = new A(x+y);
    countC++;
  }
  public static int getCount(){return countC;}
  public String toString()
  {
    return "C:"+super.toString()+" ["+this.myA+"]";
  }
}// end of class C
```

```
public class A
{
  private static int countA = 0;
  protected int x;
  public A()
   {
    countA++;
    this.x = 0;
  }
  public A(int x)
   {
    countA++;
    this.x = x;
  }
  public override string ToString()
   {
    return "A:"+this.x;
   }
  public virtual int DoIt(int z)
   {
    return this.x * z;
   }
  public static int GetCount(){ return countA; }
```


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

```
}// end of class A

public class B : A
{
  protected int y;
  private static int countB = 0;
  public B(int y)
  {
    this.y = y;
    countB++;
  }
  public B(int x, int y):base(x)
  {
    countB++;
    this.y = y;
  }
  public override string ToString()
  {
   return "B:"+base.ToString ()+":"+this.y;
  }
  public static new int GetCount(){return countB;}
  public override int DoIt(int z)
  {
    return base.DoIt(z) + this.y;
   }
} //end of class B

public class C : B
{
  private A myA;
  private static int countC = 0;
  public C(int x, int y): base(x, y)
  {
    this.myA = new A(x+y);
    countC++;
  }
  public static new int GetCount(){return countC;}
  public override string ToString()
  {
    return "C:"+base. ToString()+" ["+this.myA+"]";
  }
}// end of class C
```

א. **ציירו תרשים UML שמייצג את הקשרים בין המחלקות A, B, C.**

יש לסמן ירושה באמצעות חץ ►-- והכלה באמצעות הסימן ◆————

(1 נקודות)

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

ב. נתון קטע הקוד הבא, בפעולה הראשית במחלקה אחרת:

```
A[] arr = new A[5];
arr[0] = new A(13);
arr[1] = new B(5, 10);
arr[2] = new C(2, 7);
arr[3] = new C(3, 3);
arr[4] = new B(5);
```

```
A[] arr = new A[5];
arr[0] = new A(13);
arr[1] = new B(5, 10);
arr[2] = new C(2, 7);
arr[3] = new C(3, 3);
arr[4] = new B(5);
```

עקבו אחרי ביצוע קטע הקוד. יש לצייר את כל העצמים שנוצרו ולציין את ערכי התכונות שלהם.

(5 נקודות)

ג. לקוד של סעיף ב' התווסף הקוד הבא, רשמו מה יהיה הפלט בהתאם:

```
System.out.println("A objects: " + A.getCount());
System.out.println("B objects: " + B.getCount());
System.out.println("C objects: " + C.getCount());
```

```
Console.WriteLine("A objects: " + A.GetCount());
Console.WriteLine("B objects: " + B.GetCount());
Console.WriteLine("C objects: " + C.GetCount());
```

(3 נקודות)

ד. לקוד של סעיף ב' התווסף הקוד הבא, רשמו מה יהיה הפלט:

```
for (int k=0; k <arr.length; k++)
 System.out.println(arr[k]);
```

```
for (int k=0; k <arr.Length; k++)
 Console.WriteLine(arr[k]);
```

(5 נקודות)

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

ה. לקוד של סעיף ב' התווסף הקוד הבא, רשמו מה יהיה הפלט:

```
for (int k=0; k <arr.length; k++)
 if(arr[k] instanceof B)
   System.out.println(arr[k].doIt(k+1));
```

```
for (int k=0; k <arr.Length; k++)
 if(arr[k] is B)
   Console.WriteLine(arr[k].DoIt(k+1));
```

(4 נקודות)

11. החברת **Helpl4U** נותנת שירותי תמיכה טלפונית. מוקד השירות כולל **15** עמדות תמיכה הממוספרות **0-14**.

שירותי התמיכה מתנהלים באופן הבא:

- כל מתקשר משאיר את שמו ואת מספר הטלפון שלו לרכז מערכת.
- הרכז מצרף את פרטי המתקשר לאוסף הממתינים בעמדת התמיכה הכי פחות עמוסה.
- כאשר עמדת התמיכה מסיימת טיפול במתקשר, פרטיו נמחקים מאוסף הממתינים של העמדה.

מערכת לניהול שירות כוללת כמה מחלקות.

המחלקה **Caller** (מתקשר). למחלקה הוגדרו פעולה בונה ופעולות getName() ו- getPhone() (ב-C#: GetName() ו- GetPhone())

פעולה	תיאור
`Caller (String name, String phone)` (ב-C#: `Caller (string name, string phone)`)	הפעולה בונה מתקשר חדש ששמו name ומספר הטלפון שלו phone

המחלקה **Service** מייצגת עמדת שירות. לפניכם ממשק חלקי של המחלקה Service:

פעולה	תיאור
`void addCaller (Caller c)` (ב-C#: `AddCaller (Caller c)`)	מוסיפה מתקשר c לאוסף הממתינים בעמדה
`Caller callback()` (ב-C#: `Callback()`)	הפעולה מחזירה את המתקשר הראשון מבין הממתינים בעמדה, ומוחקת את נתוניו מאוסף הממתינים. **הנחה**: יש ממתינים בעמדה
`int getNumOfCallers()` (ב-C#: `GetNumOfCallers()`)	הפעולה מחזירה את מספר ממתינים בעמדה

א. כתבו את כותרת המחלקה Service ואת התכונות שלה

לייצוג אוסף, אפשר להשתמש במחלקות Stack, Queue, Node. אפשר להוסיף תכונות נוספות. חובה לתעד את התכונות.

(3 נקודות)

ב. כתבו את הפעולה callback() (ב-Callback :C#())

(3 נקודות)

ניהול מוקד השירות נעשה בעזרת המחלקה **Support**. לפניכם ממשק חלקי של המחלקה Support:

פעולה	תיאור
void startSupport (String name, String` C#: `void StartSupport (string-ב) `phone) (`name, string phone)	הפעולה מקבלת פרטים של מתקשר ומוסיפה אותו לסוף אוסף הממתינים של העמדה שמספר הממתינים בה הוא הנמוך ביותר. אם יש כמה עמדות כאלה, המתקשר יתווסף לראשונה מבניהן.

ג. כתבו את כותרת המחלקה Support ואת התכונות שלה.

(3 נקודות)

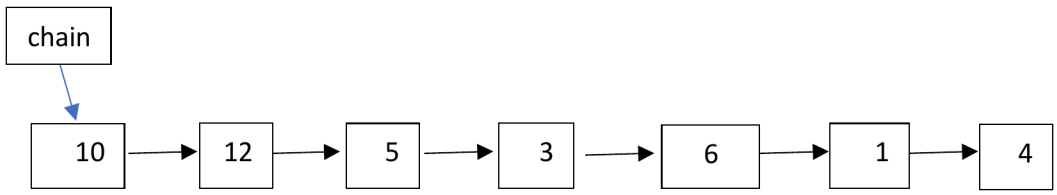
ד. כתבו את הפעולה startSupport() (ב-StartSupport :C#()).

(3 נקודות)

ה. הנהלת החברה החליטה לציין בסוף כל יום עבודה את פקיד/ת השרות שטיפל/ה בהכי הרבה לקוחות. ציינו שינוים הנדרשים במחלקה/מחלקות וכתבו במחלקה Support פעולה theBest() (ב-TheBest :C#()) המחזירה את מספר העמדה של הפקיד הטוב ביותר (אפשר להניח שיש רק פקיד אחד).

(4 נקודות)

12. נתונה השרשרת הבאה: $10 \rightarrow 12 \rightarrow 5 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 4$



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

א. עקבו אחרי זימון הפעולה הרקורסיבית one(chain) הבאה וכתבו איך תיראה השרשרת אחרי ביצוע הפעולה ? כמו כן, כתבו מה מבצעת הפעולה one באופן כללי?

יש להראות מעקב אחרי ביצוע הפעולה!

```
public static void one(Node<Integer> n){
  if(n.hasNext())
  {
    n.setValue(n.getNext().getValue());
    one(n.getNext());
  }
  else
  {
    n.setValue(0);
  }
}
```

```
public static void One(Node<int> n){
  if(n.HasNext())
  {
    n.SetValue(n.GetNext().GetValue());
    One(n.GetNext());
  }
  else
  {
    n.SetValue(0);
  }
}
```

(6 נקודות)

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

ב. נתונות הפעולה הרקורסיבית two הבאה, עקבו אחרי זימון הפעולה two(chain,10) ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה? רשמו כמו כן מה מבצעת הפעולה באופן כללי?

יש להראות מעקב אחרי ביצוע הפעולה!

```
public static void two(Node<Integer> n, int num){
  if(n.hasNext())
  {
    two(n.getNext(), num);
  }
  else
  {
    n.setValue(num);
  }
}
```

```
public static void Two(Node<int> n, int num){
  if(n.HasNext())
  {
    Two(n.GetNext(), num);
  }
  else
  {
    n.SetValue(num);
  }
}
```

(6 נקודות)

ג. נתונה הפעולה three הבאה, מה יהיה תוכן השרשרת chain אחרי זימון הפעולה three(chain)? כתבו מה מבצעת הפעולה באופן כללי?

```
public static void three(Node<Integer>n){
  int x = n.getValue();
  one(n);
  two(n,x);
}
```

```
public static void Three(Node<int>n){
  int x = n.GetValue();
  One(n);
  Two(n,x);
}
```

(4 נקודות)

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

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