GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2025 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-C# (עמוד 2) מבחן ב-JAVA (עמוד 14) בכל מבחן עשר שאלות. **חלק א' – 32 נקודות** שאלות 1–3: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 16 נקודות. **חלק ב' – 32 נקודות** שאלות 4–6: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 16 נקודות. **חלק ג' – 36 נקודות** שאלות 7–9: יש לענות על **שתי** שאלות בלבד. ערך כל השאלה 18 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 המבחן, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת המבחן). 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!** **הנחיות כלליות לנבחנים:** בנספחי הבחינה מובאים ממשקים (interface) הכוללים פונקציות ומחלקות עזר, יש להקפיד ולהשתמש בהן בשאלות הרלוונטיות בבחינה על-מנת להימנע מאיבוד ניקוד.

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

בהצלחה!

השאלות

1. הגדרה:

מחסנית של מספרים שלמים וחיוביים נקראת "מחסנית תקינה עבור הספרה K", אם יש לפחות מספר אחד שמתחיל בספרה K **וכל** המספרים שמתחילים ב-K נמצאים ברצף ולא נמצאים באף מיקום אחר במחסנית.

לדוגמה: המחסנית שלפניכם היא "מחסנית תקינה" עבור הספרות K=2,5,9 אבל לא תקינה עבור ספרות אחרות.

31
307
2
2215
3
900
987
91
40
37
4
555
41

א. כתבו פעולה המקבלת מחסנית של מספרים שלמים וחיוביים `st` וספרה `k`. הפעולה תבדוק אם מחסנית היא "מחסנית תקינה" עבור ספרה זו. אם כן - הפעולה תחזיר ערך `true`, ואם לא - הפעולה תחזיר ערך `false`.

כותרת הפעולה:

```
public static boolean isProperK(Stack<Integer> st, int k)
```

```
public static bool IsProperK(Stack<int> st, int k)
```

(8 נקודות)

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

ב. כתבו פעולה המקבלת מחסנית של מספרים שלמים `st` וספרה `k`. הפעולה תבדוק אם המחסנית היא "**מחסנית תקינה עבור k**". אם כן - הפעולה לא תבצע דבר, ואם לא - הפעולה תשנה את סדר המספרים במחסנית כך שהיא תהיה "**תקינה עבור k**".
אפשר להניח שבמחסנית יש לפחות מספר אחד אשר מתחיל ב-`k`.
כותרת הפעולה:

```
public static void fixIt(Stack<Integer> st, int k)
```

```
public static void FixIt(Stack<int> st, int k)
```

(6 נקודות)

ג. מהי סיבוכיות הפעולות שכתבתם בסעיפים א'-ב'? **הסבירו את תשובתכם.**
(2 נקודות)

2. באוניברסיטה מפתחים מערכת למעקב אחר כוכבים חדשים שמתגלים מדי יום. מספר הכוכבים הוא גדול מאוד, לכן הוחלט לייצג את המספר באמצעות שרשרת חוליות של ספרות. סדר החוליות הוא לפי סדר הופעת הספרות במספר, כך שספרת האחדות היא החוליה האחרונה בשרשרת.
לדוגמה:

- השרשרת $8 \to 5 \to 2$ מייצגת את המספר **258**.
- השרשרת $5 \to 4 \to 9 \to 0 \to 2 \to 1$ מייצגת את המספר **120,945**.

המחלקה **StarCount** אחראית על ספירת הכוכבים ביקום.

```
public class StarCount
{
    private Node<Integer> number;
}
```

```
public class StarCount
{
    private Node<int> number;
}
```

הערך ההתחלתי של מספר הכוכבים ביקום הוזן באופן ידני ובגלל טעויות הקלדה בחלק מהחוליות בשרשרת `number` הופיעו מספרים שאינם חד-ספרתיים.

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

א. כתבו פעולה פנימית המתקנת את שרשרת החוליות `number` כך שכל החוליות יכילו ספרה אחת.

חוליה המכילה מספר דו-ספרתי תפוצל לשתי חוליות, חוליה מכילה מספר תלת-ספרתי תפוצל לשלוש חוליות וכן הלאה.

לדוגמה:

השרשרת: $23 \to 5 \to 174 \to 6$

תהפוך לשרשרת: $2 \to 3 \to 5 \to 1 \to 7 \to 4 \to 6$

כותרת הפעולה:

```
public void fixNumber()
```

```
public void FixNumber()
```

(6 נקודות)

ב. בכל יום מתגלים כוכבים חדשים על ידי טלסקופים הפזורים במקומות שונים בכדור הארץ.

כדי לשמור על הנתונים מעודכנים, נדרש לאפשר ביצוע של פעולות חשבון על המספרים המיוצגים על ידי שרשרת החוליות.

הפעולה הבסיסית ביותר הנדרשת היא `addOne` - פעולה המגדילה ב-1 את מספר הכוכבים.

לדוגמה:

שרשרת החוליות לפני הפעלת הפעולה addOne	שרשרת החוליות אחרי הפעלת הפעולה addOne
$1 \to 2 \to 8$	$1 \to 2 \to 9$
$4 \to 7 \to 5 \to 9$	$4 \to 7 \to 6 \to 0$
$9 \to 9$	$1 \to 0 \to 0$

ממשו את הפעולה הפנימית **addOne** המגדילה את מספר הכוכבים ב-1.

שימו לב! מספר הכוכבים הוא עצום, הרבה מעבר ליכולת הייצוג של טיפוסי נתונים רגילים כמו double, int או long ולכן אין לחבר את חוליות השרשרת כדי ליצור מספר שלם.

(8 נקודות)

ג. מהי סיבוכיות הפעולה `addOne` שכתבתם בסעיף ב'? הסבירו את תשובתכם.

(2 נקודות)

3. נתון פרויקט שבו הוגדרו מחלקות `V, W, X, Y, Z`.

הפעולה `stam()` (ב-`Stam-C#:`()`) מומשה **בשתי** מחלקות.

במחלקה נוספת באותו פרויקט, בשם `Tester`, נכתב קטע הקוד הבא בפעולה `main`, והוא **תקין**:

```
V v1 = new V(9);
V v2 = new Y("9");
V v3 = new Z(new V(6));
W w1 = new W(z1, v2);
Y y1 = new Y("5");
Z z1 = new W(4, 1.1, 'Z', true);
Z z2 = new Z();
Z z3 = new X();
z2.stam();  // תקין ומדפיס "Good"
z3.stam();  // תקין ומדפיס "Good"
w1.stam();  // תקין ומדפיס "Bad"
```

```
V v1 = new V(9);
V v2 = new Y("9");
V v3 = new Z(new V(6));
W w1 = new W(z1, v2);
Y y1 = new Y("5");
Z z1 = new W(4, 1.1, 'Z', true);
Z z2 = new Z();
Z z3 = new X();
z2.Stam();  // תקין ומדפיס "Good"
z3.Stam();  // תקין ומדפיס "Good"
w1.Stam();  // תקין ומדפיס "Bad"
```

חמש הפקודות האלה גורמות לשגיאות:

```
X x1 = new W();     // שגיאת קומפילציה
v1.stam();          // שגיאת קומפילציה
y1.stam();          // שגיאת קומפילציה
((Z)y1).stam();     // שגיאת קומפילציה
((Z)v2).stam();     // שגיאת זמן ריצה
```

```
X x1 = new W();     // שגיאת קומפילציה
v1.Stam();          // שגיאת קומפילציה
y1.Stam();          // שגיאת קומפילציה
((Z)y1).Stam();     // שגיאת קומפילציה
((Z)v2).Stam();     // שגיאת זמן ריצה
```

א. שרטטו תרשים היררכיה UML המתאר עץ ירושה אפשרי עבור מחלקות הפרויקט.

(8 נקודות)

ב. ציינו באילו מחלקות מומשה הפעולה `stam()`.

(2 נקודות)

ג. כתבו את **כותרות** כל המחלקות ו**כותרות** הפעולות הבונות (הבנאים) הנדרשות.

אין צורך לממש את הבנאים.

(6 נקודות)

4. במערכת לניהול מסעדה יש כמה מחלקות:

לקוח **Client** המאופיין בשם (`String name`) ומספר סועדים (`int diners`),

שולחן **Table** המאופיין במספר שולחן (`int num`), מספר מקומות בשולחן (`int places`) ומספר מקומות פנויים (`int free`).

מספר המקומות בשולחן (גודל השולחן) הוא אחד משלוש אפשרויות: 2 (שולחנות קטנים), 4 (שולחנות בינוניים) ו-8 (שולחנות גדולים).

בשתי המחלקות קיימות הפעולות הבונות המקבלות את ערכי התכונות ופעולות `get/set` (ב-C#: ``Get/Set``) לכל תכונה.

המחלקה `Restaurant` מאופיינת **בתור** של לקוחות **וברשימה** מקושרת של שולחנות.

```
class Restaurant{
 private Queue<Client> clients;

 private Node<Table> tables;
 ….
}
```

```
class Restaurant{
 private Queue<Client> clients;

 private Node<Table> tables;
 ….
}
```


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

א. כתבו במחלקה `Restaurant` את הפעולה הבונה המקבלת את מספר השולחנות הקטנים, את מספר השולחנות הבינוניים ואת מספר השולחנות הגדולים שיש במסעדה.
הפעולה הבונה מייצרת מסעדה חדשה עם תור הלקוחות ריק, ובה שולחנות בכמות שהועברה כפרמטר. סדר הכנסת השולחנות לרשימה הוא לבחירתכם.
כותרת הפעולה:

```
public Restaurant(int small, int medium, int large)
```

```
public Restaurant(int small, int medium, int large)
```

(4 נקודות)

ב. כתבו פעולה במחלקה `Restaurant` המקבלת מספר סועדים ומחזירה את מספר השולחן שיש בו מספיק מקומות הפנויים. אם אין שולחן כזה, הפעולה תחזיר ערך (-1).
כותרת הפעולה:

```
public int findAvailableTable(int numOfDiners)
```

```
public int FindAvailableTable(int numOfDiners)
```

(6 נקודות)

ג. כתבו פעולה במחלקה `Restaurant` המטפלת בהושבת לקוח בשולחן. הפעולה תושיב לקוח **אחד** מתוך הלקוחות הממתינים בתור, בשולחן שבו יש מספיק מקומות פנויים, תוך שמירה על סדר התור - לקוח מתאים שהגיע קודם, יטופל קודם.
אם יימצא שולחן מתאים, יש להוציא את הלקוח מהתור, לעדכן את מספר מקומות הפנויים בשולחן ולהחזיר `true`. אם הפעולה לא תצליח להושיב אף לקוח מהלקוחות בתור, הפעולה תחזיר `false`.
כותרת הפעולה:

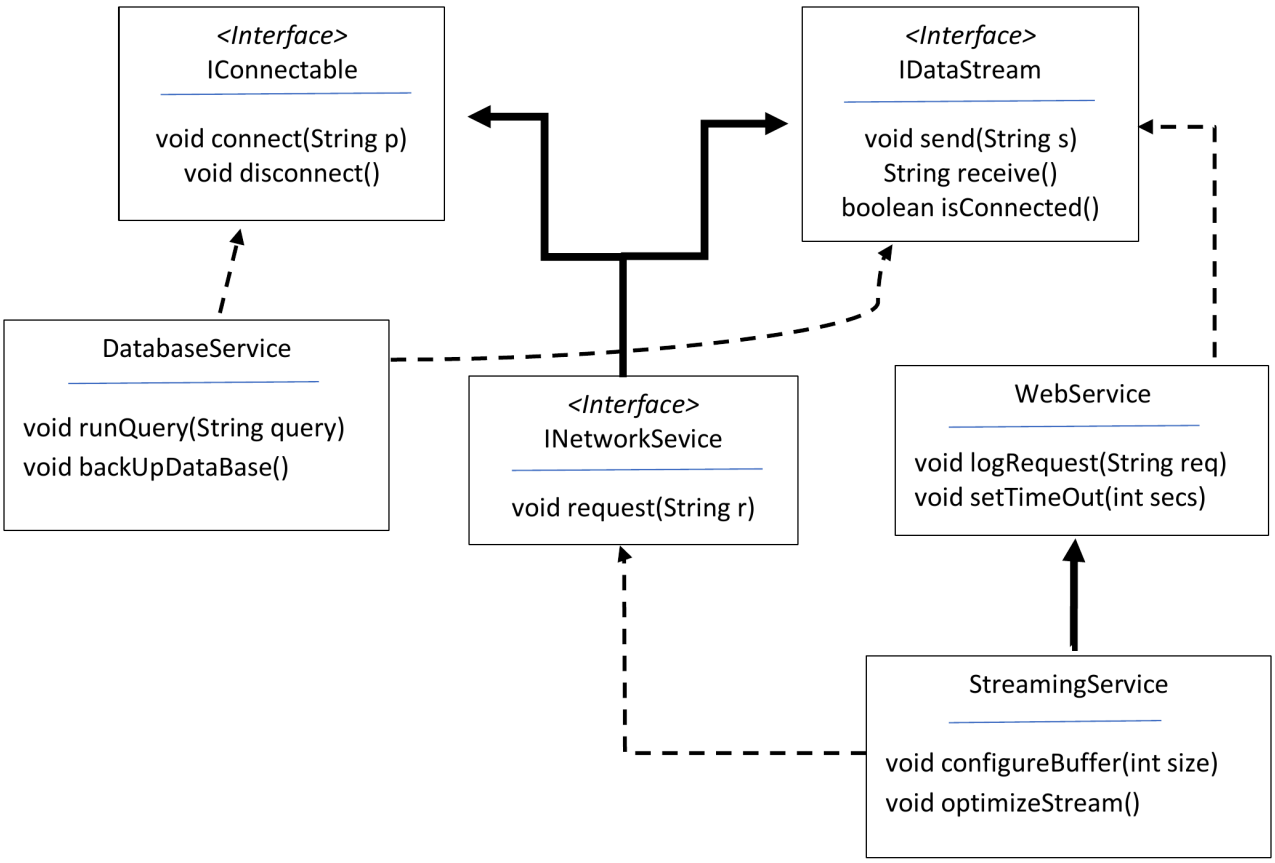
```
public boolean seatNextClient()
```

```
public bool SeatNextClient()
```

(6 נקודות)

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

5. חברת סטארטאפ מפתחת מערכת לניהול שירותי רשת. המערכת כוללת ממשקים ומחלקות המייצגים סוגים שונים של שירותים ויכולותיהם. לפניכם תרשים היררכיה המתאר את הקשר בין מחלקות וממשקים:



קשר ירושה	
קשר מימוש	

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

נתונים 12 קטעי קוד. ציינו עבור כל אחד מהם אם הוא תקין או שגוי. אם הקוד שגוי, יש לפרט ולנמק מדוע ולציין מה סוג השגיאה - שגיאת קומפילציה או שגיאת זמן ריצה. אין קשר בין הקטעים!

```
// 1
IConnectable c1 = new WebService();
c1.connect("api.com");
c1.send("hello");
// 2
IDataStream d1 = new StreamingService();
d1.connect("stream.com");
// 3
StreamingService s1 = new StreamingService();
s1.connect("video.com");
s1.request("play video");
s1.optimizeStream();
// 4
INetworkService n1 = new StreamingService();
n1.send("start stream");
n1.configureBuffer(1024);
// 5
WebService w1 = new StreamingService();
w1.receive();
w1.logRequest("get status");
// 6
IConnectable c2 = new StreamingService();
c2.disconnect();
c2.send("ping");
// 7
IConnectable c3 = new DatabaseService();
c3.connect("db.com");
c3.runQuery("SELECT * FROM users;");
((DatabaseService) c3).backupDatabase();
System.out.println(((DatabaseService) c3).receive());
// 8
INetworkService n2 = new DatabaseService();
n2.connect("db2.com");
// 9
IConnectable c4 = new WebService();
((DatabaseService) c4).runQuery("DROP TABLE posts;");
// 10
StreamingService s2 = new WebService();
// 11
INetworkService n3 = new WebService();
// 12
IConnectable[] connections = new IConnectable[2];
```


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

```
connections[0] = new WebService();
connections[1] = new DatabaseService();
((StreamingService) connections[0]).optimizeStream();
```

```
// 1
IConnectable c1 = new WebService();
c1.Connect("api.com");
c1.Send("hello");
// 2
IDataStream d1 = new StreamingService();
d1.Connect("stream.com");
// 3
StreamingService s1 = new StreamingService();
s1.Connect("video.com");
s1.Request("play video");
s1.OptimizeStream();
// 4
INetworkService n1 = new StreamingService();
n1.Send("start stream");
n1.ConfigureBuffer(1024);
// 5
WebService w1 = new StreamingService();
w1.Receive();
w1.LogRequest("get status");
// 6
IConnectable c2 = new StreamingService();
c2.Disconnect();
c2.Send("ping");
// 7
IConnectable c3 = new DatabaseService();
c3.Connect("db.com");
c3.RunQuery("SELECT * FROM users;");
((DatabaseService) c3).BackupDatabase();
Console.WriteLine(((DatabaseService) c3).Receive());
// 8
INetworkService n2 = new DatabaseService();
n2.Connect("db2.com");
// 9
IConnectable c4 = new WebService();
((DatabaseService) c4).RunQuery("DROP TABLE posts;");
// 10
StreamingService s2 = new WebService();
// 11
INetworkService n3 = new WebService();
// 12
IConnectable[] connections = new IConnectable[2];
```

```
connections[0] = new WebService();
connections[1] = new DatabaseService();
((StreamingService) connections[0]).OptimizeStream();
```

6. בשאלה זו אסור להשתמש במבנים נוספים כגון תור, מחסנית או מערך וגם אסור ליצור חוליות נוספות

נתונה המחלקה `CharList` שלפניכם המייצגת רשימת תווים:

```
public class CharList {
  private Node<Character> head;
  public CharList(){
   this.head = null;
  }
  public CharList(Node<Character> chain){
   this.head = chain;
  }
}
```

```
public class CharList {
  private Node<char> head;
  public CharList(){
   this.head = null;
  }
  public CharList(Node<char> chain){
   this.head = chain;
  }
}
```

במחלקה CharList הוגדרה פעולה המקבלת הפניה לחוליה כלשהי `chain` ותו `letter`. הפעולה תחזיר את הפנייה לחוליה הראשונה הנמצאת אחרי `chain` ומכילה את התו `letter`. אם התו אינו מופיע ברשימה אחרי ה-`chain`, הפעולה תחזיר `null`. הפעולה מוכנה ואין צורך לממש אותה!

כותרת הפעולה:

```
public Node<Character> firstAfterChain(Node<Character> chain,
                                       char letter)
```

```
public Node<char> FirstAfterChain(Node<char> chain, char letter)
```


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

א. כתבו פעולה המקבלת תו `letter`. הפעולה תחזיר את הפנייה **לחוליה האחרונה** המכילה את התו `letter`. אם התו לא מופיע ברשימה, הפעולה תחזיר `null`. כותרת הפעולה:

```
public Node<Character> last (char letter)
```

```
public Node<char> Last (char letter)
```

חובה להשתמש בפעולה firstAfterChain.

(6 נקודות)

ב. כתבו פעולה המקבלת תו `letter` המופיע **לפחות פעמיים** במקומות כלשהם ברשימה, **אך לא בתחילתה ולא בסופה**.

הפעולה תחליף בין תת-השרשרת הנמצאת לפני המופע הראשון של `letter` לבין תת-השרשרת הנמצאת אחרי המופע האחרון של התו `letter`.

לדוגמה, עבור הרשימה שלפניכם והתו `%`:

`k → a → % → b → b → % → b → % → e → r → r`

אחרי זימון הפעולה הרשימה תיראה כך:

`e → r → r → % → b → b → % → b → % → k → a`

כותרת הפעולה:

```
public void swap(char letter)
```

```
public void Swap(char letter)
```

חובה להשתמש בפעולות firstAfterChain ו- last.

(8 נקודות)

ד. מהי סיבוכיות זמן הריצה של הפעולה `swap`? הסבירו את תשובתכם.

(2 נקודות)

7.

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

א. נתונה הפעולה `why` המקבלת הפניה לשורש עץ בינארי של מספרים שלמים:

```
public static int why(BinNode<Integer> t){
  if(!t.hasLeft()&&!t.hasRight()) return t.getValue();
  if(t.hasLeft()) return why(t.getLeft());
  return why(t.getRight());
}
```

```
public static int Why(BinNode<int> t){
  if(!t.HasLeft()&&!t.HasRight()) return t.GetValue();
  if(t.HasLeft()) return Why(t.GetLeft());
  return Why(t.GetRight());
}
```

ציירו עץ בעל **שישה** צמתים שעבורו הפעולה תחזיר ערך **ארבע**. כל ערכי הצמתים שונים זה מזה.

(4 נקודות)

ב. כתבו את הפעולה `why` בצורה הלא רקורסיבית.

(4 נקודות)

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

נתונה הפעולה **what** המקבלת מספר שלם וחיובי `n`. הפעולה משתמשת בפעולה `why`.

```
public static BinNode<Integer> what (int n){
 if (n == 0)
   return null;
 else
 {
  if (n % 2 == 0)
    return new BinNode<Integer>(what(n-1), n, null);
  else
  {
    BinNode<Integer> tree = what(n-1);
    if(tree!=null)
    {
      int val = why(tree);
      tree.setRight(new BinNode<Integer>(n + val));
    }
    return tree;
  }
 }
}
```

```
public static BinNode<int> What (int n){
 if (n == 0)
   return null;
 else
 {
  if (n % 2 == 0)
    return new BinNode<int>(What(n-1), n, null);
  else
  {
    BinNode<int> tree = What(n-1);
    if(tree!=null)
    {
      int val = Why(tree);
      tree.SetRight(new BinNode<int>(n + val));
    }
    return tree;
  }
 }
}
```

ג. עקבו אחרי זימון הפעולה `what (7)` ורשמו מה תחזיר הפעולה.

יש להראות את תוכן העץ בחזרה מכל קריאה רקורסיבית.

(10 נקודות)

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

8. בפרויקט מוגדרות שש מחלקות: A, B, C, D, E ו-Test. כל אחת בקובץ נפרד.
להלן מתוארות המחלקות עם החתימות של הפעולות שיש בהן. המימוש עצמו אינו חשוב.

שימו לב שלצד כל פעולה כתוב בהערה מספר. זהו מספר הפעולה שאליו מתייחסים בהמשך.

```
public class A
{
  protected int x;
  protected int y;

  public A(){
    x = 0;
    y = 1;
  }

  public A(int a){
    this(a, a);
  }

  public A(int a,int b){
    y = a*b;
  }

  public void func(D d){} // 1
  public void func(B b){} // 2
} //end of class A

public class B extends A
{
  protected int z;

  public B(int b){
    super(b+1);
    x = b;
    z = b*2;
  }

  public void func(A a){} // 3
  public void func(B b){} // 4
} //end of class B

public class C extends A
{
  protected int t;
  public void func(A a){}// 5
```


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

```
  public void func(B b){}// 6
} //end of class C

public class D extends B
{
  public D(int d){
   super(d+2);
  }
  public void func(A a){}// 7
  public void func(B b){}// 8
} //end of class D

public class E extends C
{
  protected int w;

  public E (int num){
   super();
   x = num;
   w = x + y;
  }
public void func(A a){}// 9
} //end of class E

public class Test {
   public static void main (String []args)
   {
      A ab =new B(2);
      A ae =new E(3);
      B bd =new D(2);
      C cc =new C();
      D dd =new D(4);
      // ..... //
   }
} //end of class Test
```

```
public class A
{
  protected int x;
  protected int y;

  public A() {
   x = 0;
   y = 1;
  }
```


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

```
  public A(int a) : this(a, a){
  }

  public A(int a, int b){
   y = a * b;
  }

 public virtual void Func(D d){
 } // 1
 public virtual void Func(B b){
 } // 2
} //end of class A

public class B : A
{
  protected int z;
  public B(int b) : base(b + 1){
    x = b;
    z = b * 2;
  }

 public virtual void Func(A a){
 } // 3
 public override void Func(B b){
 } // 4
} //end of class B

public class C : A
{
 protected int t;
 public virtual void Func(A a){
 } // 5
 public override void Func(B b){
 } // 6
} //end of class C

public class D : B
{
 public D(int d): base(d+2){
 }

 public override void Func(A a) {
 } //7
 public override void Func(B b) {
 } //8
} //end of class D
```

```
public class E :C
{
 protected int w;

 public E(int num):base(){
    x = num;
    w = x + y;
 }

 public override void Func(A a){
 } // 9
} //end of class E

public class Test {
   public static void Main (string []args)
   {
      A ab =new B(2);
      A ae =new E(3);
      B bd =new D(2);
      C cc =new C();
      D dd =new D(4);
      // ..... //
   }
} //end of class Test
```

א. **שרטטו את עץ הירושה למחלקות המוגדרות בפרויקט.**

(2 נקודות)

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

ב. לפניכם טבלה ובה כל האובייקטים שהוגדרו בשיטה `main` במחלקה `Test` לעיל. כל אובייקט בשורה נפרדת. בעמודות של הטבלה רשומים השמות של התכונות שיש במחלקות שמהן נוצרו האובייקטים.

עליכם למלא את הטבלה ולכתוב בכל תא מהו הערך של התכונה שבעמודה באובייקט שבשורה. שימו לב שלא תמיד כל התאים יתמלאו. בתא שלא קיימת התכונה במחלקה, כתבו "**אין**".

להדגמה, מילאנו את הערכים המתאימים בחלק מהתאים. **עליכם להשלים את כל שאר התאים ולהעתיק את הטבלה המלאה לחוברת הבחינה.**

אובייקט	x	y	z	t	w
`A ab =new B(2);`	2	9	4	אין	אין
`A ae =new E(3);`	3			0	
`B bd =new D(2);`					
`C cc =new C();`	0				
`D dd =new D(4);`		49			

(8 נקודות)

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

ג. בכל אחד מסעיפים הבאים השורה (*) תוחלף בשורת קוד משורות הקוד 1-8. כתבו, עבור כל אחד מהסעיפים, אם הקוד תקין או לא. אם הקוד אינו תקין - יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (קומפילציה או זמן ריצה).
אם הקוד תקין, כתבו מהי מספר הפעולה `func` שתופעל.

שימו לב שאין קשר בין סעיפים!

```
1. ab.func(bd);
2. ((E)ae).func (ab);
3. ((B)dd).func (bd);
4. ((A)bd).func (ae);
5. ((E)cc).func (bd);
6. bd.func(ae);
7. cc.func(cc);
8. ((D)ab).func(dd);
```

```
1. ab.Func(bd);
2. ((E)ae).Func (ab);
3. ((B)dd).Func (bd);
4. ((A)bd).Func (ae);
5. ((E)cc).Func (bd);
6. bd.Func(ae);
7. cc.Func(cc);
8. ((D)ab).Func(dd);
```

(8 נקודות)

9. נתונות הפעולות `f1, f2` (ב-`F1, F2`): (C#)

```
public static boolean f1(Node<Integer> n, Queue<Integer> q){
 if (q.isEmpty())return true;
 if (n == null) return false;
 if (n.getValue() != q.remove())return false;
 return f1(n.getNext(), q);
}
public static Queue<Integer>f2(Node<Integer> n1,
                    Node<Integer> n2) {
 if (n2 == null) {
  return new Queue<Integer>();
 }
 if (!n2.hasNext()) {
  Queue<Integer> q = new Queue<Integer>();
  q.insert(n1.getValue());
  return q;
 }
 Queue<Integer>q = f2(n1.getNext(),n2.getNext().getNext());
 q.insert(n1.getValue());
 return q;
}
```

```
public static bool F1(Node<int> n, Queue<int> q){
 if (q.IsEmpty())return true;
 if (n == null) return false;
 if (n.GetValue() != q.Remove())return false;
 return F1(n.GetNext(), q);
}
public static Queue<int>F2(Node<int> n1,
                  Node<int> n2) {
 if (n2 == null) {
  return new Queue<int>();
 }
 if (!n2.HasNext()) {
  Queue<int> single = new Queue<int>();
  single.Insert(n1.GetValue());
  return single;
 }
 Queue<int>q = F2(n1.GetNext(),n2.GetNext().GetNext());
 q.Insert(n1.GetValue());
 return q;
}
```


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

א. נתונה שרשרת חוליות $3 \to 4 \to 5 \to 6 \to 7 \to 8 \to 9 \to n$.

תנו דוגמה לתור `q` הכולל לפחות ארבעה איברים כך שתוצאת זימון `f1(n, q)` תהיה `true`.

(4 נקודות)

ב. מהי מטרת הפעולה `f1` באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.

(3 נקודות)

ג. נתונה שרשרת חוליות $6 \to 5 \to 4 \to 3 \to 2 \to 1 \to n$.

מה תהיה תוצאת הזימון `f2(n, n)`? **יש להראות מעקב אחרי ביצוע הפעולה!**

(4 נקודות)

ד. נתונה שרשרת חוליות $7 \to 6 \to 5 \to 4 \to 3 \to 2 \to 1 \to n$.

מה תהיה תוצאת הזימון `f2(n,n)`?

(4 נקודות)

ה. מהי מטרת הפעולה `f2` באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.

(3 נקודות)

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