

מועד הבחינה:
קיץ תשפ"ה – 2025 – מועד ב'
מספר השאלון: 97105
נספח ממשקים לבחינה JAVA
נספח ממשקים לבחינה C#
מילון עזר

מבני נתונים ותכנות מונחה עצמים

הנדסאים וטכנאים – הנדסת תוכנה

הנחיות לבחינה

מספר ת"ז _____
מספר מחברת _____

- א. משך הבחינה: ארבע שעות וחצי.
- ב. מבנה השאלון בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים: ומפתח ההערכה: 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
1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית).
 2. מותר לשימוש: קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר.
אין לצרף ספרים או חוברות עם פתרונות.
 - ד. הוראות כלליות:
 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!

בהצלחה!

מבחן ב-C#

הנחיות כלליות לנבחנים:

בנספחי הבחינה מובאים ממשקים (interface) הכוללים פונקציות ומחלקות עזר, יש להקפיד ולהשתמש בהן בשאלות הרלוונטיות בבחינה על-מנת להימנע מאיבוד ניקוד.

חלק א'

ענו על שתיים מבין השאלות 1–3 (ערך כל שאלה – 16 נקודות).

שאלה 1

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

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.

(8 נק') 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 bool IsProperK(Stack<int> st, int k)
```

(6 נק') 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<int> st, int k)
```

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

שאלה 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 החולייה האחרונה בשרשרת.

לדוגמה :

- השרשרת $5 \rightarrow 8 \rightarrow 2$ מייצגת את המספר 258.
 - השרשרת $5 \rightarrow 4 \rightarrow 9 \rightarrow 0 \rightarrow 2 \rightarrow 1$ מייצגת את המספר 120,945.
- המחלקה StarCount אחראית על ספירת הכוכבים ביקום.

```
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.

(6 נק') 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: $6 \rightarrow 174 \rightarrow 5 \rightarrow 23$

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

כותרת הפעולה :

```
public void FixNumber()
```

(8 נק') 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
$8 \rightarrow 2 \rightarrow 1$	$9 \rightarrow 2 \rightarrow 1$
$9 \rightarrow 5 \rightarrow 7 \rightarrow 4$	$0 \rightarrow 6 \rightarrow 7 \rightarrow 4$
$9 \rightarrow 9$	$0 \rightarrow 0 \rightarrow 1$

ממשו את הפעולה הפנימית AddOne המגדילה את מספר הכוכבים ב-1.

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

(2 נק') ג. מהי סיבוכיות הפעולה AddOne שכתבתם בסעיף ב'? הסבירו את תשובתכם.

שאלה 3

נתון פרויקט שבו הוגדרו מחלקות V, W, X, Y, Z. הפעולה Stam() 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(); // "Good" תקין ומדפיס
z3.Stam(); // "Good" תקין ומדפיס
w1.Stam(); // "Bad" תקין ומדפיס
```

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
```

(8 נק') א. שרטטו תרשים היררכיה UML המתאר עץ ירושה אפשרי עבור מחלקות הפרויקט.

(2 נק') ב. ציינו באילו מחלקות מומשה הפעולה Stam().

(6 נק') ג. כתבו את כותרות כל המחלקות וכותרות הפעולות הבונות (הבנאים) הנדרשות.

אין צורך לממש את הבנאים.

חלק ב'

ענו על שתיים מבין השאלות 4–6 (ערך כל שאלה – 16 נקודות).

שאלה 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 לכל תכונה.

המחלקה Restaurant מאופיינת בתור של לקוחות וברשימה מקושרת של שולחנות.

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

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

(4 נק') 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)
```

(6 נק') 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 numDiners)
```

(6 נק') 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 ימצא שולחן מתאים, יש להוציא את הלקוח מהתור, לעדכן את מספר מקומות הפנויים

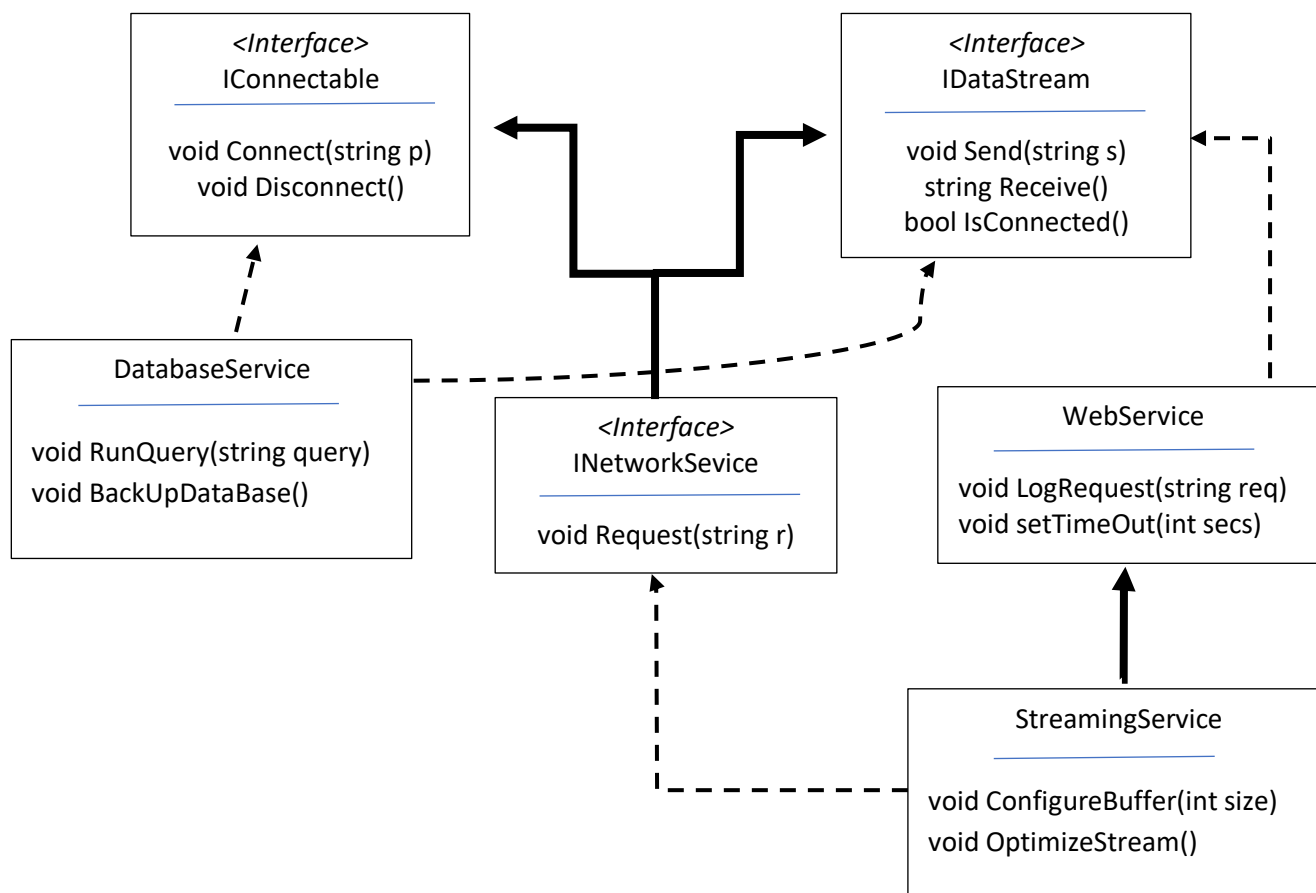
בשולחן ולהחזיר true. אם הפעולה לא תצליח להושיב אף לקוח מהלקוחות בתור, הפעולה

תחזיר false.

כותרת הפעולה:

```
public bool SeatNextClient()
```

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:



<u>קשר ירושה</u>	
<u>קשר מימוש</u>	

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	<code>IConnectable c1 = new WebService(); c1.Connect("api.com"); c1.Send("hello");</code>
2	<code>IDataStream d1 = new StreamingService(); d1.Connect("stream.com");</code>
3	<code>StreamingService s1 = new StreamingService(); s1.Connect("video.com"); s1.Request("play video"); s1.OptimizeStream();</code>
4	<code>INetworkService n1 = new StreamingService(); n1.Send("start stream"); n1.ConfigureBuffer(1024);</code>
5	<code>WebService w1 = new StreamingService(); w1.Receive(); w1.LogRequest("get status");</code>
6	<code>IConnectable c2 = new StreamingService(); c2.Disconnect(); c2.Send("ping");</code>
7	<code>IConnectable c3 = new DatabaseService(); c3.Connect("db.com"); c3.RunQuery("SELECT * FROM users;"); ((DatabaseService) c3).BackupDatabase(); Console.WriteLine(((DatabaseService) c3).Receive());</code>
8	<code>INetworkService n2 = new DatabaseService(); n2.Connect("db2.com");</code>
9	<code>IConnectable c4 = new WebService(); ((DatabaseService) c4).RunQuery("DROP TABLE posts;");</code>
10	<code>StreamingService s2 = new WebService();</code>
11	<code>INetworkService n3 = new WebService();</code>
12	<code>IConnectable[] connections = new IConnectable[2]; connections[0] = new WebService(); connections[1] = new DatabaseService(); ((StreamingService) connections[0]).OptimizeStream();</code>

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

(6 נק') 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<char> Last (char letter)
```

חובה להשתמש בפעולה FirstAfterChain.

(8 נק') ב. כתבו פעולה המקבלת תו letter המופיע לפחות פעמיים במקומות כלשהם ברשימה, אך לא

בתחילתה ולא בסופה.

הפעולה תחליף בין תת-השרשרת הנמצאת לפי המופע הראשון של letter לבין תת-השרשרת

הנמצאת אחרי המופע האחרון של 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)
```

חובה להשתמש בפעולות FirstAfterChain ו Last.

(2 נק') ד. מהי סיבוכיות זמן הריצה של הפעולה Swap? הסבירו את תשובתכם.

חלק ג'

ענו על שתיים מבין השאלות 7–9 (ערך שאלה – 18 נקודות).

שאלה 7

(4 נק') א. נתונה הפעולה Why המקבלת הפניה לשורש עץ בינארי של מספרים שלמים:

```
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 נק') ב. כתבו את הפעולה Why בצורה הלא רקורסיבית.

נתונה הפעולה What המקבלת מספר שלם וחיובי n. הפעולה משתמשת בפעולה Why.

```
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;
        }
    }
}
```

(10 נק') ג. עקבו אחרי זימון הפעולה What (7) ורשמו מה תחזיר הפעולה.

יש להראות את תוכן העץ בחזרה מכל קריאה רקורסיבית.

שאלה 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.

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

(2 נק') א. שרטטו את עץ הירושה למחלקות המוגדרות בפרויקט.

(8 נק') 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.

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

(8 נק') 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 מהו 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) ;

שאלה 9

נתונות הפעולות F1, F2:

```

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> q = new Queue<int>();
        q.insert(n1.GetValue());
        return q;
    }
    Queue<int>q = f2(n1.GetNext(),n2.GetNext().GetNext());
    q.Insert(n1.GetValue());
    return q;
}

```

(4 נק') א. נתונה שרשרת חוליות $n \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3$.

תנו דוגמה לתור q הכולל לפחות ארבעה איברים כך שתוצאת זימון $F1(n, q)$ תהיה true.

(3 נק') ב. מהי מטרת הפעולה F1 באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.

(4 נק') ג. נתונה שרשרת חוליות $n \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$.מה תהיה תוצאת הזימון $F2(n, n)$ יש להראות מעקב אחרי ביצוע הפעולה!(4 נק') ד. נתונה שרשרת חוליות $n \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$.מה תהיה תוצאת הזימון $F2(n, n)$

(3 נק') ה. מהי מטרת הפעולה F2 באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.

מבחן ב-JAVA

הנחיות כלליות לנבחנים:

בנספחי הבחינה מובאים ממשקים (interface) הכוללים פונקציות ומחלקות עזר, יש להקפיד ולהשתמש בהן בשאלות הרלוונטיות בבחינה על-מנת להימנע מאיבוד ניקוד.

מבחן ב-JAVA

חלק א'

ענו על שתיים מבין השאלות 1–3 (ערך כל שאלה – 16 נקודות).

שאלה 1

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

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.

(8 נק') 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)
```

(6 נק') 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)
```

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

שאלה 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 החולייה האחרונה בשרשרת.

לדוגמה:

- השרשרת $5 \rightarrow 8 \rightarrow 2$ מייצגת את המספר 258.
 - השרשרת $5 \rightarrow 4 \rightarrow 9 \rightarrow 0 \rightarrow 2 \rightarrow 1$ מייצגת את המספר 120,945.
- המחלקה StarCount אחראית על ספירת הכוכבים ביקום.

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

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

(6 נק') 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: $6 \rightarrow 174 \rightarrow 5 \rightarrow 23$

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

כותרת הפעולה:

```
public void fixNumber()
```

(8 נק') 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
$1 \rightarrow 2 \rightarrow 8$	$1 \rightarrow 2 \rightarrow 9$
$4 \rightarrow 7 \rightarrow 5 \rightarrow 9$	$4 \rightarrow 7 \rightarrow 6 \rightarrow 0$
$9 \rightarrow 9$	$1 \rightarrow 0 \rightarrow 0$

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

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

(2 נק') ג. מהי סיבוכיות הפעולה addOne שכתבתם בסעיף ב'? הסבירו את תשובתכם.

שאלה 3

נתון פרויקט שבו הגדרו מחלקות V, W, X, Y, Z. הפעולה `stam()` 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(); // "Good" תקין ומדפיס
z3.stam(); // "Good" תקין ומדפיס
w1.stam(); // "Bad" תקין ומדפיס
```

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
```

(8 נק') א. שרטטו תרשים היררכיה UML המתאר עץ ירושה אפשרי עבור מחלקות הפרויקט.

(2 נק') ב. ציינו באילו מחלקות מומשה הפעולה `stam()`.

(6 נק') ג. כתבו את כותרות כל המחלקות וכותרות הפעולות הבונות (הבנאים) הנדרשות.

אין צורך לממש את הבנאים.

חלק ב'

ענו על שתיים מבין השאלות 4–6 (ערך כל שאלה – 16 נקודות).

שאלה 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 לכל תכונה.

המחלקה Restaurant מאופיינת בתור של לקוחות וברשימה מקושרת של שולחנות.

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

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

(4 נק') 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)
```

(6 נק') 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 numDiners)
```

(6 נק') 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 ימצא 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

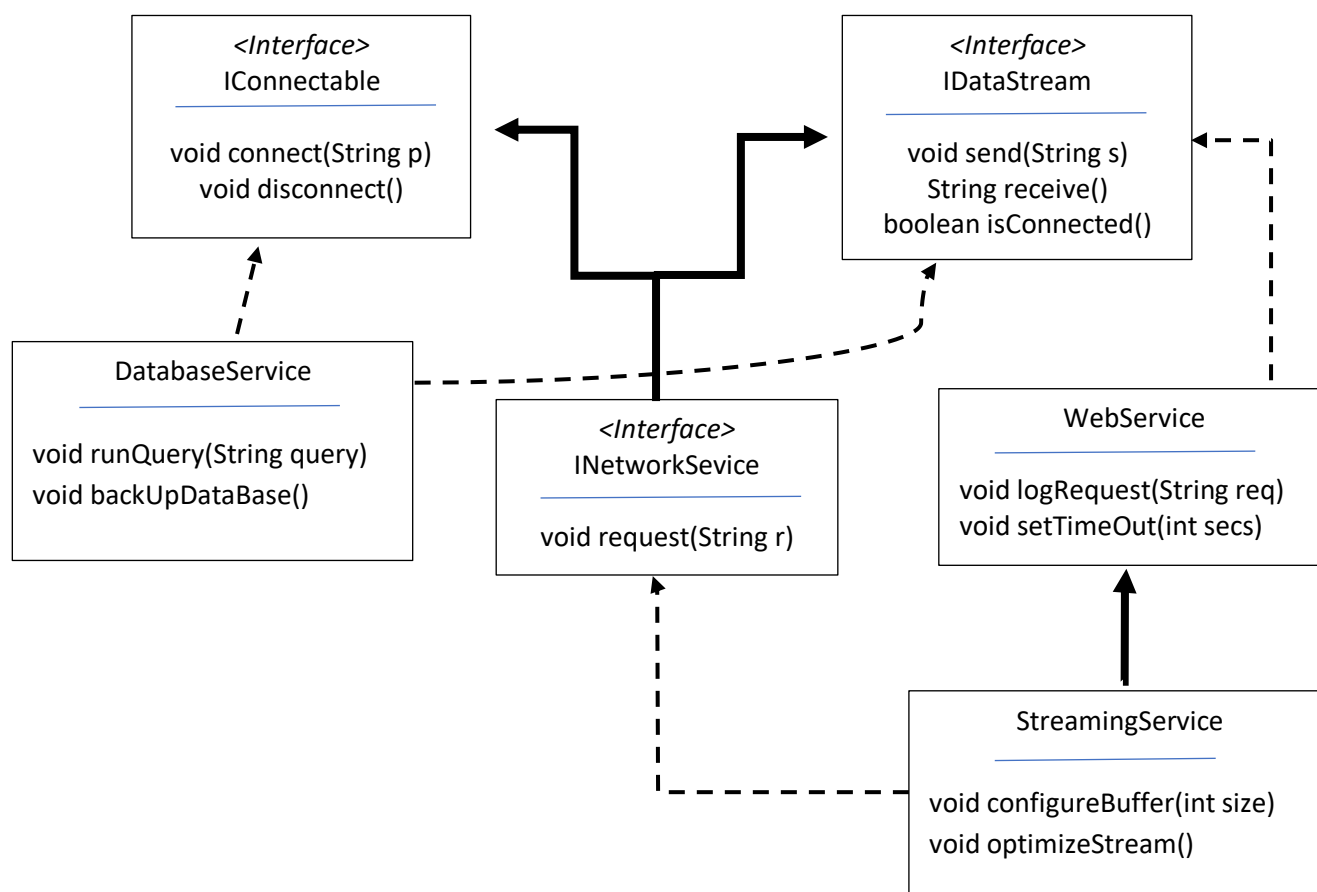
תחזיר false.

כותרת הפעולה:

```
public boolean seatNextClient()
```

שאלה 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:



<u>קשר ירושה</u>	
<u>קשר מימוש</u>	

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	<code>IConnectable c1 = new WebService(); c1.connect("api.com"); c1.send("hello");</code>
2	<code>IDataStream d1 = new StreamingService(); d1.connect("stream.com");</code>
3	<code>StreamingService s1 = new StreamingService(); s1.connect("video.com"); s1.request("play video"); s1.optimizeStream();</code>
4	<code>INetworkService n1 = new StreamingService(); n1.send("start stream"); n1.configureBuffer(1024);</code>
5	<code>WebService w1 = new StreamingService(); w1.receive(); w1.logRequest("get status");</code>
6	<code>IConnectable c2 = new StreamingService(); c2.disconnect(); c2.send("ping");</code>
7	<code>IConnectable c3 = new DatabaseService(); c3.connect("db.com"); c3.runQuery("SELECT * FROM users;"); ((DatabaseService) c3).backupDatabase(); System.out.println(((DatabaseService) c3).receive());</code>
8	<code>INetworkService n2 = new DatabaseService(); n2.connect("db2.com");</code>
9	<code>IConnectable c4 = new WebService(); ((DatabaseService) c4).runQuery("DROP TABLE posts;");</code>
10	<code>StreamingService s2 = new WebService();</code>
11	<code>INetworkService n3 = new WebService();</code>
12	<code>IConnectable[] connections = new IConnectable[2]; connections[0] = new WebService(); connections[1] = new DatabaseService(); ((StreamingService) connections[0]).optimizeStream();</code>

שאלה 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;
    }
}
```

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)
```

(6 נק') 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)
```

חובה להשתמש בפעולה firstAfterChain.

(8 נק') ב. כתבו פעולה המקבלת תו letter המופיע לפחות פעמיים במקומות כלשהם ברשימה, אך לא

בתחילתה ולא בסופה.

הפעולה תחליף בין תת-השרשרת הנמצאת לפי המופע הראשון של letter לבין תת-השרשרת

הנמצאת אחרי המופע האחרון של 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)
```

חובה להשתמש בפעולות firstAfterChain ו-last.

(2 נק') ד. מהי סיבוכיות זמן הריצה של הפעולה swap? הסבירו את תשובתכם.

חלק ג'

ענו על שתיים מבין השאלות 7–9 (ערך שאלה – 18 נקודות).

שאלה 7

(4 נק') 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());
}
```

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

(4 נק') ב. כתבו את הפעולה why בצורה הלא רקורסיבית.

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;
        }
    }
}
```

(10 נק') ג. עקבו אחרי זימון הפעולה what (7) ורשמו מה תחזיר הפעולה.

יש להראות את תוכן העץ בחזרה מכל קריאה רקורסיבית.

שאלה 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.

<pre>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</pre>	<pre>public class C extends A { protected int t; public void func(A a){} // 5 public void func(B b){} // 6 } //end of class C</pre>
<pre>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</pre>	<pre>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</pre>
<pre>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</pre>	

(2 נק') א. שרטטו את עץ הירושה למחלקות המוגדרות בפרויקט.

(8 נק') 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 שממן 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.

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

(8 נק') 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 מהו 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) ;

שאלה 9נתונות הפעולות $f1, f2$:

```

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;
}

```

(4 נק') א. נתונה שרשרת חוליות $n \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3$.

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

(3 נק') ב. מהי מטרת הפעולה $f1$ באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.(4 נק') ג. נתונה שרשרת חוליות $n \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$.מה תהיה תוצאת הזימון $f2(n, n)$ יש להראות מעקב אחרי ביצוע הפעולה!(4 נק') ד. נתונה שרשרת חוליות $n \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$.מה תהיה תוצאת הזימון $f2(n, n)$?(3 נק') ה. מהי מטרת הפעולה $f2$ באופן כללי? יש להסביר מה מבצעת הפעולה ולא כיצד פועל האלגוריתם.

נספח ממשקים מבנה הנתונים בתוכנית הלימודים-JAVA

ממשק החוליה הגנרית - $\text{Node}<T>$

המחלקה מגדירה חוליה גנרית שבה ערך מטיפוס T והפניה לחוליה העוקבת.

$\text{Node } (T \ x)$	הפעולה בונה חוליה. הערך של החוליה הוא x , ואין לה חוליה עוקבת.
$\text{Node } (T \ x, \text{Node}<T> \text{ next})$	הפעולה בונה חוליה. הערך של החוליה הוא x , והחוליה העוקבת לה היא next . ערכו של next יכול להיות null .
$T \text{ getValue}()$	הפעולה מחזירה את הערך של החוליה.
$\text{Node}<T> \text{ getNext}()$	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null .
$\text{void setValue } (T \ x)$	הפעולה משנה את הערך השמור בחוליה ל- x .
$\text{boolean hasNext}()$	הפעולה מחזירה true אם יש חוליה נוספת.
$\text{void setNext } (\text{Node}<T> \text{ next})$	הפעולה משנה את החוליה העוקבת ל- next . ערכו של next יכול להיות null .
$\text{String toString}()$	הפעולה מחזירה מחרוזת המתארת את החוליה.

יעילות הפעולות: כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$.

ממשק המחלקה הגנרית - מחסנית $\text{Stack}<T>$

המחלקה מגדירה טיפוס אוסף בעל פרוטוקול LIFO להכנסה והוצאה של ערכים.

$\text{Stack}<T>()$	הפעולה בונה מחסנית ריקה
$\text{boolean isEmpty}()$	הפעולה מחזירה "אמת" אם המחסנית הנוכחית ריקה, ואם לא – הפעולה תחזיר "שקר".
$\text{void push } (T \ x)$	הפעולה מכניסה את הערך x לראש המחסנית הנוכחית (דחיפה).
$T \text{ pop}()$	הפעולה מוציאה את הערך שבראש המחסנית הנוכחית ומחזירה אותו (שליפה). הנחה: המחסנית הנוכחית אינה ריקה.
$T \text{ top}()$	הפעולה מחזירה את הערך שבראש המחסנית הנוכחית בלי להוציאו. הנחה: המחסנית הנוכחית אינה ריקה.
$\text{String toString}()$	הפעולה מחזירה תיאור של המחסנית, כסדרה של ערכים, במבנה הזה (x_1 הוא האיבר שבראש המחסנית): $[x_1, x_2, \dots, x_n]$

יעילות הפעולות - מחלקה מיוצגת בעזרת שרשרת חוליות.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה $\text{toString}()$, המתבצעת בסדר גודל לינארי.

ממשק המחלקה הגנרית - תור <T> QUEUE

המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים.

Queue<T>()	הפעולה בונה תור ריק
boolean isEmpty()	הפעולה מחזירה "אמת" אם התור הנוכחי ריק, ואם לא – הפעולה תחזיר "שקר".
void insert (Tx)	הפעולה מכניסה את הערך x לסוף התור הנוכחי.
T remove()	הפעולה מוציאה את הערך שבראש התור הנוכחי ומחזירה אותו. הנחה: התור הנוכחי אינו ריק.
T head()	הפעולה מחזירה את ערכו של האיבר שבראש התור מבלי להוציאו. הנחה: התור הנוכחי אינו ריק.
String toString()	הפעולה מחזירה מחרוזת המתארת את התור כסדרה של ערכים, במבנה הזה (x_1 הוא האיבר שבראש התור): $[x_1, x_2, \dots, x_n]$

יעילות הפעולות - המחלקה מיוצגת בעזרת שרשרת חוליות והפניה לזנב התור.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה `toString()`, המתבצעת בסדר גודל לינארי.

ממשק המחלקה הגנרית - חוליה בינרית <T> BINNODE

המחלקה מגדירה חוליה בינרית שבה ערך מטיפוס T ושתי הפניות לחוליות בינריות.

BinNode (T x)	הפעולה בונה חוליה בינרית. ערך החוליה הוא x וערך שתי ההפניות שלה הוא null .
BinNode (BinNode<T> left, T x, BinNode<T> right)	הפעולה בונה חוליה בינרית שערכה יהיה x. left ו-right הן (הפניות אל) הילד השמאלי והימני שלה. ערכי ההפניות יכולים להיות null .
T getValue()	הפעולה מחזירה את הערך של החוליה.
void setValue (T x)	הפעולה משנה את הערך השמור בחוליה ל-x.
boolean hasLeft()	הפעולה מחזירה true אם יש חוליה משמאל.
boolean hasRight()	הפעולה מחזירה true אם יש חוליה מימין.
BinNode<T> getLeft()	הפעולה מחזירה את הילד השמאלי של החוליה. אם אין ילד שמאלי הפעולה מחזירה null .
BinNode<T> getRight()	הפעולה מחזירה את הילד הימני של החוליה. אם אין ילד ימני הפעולה מחזירה null .
void setLeft (BinNode<T> left)	הפעולה מחליפה את הילד השמאלי בחוליה left.
void setRight (BinNode<T> right)	הפעולה מחליפה את הילד הימני בחוליה right.
String toString()	הפעולה מחזירה מחרוזת המתארת את הערך השמור בחוליה.

יעילות הפעולות: כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$.

נספח ממשקים מבנה הנתונים בתוכנית הלימודים – C#

ממשק החוליה הגנרית $\text{NODE}<T>$

המחלקה מגדירה חוליה גנרית שבה ערך מטיפוס T והפניה לחוליה העוקבת.

$\text{Node } (T \ x)$	הפעולה בונה חוליה. הערך של החוליה הוא x , ואין לה חוליה עוקבת.
$\text{Node } (T \ x, \text{Node}<T> \text{ next})$	הפעולה בונה חוליה. הערך של החוליה הוא x , והחוליה העוקבת לה היא next . ערכו של next יכול להיות null .
$T \text{ GetValue}()$	הפעולה מחזירה את הערך של החוליה.
$\text{Node}<T> \text{ GetNext}()$	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null .
$\text{void SetValue } (T \ x)$	הפעולה משנה את הערך השמור בחוליה ל- x .
$\text{bool HasNext}()$	הפעולה מחזירה true אם יש חוליה נוספת.
$\text{void SetNext } (\text{Node}<T> \text{ next})$	הפעולה משנה את החוליה העוקבת ל- next . ערכו של next יכול להיות null .
$\text{string ToString}()$	הפעולה מחזירה מחרוזת המתארת את החוליה.

יעילות הפעולות: כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$.

ממשק המחלקה מחסנית $\text{STACK}<T>$

המחלקה מגדירה טיפוס אוסף בעל פרוטוקול LIFO להכנסה והוצאה של ערכים.

$\text{Stack}<T>()$	הפעולה בונה מחסנית ריקה.
$\text{bool IsEmpty}()$	הפעולה מחזירה "אמת" אם המחסנית הנוכחית ריקה, ואם לא – הפעולה תחזיר "שקר".
$\text{void Push } (T \ x)$	הפעולה מכניסה את הערך x לראש המחסנית הנוכחית (דחיפה).
$T \text{ Pop}()$	הפעולה מוציאה את הערך שבראש המחסנית הנוכחית ומחזירה אותו (שליפה). הנחה: המחסנית הנוכחית אינה ריקה.
$T \text{ Top}()$	הפעולה מחזירה את הערך שבראש המחסנית הנוכחית מבלי להוציאו. הנחה: המחסנית הנוכחית אינה ריקה.
$\text{string ToString}()$	הפעולה מחזירה תיאור של המחסנית, כסדרה של ערכים, במבנה הזה (x_1 הוא האיבר שבראש המחסנית): $[x_1, x_2, \dots, x_n]$

יעילות הפעולות - מחלקה מיוצגת בעזרת שרשרת חוליות.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה $\text{ToString}()$, המתבצעת בסדר גודל לינארי.

ממשק המחלקה תור $\text{QUEUE}<T>$

המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים.

<code>Queue<T>()</code>	הפעולה בונה תור ריק.
<code>bool IsEmpty()</code>	הפעולה מחזירה "אמת" אם התור הנוכחי ריק, ואם לא – הפעולה תחזיר "שקר".
<code>void Insert (T x)</code>	הפעולה מכניסה את הערך x לסוף התור הנוכחי.
<code>T Remove()</code>	הפעולה מוציאה את הערך שבראש התור הנוכחי ומחזירה אותו. הנחה: התור הנוכחי אינו ריק.
<code>T Head()</code>	הפעולה מחזירה את ערכו של האיבר שבראש התור מבלי להוציאו. הנחה: התור הנוכחי אינו ריק.
<code>string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את התור כסדרה של ערכים, במבנה הזה (x_1 הוא האיבר שבראש התור): $[x_1, x_2, \dots, x_n]$

יעילות הפעולות - המחלקה מיוצגת בעזרת שרשרת חוליות והפניה לזנב התור.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה `ToString()`, המתבצעת בסדר גודל לינארי.ממשק המחלקה חוליה בינרית $\text{BINNODE}<T>$ המחלקה מגדירה חוליה בינרית שבה ערך מטיפוס T ושתי הפניות לחוליות בינריות.

<code>BinNode (T x)</code>	הפעולה בונה חוליה בינרית. ערך החוליה הוא x וערך שתי ההפניות שלה הוא <code>null</code> .
<code>BinNode (BinNode<T> left, T x, BinNode<T> right)</code>	הפעולה בונה חוליה בינרית שערכה יהיה x . <code>left</code> ו- <code>right</code> הן (הפניות אל) הילד השמאלי והימני שלה. ערכי ההפניות יכולים להיות <code>null</code> .
<code>T GetValue()</code>	הפעולה מחזירה את הערך של החוליה.
<code>void SetValue (T x)</code>	הפעולה משנה את הערך השמור בחוליה ל- x .
<code>bool HasLeft()</code>	הפעולה מחזירה <code>true</code> אם יש חוליה משמאל.
<code>bool HasRight()</code>	הפעולה מחזירה <code>true</code> אם יש חוליה מימין.
<code>BinNode<T> GetLeft()</code>	הפעולה מחזירה את הילד השמאלי של החוליה. אם אין ילד שמאלי הפעולה מחזירה <code>null</code> .
<code>BinNode<T> GetRight()</code>	הפעולה מחזירה את הילד הימני של החוליה. אם אין ילד ימני הפעולה מחזירה <code>null</code> .
<code>void SetLeft (BinNode<T> left)</code>	הפעולה מחליפה את הילד השמאלי בחוליה <code>left</code> .
<code>void SetRight (BinNode<T> right)</code>	הפעולה מחליפה את הילד הימני בחוליה <code>right</code> .
<code>string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את הערך השמור בחוליה.

יעילות הפעולות: כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$.

97105 מבני נתונים ותכנות מונחה עצמיים – מועד ב' קיץ 2025

שאלה	סעיף	תת-סעיף	ניקוד	הערות
1	א	-	8	• בדיקה ספרה שמלית ביותר – 2 נק'
	ב	-	6	חריגה – להוריד 1 נק' איבד איברים – להוריד 2 נק'
	ג	-	2	בלי הסבר – לא לתת נקודות
2	א	-	6	• יצירת חוליות והוספה לשרשרת 4- נק' חריגה – להוריד 2 נקודות לולאה אין סופית – להוריד 2 נק'
	ב	-	8	• בניית מבנה של סדר הפוך – 2 נק' • הוספה 1 – 1 נק' • שימוש ב-CARRY – 3 נק' • החזרה לסדר מקורי – 2 נק'
	ג		2	בלי הסבר – לא לתת נקודות
3	א	-	8	
	ב	-	2	
	ג		6	• כל מחלקה 1 נק' לא כתב בנאים ריקים – להוריד 1 נק'
4	א	--	4	• יצירה תור ריק – 1 נק' • יצירה שרשרת 3 - נק'
	ב	-	6	• סריקה – 1 נק' • בדיקה מספר מקומות ריקים – 2 נק' • החזרה מספר שולחן או (1-) - 1 נק'
	ג	-	6	• לולאה על תור לקוחות – 2 נק' • שימוש בפעולה של סעיף א' – 1 נק' • סריקה על שרשרת חוליות ועדכון מספר מקומות פנויים – 1 נק' • החזרה לתור מקורי – 1 נק'
			16	• כל סעיף – 1 נקודה
6	א	-	6	מצא מיקום ראשון בשרשרת אבל לא מיקום ראשון אחרי CHAIN – להוריד 2 נק' מצא מיקום אחרון – להוריד 2 נק'
	ב	-	8	• מציאה מיקום ראשון של התו – 2 נק' • מציאה מיקום אחרון של התו – 2 נק' • החלפה קטעים – 4 נק'
	ג	-	2	בלי הסבר – לא לתת נקודות
7	א	-	4	צייר עץ לא של 6 צמתים – להוריד 1 נק'
	ב	-	4	
	ג	-	10	בלי מעקב – לתת מקסימום 5 נקודות.

	2	-	א	8
כל ערך – 0.5 נקודות	8	-	ב	
כל סעיף – 1 נקודה	8	-	ג	
	4	-	א	9
	3	-	ב	
בלי מעקב לא לתת נקודות	4	-	ג	
	4	-	ד	
בלי הסבר – לא לתת נקודות	3	-	ה	

חלק א

שאלה 1

נושא השאלה המרכזי - פונקציות עם מחסנית בשני סעיפים, דרישה לחישוב סיבוכיות על פעולות.

שאלה 2

נושא השאלה המרכזי - רשימות מקושרות, כתיבת פעולות ודרישה לחישוב סיבוכיות על פעולות

שאלה 3

נושא השאלה המרכזי - קטע קוד עם יצירת עצמים שונים חלק מיצירת העצמים שגוי, דרישה לשרטוט עץ ירושה אפשרי

חלק ב

שאלה 4

נושא השאלה המרכזי - נתונות שתי מחלקות בתיאור מילולי ופירוט תכונות, בשאלה יש שילוב של תור ורשימה ופעולות בשני מבני הנתונים הללו על עצמים של מהמחלקות הנתונות. בשאלה נכללות דרישות לכתיבת פעולות בונות ופעולות "רגילות" משולבים במחלקות ומבני הנתונים (תור, רשימה)

שאלה 5

נתון שרטוט קשרים-תרשים היררכי בין מחלקות וממשקים, בשאלה טבלה הכוללת בכל שורה קטעי קוד, יש לזהות האם הקוד שגוי או תקין ולנמק יש לציין בשגיאה האם מדובר בשגיאת ריצה או שגיאת תחבירית בהלימה לשרטוט ההיררכי של מחלקות וממשקים שנתון.

שאלה 6

נושא השאלה המרכזי - בשאלה קטע קוד ומטלות כתיבת פעולות. השאלה מגבילה שימוש במבני נתונים מסויימים ומיגבלות נוספות. נדרש לכתוב פעולות על רשימות. יש לחשב סיבוכיות של פעולות אם נדרש בפעולה \ פעולות בקוד.

חלק ג

שאלה 7

השאלה עוסקת בעצים בינאריים, ציור עץ בינארי על פי דרישה, כתיבה פעולה לפי דרישה מוגדרת, ביצוע מעקב אחר זימון פעולה.

שאלה 8

שאלה ממוקדת בתמ"ע : כולל קוד של מספר מחלקות עם קשרי ירושה, main שכולל הוראות ליצירת עצמים. יש לצייר עץ ירושה ולעקוב אחר ריצת הקוד, כולל מעקב בטבלה.

שאלה 9

שאלה ממוקדת במבנה תור ומבנה רשימה. השאלה כוללת מעקב ביצוע, מה מטרת פעולה, הדרישה למעקב לצד הדרישה מה פונקציה מסויימת מבצעת.