



מספר מחברת: _____

מספר ת"ז: _____

מועד הבחינה:
קיץ תשפ"ד – 2024 – מועד א'
מספר השאלון: 97105
נספח ממשקים לבחינה JAVA
נספח ממשקים לבחינה C#
מילון עזר

מבני נתונים ותכנות מונחה עצמים

הנדסאים וטכנאים – הנדסת תוכנה

הנחיות לבחינה

- א. משך הבחינה: ארבע שעות וחצי.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה 2 מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:
מבחן ב-Java (עמוד 2)
מבחן ב-C# (עמוד 12)
בכל מבחן 10 שאלות.
חלק א' – 48 נקודות
שאלות 1-5: מתוכן יש לענות על 3 שאלות בלבד. ערך כל שאלה 16 נקודות.
חלק ב' – 36 נקודות
שאלות 6-10: מתוכן יש לענות על 2 שאלות בלבד. ערך כל שאלה 18 נקודות.
חלק ג' – 16 נקודות
שאלות 11-12: מתוכן יש לענות על שאלה 1 בלבד. ערך שאלה 16 נקודות.
בסך הכול: 100 נקודות.
- ג. חומר עזר מותר לשימוש:
 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 כלליות:
 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!

בהצלחה!

בשאלון זה 21 עמודי בחינה ו-7 עמודי נספחים.

מבחן ב-JAVA**חלק א'**

ענו על 3 מבין השאלות 1-5 (ערך כל שאלה – 16 נקודות).

1.

א. (12 נק') מחסנית של מספרים שלמים נקראת "מאוזנת" אם יש לה אותה כמות מספרים גם חיוביים, גם שליליים וגם אפסים.

כתבו פעולה המקבלת מחסנית של מספרים שלמים ובודקת אם היא "מאוזנת". יש לשמור על תוכן של המחסנית אחרי ביצוע הפעולה.
 כותרת הפעולה:

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

ב. (4 נק') מהי הסיבוכיות של הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

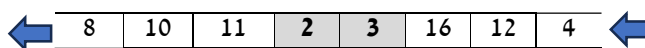
2.

א. (12 נק') כתבו פעולה המקבלת תור של מספרים שלמים ו-2 מספרים נוספים x ו- y .

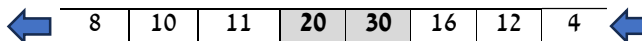
אם אורך התור זוגי, הפעולה תחליף את 2 האיברים האמצעיים לערכים x ו- y . אם אורך התור אי-זוגי, הפעולה תיצור מספר אקראי בין x ל- y (כולל) ותחליף את האיבר האמצעי של התור לערך אקראי שנוצר. שאר איברי התור יישארו ללא שינוי.

לדוגמה:

עבור התור q הבא ופרמטרים $x=20, y=30$



התור q אחרי זימון הפעולה יהיה:



ב. (4 נק') מהי הסיבוכיות של הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

3. **סדרה יורדת** היא הסדרה שכל איבר החל באיבר ה-3 שווה להפרש של 2 קודמיו בערך מוחלט. לדוגמה:

אם 2 האיברים של הסידרה הם 55 ו-34 אז סדרה יורדת תהיה:

55 34 21 13 8 5 3 2 1 1 0 1 1

אם 2 האיברים של הסידרה הם 15 ו-40 אז סדרה יורדת תהיה:

15 40 25 10 15 5 10 5 5 0 ...

תזכורת: לחישוב הפרש יש להשתמש בפונקציה של ערך מוחלט (Math.abs)

א. (6 נק') כתבו פעולה המקבלת הפנייה לשרשרת חוליות של מספרים שלמים. הפעולה תבדוק אם ערכים של השרשרת הם **סדרה יורדת**. אם כן – הפעולה תחזיר ערך `true`, ולא – הפעולה תחזיר ערך `false`.

כותרת הפעולה:

```
public static boolean isDownList (Node<Integer> chain)
```

ב. (6 נק') כתבו פעולה המקבלת מספר שלם וחיובי `num` וזוג מספרים שלמים `a` ו-`b`. הפעולה תחזיר

הפניה לחוליה הראשונה של שרשרת חוליות שאבריה הם `num` המספרים הראשונים של

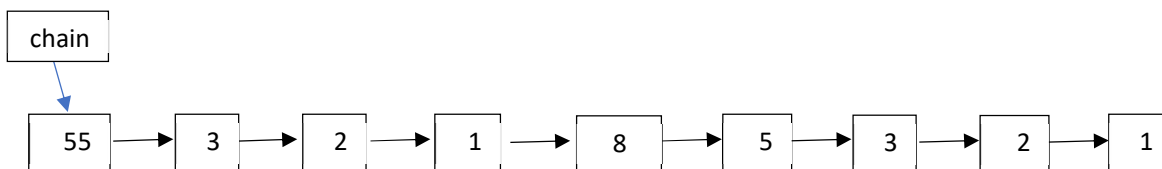
סדרה יורדת המתחילה במספרים `a` ו-`b`.

כותרת הפעולה:

```
public static Node<Integer> buildDown (int num, int a, int b)
```

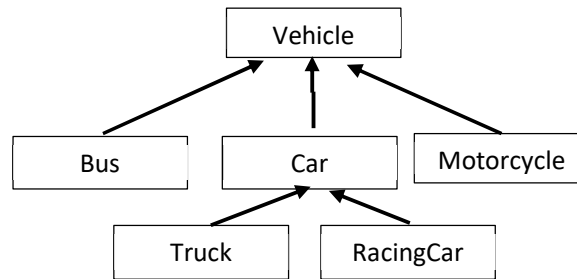
לדוגמה: עבור הזימון `buildDown(9, 55, 34)` הפעולה תחזיר הפניה לחוליה

הראשונה של שרשרת החוליות הבאה:



ג. (4 נק') מהי הסיבוכיות של הפעולות שכתבתם בסעיפים א' ו-ב'? הסבירו את תשובתכם.

א. (8 נק') נתון עץ הירושה הבא המתאר סוגי כלי תחבורה שונים:



לפניכם קטע קוד מתוך הפעולה הראשית:

```

Vehicle v1 = new Vehicle();
Vehicle v2 = new Car();
Vehicle v3 = new Truck();
Car c = new Truck ();
  
```

לפניכם 8 קטעי קוד. עבור כל אחד מהסעיפים יש לציין, אם הקוד תקין או לא. אם הקוד אינו תקין, יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (שגיאת קומפילציה / הידור או שגיאת זמן ריצה).

```

1. Car c1 = new Vehicle ();
2. Car c2 = v1;
3. Car c3 = (Car) v2;
4. Car c4 = (Car) v3;
5. Car c5 = (Car) v1;
6.
   Motorcycle m = new Motorcycle();
   Vehicle v6 = m;
   Car c6 = (Car) s6;
7. Vehicle v7 = (Car) (new Truck ());
8. Car c8 = (Vehicle) (new Truck ());
  
```

ב. (4 נק') רוצים לכתוב פעולה אשר מקבלת מערך של כלי תחבורה (מערך הפניות לעצמים מטיפוס

Vehicle) ומחזירה את המהירות הממוצעת של כל האופנועים (Motorcycle) ומכוניות המירוץ (RacingCar). באיזו מחלקה או מחלקות יש להוסיף את הפעולה `double speed()` המחזירה את המהירות? הסבירו את תשובתכם, אין צורך לממש את הפעולה `speed()`.

ג. (4 נק') כתבו פעולה המקבלת מערך כלי תחבורה ומחזירה את האופנוע שמהירותו היא הגבוהה ביותר. אם במערך אין אף אופנוע, הפעולה תחזיר `null`.

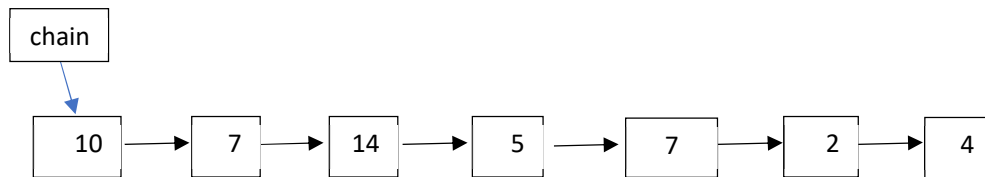
כותרת הפעולה:

```
public static Motorcycle maxSpeed(Vehicle[] arr)
```

א. (6 נק') נתונה הפעולה why הבאה:

```
public static Node<Integer> why(Node<Integer> chain, boolean flag)
{
    int count = 0;
    Node<Integer> p = chain;
    while(p != null)
    {
        int x = p.getValue();
        boolean f = (x % 2 == 0);
        if( f == flag)
            count ++;
        p = p.getNext();
    }
    return new Node<Integer>(count);
}
```

נתונה שרשרת חוליות הבאה:



מה תחזיר הפעולה ?why(chain, false)

יש להראות מעקב אחרי ביצוע הפעולה why.

ב. (6 נק') נתונה הפעולה what הבאה:

```
public static void what(Node<Integer> chain)
{
    Node<Integer> p = chain;
    Node<Integer> temp;
    while (p != null)
    {
        if(p.getValue()%2 ==0)
            temp = why(p.getNext(), false);
        else
            temp = why(p.getNext(), true);
        temp.setNext(p.getNext());
        p.setNext(temp);
        p = temp.getNext();
    }
}
```

עקבו אחרי ביצוע זימון הפעולה what(chain) ורשמו איך תראה השרשרת אחרי זימון הפעולה.

יש להראות מעקב אחרי ביצוע הפעולה what, אין צורך במעקב אחרי why.

ג. (4 נק') מה מבצעות הפעולות what(chain) ו-why(chain, flag) באופן כללי?

חלק ב'

ענו על 2 מבין השאלות 6-10 (ערך כל שאלה – 18 נקודות).

.6

א. (7 נק') כתבו פעולה המקבלת מחסנית של מספרים שלמים st ומספר נוסף num.

הפעולה תמחק ממחסנית st את כל המופעים של num.

כותרת הפעולה:

```
public static void remove(Stack<Integer> st, int num)
```

ב. (7 נק') כתבו פעולה המקבלת 2 מחסניות של מספרים שלמים st1 ו-st2. הפעולה תחזיר מחסנית

חדשה הכוללת כל ערך מ-2 המחסניות. כל ערך במחסנית החדשה צריך להופיע פעם אחת.

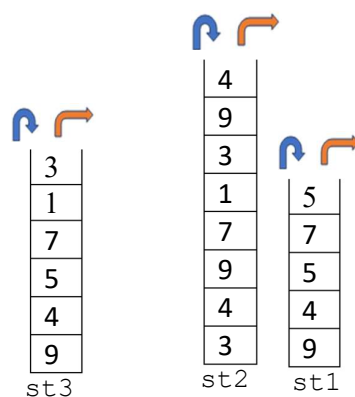
סדר האיברים במחסנית החדשה לא חשוב.

לדוגמה: עבור 2 מחסניות הבאות st1 ו-st2 הפעולה תחזיר מחסנית st3.

כותרת הפעולה:

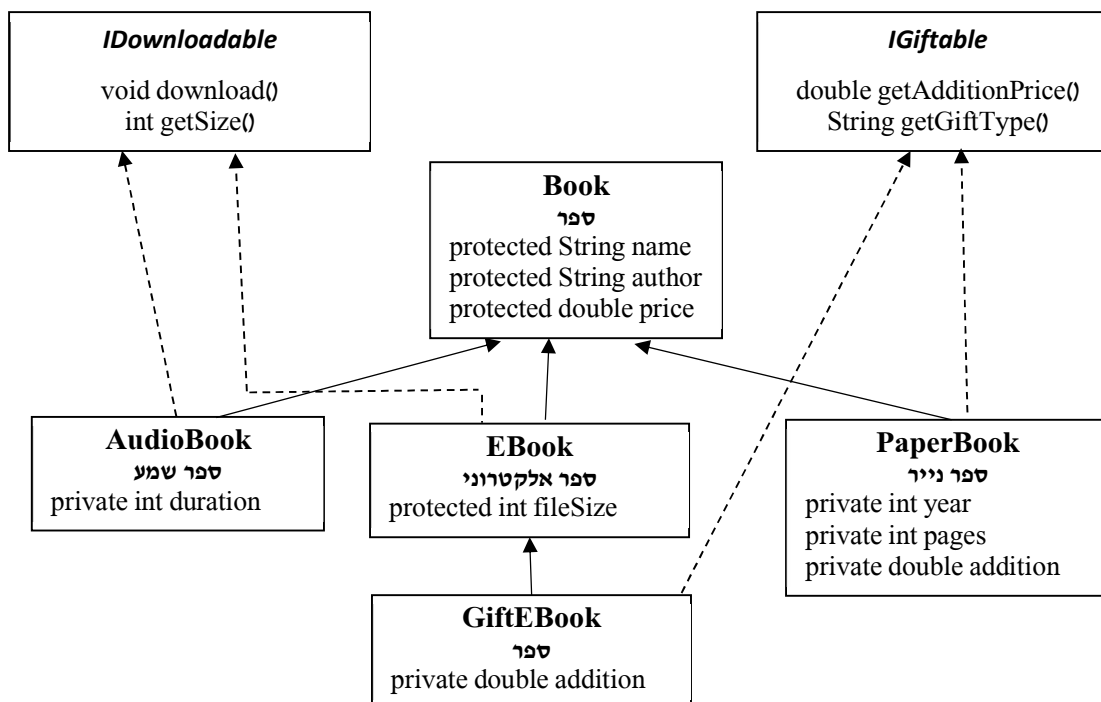
```
public static Stack<Integer> commonValues(Stack<Integer> s1,
```

```
Stack<Integer> s2)
```



ג. (4 נק') מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'-ב'? הסבירו את תשובתכם.

7. החנות "ספר לכל" מוכרת סוגים שונים של ספרים: ספרים מודפסים, ספרי שמע, ספרים אלקטרוניים. חלק מהמוצרים ניתן לרכוש באריות מתנה (בעבור תשלום נוסף). לשם קידום מכירות ספרים, חנות "ספר לכל" פיתחה מערכת לניהול מלאי הספרים הכוללת את המחלקות הבאות: GiftEBook, PaperBook, EBook, AudioBook, Book ו-IGiftable. IDownloadable-1.
- לפניכם תרשים UML המתאר קשרים בין המחלקות והממשקים:



- א. (5 נק') כתבו עבור כל מחלקה, את כותרת המחלקה ואת הכותרות של כל הפעולות שחייבות להיות במחלקה.
- ב. (2 נק') המחיר של כל ספר תלוי בסוג הספר. לשם כך הוחלט להגדיר 3 קבועים המציינים מקדמי מחיר עבור ספרים אלקטרוניים, ספרי שמע וספרים מודפסים:
- ```
final double EBOOK_FACTOR = 0.8;
final double AUDIOBOOK_FACTOR = 1.2;
final double PAPERBOOK_FACTOR = 1.0;
```
- ג. (4 נק') באיזו מחלקה או באילו מחלקות יש להוסיף את הקבועים הנ"ל? הסבירו את תשובתכם. כתבו פעולה או פעולות המחזירה את מחיר הספר (ללא אריות מתנה), עבור כל סוג ספר, יש לציין באיזו מחלקה צריך להוסיף את הפעולה או את הפעולות.
- ד. (3 נק') כתבו פעולה חיצונית המקבלת מערך ספרים המתאר את מלאי החנות וסכום כסף שלקוח רוצה להשקיע. הפעולה תדפיס את פרטי הספרים שמחירם אינו עולה על הסכום שהלקוח רוצה להשקיע.
- ה. (3 נק') כתבו פעולה חיצונית המקבלת מערך ספרים המתאר את מלאי החנות. הפעולה תדפיס את מספר הספרים שאפשר לרכוש אותם באריות מתנה.

8. הגדרה: עץ "שרשרת עולה" הוא עץ בינארי של מספרים שלמים העונה על הכלל הבא:

- לכל צומת פרט לעלים יש בדיוק בן אחד.
  - ערכים של כל הצמתים הם מספרים חיוביים זוגיים.
  - הערך של צומת הבן גדול מהערך של צומת האב.
- א. (4 נק') ציירו עץ "שרשרת עולה" הכולל לפחות 6 צמתים.
- ב. (14 נק') כתבו פעולה המקבלת הפנייה לשורש של עץ בינארי של מספרים שלמים. הפעולה תבדוק אם העץ הוא "שרשרת עולה". אם כן – הפעולה תחזיר ערך true, ולא – הפעולה תחזיר ערך false.

9. הערה כללית לשאלה:

- אין להשתמש במבני נתונים נוספים.
  - אפשר להניח שקיימת פעולה sizeQ המחזירה את מספר האיברים בתור. כותרת הפעולה:
- ```
public static int sizeQ(Queue<Integer> q)
```
- א. (7 נק') כתבו פעולה המקבלת תור של מספרים שלמים וחיוביים q ומספר שלם חיובי pos. הפעולה תחזיר איבר אשר נמצא במקום pos בתור בלי להוציא אותו מהתור. אם אין מקום pos בתור, הפעולה תחזיר ערך (-1).
- הערה: האיבר הנמצא בראש התור הוא איבר במקום 1, איבר אחריו – במקום 2 וכו'. כותרת הפעולה:
- ```
public static int valueAt(Queue<Integer> q, int pos)
```
- ב. (7 נק') כתבו פעולה המקבלת 2 תורים של מספרים שלמים q1 ו-q2. הפעולה תיצור ותחזיר תור חדש לפי כלל הבא:
- קודם כול מכניסים איבר ראשון מהתור q1, ואחריו – איבר האחרון מהתור q2.
  - איבר 2 מהתור q1, איבר לפני האחרון מהתור q2.
  - וכך הלאה.
- אפשר להניח שאורכם של 2 התורים שווה.
- כותרת הפעולה:

```
public static Queue<Integer> merge (Queue<Integer>q1,
Queue<Integer>q2)
```

ג. (4 נק') מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א' ו-ב'? הסבירו את תשובתכם.

```

public class AA{
 public AA(){
 System.out.println("AA constructor");
 }
 public void one() {
 System.out.println("one in AA");
 }
 public void two() {
 one();
 }
}
public class BB extends AA{
 public BB(){
 System.out.println("BB constructor");
 }
 public void one(){
 System.out.println("one in BB");
 }
 public void two() {
 System.out.println("two in BB");
 }
 public void three() {
 super.two();
 }
}

```

א. (14 נק') נתון קטע הקוד הבא (מספרי שורות לא חלק של הקוד)

```

(1) AA a = new AA();
(2) BB b = new BB();
(3) AA ab = new BB();
(4) a.one();
(5) ab.one();
(6) b.one();
(7) a.two();
(8) ab.two();
(9) b.two();
(10) ((BB) ab).three();
(11) b.three();

```

1. עקבו אחרי ביצוע קטע הקוד ורשמו מה יהיה הפלט

2. האם אפשר להוסיף לקטע הקוד את 2 הפקודות הבאות? הסבירו את תשובתכם

```
a.three();
```

```
((BB) a).three();
```

ב. (4 נק') כתבו פעולה המקבלת מערך הפניות לעצמים מסוג Object. הפעולה תדפיס כמה עצמים

הם מסוג BB, כמה עצמים הם מסוג AA ולא סוג BB וכמה עצמים הם לא מסוג AA.

## חלק ג'

ענו על אחת מבין השאלות 11-12 (ערך שאלה – 16 נקודות).

11. ארגון סטודנטים מתנדבים "עזרה לגיל זהב" מחזיק מאגר מידע של סטודנטים הרוצים להתנדב ולעזור לקשישים, בצורה ממוחשבת. עבור כל סטודנט נשמרים הנתונים הבאים:

- מספר תעודת הזהות (מחרוזת)
- שם (מחרוזת)
- יישוב מגורים (מחרוזת)
- אם יש לו רכב (ערך בוליאני, true – כן, false – לא)
- שפת האם (מחרוזת)
- שפה נוספת שסטודנט שולט בה ברמה גבוהה (מחרוזת)

א. (2 נק') כתבו את כותרת המחלקה ואת התכונות עבור כל אחת מהמחלקות הבאות:

- Student המתארת סטודנט.
  - StudentsData המתארת מידע על כל הסטודנטים.
- אפשר להניח שבמחלקה Student קיימים בנאים, פעולות get ו-set לכל תכונה.
- לייצוג אוסף אפשר להשתמש במחלקות Stack, Queue, Node. אפשר להוסיף תכונות נוספות. חובה לתעד את התכונות.**

ב. (3 נק') בסיום שנת הלימודים יש לעדכן את המערכת ולהוציא מהמאגר (אוסף הנתונים) את הסטודנטים שסיימו ללמוד. כתבו פעולה במחלקה StudentsData המקבלת מספר תעודת זהות של מתנדב ומוציאה אותו מהמאגר. כותרת הפעולה:

```
public void eraseStudent(String id)
```

ג. (4 נק') מדי פעם מתקבלת בקשה לעזרה מאזרח וותיק אשר מדבר בשפה מסוימת והמתגורר ביישוב מסוים. כתבו במחלקה StudentsData את הפעולה:

```
public void print(String city, String lang)
```

הפעולה תדפיס את שמות המתנדבים אשר מדברים בשפה lang ומתגוררים ביישוב city. אם אין אף מתנדב שגר ביישוב city ומדבר בשפה lang, הפעולה תדפיס את דוברי השפה lang שהם בעלי רכב.

ד. (7 נק') הנהלת הארגון רוצה לארגן כנס למתנדבים. הוחלט שהכנס יתקיים בעיר שמתגוררים בה הכי הרבה מתנדבים חברי הארגון.

1. כתבו פעולה במחלקה StudentsData המקבלת שם עיר ומחזירה את מספר המתנדבים המתגוררים בה. כותרת הפעולה:

```
public int numStudents(String city)
```

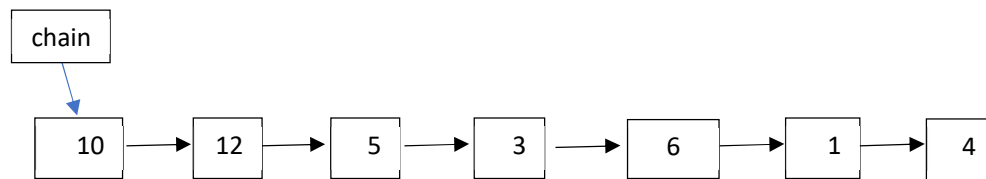
2. כתבו פעולה במחלקה StudentsData המקבלת מערך של שמות ערים (מערך מסוג מחרוזת) ומחזירה את שם העיר שגרים בה הכי הרבה מתנדבים. כותרת הפעולה:

```
public String cityName(String[] cities)
```

**אפשר להניח שיש רק עיר אחת כזאת.**

א. (4 נק') נתונות פעולה רקורסיבית הבאה:

```
public static void one(Node<Integer> n){
 if(n!=null)
 {
 if(n.getValue()% 2==0)
 {
 n.setNext(new Node<Integer> (0,n.getNext()));
 n = n.getNext();
 }
 one(n.getNext());
 }
}
```

עקבו אחרי זימון הפעולה `one(chain)` ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה.ב. (2 נק') מה מבצעת הפעולה `one` באופן כללי?ג. (4 נק') נתונות פעולה רקורסיבית `two` הבאה:

```
public static void two(Node<Integer> n){
 if(n.getNext()!=null)
 {
 if(n.getNext().getValue()%2 == 0)
 n.setNext(n.getNext().getNext());
 else
 n=n.getNext();
 two(n);
 }
}
```

עקבו אחרי זימון הפעולה `two(chain)` ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה.ד. (2 נק') מה מבצעת הפעולה `two` באופן כללי?ה. (4 נק') נתונה הפעולה `three` המשתמשת בפעולה `two`.

```
public static Node<Integer> three(Node<Integer>n){
 if(n == null) return null;
 if(n.getValue()%2 == 0)
 return three(n.getNext());
 else
 {
 two(n);
 return n;
 }
}
```

מה יהיה התוכן של `chain` של הרשימה אחרי ביצוע הפקודה הבאה:`chain = three(chain);`מה מבצעת הפעולה `three` באופן כללי?

## מבחן ב-C#

### חלק א'

ענו על 3 מבין השאלות 1-5 (ערך כל שאלה – 16 נקודות).

1.

א. (12 נק') מחסנית של מספרים שלמים נקראת "מאוזנת" אם יש לה אותה כמות מספרים גם חיוביים, גם שליליים וגם אפסים.  
כתבו פעולה המקבלת מחסנית של מספרים שלמים ובודקת אם היא "מאוזנת". יש לשמור על תוכן של המחסנית אחרי ביצוע הפעולה.  
כותרת הפעולה:

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

ב. (4 נק') מהי הסיבוכיות של הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

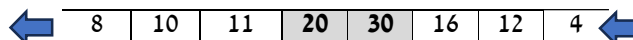
2.

א. (12 נק') כתבו פעולה המקבלת תור של מספרים שלמים ו-2 מספרים נוספים  $x$  ו- $y$ . אם אורך התור זוגי, הפעולה תחליף את 2 האיברים האמצעיים לערכים  $x$  ו- $y$ . אם אורך התור אי-זוגי, הפעולה תיצור מספר אקראי בין  $x$  ל- $y$  (כולל) ותחליף את האיבר האמצעי של התור לערך אקראי שנוצר. שאר איברי התור יישארו ללא שינוי.  
לדוגמה:

עבור התור  $q$  הבא ופרמטרים  $x=20, y=30$



התור  $q$  אחרי זימון הפעולה יהיה:



ב. (4 נק') מהי הסיבוכיות של הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

3. **סדרה יורדת** היא הסדרה שכל איבר החל באיבר ה-3 שווה להפרש של 2 קודמיו בערך מוחלט. לדוגמה:

אם 2 האיברים של הסידרה הם 55 ו-34 אז סדרה יורדת תהיה:

55 34 21 13 8 5 3 2 1 0 1 1 ....

אם 2 האיברים של הסידרה הם 15 ו-40 אז סדרה יורדת תהיה:

15 40 25 10 15 5 10 5 0 ...

**תזכורת:** לחישוב הפרש יש להשתמש בפונקציה של ערך מוחלט ( $\text{Math.abs}$ )

א. (6 נק') כתבו פעולה המקבלת הפנייה לשרשרת חוליות של מספרים שלמים. הפעולה תבדוק אם ערכים של השרשרת הם **סדרה יורדת**. אם כן – הפעולה תחזיר ערך `true`, ולא – הפעולה

תחזיר ערך `false`

כותרת הפעולה:

```
public static bool IsDownList (Node<int> chain)
```

ב. (6 נק') כתבו פעולה המקבלת מספר שלם וחיובי `num` וזוג מספרים שלמים `a` ו-`b`. הפעולה תחזיר

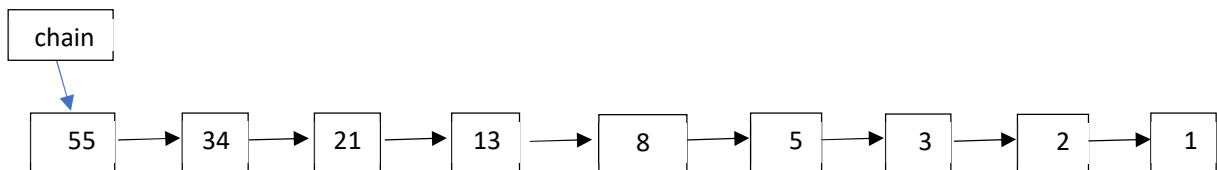
הפניה לחוליה הראשונה של שרשרת חוליות שאבריה הם `num` המספרים הראשונים של

סדרה יורדת המתחילה במספרים `a` ו-`b`. כותרת הפעולה:

```
public static Node<int> BuildDown (int num, int a, int b)
```

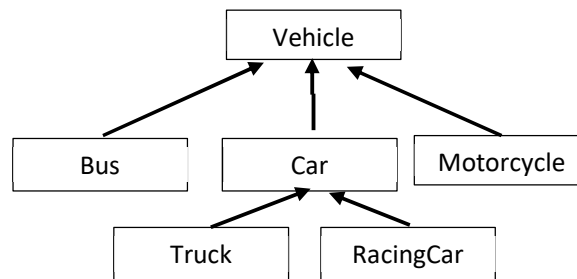
**לדוגמה:** עבור הזימון `BuildDown(9, 55, 34)` הפעולה תחזיר הפנייה לחוליה

הראשונה של שרשרת החוליות הבאה:



ג. (4 נק') מהי הסיבוכיות של הפעולות שכתבתם בסעיפים א' ו-ב'? **הסבירו את תשובתכם.**

א. (8 נק') נתון עץ הירושה הבא המתאר סוגי כלי תחבורה שונים:



לפניכם קטע קוד מתוך הפעולה הראשית:

```
Vehicle v1 = new Vehicle();
Vehicle v2 = new Car();
Vehicle v3 = new Truck();
Car c = new Truck ();
```

לפניכם 8 קטעי קוד. עבור כל אחד מהסעיפים יש לציין, אם הקוד תקין או לא.  
אם הקוד אינו תקין – יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (שגיאת קומפילציה / הידור או שגיאת זמן ריצה).

```
1. Car c1 = new Vehicle ();
2. Car c2 = v1;
3. Car c3 = (Car) v2;
4. Car c4 = (Car) v3;
5. Car c5 = (Car) v1;
6.
 Motorcycle m = new Motorcycle();
 Vehicle v6 = m;
 Car c6 = (Car) s6;
7. Vehicle v7 = (Car) (new Truck ());
8. Car c8 = (Vehicle) (new Truck ());
```

ב. (4 נק') רוצים לכתוב פעולה אשר מקבלת מערך של כלי תחבורה (מערך הפניות לעצמים מטיפוס Vehicle) ומחזירה את המהירות הממוצעת של כל האופנועים (Motorcycle) ומכוניות המרוץ (RacingCar).

באיזו מחלקה או מחלקות יש להוסיף את הפעולה double Speed() המחזירה את

המהירות? הסבירו את תשובתכם, אין צורך לממש את הפעולה Speed().

ג. (4 נק') כתבו פעולה המקבלת מערך כלי תחבורה ומחזירה את האופנוע שמהירותו היא הגבוהה ביותר. אם במערך אין אף אופנוע, הפעולה תחזיר null.  
כותרת הפעולה:

```
public static Motorcycle MaxSpeed(Vehicle[] arr)
```

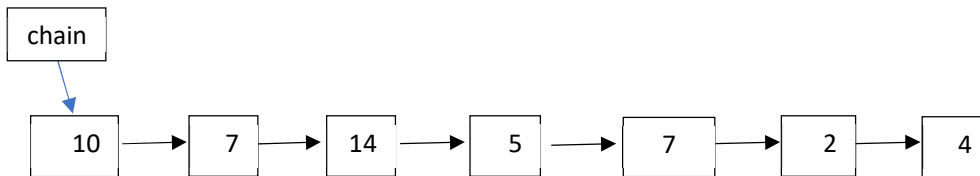
א. (6 נק') נתונה הפעולה Why הבאה:

```

public static Node<int> Why(Node<int> chain, bool flag)
{
 int count = 0;
 Node<int> p = chain;
 while(p != null)
 {
 int x = p.GetValue();
 bool f = (x % 2 == 0);
 if(f == flag)
 count ++;
 p = p.GetNext();
 }
 return new Node<int>(count);
}

```

נתונה שרשרת חוליות הבאה:



מה תחזיר הפעולה Why(chain, false)? יש להראות מעקב אחרי ביצוע הפעולה Why.

ב. (6 נק') נתונה הפעולה What הבאה:

```

public static void What(Node<int> chain)
{
 Node<int> p = chain;
 Node<int> temp;
 while (p != null)
 {
 if(p.GetValue() % 2 == 0)
 temp = Why(p.GetNext(), false);
 else
 temp = Why(p.GetNext(), true);
 temp.SetNext(p.GetNext());
 p.SetNext(temp);
 p = temp.GetNext();
 }
}

```

עקבו אחרי ביצוע זימון הפעולה What(chain) ורשמו איך תראה השרשרת אחרי זימון

הפעולה. יש להראות מעקב אחרי ביצוע הפעולה What, אין צורך במעקב אחרי Why.

ג. (4 נק') מה מבצעות הפעולות What(chain) ו-Why(chain, flag) באופן כללי?

## חלק ב'

ענו על 2 מבין השאלות 6-10 (ערך כל שאלה – 18 נקודות).

.6

א. (7 נק') כתבו פעולה המקבלת מחסנית של מספרים שלמים st ומספר נוסף num. הפעולה תמחק

ממחסנית st את כל המופעים של num.

כותרת הפעולה:

```
public static void Remove(Stack<int> st, int num)
```

ב. (7 נק') כתבו פעולה המקבלת 2 מחסניות של מספרים שלמים st1 ו-st2. הפעולה תחזיר מחסנית

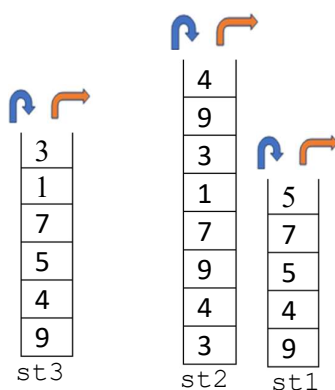
חדשה הכוללת כל ערך מ-2 המחסניות. כל ערך במחסנית החדשה צריך להופיע פעם אחת.

סדר האיברים במחסנית החדשה לא חשוב.

לדוגמה: עבור 2 מחסניות הבאות st1 ו-st2 הפעולה תחזיר מחסנית st3.

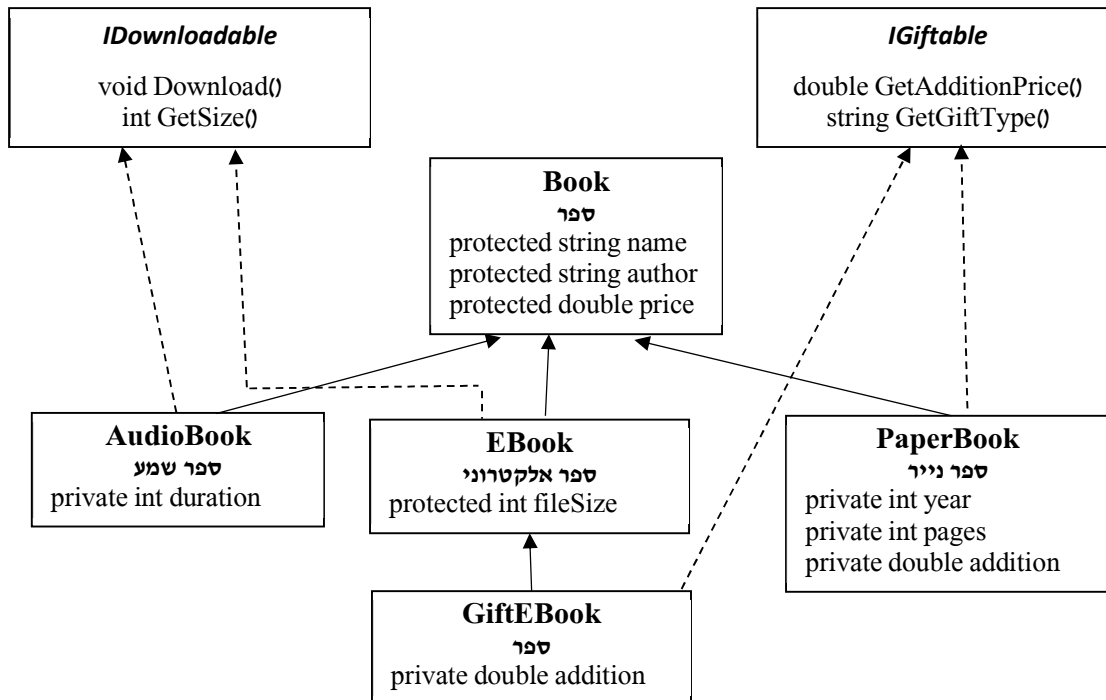
כותרת הפעולה:

```
public static Stack<int> CommonValues(Stack<int> s1,
 Stack<int> s2)
```



ג. (4 נק') מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'-ב'? הסבירו את תשובתכם.

7. החנות "ספר לכל" מוחרת סוגים שונים של ספרים: ספרים מודפסים, ספרים שמע, ספרים אלקטרוניים. חלק מהמוצרים ניתן לרכוש באריות מתנה (בעבור תשלום נוסף). לשם קידום מכירות ספרים, חנות "ספר לכל" פיתחה מערכת לניהול מלאי הספרים הכוללת את המחלקות הבאות GiftEBook, PaperBook, EBook, AudioBook, Book ו-2 ממשקים IGiftable ו-IDownloadable. לפניהם תרשים UML המתאר קשרים בין המחלקות וממשקים:



- א. (5 נק') כתבו עבור כל מחלקה, את כותרת המחלקה ואת הכותרות של כל הפעולות שחייבות להיות במחלקה.
- ב. (2 נק') המחיר של כל ספר תלוי בסוג הספר. לשם כך הוחלט להגדיר 3 קבועים המציינים מקדמי מחיר עבור ספרים אלקטרוניים, ספרי שמע וספרים מודפסים:
- ```
const double EBOOK_FACTOR = 0.8;
const double AUDIOBOOK_FACTOR = 1.2;
const double PAPERBOOK_FACTOR = 1.0;
```
- באיזו מחלקה או באילו מחלקות יש להוסיף את הקבועים הנ"ל? הסבירו את תשובתכם
- ג. (4 נק') כתבו פעולה או פעולות המחזירה את מחיר הספר (ללא אריות מתנה), עבור כל סוג ספר, יש לציין באיזו מחלקה צריך להוסיף את הפעולה או את הפעולות.
- ד. (3 נק') כתבו פעולה חיצונית המקבלת מערך ספרים המתאר את מלאי החנות וסכום כסף שלקוח רוצה להשקיע. הפעולה תדפיס את פרטי הספרים שמחירם אינו עולה על הסכום שהלקוח רוצה להשקיע.
- ה. (3 נק') כתבו פעולה חיצונית המקבלת מערך ספרים המתאר את מלאי החנות. הפעולה תדפיס את מספר הספרים שאפשר לרכוש אותם באריות מתנה.

8. הגדרה: עץ "שרשרת עולה" הוא עץ בינארי של מספרים שלמים העונה על הכלל הבא:

- לכל צומת פרט לעלים יש בדיוק בן אחד.
 - ערכים של כל הצמתים הם מספרים חיוביים זוגיים.
 - הערך של צומת הבן גדול מהערך של צומת האב.
- א. (4 נק') ציירו עץ "שרשרת עולה" הכולל לפחות 6 צמתים.
- ב. (14 נק') כתבו פעולה המקבלת הפנייה לשורש של עץ בינארי של מספרים שלמים. הפעולה תבדוק אם העץ הוא "שרשרת עולה".
- אם כן – הפעולה תחזיר ערך true, ולא – הפעולה תחזיר ערך false.

9. הערה כללית לשאלה:

- אין להשתמש במבני נתונים נוספים.
 - אפשר להניח שקיימת פעולה SizeQ המחזירה את מספר האיברים בתור. כותרת הפעולה:
- ```
public static int SizeQ(Queue<int> q)
```
- א. (7 נק') כתבו פעולה המקבלת תור של מספרים שלמים וחיוביים q ומספר שלם חיובי pos. הפעולה תחזיר איבר אשר נמצא במקום pos בתור בלי להוציא אותו מהתור.
- אם אין מקום pos בתור, הפעולה תחזיר ערך (-1).
- הערה: האיבר הנמצא בראש התור הוא איבר במקום 1, איבר אחריו – במקום 2 וכו'.
- כותרת הפעולה:
- ```
public static int ValueAt(Queue<int> q, int pos)
```
- ב. (7 נק') כתבו פעולה המקבלת 2 תורים של מספרים שלמים q1 ו-q2. הפעולה תיצור ותחזיר תור חדש לפי כלל הבא:
- קודם כול מכניסים איבר ראשון מהתור q1, ואחריו – איבר האחרון מהתור q2
 - איבר 2 מהתור q1, איבר לפני האחרון מהתור q2
 - וכך הלאה
- אפשר להניח שאורכם של 2 התורים שווה.
- כותרת הפעולה:
- ```
public static Queue<int> Merge (Queue<int>q1, Queue<int>q2)
```

ג. (4 נק') מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א' ו-ב'? הסבירו את תשובתכם

```

public class AA{
 public AA(){
 Console.WriteLine("AA constructor");
 }
 public virtual void One() {
 Console.WriteLine("One in AA");
 }
 public virtual void Two() {
 One();
 }
}
public class BB : AA{
 public BB(){
 Console.WriteLine("BB constructor");
 }
 public override void One(){
 Console.WriteLine("One in BB");
 }
 public override void Two() {
 Console.WriteLine("Two in BB");
 }
 public void Three() {
 base.Two();
 }
}

```

א. (14 נק') נתון קטע הקוד הבא (מספרי שורות לא חלק של הקוד):

```

(1) AA a = new AA();
(2) BB b = new BB();
(3) AA ab = new BB();
(4) a.One();
(5) ab.One();
(6) b.One();
(7) a.Two();
(8) ab.Two();
(9) b.Two();
(10) ((BB) ab).Three();
(11) b.Three();

```

1. עקבו אחרי ביצוע קטע הקוד ורשמו מה יהיה הפלט.

2. האם אפשר להוסיף לקטע הקוד את 2 הפקודות הבאות? הסבירו את תשובתכם.

```

a.Three();
((BB) a).Three();

```

ב. (4 נק') כתבו פעולה המקבלת מערך הפניות לעצמים מסוג Object. הפעולה תדפיס כמה עצמים

הם מסוג BB, כמה עצמים הם מסוג AA ולא סוג BB וכמה עצמים הם לא מסוג AA.

## חלק ג'

ענו על 1 מבין השאלות 11-12 (ערך שאלה – 16 נקודות).

11. ארגון סטודנטים מתנדבים "עזרה לגיל זהב" מחזיק מאגר מידע של סטודנטים הרוצים להתנדב ולעזור לקשישים, בצורה ממוחשבת. עבור כל סטודנט נשמרים נתונים הבאים:

- מספר תעודת הזהות (מחרוזת)
- שם (מחרוזת)
- יישוב מגורים (מחרוזת)
- אם יש לו רכב (ערך בוליאני, true – כן, false – לא)
- שפת האם (מחרוזת)
- שפה נוספת שסטודנט שולט בה ברמה גבוהה (מחרוזת)

א. (2 נק') כתבו את כותרת המחלקה ואת התכונות עבור כל אחת מהמחלקות הבאות:

- Student המתארת סטודנט
  - StudentsData המתארת מידע על כל סטודנטים
- אפשר להניח שבמחלקה Student קיימים בנאים, פעולות Get ו-Set לכל תכונה.
- לייצוג אוסף אפשר להשתמש במחלקות Stack, Queue, Node. אפשר להוסיף תכונות נוספות. חובה לתעד את התכונות.**

ב. (3 נק') בסיום שנת לימודים יש לעדכן מערכת ולהוציא ממאגר (אוסף נתונים) את הסטודנטים שסיימו ללמוד.

כתבו פעולה במחלקה StudentsData המקבלת מספר תעודת הזהות של המתנדב ומוציאה אותו ממאגר. כותרת הפעולה:

```
public void EraseStudent(string id)
```

ג. (4 נק') מדי פעם מתקבלת בקשה לעזרה מאזרח וותיק אשר מדבר בשפה מסוימת והמתגורר ביישוב מסוים. כתבו במחלקה StudentsData את הפעולה:

```
public void Print(string city, string lang)
```

הפעולה תדפיס את שמות המתנדבים אשר מדברים בשפה lang ומתגוררים ביישוב city. **אם אין אף מתנדב שגר ביישוב city ומדבר בשפה lang, הפעולה תדפיס את דוברי השפה lang שהם בעלי רכב.**

ד. (7 נק') הנהלת הארגון רוצה לארגן כנס למתנדבים. הוחלט שהכנס יתקיים בעיר שמתגוררים בה הכי הרבה מתנדבים חברי הארגון.

1. כתבו פעולה במחלקה StudentsData המקבלת שם של עיר ומחזירה את מספר המתנדבים המתגוררים בה. כותרת הפעולה:

```
public int NumStudents(string city)
```

2. כתבו פעולה במחלקה StudentsData המקבלת מערך של שמות ערים (מערך מסוג מחרוזת) ומחזירה את שם העיר שגרים בה הכי הרבה מתנדבים. כותרת הפעולה:

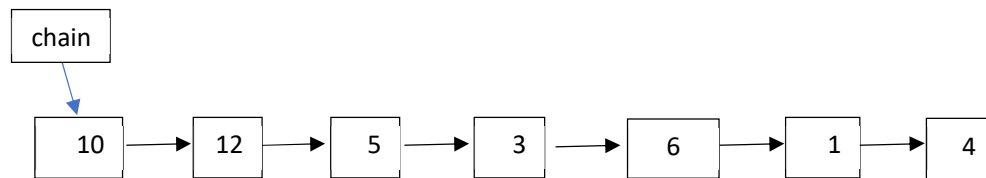
```
public string CityName(string[] cities)
```

**אפשר להניח שיש רק עיר אחת כזאת.**

א. (4 נק') נתונות פעולה רקורסיבית הבאה:

```
public static void One(Node<int> n){
 if(n!=null)
 {
 if(n.GetValue()% 2==0)
 {
 n.SetNext(new Node<int> (0,n.GetNext()));
 n = n.GetNext();
 }
 One(n.GetNext());
 }
}
```

עקבו אחרי זימון הפעולה one(chain) ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה.



ב. (2 נק') מה מבצעת הפעולה One באופן כללי?

ג. (4 נק') נתונה פעולה רקורסיבית Two הבאה:

```
public static void Two(Node<int> n){
 if(n.GetNext()!=null)
 {
 if(n.GetNext().GetValue()%2 == 0)
 n.SetNext(n.GetNext().GetNext());
 else
 n=n.GetNext();
 Two(n);
 }
}
```

עקבו אחרי זימון הפעולה Two(chain) ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה.

ד. (2 נק') מה מבצעת הפעולה Two באופן כללי?

ה. (4 נק') נתונה הפעולה Three המשתמשת בפעולה Two.

```
public static Node<int> Three(Node<int>n){
 if(n == null) return null;
 if(n.GetValue()%2 == 0)
 return Three(n.GetNext());
 else
 {
 Two(n);
 return n;
 }
}
```

מה יהיה תוכן של הרשימה chain אחרי ביצוע הפקודה הבאה:

chain = Three(chain);

מה מבצעת הפעולה Three באופן כללי?

**בהצלחה!**

© כל הזכויות שמורות למה"ט

## נספח לשאלון 97105 – מבני נתונים ותכנות מונחה עצמים – JAVA

נספח ממשקים מבנה הנתונים בתוכנית הלימודים

### ממשק המחלקה חוליה הגנרית- $\text{Node}<T>$

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

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

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

## ממשק המחלקה הגנרית - מחסנית $\text{Stack}\langle T \rangle$

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

|                          |                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Stack()</b>           | הפעולה בונה מחסנית ריקה.                                                                                                |
| <b>boolean isEmpty()</b> | הפעולה מחזירה "אמת" אם המחסנית הנוכחית ריקה, "שקר" אם היא אינה ריקה.                                                    |
| <b>void push (T x)</b>   | הפעולה מכניסה את הערך $x$ לראש המחסנית הנוכחית (דחיפה).                                                                 |
| <b>T pop()</b>           | הפעולה מוציאה את הערך שבראש המחסנית הנוכחית ומחזירה אותו (שליפה).<br><b>הנחה:</b> המחסנית הנוכחית אינה ריקה.            |
| <b>T top()</b>           | הפעולה מחזירה את הערך שבראש המחסנית הנוכחית מבלי להוציאו.<br><b>הנחה:</b> המחסנית הנוכחית אינה ריקה.                    |
| <b>String toString()</b> | הפעולה מחזירה תיאור של המחסנית, כסדרה של ערכים, במבנה הזה ( $x_1$ הוא האיבר שבראש המחסנית):<br>$[x_1, x_2, \dots, x_n]$ |

יעילות הפעולות- מחלקה מיוצגת בעזרת שרשרת חוליות.

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

## ממשק המחלקה הגנרית- תור $\text{Queue}\langle T \rangle$

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

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

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.

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

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

## נספח לשאלון 97105 – מבני נתונים ותכנות ונחה עצמים – #C

נספח ממשקים מבנה הנתונים בתוכנית הלימודים

### ממשק המחלקה חוליה הגנרית - $\text{Node}<\text{T}>$

המחלקה מגדירה חוליה גנרית שבה ערך מטיפוס  $\text{T}$  והפניה לחוליה העוקבת.

|                                    |                                                                                                    |
|------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Node</b> (T x)                  | הפעולה בונה חוליה. הערך של החוליה הוא x, ואין לה חוליה עוקבת.                                      |
| Node (T x, Node<T> next)           | הפעולה בונה חוליה. הערך של החוליה הוא x, והחוליה העוקבת לה היא next. ערכו של next יכול להיות null. |
| <b>T</b> GetValue()                | הפעולה מחזירה את הערך של החוליה.                                                                   |
| <b>Node&lt;T&gt;</b> GetNext()     | הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null.                            |
| <b>void</b> SetValue (T x)         | הפעולה משנה את הערך השמור בחוליה ל-x.                                                              |
| <b>bool</b> HasNext()              | הפעולה מחזירה true אם יש חוליה נוספת.                                                              |
| <b>void</b> SetNext (Node<T> next) | הפעולה משנה את החוליה העוקבת ל-next. ערכו של next יכול להיות null.                                 |
| <b>override string</b> ToString()  | הפעולה מחזירה מחרוזת המתארת את החוליה.                                                             |

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

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

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

|                                   |                                                                                                                       |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Stack</b> ()                   | הפעולה בונה מחסנית ריקה.                                                                                              |
| <b>bool</b> IsEmpty()             | הפעולה מחזירה "אמת" אם המחסנית הנוכחית ריקה, "שקר" אם היא אינה ריקה.                                                  |
| <b>void</b> Push (T x)            | הפעולה מכניסה את הערך x לראש המחסנית הנוכחית (דחיפה).                                                                 |
| <b>T</b> Pop()                    | הפעולה מוציאה את הערך שבראש המחסנית הנוכחית ומחזירה אותו (שליפה).<br><b>הנחה:</b> המחסנית הנוכחית אינה ריקה.          |
| <b>T</b> Top()                    | הפעולה מחזירה את הערך שבראש המחסנית הנוכחית בלי להוציאו.<br><b>הנחה:</b> המחסנית הנוכחית אינה ריקה.                   |
| <b>override string</b> ToString() | הפעולה מחזירה תיאור של המחסנית, כסדרה של ערכים, במבנה הזה $x_1$ הוא האיבר שבראש המחסנית):<br>$[x_1, x_2, \dots, x_n]$ |

יעילות הפעולות- מחלקה מיוצגת בעזרת שרשרת חוליות.

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

## ממשק המחלקה הגנרית - תור $Queue<T>$

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

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

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 $BinNode<T>$

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

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

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

**מילון עזר – בחינת מה"ט 97105 – מועד א' קיץ 24 – תשפ"ד**

**חלק א' - القسم أ**

| رقم السؤال | الكلمة/ التعبير بالعبرية | الكلمة/التعبير بالعربية  |
|------------|--------------------------|--------------------------|
| 1          | "מאוזנת"                 | "متوازنة"                |
| 1          | תוכן                     | محتوى                    |
| 2          | תחליף                    | استبدال                  |
| 3          | אין מילים                | لا توجد كلمات/بدون كلمات |
| 4          | כלי תחבורה               | وسائل مواصلات            |
| 4          | אופנוע/אופנועים          | دراجة نارية/دراجات نارية |
| 4          | מכונית מירוץ             | سيارة سباق               |
| 5          | אין מילים                | لا توجد كلمات/بدون كلمات |

**חלק ב' - القسم ب**

| رقم السؤال | الكلمة/ التعبير بالعبرية | الكلمة/التعبير بالعربية  |
|------------|--------------------------|--------------------------|
| 6          | המופעים                  | العروض                   |
| 7          | הפרש                     | فرق                      |
| 7          | סדרה                     | سلسلة                    |
| 7          | סדרה יורדת               | سلسلة تنازلية            |
| 7          | זוג                      | زوج                      |
| 8          | אין מילים                | لا توجد كلمات/بدون كلمات |
| 9          | אין מילים                | لا توجد كلمات/بدون كلمات |
| 10         | אין מילים                | لا توجد كلمات/بدون كلمات |

**חלק ג' - القسم ج**

| رقم السؤال | الكلمة/ التعبير بالعبرية | الكلمة/التعبير بالعربية  |
|------------|--------------------------|--------------------------|
| 11         | ארגון סטודנטים מתנדבים   | منظمة طلاب متطوعين       |
| 11         | "עזרה לגיל הזהב"         | "مساعدة لكبار السن"      |
| 11         | לקשישים                  | للمسنين                  |
| 11         | אזרח ותיק                | مواطن مُسن (كبير السن)   |
| 11         | דוברי השפה               | المتحدثين باللغة         |
| 12         | אין מילים                | لا توجد كلمات/بدون كلمات |