

## מבני נתונים ותכנות מונחה עצמים

### הנדסאים וטכנאים – הנדסת תוכנה

#### הנחיות לבחינה

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:

מבחן ב- Java (עמוד 2)

מבחן ב- C# (עמוד 15)

בכל מבחן 11 שאלות.

חלק א' – 45 נקודות

שאלות 1-4: יש לענות על שלוש שאלות בלבד. ערך כל שאלה 15 נקודות.

חלק ב' – 30 נקודות

שאלות 5-8: יש לענות על שתי שאלות בלבד. ערך כל שאלה 15 נקודות.

חלק ג' – 25 נקודות

שאלות 9-11: יש לענות על שתי שאלות בלבד. ערך כל שאלה 12 נקודות.

נקודה אחת תינתן על הערכה.

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 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!

## מבחן ב-JAVA

### חלק א'

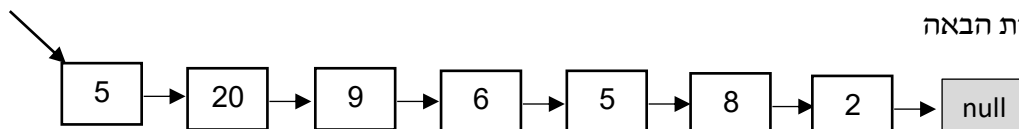
ענו על שלוש מבין השאלות 1-4 (ערך כל שאלה – 15 נקודות).

#### שאלה 1

"שרשרת ההפרשים" של שרשרת מטיפוס שלם היא שרשרת חוליות אשר כל אחד מאיבריה הוא ההפרש בערך מוחלט בין כל זוג איברים סמוכים מהשרשרת המקורית.

לדוגמה:

עבור השרשרת הבאה



שרשרת ההפרשים תהיה



$$|20-5|=15, |9-20|=11, |6-9|=3, |5-6|=1, |8-5|=3, |2-8|=6$$

7 נק') א. כתבו פעולה חיצונית differenceList המקבלת הפניה לחוליה ראשונה של שרשרת חוליות ומחזירה הפניה לחוליה הראשונה של "שרשרת ההפרשים".

8 נק') ב. כתבו פעולה חיצונית בשם theSurvives המקבלת הפניה לחוליה ראשונה של שרשרת חוליות. על הפעולה לחשב לכל שרשרת הפרשים את שרשרת ההפרשים שלה עד שמתקבלת רשימה עם איבר אחד. איבר זה הוא ה"שורד". על הפעולה להדפיס את כל השרשראות המתקבלות ולהחזיר את ערכו של האיבר ה"שורד".

מומלץ להשתמש בפעולה שכתבתם בסעיף א'.

לדוגמה:

5 , 20 , 9 , 6 , 5 , 8 , 2

15 , 11 , 3 , 1 , 3 , 6

4 , 8 , 2 , 2 , 3

4 , 6 , 0 , 1

2 , 6 , 1

4 , 5

1

הפעולה תחזיר 1

## שאלה 2

(12 נק') א. כתבו פעולה חיצונית המקבלת מחסנית שאיבריה מטיפוס שלם. הפעולה תחזיר true אם ערכו שלכל איבר במחסנית (פרט לאיבר הנמצא בראש המחסנית) גדול מסכום ערכם של כל האיברים שנמצאים לפניו במחסנית, אם לא הפעולה תחזיר false. אפשר להשתמש במבני נתונים נוספים.

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

## שאלה 3

נתונות ארבע המחלקות הבאות :

```
public class A {
    protected int x;
    public A() {
        x = 0;
    }
    public void method(int num) {
        x += num;
    }
    public String toString() {
        return "" + x;
    }
} // end of class A

public class B extends A{
    private int y;
    public B()
    {
        super();
        y = 1;
    }
    public void method(int num) {
        x -= num;
        y += num;
    }
    public String toString() {
        return "" + x + "." + y;
    }
} // end of class B
```

```
public class C extends A{
    private int y;
    public C()
    {
        super();
        y = 10;
    }
    public void method(int num) {
        x += num;
        y += num;
    }
    public String toString() {
        return "" + x + "/" + y;
    }
} // end of class C
```

```
public class Test
{
    public static void main (String [] args)
    {
        A [] array = new A[4];
        array[0] = new C();
        array[1] = new B();
        array[2] = new A();
        array[3] = new B();

        for (int i = 0; i < array.length; i ++)
        {
            System.out.println ("array[" + i + "] = " + array[i]);
            array[i].method(i+2);
            System.out.println ("array[" + i + "] = " + array[i]);
        }
    }
}
```

עקבו אחרי ביצוע הפעולה הראשית של המחלקה Test וכתבו מה יהיה הפלט של הפעולה.  
חובה להציג באיור את כל העצמים שנוצרו ואת התכונות שלהם.

נתונות שלוש המחלקות הבאות:

```

public class A {
    protected int num;
    public A(int n)
    {
        num = n;
    }
    public int getNum()
    {
        return num;
    }

    public boolean f(A a)
    {
        System.out.println("F in A");
        return num == a.num * 2;
    }
} // end of class A

public class B extends A {
    public B(int n)
    {
        super(n);
    }
    public boolean f(B b)
    {
        System.out.println("F in B");
        return num == b.num;
    }
} // end of class B

public class C extends A {
    public C (int n)
    {
        System.out.println("F in C");
        super(n);
    }
    public boolean f(A a)
    {
        return a instanceof C && num == a.num;
    }
} //end of class C

```

(3 נק') א. האם קיימת דריסה (overriding) או העמסה (overloading) של הפעולות? הסבירו את תשובתכם.

```

public class Driver
{
    public static void main(String[] args)
    {
        A y1 = new B(10);
        B y2 = new B(10);
        A z1 = new C(10);
        C z2 = new C(10);
        (****)
    }
}

```

(12 נק') ב. בכל אחד מסעיפים הבאים השורה (\*\*\*\*) תוחלף בשורת קוד משורות הקוד 1-6.

כתבו עבור כל אחד מהסעיפים, אם קוד תקין או לא. אם הקוד תקין, יש לציין מה יהיה הפלט, אם הקוד אינו תקין – יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (קומפילציה או זמן ריצה).

**שימו לב שאין קשר בין סעיפים!**

1. `System.out.println(y1.getNum() == ( (B) z1 ).getNum() );`
2. `System.out.println(y1.f(y2));`
3. `System.out.println(y2.f(y1));`
4. `System.out.println(z1.f(z2));`
5. `System.out.println(z1.f(y1));`
6. `System.out.println(z2.f( (C) y2 ));`

## חלק ב'

ענו על שתיים מבין השאלות 5-8 (ערך כל שאלה – 15 נקודות).

### שאלה 5

פירמידה של ליצנים מתארת ליצנים העומדים זה על זה.

נתונה המחלקה Clown (ליצן) הבאה:

```
class Clown
{
    private String name; // שם ליצן
    private int weight; // משקל ליצן
}
```

אפשר להניח כי במחלקה הוגדרו בנאי ופעולות get לכל תכונה, אין צורך לממש.

(3 נק') א. הגדירו מחלקה Pyramid המתארת פירמידה של ליצנים. יש לבחור תכונות של המחלקה ולכתוב בנאי ליצירת פירמידה ריקה.

שימו לב, שמספר הליצנים בפירמידה אינו ידוע מראש. אפשר להשתמש בכל אחת ממבני הנתונים הבאים: מחסנית, תור, שרשרת חוליות.

פירמידה תיקרא **פירמידה יציבה** אם הליצן שבתחתית הפירמידה הוא הכבד ביותר, מעליו האדם שמשקלו הוא במקום השני וכך הלאה עד שבראש הפירמידה נמצא הליצן בעל המשקל הקל ביותר.

(6 נק') ב. כתבו פעולה פנימית הבודקת אם הפירמידה היא "פירמידה יציבה". כותרת הפעולה:

```
public boolean isStable()
```

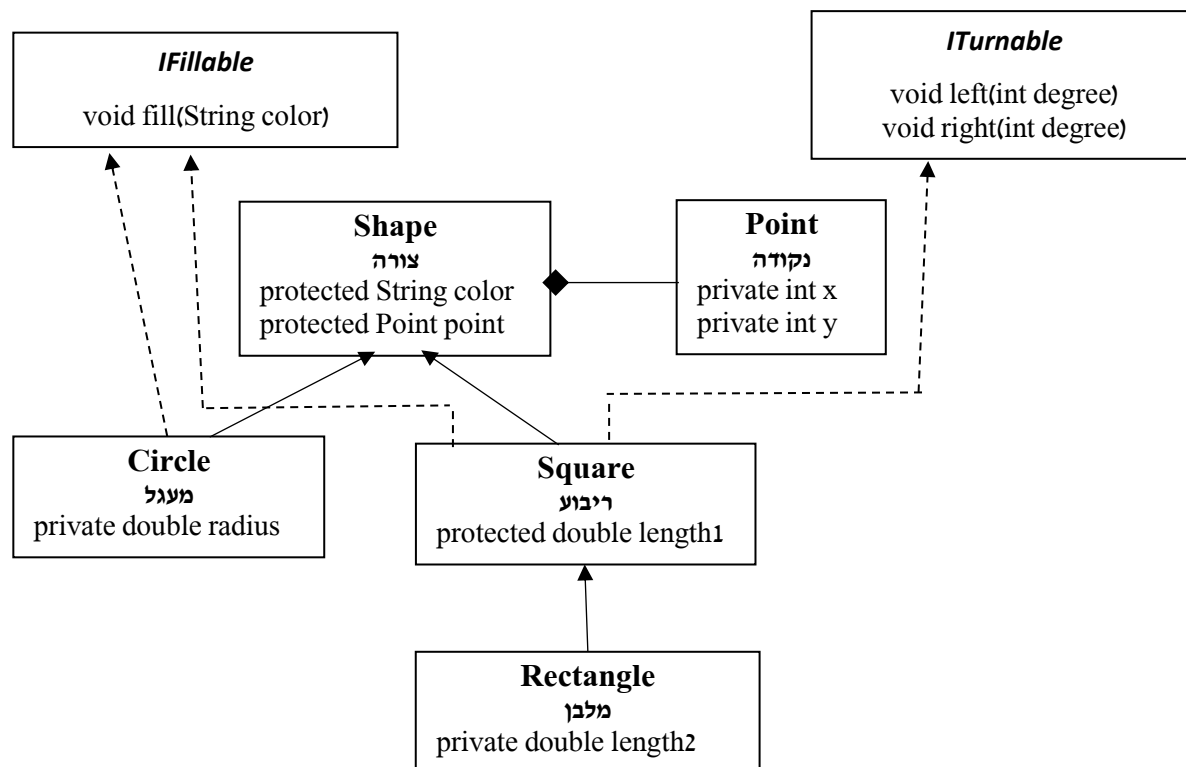
(6 נק') ג. כתבו פעולה בוליאנית המקבלת ליצן. על הפעולה להוסיף את הליצן לפירמידה כך שהיא תישאר יציבה. אם הדבר אפשרי – הפעולה מוסיפה את הליצן ומחזירה true, ואם לא – היא מחזירה false. כותרת הפעולה:

```
public boolean addClown(Clown c)
```

שימו לב: אם הפירמידה לא יכולה להישאר יציבה או שהיא אינה יציבה מלכתחילה – אין להוסיף ליצן.

## שאלה 6

נתונה היררכיית המחלקות הבאה הכוללת חמש מחלקות Rectangle, Square, Circle, Shape, Point ושני ממשקים IFillable ו-ITurnable.



5 נק') א. כתבו עבור כל מחלקה, את כותרת המחלקה ואת הכותרות של כל הפעולות שחייבות להיות במחלקה. נתון קטע קוד בפעולה הראשית (main):

```
Shape s = new Square(new Point(50,50),10);
Shape c = new Circle(new Point(50,50),8);
ITurnable t; IFillable f;
(***)
```

3 נק') ב. כתבו כותרות לכל הבנאים הנדרשים להרצה תקינה של קטע הקוד.

7 נק') ג. בהתייחס לכל אחד מהקטעים הבאים, כתבו מה יקרה בעקבות הוספתו לשיטה main שלעיל, במקום שמסומן בכוכביות \*\*\*.

אם הקוד אינו תקין יש להסביר ולציין את סוג השגיאה (קומפילציה או זמן ריצה).

- (1) `f = (IFillable) c;`
- (2) `t = s;`  
`t.fill("Green");`
- (3) `f = c;`  
`f.fill("Red");`
- (4) `((Circle)s).left(45);`
- (5) `Shape d = c;`  
`d.fill("Yellow");`
- (6) `f = new Rectangle(new Point(10,10),10,20);`  
`f.left(Math.PI);`
- (7) `c = s;`  
`c.left(90);`



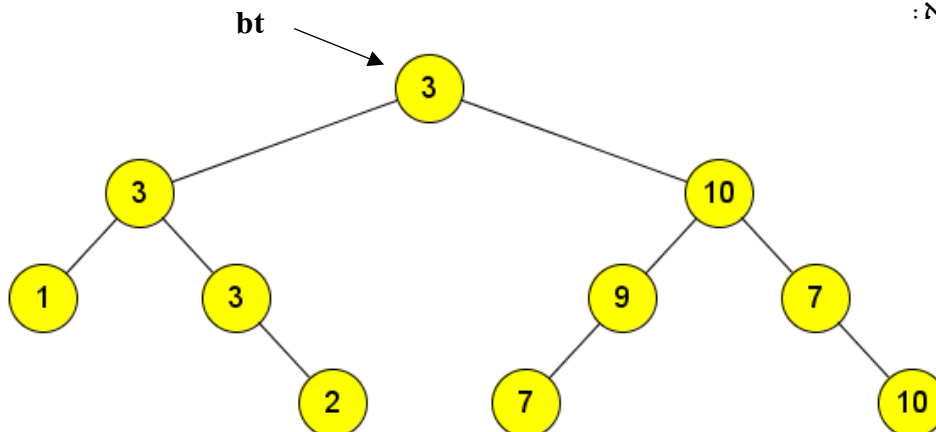
```

public static void what1(BinNode<Integer> bt)
{
    if(bt!=null){
        int x=bt.getValue();
        if(what2(bt.getLeft(),x) && what2(bt.getRight(),x))
            System.out.println(x);
        what1(bt.getLeft());
        what1(bt.getRight());
    }
}

public static boolean what2(BinNode<Integer> bt, int x)
{
    if(bt==null)return true;
    if(bt.getValue()==x)return false;
    return what2(bt.getLeft(), x) && what2(bt.getRight(), x);
}

```

נתון העץ הבינרי הבא:



- (3 נק') א. מה תהיה תוצאת זימון הפעולה `what2(bt, 10)`? **חובה להראות את המעקב!**
- (1 נק') ב. תנו דוגמה של הפרמטר `x`, שעבורו זימון הפעולה `what2(bt, x)` יחזיר תוצאה שונה מזו שהתקבלה בסעיף א'.
- (3 נק') ג. מה מבצעת הפעולה `what2(BinNode<Integer> bt, int x)` באופן כללי?
- (5 נק') ד. מה תהיה תוצאת הזימון `what1(bt)` לעץ הנתון? **חובה להראות את המעקב!**
- הערה: אין צורך במעקב אחרי הפעולה `what2`
- (3 נק') ה. מה מבצעת הפעולה `what1(bt)` באופן כללי?

## שאלה 8

(4 נק') א. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה סכום איברים שנמצאים בתור.

כותרת הפעולה

```
public static int sumQueue(Queue<Integer> q)
```

(4 נק') ב. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה את הערך של המספר האחרון שנמצא בתור (את הערך הנמצא בסוף התור).

כותרת הפעולה

```
public static int lastValue(Queue<Integer> q)
```

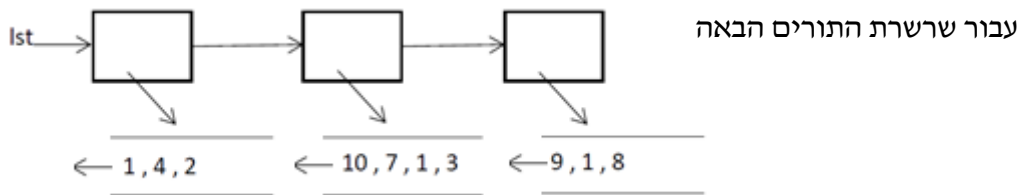
**שרשרת תורים** היא שרשרת חוליות מסוג `Node<Queue<Integer>>`. כלומר, הערך (value) של כל חוליה הוא הפניה לתור של מספרים שלמים.

(4 נק') ג. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת תורים.

**הפעולה תחזיר תור** של מספרים שלמים באופן הבא:

עבור כל תור שנמצא בשרשרת תורים שהערך הנמצא בראשו הוא **אי זוגי**, יוכנס לתור החדש ערך השווה לסכום האיברים בתור, ואילו עבור כל תור שערך הנמצא בראשו הוא **זוגי** הפעולה תכניס לתור החדש את הערך הנמצא בסוף התור.

**לדוגמה:**



**הפעולה תחזיר תור חדש הבא:**

← 7, 3, 18

- עבור החוליה הראשונה יתקבל **סכום** של איברים כי הערך בראש התור הוא **אי זוגי**.  
סכום הוא  $1+4+2 = 7$
- עבור החוליה השנייה יתקבל **מספר בסוף** התור (3) כי הערך בראש התור הוא **זוגי**.
- עבור החוליה השלישית יתקבל **סכום** של איברים כי ערך בראש התור הוא **אי זוגי**.  
סכום הוא  $9+1+8 = 18$

(3 נק') ד. מהי סיבוכיות זמן הריצה של הפעולה שכתבתם בסעיף ג'? הסבירו את תשובתכם!

## חלק ג'

ענו על שתיים מבין השאלות 9-11 (ערך כל שאלה – 12 נקודות).

### שאלה 9

לשם קידום מכירת דירות חברת "קנה ביתך" פותח פרויקט לניהול שיווק דירות הכולל את המחלקות הבאות:

Appartment – דירה רגילה.

GardenApp – דירת גן.

Penthouse – דירת פנטהאוז.

Luxary – דירת פאר.

לדירה (Appartment) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, String.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.

לדירה גן (GardenApp) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, String.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- gardenArea – שטח גינה, מטיפוס שלם, int.

לדירת פנטהאוז (Penthouse) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, String.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- numTerr – מספר מרפסות, מטיפוס שלם, int.
- terraceArea – השטח הכולל של המרפסות, מטיפוס שלם, int.
- seaView – נוף לים, כן/לא, מטיפוס boolean.

לדירת מיני-פנטהאוז (Luxary) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, String.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- terraceArea – שטח מרפסת, מטיפוס שלם, int.

(2 נק') א. סרטטו תרשים UML המתאר את הקשר בין המחלקות:

**Appartment, GardenApp, Penthouse, Luxary**

באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

(2 נק') ב. לכל אחת מהמחלקות **Appartment, GardenApp, Penthouse, Luxary** כתבו:

– כותרת המחלקה.

– תכונות.

למחלקה Appartment הוגדרה פעולה הבונה (בנאי, constructor) המייצרת דירה רגילה מוכנה למכירה:

```
public Appartment(int floor, int numApp, int area)
{
    this.owner = "FREE";
    this.floor = floor;
    this.numApp = numApp;
    this.area = area;
}
```

ופעולה הקובעת את שם בעל הדירה אחרי המכירה:

```
public void setOwner(String name)
{
    this.owner = name;
}
```

המחיר של כל דירה תלוי בשטח הדירה, בשטח המרפסת (אם יש) ובשטח הגינה (אם יש). לשם כך הוחלט להגדיר שלושה קבועים המציינים מחיר למטר מרובע של דירה, מרפסת וגן, בהתאם:

```
final int COST_APP = 1000;
final int COST_TERRACE = 300;
final int COST_GARDEN = 50;
```

(1 נק') ג. באיזו מחלקה או באילו מחלקות יש להוסיף את הקבועים הנ"ל? הסבירו את תשובתכם.

כדי לקדם את המכירות של דירות פנטהאוז הוחלט לתת הנחה של 5% לדירות שאין להן נוף לים.

(4 נק') ד. כתבו פעולה או פעולות לחישוב ולהחזרה של מחיר הדירה. יש לציין באיזו מחלקה צריך להוסיף את הפעולה או את הפעולות.

(3 נק') ה. כתבו פעולה חיצונית המקבלת מערך דירות המתאר בניין מסוים וסכום כסף שלקוח רוצה להשקיע. הפעולה תדפיס פרטי דירות שפנויות למכירה ומחירן אינו עולה על הסכום שאותו הלוקח רוצה להשקיע.

**הערה:** הניחו שהפעולות get ו-toString מוגדרות בעבור כל תכונה בכל אחת מהמחלקות הפרויקט.

## שאלה 10

במסגרת המלחמה בתאונות הדרכים, הציבו בכבישים ברחבי הארץ מצלמות אוטומטיות (מכמונות) למדידת מהירות כלי הרכב.

בכל כביש מציבים מצלמה אחת לכל היותר.

מספר המצלמות שמפזרים בכבישים עשוי להשתנות מעת לעת עפ"י החלטת המשטרה.

במשטרת התנועה מנהלים מערכת ממוחשבת המרכזת את המידע לגבי מיקום המצלמות, וכן מידע על מכוניות אשר צולמו על ידי המצלמה.

**מצלמה מהירות SpeedCamera מיוצגת באמצעות :**

- קוד המצלמה.
- מספר הכביש שבו היא מוצבת.
- המהירות המקסימאלית המותרת בכביש זה.
- אוסף מספרי כלי הרכב אשר נסעו במהירות גבוהה מהמהירות המותרת בכביש זה.

**(1 נק')** א. כתבו את כותרת המחלקה SpeedCamera ואת התכונות שלה.

**לייצוג אוסף אפשר להשתמש במחלקות Stack, Queue, Node.**

**אפשר להוסיף תכונות נוספות.**

**חובה לתעד את התכונות.**

**(1 נק')** ב. כתבו את הבנאי של המחלקה המקבל את קוד המצלמה, את מספר הכביש ואת המהירות המותרת. הבנאי מאתחל אוסף מספרי כלי הרכב להיות ריק.

**(2 נק')** ג. כתבו פעולה addCar. הפעולה מקבלת את מספר מכונית שצולמה ואת המהירות שלה. אם המהירות גבוהה מהמותר בכביש זה, הפעולה תוסיף מכונית זו לאוסף של המכוניות שנסעו מהר מהמותר.

**הערה :** הניחו שבמחלקה SpeedCamera קיימות פעולות get ואין צורך לממש אותן.

בכל יום מציבה המשטרה 100 מצלמות לכל היותר (ייתכן אף פחות). **רשת המצלמות** המוצבות בכבישים ביום נתון, מיוצגת באמצעות המחלקה SpeedCameraNetwork.

**(1 נק')** ד. כתבו את כותרת המחלקה SpeedCameraNetwork ואת התכונות שלה.

**(1 נק')** ה. כתבו את הפעולה addSpeedCamera המוסיפה את מצלמת המהירות sc לרשת המצלמות. אפשר להניח שהמצלמה sc אינה מופיע ברשת המצלמות.

במסגרת פעולות האכיפה מעוניינים לדעת מהם הכבישים המצריכים אכיפה מוגברת. כביש הדורש אכיפה מוגברת הוא כביש שבוצע בו מעל 200 עבירות מהירות ליום.

**(3 נק')** ו. כתבו פעולה המדפיסה את מספרי הכבישים שבהם יש לבצע אכיפה מוגברת. ציינו באיזו מחלקה יש לכתוב את הפעולה.

במשטרה רוצים לבדוק אם רכב מסוים צולם כאשר נסע במהירות הגבוהה מהמותר.

**(3 נק')** ז. כתבו פעולה המקבלת מספר רכב. הפעולה תבדוק אם הרכב צולם כאשר נסע במהירות גבוהה מהמותר. אם כן – הפעולה תדפיס את מספר המצלמה אשר צילמה את העבירה ותחזיר true, ואם לא – הפעולה תחזיר false.

**הערה :** אם רכב עשה כמה עבירות יש להדפיס את המספרים של כל המצלמות שצילמו את העבירות.

## שאלה 11

נתונה פעולה הבאה:

```

public static boolean what(BinNode<Integer> t)
{
    if (t==null) return true;
    int x = t.getValue();
    if (t.getLeft()!=null && t.getLeft().getValue()>=x)
        return false;
    if (t.getRight()!=null && t.getRight().getValue()<x)
        return false;
    return what(t.getLeft()) && what(t.getRight());
}

```

נתון עץ בינארי  $t$  שעבורו הפעולה  $\text{what}(t)$  מחזירה ערך  $\text{true}$ . עבור כל אחת מטענות הבאות ענו אם היא נכונה או לא. אם הטענה אינה נכונה, יש לצייר דוגמת עץ סותרת.

1. אם בשורש העץ הופיע המספר 13, יכול להופיע שוב רק כבנו הימני של השורש.
2. אם בשורש העץ הופיע המספר 13, יכול להופיע שוב רק בתת-עץ הימני של השורש.
3. אם בשורש העץ הופיע המספר 13, יכול להופיע במקומות רבים בעץ, כולל תת העץ השמאלי, אך לא כבנו של צומת שערכו 13 או פחות.
4. לא ייתכנו בעץ  $t$  שני אחים בעלי ערך זהה.
5. אם אין בעץ  $t$  אף בן ימני (עץ 'שורוך'), סריקתו בסדר תחילי תדפיס ערכים ממוינים מהגדול לקטן.
6. סריקת העץ בסדר תוכי תדפיס תמיד ערכים ממוינים.

## מבחן ב- C#

### חלק א'

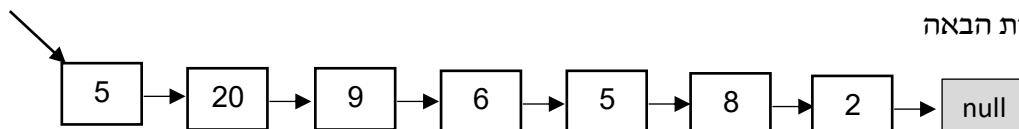
ענו על שלוש מבין השאלות 1-4 (ערך כל שאלה – 15 נקודות).

#### שאלה 1

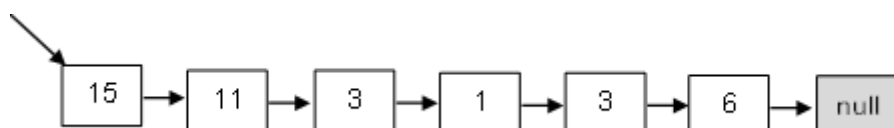
"שרשרת ההפרשים" של שרשרת מטיפוס שלם היא שרשרת חוליות אשר כל אחד מאיבריה הוא ההפרש בערך מוחלט בין כל זוג איברים סמוכים מהשרשרת המקורית

לדוגמה:

עבור השרשרת הבאה



שרשרת ההפרשים תהיה



$$|20-5|=15, |9-20|=11, |6-9|=3, |5-6|=1, |8-5|=3, |2-8|=6$$

7 נק') א. כתבו פעולה חיצונית DifferenceList המקבלת הפניה לחוליה ראשונה של שרשרת חוליות ומחזירה הפניה לחוליה הראשונה של "שרשרת ההפרשים".

8 נק') ב. כתבו פעולה חיצונית בשם TheSurvives המקבלת הפניה לחוליה ראשונה של שרשרת חוליות. על הפעולה לחשב לכל שרשרת הפרשים את שרשרת ההפרשים שלה עד שמתקבלת רשימה עם איבר אחד. איבר זה הוא ה"שורד". על הפעולה להדפיס את כל השרשראות המתקבלות ולהחזיר את ערכו של האיבר ה"שורד".

לדוגמה:

5 , 20 , 9 , 6 , 5 , 8 , 2

15 , 11 , 3 , 1 , 3 , 6

4 , 8 , 2 , 2 , 3

4 , 6 , 0 , 1

2 , 6 , 1

4 , 5

1

הפעולה תחזיר 1

## שאלה 2

(12 נק') א. כתבו פעולה חיצונית המקבלת מחסנית שאיבריה מטיפוס שלם. הפעולה תחזיר true אם ערכו שלכל איבר במחסנית (פרט לאיבר הנמצא בראש המחסנית) גדול מסכום ערכם של כל האיברים שנמצאים לפניו במחסנית, אם לא הפעולה תחזיר false. אפשר להשתמש במבני נתונים נוספים.

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

## שאלה 3

נתונות ארבע המחלקות הבאות :

```
public class A {
    protected int x;
    public A() {
        x = 0;
    }
    public virtual void Method(int num) {
        x += num;
    }
    public override string ToString() {
        return "" + x;
    }
} // end of class A

public class B : A{
    private int y;
    public B()
    {
        y = 1;
    }
    public override void Method(int num) {
        x -= num;
        y += num;
    }
    public override string ToString () {
        return "" + x + "." + y;
    }
} // end of class B
```



```
public class C : A{
    private int y;
    public C()
    {

y = 10;
    }
    public override void Method(int num)    {
        x += num;
        y += num;
    }
    public override string ToString()    {
        return "" + x + "/" + y;
    }
} // end of class C


public class Test
{
    public static void Main (string [] args)
    {
        A [] array = new A[4];
        array[0] = new C();
        array[1] = new B();
        array[2] = new A();
        array[3] = new B();

        for (int i = 0; i < array.length; i ++)
        {
            Console.WriteLine("array[" + i + "] = " + array[i]);
            array[i].Method(i+2);
            Console.WriteLine("array[" + i + "] = " + array[i]);
        }
    }
}
```

עקבו אחרי ביצוע הפעולה הראשית של המחלקה Test כתבו מה יהיה הפלט של הפעולה.  
חובה להציג באיור את כל העצמים שנוצרו ואת התכונות שלהם.

נתונות שלוש המחלקות הבאות:

```

public class A {
    protected int num;
    public A(int n)
    {
        num = n;
    }

    public int GetNum()
    {
        return num;
    }

    public virtual bool F(A a)
    {
        Console.WriteLine("F in A");
        return num == a.num * 2;
    }
} // end of class A

public class B : A {
    public B(int n): base(n)
    {
    }

    public bool F(B b)
    {
        Console.WriteLine("F in B");
        return num == b.num;
    }
} // end of class B

public class C : A {
    public C (int n): base(n)
    {
    }

    public override bool F(A a)
    {
        Console.WriteLine("F in C");
        return a is C && num == ((C)a).num;
    }
} //end of class C

```

(3 נק') א. האם קיימת דריסה (overriding) או העמסה (overloading) של הפעולות? הסבירו את תשובתכם.

```
public class Driver
{
    public static void Main(string[] args)
    {
        A y1 = new B(10);
        B y2 = new B(10);
        A z1 = new C(10);
        C z2 = new C(10);
        (****)
    }
}
```

(12 נק') ב. בכל אחד מסעיפים הבאים השורה (\*\*\*\*) תוחלף בשורת קוד משורות הקוד 1-6.

כתבו עבור כל אחד מהסעיפים, אם קוד תקין או לא. אם הקוד תקין, יש לציין מה יהיה הפלט, אם הקוד אינו תקין – יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (קומפילציה או זמן ריצה).

**שימו לב שאין קשר בין סעיפים!**

1. Console.WriteLine(y1.GetNum() == (B) z1).GetNum());
2. Console.WriteLine(y1.F(y2));
3. Console.WriteLine(y2.F(y1));
4. Console.WriteLine(z1.F(z2));
5. Console.WriteLine(z1.F(y1));
6. Console.WriteLine(z2.F((C) y2));

## חלק ב'

ענו על שתיים מבין השאלות 5-8 (ערך כל שאלה – 15 נקודות).

### שאלה 5

פירמידה של ליצנים מתארת ליצנים העומדים זה על זה.

נתונה המחלקה Clown (ליצן) הבאה:

```
class Clown
{
    private string name; // שם ליצן
    private int weight; // משקל ליצן
}
```

אפשר להניח כי במחלקה הוגדרו בנאי ופעולות Get לכל תכונה, אין צורך לממש.

(3 נק') א. הגדירו מחלקה Pyramid המתארת פירמידה של ליצנים. יש לבחור תכונות של המחלקה ולכתוב בנאי ליצירת פירמידה ריקה.

**שימו לב, מספר ליצנים בפירמידה אינו ידוע מראש. אפשר להשתמש בכל אחד ממבני הנתונים הבאים: מחסנית, תור, שרשרת חוליות.**

פירמידה תיקרא **פירמידה יציבה** אם הליצן שבתחתית הפירמידה הוא הכבד ביותר, מעליו האדם שמשקלו הוא במקום השני וכך הלאה עד שבראש הפירמידה נמצא הליצן בעל המשקל הקל ביותר.

(6 נק') ב. כתבו פעולה פנימית הבודקת אם הפירמידה היא "פירמידה יציבה". כותרת הפעולה:

```
public bool IsStable()
```

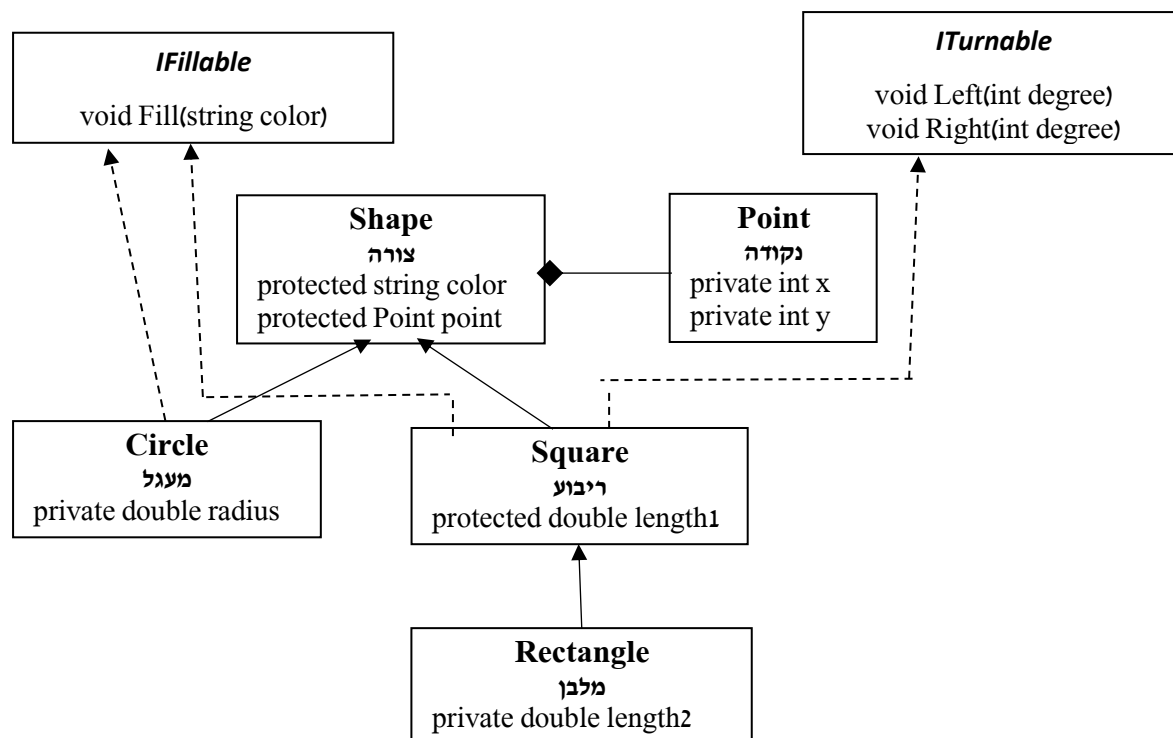
(6 נק') ג. כתבו פעולה בוליאנית המקבלת ליצן. על הפעולה להוסיף את הליצן לפירמידה כך שהיא תישאר יציבה. אם הדבר אפשרי – הפעולה מוסיפה את הליצן ומחזירה true, ואם לא – היא מחזירה false. כותרת הפעולה:

```
public bool AddClown(Clown c)
```

**שימו לב: אם הפירמידה אינה יכולה להישאר יציבה או שהיא אינה יציבה מלכתחילה – אין להוסיף ליצן.**

## שאלה 6

נתונה היררכיית המחלקות הבאה הכוללת חמש מחלקות Rectangle, Square, Circle, Shape, Point ושני ממשקים IFillable ו-ITurnable.



5 נק') א. כתבו עבור כל מחלקה, את כותרת המחלקה ואת הכותרות של כל הפעולות שחייבות להיות במחלקה. נתון קטע קוד בפעולה הראשית (Main):

```

Shape s = new Square(new Point(50,50),10);
Shape c = new Circle(new Point(50,50),8);
ITurnable t;
IFillable f;
(***)

```

3 נק') ב. כתבו כותרות לכל הבנאים הנדרשים להרצה תקינה של קטע הקוד הנתון.

7 נק') ג. בהתייחס לכל אחד מהקטעים הבאים, כתבו מה יקרה בעקבות הוספתו לשיטה Main שלעיל, במקום שמסומן בכוכביות \*\*\*.

אם הקוד אינו תקין יש להסביר ולציין את סוג השגיאה (קומפילציה או זמן ריצה)

- (1) `f=(IFillable) c;`
- (2) `t = s;`  
`t.Fill("Green");`
- (3) `f = c;`  
`f.Fill("Red");`
- (4) `((Circle)s).Left (45);`
- (5) `Shape d = c;`  
`d.Fill("Yellow");`
- (6) `f = new Rectangle(new Point(10,10),10,20);`  
`f.Left(Math.PI);`
- (7) `c = s;`  
`c.Left (90);`

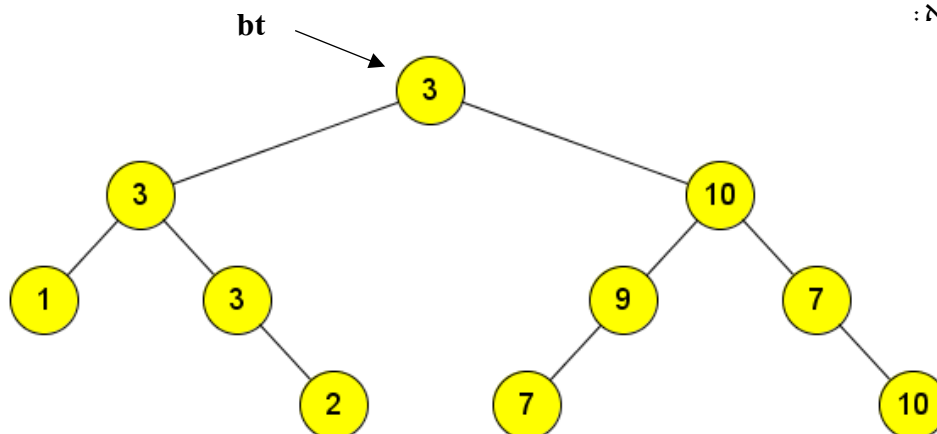
```

public static void What1(BinNode<int> bt)
{
    if(bt!=null){
        int x=bt.GetValue();
        if(What2(bt.GetLeft(),x) && What2(bt.GetRight(),x))
            Console.WriteLine(x);
        What1(bt.GetLeft());
        What1(bt.GetRight());
    }
}

public static bool What2(BinNode<int> bt, int x)
{
    if(bt==null) return true;
    if(bt.GetValue()==x) return false;
    return What2(bt.GetLeft(), x) && What2(bt.GetRight(), x);
}

```

נתון העץ הבינרי הבא:



(3 נק') א. מה תהיה תוצאת זימון הפעולה What2(bt, 10)? חובה להראות את המעקב!

(1 נק') ב. תנו דוגמה של הפרמטר x, שעבורו זימון הפעולה What2(bt, x) יחזיר תוצאה שונה מזו שהתקבלה בסעיף א'.

(3 נק') ג. מה מבצעת הפעולה What2(BinNode&lt;int&gt; bt, int x) באופן כללי?

(5 נק') ד. מה תהיה תוצאת הזימון What1(bt) לעץ הנתון bt? חובה להראות את המעקב!

הערה: אין צורך במעקב אחרי הפעולה What2

(3 נק') ה. מה מבצעת הפעולה What1(bt) באופן כללי?

## שאלה 8

(4 נק') א. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה סכום איברים שנמצאים בתור.

כותרת הפעולה

```
public static int SumQueue(Queue<int> q)
```

(4 נק') ב. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה את הערך של המספר האחרון שנמצא בתור (את הערך הנמצא בסוף התור).

כותרת הפעולה

```
public static int LastValue(Queue<int> q)
```

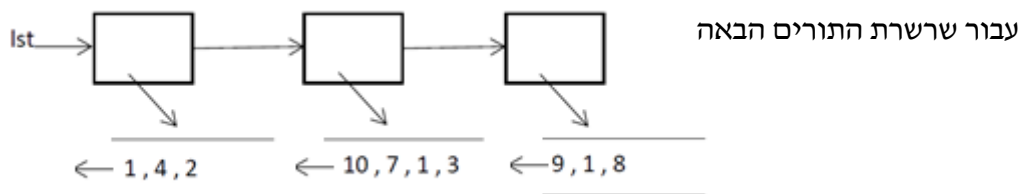
**שרשרת תורים** היא שרשרת חוליות מסוג `Node<Queue<int>>`. כלומר, הערך (value) של כל חוליה הוא הפניה לתור של מספרים שלמים.

(4 נק') ג. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת תורים.

**הפעולה תחזיר תור** של מספרים שלמים באופן הבא:

עבור כל תור שנמצא בשרשרת תורים שהערך הנמצא בראשו הוא **אי זוגי**, יוכנס לתור החדש ערך השווה לסכום האיברים בתור ואילו עבור כל תור שערך הנמצא בראשו הוא **זוגי** הפעולה תכניס לתור החדש את הערך הנמצא בסוף התור.

**לדוגמה:**



**הפעולה תחזיר תור חדש הבא:**

← 7, 3, 18

- עבור החוליה הראשונה יתקבל **סכום** של איברים כי הערך בראש התור הוא **אי זוגי**.  
הסכום הוא  $1+4+2 = 7$ .
- עבור החוליה השנייה יתקבל **מספר בסוף** התור (3) כי הערך בראש התור הוא **זוגי**.
- עבור החוליה השלישית יתקבל **סכום** של איברים כי הערך בראש התור הוא **אי זוגי**.  
הסכום הוא  $9+1+8 = 18$ .

(3 נק') ד. מהי סיבוכיות זמן הריצה של הפעולה שכתבתם בסעיף ג'? הסבירו את תשובתכם!

## חלק ג'

ענו על שתיים מבין השאלות 9-11 (ערך כל שאלה – 12 נקודות).

### שאלה 9

לשם קידום מכירת דירות של חברת "קנה ביתך" פותח פרויקט לניהול שיווק דירות הכולל את המחלקות הבאות:

Appartment – דירה רגילה.

GardenApp – דירת גן.

Penthouse – דירת פנטהאוז.

Luxary – דירת פאר.

לדירה (Appartment) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, string.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.

לדירה גן (GardenApp) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, string.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- gardenArea – שטח גינה, מטיפוס שלם, int.

לדירת פנטהאוז (Penthouse) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, string.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- numTerr – מספר מרפסות, מטיפוס שלם, int.
- terraceArea – השטח הכולל של המרפסות, מטיפוס שלם, int.
- seaView – נוף לים, כן/לא, מטיפוס bool.

לדירת מיני-פנטהאוז (Luxary) יש התכונות הבאות:

- owner – שם בעל דירה, מטיפוס מחרוזת, string.
- floor – מספר קומה, מטיפוס שלם, int.
- numApp – מספר דירה, מטיפוס שלם, int.
- area – שטח דירה, מטיפוס שלם, int.
- terraceArea – שטח מרפסת, מטיפוס שלם, int.



(2 נק') א. סרטטו תרשים UML המתאר את הקשר בין המחלקות:

**Appartment, GardenApp, Penthouse, Luxary**

באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

(2 נק') ב. לכל אחת מהמחלקות **Appartment, GardenApp, Penthouse, Luxary** כתבו:

– כותרת המחלקה.

– תכונות.

למחלקה Appartment הוגדרה פעולה הבונה (בנאי, constructor) המייצרת דירה רגילה מוכנה למכירה:

```
public Appartment(int floor, int numApp, int area)
{
    this.owner = "FREE";
    this.floor = floor;
    this.numApp = numApp;
    this.area = area;
}
```

ופעולה הקובעת את שם בעל הדירה אחרי המכירה:

```
public void SetOwner(string name)
{
    this.owner = name;
}
```

המחיר של כל דירה תלוי בשטח הדירה, בשטח המרפסת (אם יש) ובשטח הגינה (אם יש). לשם כך הוחלט להגדיר שלושה קבועים המציינים מחיר **למטר מרובע** של דירה, מרפסת וגן, בהתאם:

```
const int COST_APP = 1000;
const int COST_TERRACE = 300;
const int COST_GARDEN = 50;
```

(1 נק') ג. באיזו מחלקה או באילו מחלקות יש להוסיף את הקבועים הנ"ל? הסבירו את תשובתכם.

כדי לקדם את המכירות של דירות הפנטהאוז הוחלט לתת הנחה של 5% לדירות שאין להן נוף לים.

(4 נק') ד. כתבו פעולה או פעולות לחישוב ולהחזרת מחיר הדירה. יש לציין באיזו מחלקה צריך להוסיף את הפעולה או את הפעולות.

(3 נק') ה. כתבו פעולה חיצונית המקבלת מערך דירות המתאר בניין מסוים וסכום כסף שלקוח רוצה להשקיע. הפעולה תדפיס פרטי דירות שפנויות למכירה ומחירן אינו עולה על הסכום שאותו הלוקח רוצה להשקיע.

**הערה:** הניחו שהפעולות Get ו-ToString מוגדרות בעבור כל תכונה בכל אחת מהמחלקות הפרויקט.

## שאלה 10

במסגרת המלחמה בתאונות הדרכים, הציבו בכבישים ברחבי הארץ מצלמות אוטומטיות (מכמונות) למדידת מהירות כלי הרכב. בכל כביש מציבים מצלמה אחת לכל היותר. מספר המצלמות שמפזרים בכבישים עשוי להשתנות מעת לעת עפ"י החלטת המשטרה. במשטרת התנועה מנהלים מערכת ממוחשבת המרכזת את המידע לגבי מיקום המצלמות, וכן מידע על מכוניות אשר צולמו על ידי המצלמות.

**מצלמה מהירות SpeedCamera מיוצגת באמצעות :**

- קוד המצלמה
- מספר הכביש שבו היא מוצבת
- המהירות המקסימאלית המותרת בכביש זה
- אוסף מספרי כלי הרכב אשר נסעו במהירות גבוהה מהמהירות המותרת בכביש זה.

**(1 נק') א.** כתבו את כותרת המחלקה SpeedCamera ואת התכונות שלה.

**לייצוג אוסף אפשר להשתמש במחלקות Stack, Queue, Node.**

**אפשר להוסיף תכונות נוספות.**

**חובה לתעד את התכונות.**

**(1 נק') ב.** כתבו את הבנאי של המחלקה המקבל את קוד מצלמה, את מספר כביש ואת המהירות המותרת. הבנאי מאתחל אוסף מספרי כלי הרכב להיות ריק.

**(2 נק') ג.** כתבו פעולה AddCar. הפעולה מקבלת את מספר מכונית שצולמה ואת המהירות שלה. אם המהירות גבוהה מהמותר בכביש זה, הפעולה תוסיף מכונית זו לאוסף של המכוניות שנסעו מהר מהמותר.

**הערה :** הניחו שבמחלקה SpeedCamera קיימות פעולות Get ואין צורך לממש אותן.

בכל יום מציבה המשטרה 100 מצלמות לכל היותר (יתכן אף פחות). **רשת המצלמות** המוצבות בכבישים ביום נתון, מיוצגת באמצעות המחלקה SpeedCameraNetwork.

**(1 נק') ד.** כתבו את כותרת המחלקה SpeedCameraNetwork ואת התכונות שלה.

**(1 נק') ה.** כתבו את הפעולה AddSpeedCamera המוסיפה את מצלמת המהירות sc לרשת המצלמות. אפשר להניח שהמצלמה sc אינה מופיעה ברשת המצלמות.

במסגרת פעולות האכיפה מעוניינים לדעת מהם הכבישים המצריכים אכיפה מוגברת. כביש הדורש אכיפה מוגברת הוא כביש שבוצע בו מעל 200 עבירות מהירות ליום.

**(3 נק') ו.** כתבו פעולה המדפיסה את מספרי הכבישים שבהם יש לבצע אכיפה מוגברת. ציינו באיזו מחלקה יש לכתוב את הפעולה.

במשטרה רוצים לבדוק אם רכב מסוים צולם כאשר נסע במהירות הגבוהה מהמותר.

**(3 נק') ז.** כתבו פעולה המקבלת מספר רכב. הפעולה תבדוק אם הרכב צולם כאשר נסע במהירות גבוהה מהמותר. אם כן – הפעולה תדפיס את מספר המצלמה אשר צילמה את העבירה ותחזיר true, ואם לא – הפעולה תחזיר false.

**הערה :** אם רכב עשה כמה עבירות יש להדפיס את המספרים של כל המצלמות שצילמו את העבירות.

נתונה פעולה הבאה:

```
public static bool What(BinNode<int> t)
{
    if (t == null) return true;
    int x = t.GetValue();
    if (t.GetLeft() != null && t.GetLeft().GetValue() >= x)
        return false;
    if (t.GetRight() != null && t.GetRight().GetValue() < x)
        return false;
    return What(t.GetLeft()) && What(t.GetRight());
}
```

נתון עץ בינארי  $t$  שעבורו הפעולה  $What(t)$  מחזירה ערך  $true$ . עבור כל אחת מטענות הבאות ענו האם היא נכונה או לא. אם הטענה אינה נכונה, יש לצייר דוגמת עץ סותרת.

1. אם בשורש העץ הופיע המספר 13, 13 יכול להופיע שוב רק כבנו הימני של השורש.
2. אם בשורש העץ הופיע המספר 13, 13 יכול להופיע שוב רק בתת-עץ הימני של השורש.
3. אם בשורש העץ הופיע המספר 13, 13 יכול להופיע במקומות רבים בעץ, כולל תת העץ השמאלי, אך לא כבנו של צומת שערכו 13 או פחות.
4. לא ייתכנו בעץ  $t$  שני אחים בעלי ערך זהה.
5. אם אין בעץ  $t$  אף בן ימני, סריקתו בסדר תחילי תדפיס ערכים ממוינים מהגדול לקטן.
6. סריקת העץ בסדר תוכי תדפיס תמיד ערכים ממוינים.

**בהצלחה!**

© כל הזכויות שמורות למה"ט

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

נספח ממשקים מבנה הנתונים בתוכנית הלימודים

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

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 - $\text{Stack}\langle T \rangle$ מחסנית

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). <b>הנחה:</b> המחסנית הנוכחית אינה ריקה.
<b>T top()</b>	הפעולה מחזירה את הערך שבראש המחסנית הנוכחית מבלי להוציאו. <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): $[x_1, x_2, \dots, x_n]$

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. <b>הנחה:</b> התור הנוכחי אינו ריק.
<b>T head()</b>	הפעולה מחזירה את ערכו של האיבר שבראש התור מבלי להוציאו. <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): $[x_1, x_2, \dots, x_n]$

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.

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

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>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 - $\text{Stack}<T>$ מחסנית

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). <b>הנחה:</b> המחסנית הנוכחית אינה ריקה.
<b>T</b> Top()	הפעולה מחזירה את הערך שבראש המחסנית הנוכחית בלי להוציאו. <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): $[x_1, x_2, \dots, x_n]$

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 $\text{Queue}<\text{T}>$

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. <b>הנחה</b> : התור הנוכחי אינו ריק.
<b>T Head()</b>	הפעולה מחזירה את ערכו של האיבר שבראש התור מבלי להוציאו. <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): $[x_1, x_2, \dots, x_n]$

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.

**מבני נתונים ותכנות מונחה עצמיים 97105 – מועד א' קיץ 2022**

שאלה	סעיף	תת-סעיף	ניקוד	הערות
1	א	-	7	חריגה – להוריד 2 נקודות לולאה אין סופית – להוריד 2 נק'
	ב	-	8	חריגה – להוריד 2 נקודות לולאה אין סופית – להוריד 2 נק'
2	-	-	12	הערה: אין חשיבות לסדר איברים בתוך מחסנית
			3	בלי הסבר – לא לתת נקודות
3		-	15	בלי מעקב לא לתת נקודות
4	א	-	3	• כל סעיף – נקודה אחת (העמסה ב-B, דריסה והעמסה ב-C)
	ב	-	12	• כל סעיף – 2 נק'
5	א	-	3	
	ב		6	אם לא שמר על המחסנית – להוריד 1 נקודה אם פנה לתכונות של ליצן בלי שימוש ב-GET, להוריד 1 נקודה
	ג	-	6	אם לא בדק שפירמידה יציבה לפני ההכנסה, להוריד 1 נקודה אם לא בדק שפירמידה יציבה אחרי ההכנסה, להוריד 1 נקודה
6	א	-	5	• כל מחלקה – נקודה 1
	ב	-	3	אם לא כתב בנאי של SHAPE – להוריד 1 נקודה
	ג		7	• כל סעיף – נקודה אחת



שאלה	סעיף	תת-סעיף	ניקוד	הערות
7	א	-	3	בלי מעקב לא לתת נקודות
	ב	-	1	
	ג		3	
	ד		5	בלי מעקב לא לתת נקודות
	ה		3	
8	א	-	4	אם לא שמר תור, להוריד 1 נקודה
	ב	-	4	אם לא שמר תור, להוריד 1 נקודה
	ג	-	4	
	ד	-	3	בלי הסבר – לא לתת נקודות. ההסבר חייב להתייחס גם למספר חוליות בתור וגם למספר איברים במחסנית
9	א	-	2	
	ב		2	
	ג	-	1	
	ד		4	
	ה		3	
10	א	-	1	
	ב	-	1	
	ג		2	
	ד		1	
	ה		1	
	ו		3	
	ז		3	
11			12	• כל סעיף 2 נקודות