

## מבני נתונים ותכנות מונחה עצמים

### הנדסאים וטכנאים – הנדסת תוכנה

#### הנחיות לבחינה

ארבע שעות וחצי.

א. משך הבחינה:

בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:

ב. מבנה השאלון

ומפתח ההערכה:

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

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.

1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות.

ד. הוראות כלליות:

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 לך

להמשיך בפתרון השאלה. נמקו את בחירתכם.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!

בהצלחה!

בשאלון זה 27 עמודי בחינה ו- 4 עמודי נספחים.

## מבחן ב-JAVA

### חלק א'

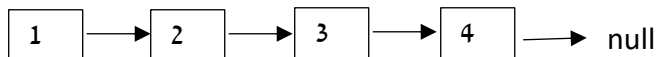
ענו על שלוש מבין השאלות 1-4 (ערך כל שאלה – 15 נקודות).

#### שאלה 1

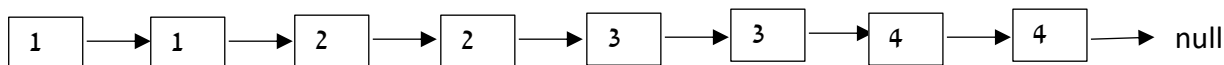
(8 נק') א. כתבו פעולה המקבלת הפנייה לחוליה ראשונה בשרשרת חוליות של מספרים שלמים ו"מכפילה" את

השרשרת בצורה הבאה:

אם לפני זימון הפעולה השרשרת הייתה:



אז אחרי זימון הפעולה היא תהיה:

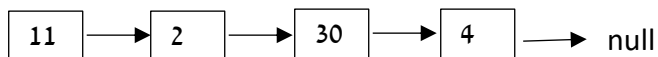


כותרת הפעולה: `public static void first(Node<Integer> chain)`

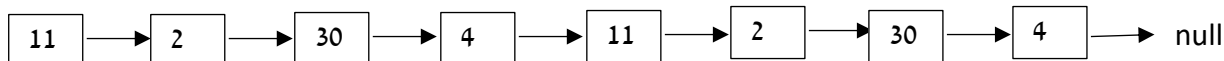
(7 נק') ב. כתבו פעולה המקבלת הפנייה לחוליה ראשונה בשרשרת חוליות של מספרים שלמים ו"מכפילה" את

השרשרת בצורה הבאה:

אם לפני זימון הפעולה השרשרת הייתה



אז אחרי זימון הפעולה היא תהיה



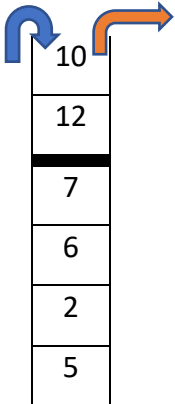
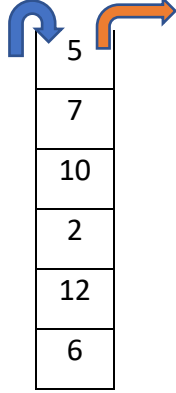
כותרת הפעולה: `public static void second(Node<Integer> chain)`

## שאלה 2

(12 נק') א. כתבו פעולה המקבלת מחסנית של מספרים שלמים. הפעולה תמייך מחדש את האיברים במחסנית לפי הסדר הבא: איברים הגדולים מממוצע האיברים במחסנית, יישמרו בחלק העליון של המחסנית ואילו איברים קטנים או שווים לממוצע האיברים במחסנית יישמרו בחלק התחתון.

לדוגמה:

ממוצע איברי המחסנית הוא:  $7 = (5+7+10+2+12+6) / 6$

המחסנית אחרי הפעולה	המחסנית לפני הפעולה
	

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

### שאלה 3

(9 נק') א. נתון קטע קוד שרץ ללא שגיאות:

```
A x = new B(t, s);
h.g(n, d);
```

עבור כל אחד מההיגדים 1-6 ציינו האם הוא נכון או אינו נכון. הסבירו את תשובתכם.

1. הפעולה g חייבת להיות ציבורית (public).
2. t הוא בהכרח משתנה מטיפוס פשוט (כמו: int, double וכד').
3. h הוא בהכרח אובייקט.
4. d יכול להיות שם של פעולה.
5. B היא בהכרח מחלקה שיורשת מ-A (גם אם לא ישירות).
6. ייתכן ש-A היא מחלקה שיורשת מ-B.

(6 נק') ב. (אין קשר לסעיף א')

נתונה המחלקה A הבאה:

```
public class A
{
    protected void a1()
    {
        System.out.println ("Hello A");
    }
    protected void a2()
    {
        a1();
    }
}
```

בקובץ נפרד, הגדירו מחלקה נוספת B שיורשת מהמחלקה A. במחלקה B

ביצעו דריסה לפעולה a1 והיא מדפיסה "Bye B".

מה יקרה אם נפעיל על אובייקט מסוג B את הפעולה a2?

1. הפלט יהיה: Hello A
2. הפלט יהיה: Bye B
3. הפלט יהיה:

Hello A

Bye B

4. הפלט יהיה:

Bye B

Hello A

## שאלה 4

נתונות שתי מחלקות A ו-B:

```

public class A
{
    public static int countA = 0;
    private int myVal;
    protected String myString;
    public A() { myVal = 1; }
    public A (int val) { myVal = val; myString = "GOOD!"; }
    public void func() { System.out.println("YES"); }
}
public class B extends A
{
    private double x;
    public boolean goodCode() { return x > 15.0; }
}

```

(5 נק') א. לפניכם ארבעה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון ונמקו את קביעתכם:

1. המחלקה B לא תעבור קומפילציה כי אין לה אף בנאי.
  2. המחלקה A יורשת את הפעולה goodCode מהמחלקה B.
  3. המחלקה B יורשת את כל התכונות ואת כל הפעולות של המחלקה A.
  4. המחלקה B יכולה לגשת ישירות לתכונות של המחלקה A.
  5. המחלקה A יכולה לגשת לתכונה x של המחלקה B.
- (2 נק') ב. כתבו במחלקה B פעולה בונה שמקבלת כפרמטר מספר שלם ומספר ממשי ומאתחלת את התכונות בהתאם.
- (4 נק') ג. לפניכם שתי הוראות מהתוכנית הראשית:

```

A one = new B(127, 1.4);
A two = new A(613);

```

בעבור כל אחת מההוראות שלפניכם קבעו אם היא תקינה או אינה תקינה. אם ההוראה אינה תקינה, נמקו את קביעתכם וצינו אם זו שגיאת ריצה או שגיאת הידור (קומפילציה).

1. boolean myBool = one.goodCode();
2. boolean myBool = two.goodCode();
3. boolean myBool = ((B) one).goodCode();
4. boolean myBool = ((B) two).goodCode();

(4 נק') ד. בתוכנית מסוימת מעוניינים לדעת כמה פעמים הפעולה func הופעלה מעצם מטיפוס A שאינו B,

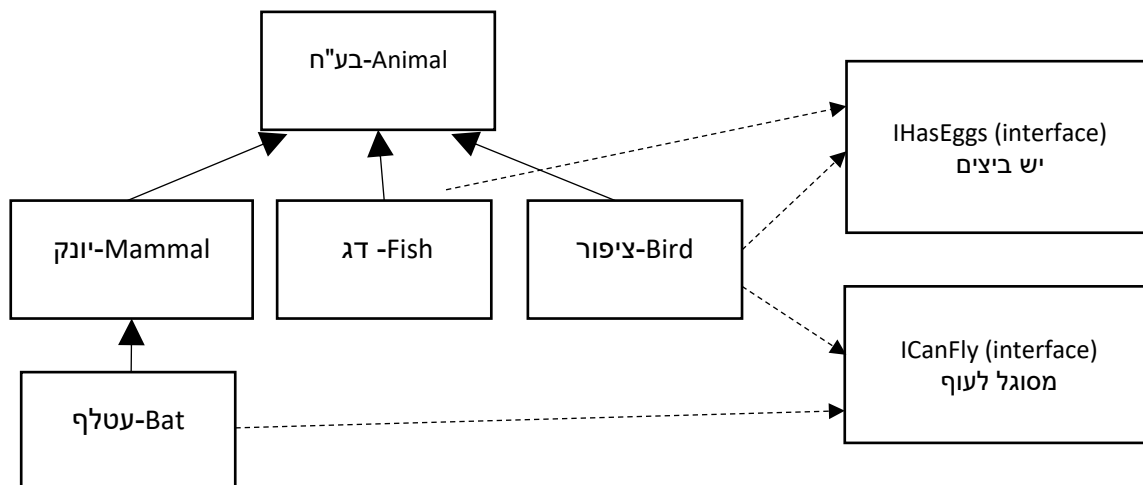
- וכמה פעמים היא הופעלה מעצם מטיפוס B.
- אם אפשר לקבל מידע זה – כתבו איזה שינויים יש לבצע במחלקות A ו-B.
- אם אי אפשר לקבל את המידע, הסבירו מדוע.

## חלק ב'

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

### שאלה 5

נתונה היררכיית המחלקות הבאה:



בממשק IHasEggs נמצאת כותרת הפעולה: `public void layingEggs()`

בממשק ICanFly נמצאת כותרת הפעולה: `public void fly()`

בכל מחלקה מוגדרת פעולה בונה ללא פרמטרים.

(5 נק') א. לכל מחלקה, כתבו את כותרת המחלקה.

(6 נק') ב. עבור הפעולות הבאות קבעו אם הן חוקיות. אם הפעולה אינה חוקית, יש להסביר את הסיבה לזה.

```

1. Animal a=new Bat();
2. Fish f=new IHasEggs();
3. ICanFly cf=new Bird();
   cf=new Bat();
4. Mammal m=new Bat();
   m.fly();
5. Animal a=new Bird();
   ((Bird)a).fly();
6. IHasEggs he=new Bird();
   he.fly();
    
```

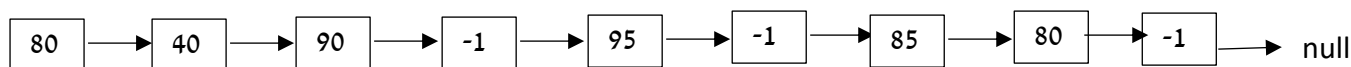
(4 נק') ג. כתבו פעולה המקבלת כפרמטר מערך של בעלי חיים (מערך הפניות לעצמים מסוג Animal), ומחזירה

את מספר בעלי החיים המסוגלים לעוף הנמצאים במערך.

## שאלה 6

בבית ספר "שזר" מאוכסן מידע על ציוני התלמידים במדעי המחשב כשרשרת חוליות המכילה ציונים של כל התלמידים. לכל תלמיד יש מספר שונה של ציונים. הציונים של כל תלמיד מופרדים ב-1.

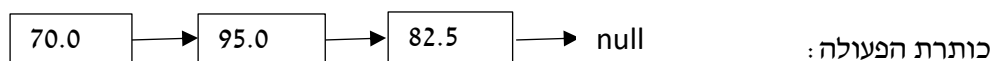
לדוגמה:



שרשרת זו מציגת ש:

- לתלמיד הראשון יש שלושה ציונים (80,40,90).
- לתלמיד השני יש ציון אחד (95).
- לתלמיד השלישי יש שני ציונים (85,80).

9 נק') א. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת ציונים ומחזירה הפניה לחוליה ראשונה של שרשרת חדשה הכוללת ציוני ממוצע של כל תלמיד. עבור השרשרת הנ"ל תחזיר הפעולה את השרשרת הבאה:



כותרת הפעולה:

```
public static Node<Double> averageList(Node<Integer> lst)
```

6 נק') ב. כדי לשפר את ציוני המגן הוחלט לבטל לכל תלמיד את הציון הנמוך ביותר שלו (כמובן שביטול הציון מתייחס רק לתלמידים שיש להם יותר מציון אחד).

לדוגמה:

אחרי ה"שיפור" אצל התלמיד הראשון יתבטל ציון 40, אצל התלמיד השלישי יתבטל ציון 80. כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת ציונים ומדפיסה את המספר הסידורי של התלמיד ואת הציון הממוצע שלו לפני ואחרי השיפור. עבור המידע הנ"ל הפעולה תדפיס:

1	70.0	85.0
2	95.0	95.0
3	82.5	85.0

כותרת הפעולה:

```
public static void print(Node<Integer> lst)
```

## שאלה 7

נגדיר טיפוס נתונים חדש בשם TStack (תלת-מחסנית), כמבנה המכיל שלוש מחסניות של מספרים שלמים: S0, S1, S2.

במחלקה TStack הוגדרו הפעולות הבאות:

• הפעולה: `void move(int from, int to)`

הפעולה המעבירה איבר מראש מחסנית מספר `from` למחסנית מספר `to`.

לדוגמה:

`move(0, 1)` מעביר את האיבר הנמצא בראש המחסנית מספר 0 לראש מחסנית מספר 1.

מעברים חוקיים הם רק מ- S0 ל- S1, מ- S1 ל- S2, ומ- S2 ל- S0, כלומר, אפשר לבצע רק את מעברים הבאים:

`move(0, 1);`

`move(1, 2);`

`move(2, 0);`

אם הפרמטרים `from` ו-`to` אינם מקיימים מעבר חוקי, הפעולה אינה מבצעת דבר.

הנחה: המחסנית מספר `from` אינה ריקה.

• הפעולה: `boolean bigOrEqual(int from, int toCompare)`

היא הפעולה המחזירה `true` אם האיבר בראש המחסנית מספר `from` גדול או שווה לאיבר בראש המחסנית מספר `toCompare` ו-`false` אם המצב שונה.

הנחה: שתי המחסניות לא ריקות.

• הפעולה: `boolean isEmpty(int stackId)`

היא הפעולה המחזירה `true` אם המחסנית מספר `stackId` ריקה ו-`false` אם המחסנית אינה ריקה.

(6 נק') א. נתונה תלת-מחסנית, שבה מחסנית מספר 0 מכילה מספרים שלמים לא ממוינים, ושתי מחסניות אחרות ריקות. כתבו פעולה `void maximum()`, המעבירה את האיבר הגדול ביותר ממחסנית S0 לראש המחסנית S1.

הערה: יש להשתמש רק בשלוש הפעולות המוגדרות במחלקה TStack! אסור להשתמש במחלקה

Stack או כל מבנה אחר.

(6 נק') ב. נתונה תלת-מחסנית, שבה בכל המחסניות נמצאים מספרים שלמים לא ממוינים.

כתבו פעולה `void sort()`, המשנה את סדר האיברים בתלת-מחסנית, כך שבאחת מהמחסניות האיברים יהיו ממוינים בסדר עולה (האיבר הגדול ביותר נמצא בתחתית המחסנית) ושתי מחסניות האחרות תהיינה ריקות.

הערה: יש להשתמש רק בשלוש הפעולות המוגדרות במחלקה TStack ובפעולה `maximum()`!

אסור להשתמש במחלקה Stack או בכל מבנה אחר.

(3 נק') ג. מהי סיבוכיות זמן הריצה של האלגוריתם בסעיף ב', בהנחה שבתלת-מחסנית יש N איברים?

הסיבוכיות של כל הפעולות `move`, `bigOrEqual`, `isEmpty` היא O(1), הסבירו את תשובתכם.

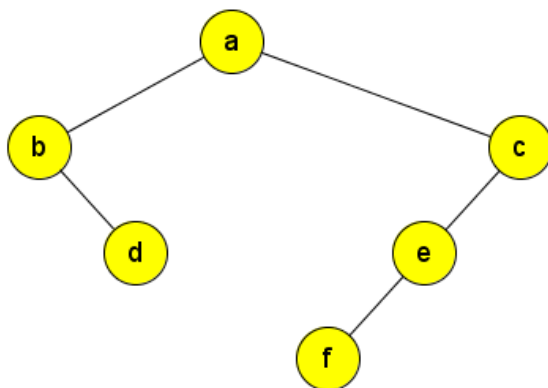


## שאלה 8

נתונה הפעולה חיצונית **mystery** הבאה שמקבלת כפרמטר הפניה לשורש של עץ בינרי:

```
public static void mystery(BinNode<Character> t)
{
    if (t != null)
    {
        System.out.print (t.getValue() + " ");
        Mystery (t.getLeft());
        System.out.print (t. getValue ()+ " ");
        Mystery (t. getRight());
        System.out.print (t.getValue()+ " ");
    }
}
```

(8 נק') א. מה תדפיס הפעולה **mystery** כאשר היא תקבל את השורש של העץ הבינרי **t** הבא?  
חובה להראות את המעקב!



(7 נק') ב. ציירו עץ שעבורו תדפיס הפעולה **mystery** שורה הבאה:

**b a c c d d d c a e e f f f e a b k k k b**

## חלק ג'

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

#### שאלה 9

בחנות למכירת מכשירים אלקטרוניים אפשר לשלם באמצעי התשלום האלה: מזומן, המחאה (צ'ק), כרטיס אשראי. את הסכום המשולם בהמחאות ובאשראי אפשר לחלק לכמה תשלומים, התשלומים לא חייבים להיות שווים.

התשלום בעבור כל קנייה יכול להתבצע באמצעי תשלום אחד או יותר. במקרה של שימוש ביותר מאמצעי תשלום אחד, ששילוב אמצעי התשלום ייתן את הסכום הנדרש.

#### לדוגמה:

לפניכם כמה שילובים אפשריים של אמצעי תשלום בעבור קניה בסכום של 1000 ש"ח.

- לשלם את כל הסכום באמצעי תשלום אחד: מזומן או המחאה או כרטיס אשראי.
- לשלם 200 ₪ במזומן ו-800 ₪ בכרטיס אשראי ב-4 תשלומים שווים (כל חודש יחויב סכום של 200 ₪).
- לשלם 100 ₪ במזומן, 500 ₪ באמצעות שתי המחאות: הראשונה על סך 200 ₪ והשנייה על סך 300 ₪, ו-400 ₪ בכרטיס אשראי.

החנות זקוקה לתוכנה כדי לנהל את תשלומי הקונים.

בעבור כל קנייה, המידע המתקבל בחנות הוא:

תאריך הקניה, הסכום לתשלום בעבור הקנייה, פירוט שילוב אמצעי התשלום ומספר אמצעי התשלום שבאמצעותם נערכה הקנייה.

בעבור תשלום במזומן – הסכום לתשלום.

בעבור תשלום בהמחאה – הסכום לתשלום, מספר ההמחאה, שם הבנק והתאריך הרשום על ההמחאה.

בעבור תשלום בכרטיס אשראי – הסכום לתשלום, מספר כרטיס האשראי, תוקף הכרטיס והתאריך שבו יחויב בעל כרטיס האשראי.

בחנות יישמר המידע עבור הקניות שנעשו בחנות ומספר הקניות שבוצעו.

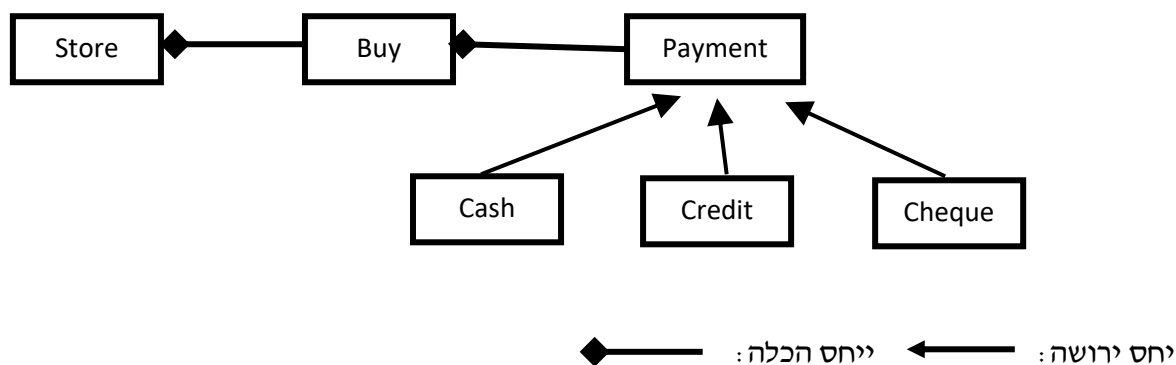
#### הפעולות הנפוצות בניהול מערכת תשלומים:

- בדיקה שסכום כל התשלומים בעבור קנייה אחת שווה לסכום הקנייה.
- הדפסת פרטי הקניות שהתשלום עבורן התבצע בכרטיס אשראי מסוים (לפחות אחד התשלומים)
- דיווח של מספר הקניות שבוצעו במזומן בלבד.

כדי לנהל מערכת תשלומים מפתחים פרויקט הכולל את המחלקות הבאות:

- Store – המייצגת את החנות.
- Buy – המייצגת קניה.
- Payment – המייצגת תשלום.
- Cash – המייצגת תשלום במזומן.
- Credit – המייצגת תשלום בכרטיס אשראי.
- Cheque – המייצגת תשלום בהמחאה.

לפניכם תרשים של היררכיית המחלקות הנדרשות בעבור כתיבת התוכנה.



(6 נק') א. כתבו את כותרות המחלקות הנזכרות בתרשים ולכל מחלקה הוסיפו את התכונות הנדרשות. הוסיפו תיעוד!

**אפשר להניח שקיימת מחלקה Date, המייצגת תאריך.**

(6 נק') ב. כתבו את הפעולות הנזכרות לעיל וציינו באיזו מחלקה צריכה להיות כל פעולה:

1. בדיקה שסכום כל התשלומים בעבור קנייה אחת שווה לסכום הקנייה.

```
public boolean check()
```

2. פעולה המקבלת מספר כרטיס אשראי ומדפיסה את פרטי הקניות שהיה בהן תשלום בכרטיס אשראי זה (לפחות אחד התשלומים).

```
public void print (String creditNum)
```

3. פעולה המחזירה את מספר הקניות שבוצעו במזומן בלבד.

```
public int cashPayments()
```

**הערות:**

- אפשר להניח כי בכל מחלקה הוגדרו פעולות get/set ו- toString (אין צורך לממש).
- אם כתבתם פעולות עזר, יש לציין באיזו מחלקה נמצאת כל הפעולה.

## שאלה 10

פרסומת ברדיו מאופיינת במשך זמן הפרסומת בשניות `length`, בשם המוצר המפורסם `product`, בשם החברה המפרסמת `company` ובמחיר הפרסומת `price`.  
**המחלקה Advert כוללת את התכונות הבאות:**

```
public class Advert
{
    private int length;
    private String product;
    private String company;
    private double price;
```

במחלקה הוגדרו פעולה בונה (בנאי- constructor), פעולות `set/get` לכל התכונות

בשעת שידור אחת מותר לפרסם מקסימום 15 פרסומות במשך **חמש דקות** בסה"כ. מספר הפרסומות המדויק בשעת השידור לא ידוע מראש, מכיוון שהוא תלוי באורכן של הפרסומות המשודרות באותה שעה. שעות שידור הן מספרים שלמים: 0,1,2...23  
**המחלקה שעת שידור, AdvertHour, מכילה את הפרסומות המשודרות בשעה מסוימת.**

חלק מפעולות הממשק של המחלקה `AdvertHour` מתוארות בטבלה הבאה:

שם הפעולה	תיאור הפעולה
<b>freeTime</b>	הפעולה מחזירה את הזמן (בשניות) שנותר פנוי לפרסומת בשעת השידור.
<b>isPossible</b>	הפעולה מקבלת פרסומת <code>adv</code> ומחזירה אמת אם אפשר להוסיף את הפרסומת לשעת השידור.
<b>addAdvert</b>	הפעולה מקבלת פרסומת <code>adv</code> ומוסיפה אותה לשעת השידור. הפרסומת <code>adv</code> תהיה פרסומת האחרונה שתשודר. אם אי אפשר להוסיף פרסומת, הפעולה לא מבצעת דבר.

(1 נק') א. כתבו את כותרת המחלקה `AdvertHour` ואת התכונות שלה.  
**חובה לתעד את התכונות.**

(6 נק') ב. כתבו את הבנאי של המחלקה ואת שלוש הפעולות `addAdvert`, `isPossible`, `freeTime`.

**המחלקה ManageDay מנהלת את יום השידור תוך שימוש במחלקות AdvertHour ו- Advert.**

(1 נק') ג. כתבו את כותרת המחלקה `ManageDay` ואת התכונות שלה.  
**חובה לתעד את התכונות.**

(4 נק') ד. כתבו פעולה `benefitDay` במחלקה `ManageDay` המחזירה את הרווח הכולל שהתקבל משידור הפרסומות ביום זה.

נתונות שלוש המחלקות :MultiOne,SingleOne,Basis

```

class Basis    {
    protected int num1;
    public Basis()
    {
    }
    public Basis(int n)
    {
        this.num1 = n;
    }
    public void print()
    {
        System.out.println(this.num1);
    }
}

class SingleOne extended Basis    {
    protected int num2;
    public SingleOne(int n1, int n2)
    {
        super(n1);
        this.num2 = n2;
    }
    public void print()
    {
        super.print();
        System.out.println(this.num2);
    }
}

class MultiOne extends Basis    {
    private int count = 0;
    private Basis[] arr;
    public MultiOne()
    {
        this.arr = new Basis[5];
    }
}

```

```

    }
    public void print()
    {
        for (int i = 0; i < count; i++)
            arr[i].print();
    }
    public void add(Basis b) {
        if(count<arr.length)
        {
            arr[count] = b;
            count++;
        }
    }
}

```

**(2 נק')** א. ציירו תרשים UML שמייצג את הקשרים בין המחלקות Basis, SingleOne, MultiOne.

יש לסמן ירושה באמצעות החץ  והכללה באמצעות הסימן 

**(10 נק')** ב. נתונה המחלקה Test ובה הפעולה הראשית.

עקבו בעזרת טבלת מעקב אחר הפעולה Main במחלקה Test, ורשום את הפלט.  
על הטבלה לכלול את ערכי כל המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```

class Test
{
    public static void Main(String[] args)
    {
        MultiOne container = new MultiOne();
        SingleOne s1 = new SingleOne(11, 35);
        container.add(s1);
        s1 = new SingleOne(47, 22);
        container.add(s1);
        s1 = new SingleOne(8, 17);
        container.add(s1);
        MultiOne subContainer = new MultiOne();
        s1 = new SingleOne(53, 40);
        subContainer.add(s1);
        s1 = new SingleOne(21, 13);
        subContainer.add(s1);
        s1 = new SingleOne(39, 62);
        subContainer.add(s1);
        container.add(subContainer);
        container.print();
    }
}

```

## מבחן ב- C#

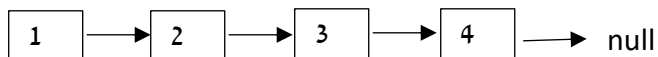
### חלק א'

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

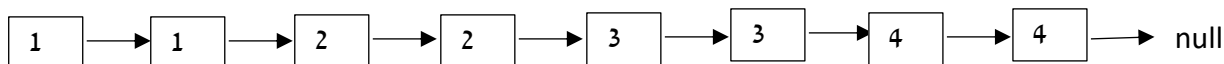
#### שאלה 1

(8 נק') א. כתבו פעולה המקבלת הפניה לחוליה ראשונה בשרשרת חוליות של מספרים שלמים ו"מכפילה" את השרשרת בצורה הבאה:

אם לפני זימון הפעולה השרשרת הייתה:



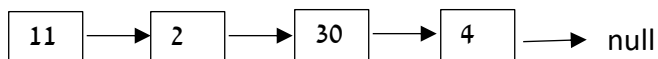
אז אחרי זימון הפעולה היא תהיה:



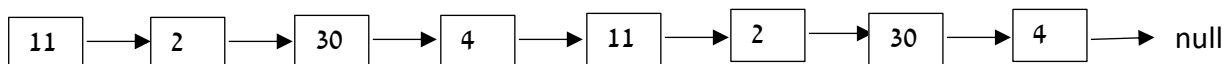
כותרת הפעולה: `public static void First(Node<int> chain)`

(7 נק') ב. כתבו פעולה המקבלת הפניה לחוליה ראשונה בשרשרת חוליות של מספרים שלמים ו"מכפילה" את השרשרת בצורה הבאה:

אם לפני זימון הפעולה השרשרת הייתה:



אז אחרי זימון הפעולה היא תהיה:



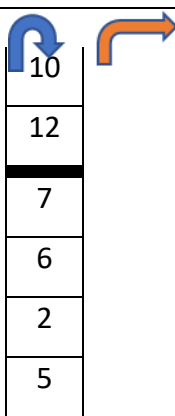
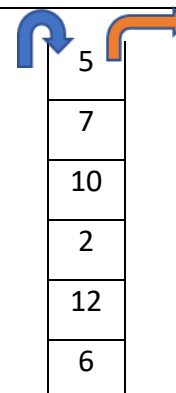
כותרת הפעולה: `public static void Second(Node<int> chain)`

## שאלה 2

(12 נק') א. כתבו פעולה המקבלת מחסנית של מספרים שלמים. הפעולה תמייין מחדש את האיברים במחסנית לפי הסדר הבא: איברים הגדולים מממוצע האיברים במחסנית, יישמרו בחלק העליון של המחסנית ואילו איברים קטנים או שווים לממוצע האיברים במחסנית יישמרו בחלק התחתון.

לדוגמה:

ממוצע איברי המחסנית הוא:  $7 = (5+7+10+2+12+6) / 6$

המחסנית אחרי הפעולה	המחסנית לפני הפעולה
	

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



## שאלה 3

(9 נק') א. נתון קטע קוד שרץ ללא שגיאות:

```
A x = new B(t, s);
h.G(n, d);
```

עבור כל אחד מההיגדים 1-6 ציינו אם הוא נכון או אינו נכון. הסבירו את תשובתכם.

1. הפעולה g חייבת להיות ציבורית (public).
2. t הוא בהכרח משתנה מטיפוס פשוט (כמו: int, double וכד').
3. h הוא בהכרח אובייקט.
4. d יכול להיות שם של פעולה.
5. B היא בהכרח מחלקה שיורשת מ-A (גם אם לא ישירות).
6. ייתכן ש-A היא מחלקה שיורשת מ-B.

(6 נק') ב. (אין קשר לסעיף א')

נתונה המחלקה A הבאה:

```
public class A
{
    protected virtual void A1()
    {
        Console.WriteLine ("Hello A");
    }
    protected void A2()
    {
        A1();
    }
}
```

בקובץ נפרד, הגדירו מחלקה נוספת B שיורשת מהמחלקה A.  
 במחלקה B ביצעו דריסה לפעולה A1 והיא מדפיסה "Bye B".  
 מה יקרה אם נפעיל על אובייקט מסוג B את הפעולה A2?

1. הפלט יהיה: Hello A

2. הפלט יהיה: Bye B

3. הפלט יהיה:

Hello A

Bye B

4. הפלט יהיה:

Bye B

Hello A

## שאלה 4

נתונות שתי מחלקות A ו-B:

```

public class A
{
    public static int countA = 0;
    private int myVal;
    protected string myString;
    public A(){ myVal = 1; myString = "BAD"; }
    public A (int val) { myVal = val; myString = "GOOD!"; }
    public int Func() { return 1; }
}
public class B : A
{
    private double x;
    public bool GoodCode() { return x > 15.0; }
}

```

(5 נק') א. לפניכם ארבעה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון ונמקו את קביעתכם:

1. המחלקה B לא תעבור קומפילציה כי אין לה אף בנאי.
  2. המחלקה A יורשת את הפעולה GoodCode מהמחלקה B.
  3. המחלקה B יורשת את כל התכונות ואת כל הפעולות של המחלקה A.
  4. המחלקה B יכולה לגשת ישירות לתכונות של המחלקה A.
  5. המחלקה A יכולה לגשת לתכונה x של המחלקה B.
- (2 נק') ב. כתבו במחלקה B פעולה בונה שמקבלת כפרמטר מספר שלם ומספר ממשי ומאתחלת את התכונות בהתאם.
- (4 נק') ג. לפניכם שתי הוראות מהתוכנית הראשית:

```

A one = new B(127, 1.4);
A two = new A(613);

```

בעבור כל אחת מההוראות שלפניכם קבעו אם היא תקינה או אינה תקינה. אם ההוראה אינה תקינה, נמקו את קביעתכם וצינו אם זו שגיאת ריצה או שגיאת הידור (קומפילציה).

1. bool myBool = one. GoodCode ();
2. bool myBool = two. GoodCode ();
3. bool myBool = ((B) one). GoodCode ();
4. bool myBool = ((B) two). GoodCode ();

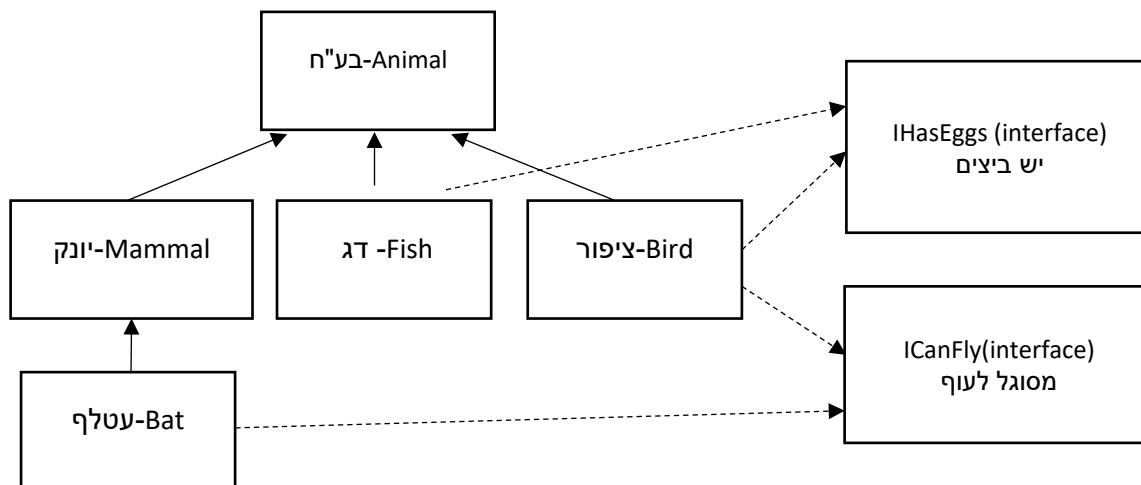
- (4 נק') ד. בתוכנית מסוימת מעוניינים לדעת כמה פעמים הפעולה Func הופעלה מעצם מטיפוס A שאינו B, וכמה פעמים היא הופעלה מעצם מטיפוס B.
- אם אפשר לקבל מידע זה – כתבו איזה שינויים יש לבצע במחלקות A ו-B.
- אם אי אפשר לקבל את המידע, הסבירו מדוע.

## חלק ב'

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

### שאלה 5

נתונה היררכיית המחלקות הבאה:



בממשק IHasEggs נמצאת כותרת הפעולה:

בממשק ICanFly נמצאת כותרת הפעולה:

בכל מחלקה מוגדרת פעולה בונה ללא פרמטרים.

(5 נק') א. לכל מחלקה, כתבו את כותרת המחלקה.

(6 נק') ב. עבור הפעולות הבאות קבעו אם הן חוקיות. אם הפעולה אינה חוקית, יש להסביר את הסיבה לזה:

```

1. Animal a=new Bat();
2. Fish f=new IHasEggs();
3. ICanFly cf=new Bird();
   cf=new Bat();
4. Mammal m=new Bat();
   m.Fly();
5. Animal a=new Bird();
   ((Bird)a).Fly();
6. IHasEggs he=new Bird();
   he.Fly();
    
```

(4 נק') ג. כתבו פעולה המקבלת כפרמטר מערך של בעלי חיים (מערך הפניות לעצמים מסוג Animal),

ומחזירה את מספר בעלי החיים המסוגלים לעוף הנמצאים במערך.

## שאלה 6

בבית ספר "שזר" מאוכסן מידע על ציוני התלמידים במדעי המחשב כשרשרת חוליות המכילה ציונים של כל התלמידים. לכל תלמיד יש מספר שונה של ציונים. הציונים של כל תלמיד מופרדים ב-1.

**לדוגמה:**



שרשרת זו מציינת ש:

- לתלמיד הראשון יש שלושה ציונים (80,40,90).
- לתלמיד השני יש ציון אחד (95).
- לתלמיד השלישי יש שני ציונים (85,80).

**(9 נק')** א. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת ציונים ומחזירה הפניה לחוליה ראשונה של

שרשרת חדשה הכוללת ציוני ממוצע של כל תלמיד.

עבור השרשרת הנ"ל תחזיר הפעולה את השרשרת הבאה:



כותרת הפעולה:

```
public static Node<double> AverageList(Node<int> lst)
```

**(6 נק')** ב. כדי לשפר את ציוני המגן הוחלט לבטל לכל תלמיד את הציון הנמוך ביותר שלו (מובן שביטול הציון

מתייחס רק לתלמידים שיש להם יותר מציון אחד)

**לדוגמה:**

אחרי ה"שיפור" אצל התלמיד הראשון יתבטל ציון 40, אצל התלמיד השלישי יתבטל ציון 80.

כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת הציונים ומדפיסה את המספר

הסידורי של התלמיד ואת הציון הממוצע שלו לפני ואחרי השיפור.

עבור המידע הנ"ל הפעולה תדפיס:

1	70.0	85.0
2	95.0	95.0
3	82.5	85.0

כותרת הפעולה:

```
public static void Print(Node<int> lst)
```

## שאלה 7

נגדיר טיפוס נתונים חדש בשם **TStack (תלת-מחסנית)**, כמבנה המכיל שלוש מחסניות של מספרים שלמים: **S0, S1, S2**.

במחלקה **TStack** הוגדרו הפעולות הבאות:

• הפעולה `void Move(int from, int to)`

הפעולה המעבירה איבר מראש מחסנית מספר from למחסנית מספר to.

לדוגמה:

`Move(0, 1)` מעביר את האיבר הנמצא בראש המחסנית מספר 0 לראש מחסנית מספר 1.

**מעברים חוקיים** הם רק מ- `S0` ל- `S1`, מ- `S1` ל- `S2`, ומ- `S2` ל- `S0`. כלומר, אפשר לבצע רק את המעברים הבאים:

`Move(0, 1);`

`Move(1, 2);`

`Move(2, 0);`

אם הפרמטרים `from` ו-`to` אינם מקיימים מעבר חוקי, הפעולה אינה מבצעת דבר.

**הנחה:** המחסנית מספר from אינה ריקה.

• הפעולה `bool BigOrEqual(int from, int toCompare)`

היא הפעולה המחזירה `true` אם האיבר בראש המחסנית מספר from גדול או שווה לאיבר בראש המחסנית מספר toCompare ו-`false` אם המצב הוא שונה.

**הנחה:** שתי המחסניות לא ריקות.

• הפעולה `boolean IsEmpty(int stackId)`

היא הפעולה המחזירה `true` אם המחסנית מספר stackId ריקה ו-`false` אם המחסנית אינה ריקה.

**(6 נק')** א. נתונה **תלת-מחסנית**, שבה מחסנית מספר 0 מכילה מספרים שלמים לא ממוינים ושתי מחסניות אחרות ריקות.

כתבו פעולה `void Maximum()`, המעבירה את האיבר הגדול ביותר ממחסנית `S0` לראש המחסנית `S1`.

**הערה:** יש להשתמש רק בשלוש הפעולות המוגדרות במחלקה **TStack**! אסור להשתמש במחלקה

**Stack** או כל מבנה אחר.

**(6 נק')** ב. נתונה **תלת-מחסנית**, שבה בכל המחסניות נמצאים מספרים שלמים לא ממוינים.

כתבו פעולה `void Sort()`, המשנה את סדר האיברים בתלת-מחסנית, כך שבאחת מהמחסניות

האיברים יהיו ממוינים בסדר עולה (האיבר הגדול ביותר נמצא בתחתית המחסנית) ושתי מחסניות האחרות תהיינה ריקות.

**הערה:** יש להשתמש רק בשלוש הפעולות המוגדרות במחלקה **TStack** ובפעולה `!maximum()`

אסור להשתמש במחלקה **Stack** או בכל מבנה אחר.

**(3 נק')** ג. מהי סיבוכיות זמן הריצה של האלגוריתם בסעיף ב', בהנחה שבתלת-מחסנית יש `N` איברים?

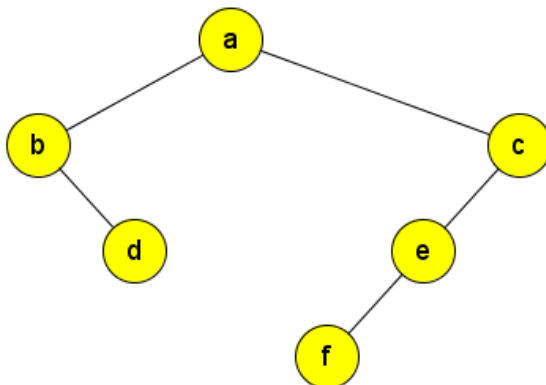
הסיבוכיות של כל הפעולות `Move`, `BigOrEqual`, `IsEmpty` היא  $O(1)$ . **הסבירו את תשובתכם.**

## שאלה 8

נתונה הפעולה החיצונית **Mystery** הבאה שמקבלת כפרמטר הפניה לשורש של עץ בינרי:

```
public static void Mystery(BinNode<char> t)
{
    if (t != null)
    {
        Console.Write (t.GetValue() + " ");
        Mystery (t.GetLeft());
        Console.Write (t. GetValue ()+ " ");
        Mystery (t. GetRight());
        Console.Write (t.GetValue()+ " ");
    }
}
```

(8 נק') א. מה תדפיס הפעולה **Mystery** כאשר היא תקבל את השורש של העץ הבינרי **t** הבא?  
חובה להראות את המעקב !



(7 נק') ב. ציירו עץ שעבורו תדפיס הפעולה **Mystery** היא שורה הבאה:

**b a c c d d d c a e e f f f e a b k k k b**

## חלק ג'

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

#### שאלה 9

בחנות למכירת מכשירים אלקטרוניים אפשר לשלם באמצעי התשלום האלה: מזומן, המחאה (צ'ק), כרטיס אשראי. את הסכום המשולם בהמחאות ובאשראי אפשר לחלק לכמה תשלומים, התשלומים לא חייבים להיות שווים.

התשלום בעבור כל קנייה יכול להתבצע באמצעי תשלום אחד או יותר. במקרה של שימוש ביותר מאמצעי תשלום אחד, שילוב אמצעי התשלום ייתן את הסכום הנדרש.

#### לדוגמה:

לפניכם כמה שילובים אפשריים של אמצעי תשלום בעבור קנייה בסכום של 1,000 ש"ח.

- לשלם את כל הסכום באמצעי תשלום אחד: מזומן, המחאה או כרטיס אשראי.
- לשלם 200 ש"ח במזומן ו- 800 ש"ח בכרטיס אשראי בארבעה תשלומים שווים (כל חודש יחויב סכום של 200 ש"ח).
- לשלם 100 ₪ במזומן, 500 ₪ באמצעות שתי המחאות: הראשונה על סך 200 ₪ והשנייה על סך 300 ₪ ו- 400 ₪ בכרטיס אשראי.

החנות זקוקה לתוכנה כדי לנהל את תשלומי הקונים.

בעבור כל קנייה, המידע המתקבל בחנות הוא:

תאריך הקניה, הסכום לתשלום בעבור הקנייה, פירוט שילוב אמצעי התשלום ומספר אמצעי התשלום שבאמצעותם נערכה הקנייה.

בעבור תשלום במזומן – הסכום לתשלום.

בעבור תשלום בהמחאה – הסכום לתשלום, מספר ההמחאה, שם הבנק והתאריך הרשום על ההמחאה.

בעבור תשלום בכרטיס אשראי – הסכום לתשלום, מספר כרטיס האשראי, תוקף הכרטיס והתאריך שבו יחויב בעל כרטיס האשראי.

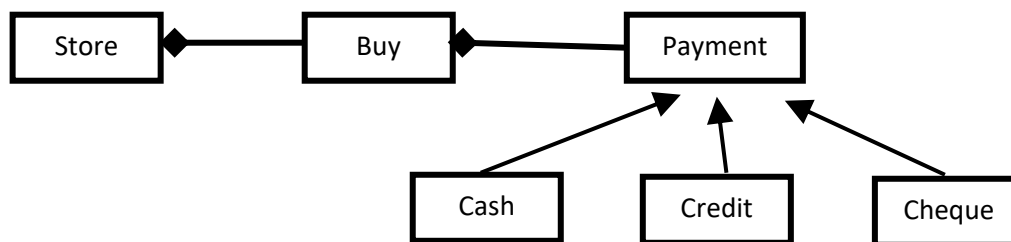
בחנות יישמר המידע עבור הקניות שנעשו בחנות ומספר הקניות שבוצעו.

#### הפעולות הנפוצות בניהול מערכת תשלומים:

- בדיקה שסכום כל התשלומים בעבור קנייה אחת שווה לסכום הקנייה.
  - הדפסת פרטי הקניות שהתשלום עבורן התבצע בכרטיס אשראי מסוים (לפחות אחד התשלומים).
  - דיווח של מספר הקניות שבוצעו במזומן בלבד.
- כדי לנהל מערכת תשלומים מפתחים פרויקט הכולל את המחלקות הבאות:

- Store – המייצגת את החנות.
- Buy – המייצגת קנייה.
- Payment – המייצגת תשלום.
- Cash – המייצגת תשלום במזומן.
- Credit – המייצגת תשלום בכרטיס אשראי.
- Cheque – המייצגת תשלום בהמחאה.

לפניכם תרשים של היררכיית המחלקות הנדרשות בעבור כתיבת התוכנה:



יחס ירשה: ← יחס הכלה: —◆

(6 נק') א. כתבו את כותרות המחלקות הנזכרות בתרשים ולכל מחלקה הוסיפו את התכונות הנדרשות. הוסיפו תיעוד!

**אפשר להניח שקיימת מחלקה Date, המייצגת תאריך.**

(6 נק') ב. כתבו את הפעולות הנזכרות לעיל וציינו באיזו מחלקה צריכה להיות כל פעולה:

1. בדיקה ששכום כל התשלומים בעבור קנייה אחת שווה לשכום הקנייה.

```
public bool Check()
```

2. פעולה המקבלת מספר כרטיס אשראי ומדפיסה את פרטי הקניות שהיה בהן תשלום בכרטיס אשראי זה (לפחות אחד התשלומים).

```
public void Print (string creditNum)
```

3. פעולה המחזירה את מספר הקניות שבוצעו במזומן בלבד.

```
public int CashPayments()
```

**הערות:**

- אפשר להניח כי בכל מחלקה הוגדרו פעולות Get/Set ו-ToString (אין צורך לממש).
- אם כתבתם פעולות עזר, יש לציין באיזו מחלקה נמצאת כל פעולה.



## שאלה 10

פרסומת ברדיו מאופיינת במשך זמן הפרסומת בשניות length, בשם המוצר המפורסם product, בשם החברה המפרסמת company ובמחיר הפרסומת price. המחלקה Advert כוללת את התכונות הבאות:

```
public class Advert
{
    private int length;
    private string product;
    private string company;
    private double price;
```

במחלקה הוגדרו פעולה בונה (בנאי- constructor) ופעולות Set/Get לכל התכונות.

בשעת שידור אחת מותר לפרסם מקסימום 15 פרסומות במשך חמש דקות בסה"כ. מספר הפרסומות המדויק בשעת השידור לא ידוע מראש, מכיוון שהוא תלוי באורכן של הפרסומות המשודרות באותה שעה. שעות שידור הן מספרים שלמים: 0,1,2...23. המחלקה שעת שידור, AdvertHour, מכילה את הפרסומות המשודרות בשעה מסוימת.

חלק מפעולות הממשק של המחלקה AdvertHour מתוארות בטבלה הבאה:

שם הפעולה	תיאור הפעולה
<b>FreeTime</b>	הפעולה מחזירה את הזמן (בשניות) שנותר פנוי לפרסומת בשעת השידור.
<b>IsPossible</b>	הפעולה מקבלת פרסומת adv ומחזירה אמת אם אפשר להוסיף את הפרסומת לשעת השידור.
<b>AddAdvert</b>	הפעולה מקבלת פרסומת adv ומוסיפה אותה לשעת השידור. הפרסומת adv תהיה הפרסומת האחרונה שתשודר. אם אי אפשר להוסיף פרסומת, הפעולה לא מבצעת דבר.

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

חובה לתעד את התכונות.

(6 נק') ב. כתבו את הבנאי של המחלקה ואת שלוש הפעולות FreeTime, IsPossible, AddAdvert.

המחלקה ManageDay מנהלת את יום השידור תוך שימוש במחלקות AdvertHour ו- Advert.

(1 נק') ג. כתבו את כותרת המחלקה ManageDay ואת התכונות שלה.

חובה לתעד את התכונות.

(4 נק') ד. כתבו פעולה BenefitDay במחלקה ManageDay המחזירה את הרווח הכולל שהתקבל משידור הפרסומות ביום זה.

## שאלה 11

נתונות שלוש המחלקות :MultiOne,SingleOne,Basis

```

class Basis      {
    protected int num1;
    public Basis()
    {
    }
    public Basis(int n)
    {
        this.num1 = n;
    }
    public virtual void Print()
    {
        Console.WriteLine(this.num1);
    }
}

class SingleOne:Basis    {
    protected int num2;
    public SingleOne(int n1, int n2):base(n1)
    {
        this.num2 = n2;
    }
    public override void Print()
    {
        base.Print();
        Console.WriteLine(this.num2);
    }
}

class MultiOne : Basis      {
    private int count = 0;
    private Basis[] arr;
    public MultiOne()
    {
        this.arr = new Basis[5];
    }
    public override void Print()

```

```

{
    for (int i = 0; i < count; i++)
        arr[i].Print();
}
public void Add(Basis b)
{
    if(count<arr.Length)
    {
        arr[count] = b;
        count++;
    }
}
}

```

(2 נק') א. ציירו תרשים UML שמייצג את הקשרים בין המחלקות Basis, SingleOne, MultiOne.

יש לסמן ירושה באמצעות החץ —————> והכלה באמצעות הסימן —————>

(10 נק') ב. תונה המחלקה Test ובה הפעולה הראשית.

עקבו בעזרת טבלת מעקב אחר הפעולה Main במחלקה Test, ורשמו את הפלט.

על הטבלה לכלול את ערכי כל המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```

class Test
{
    public static void Main(string[] args)
    {
        MultiOne container = new MultiOne();
        SingleOne s1 = new SingleOne(11, 35);
        container.Add(s1);
        s1 = new SingleOne(47, 22);
        container.Add(s1);
        s1 = new SingleOne(8, 17);
        container.Add(s1);
        MultiOne subContainer = new MultiOne();
        s1 = new SingleOne(53, 40);
        subContainer.Add(s1);
        s1 = new SingleOne(21, 13);
        subContainer.Add(s1);
        s1 = new SingleOne(39, 62);
        subContainer.Add(s1);
        container.Add(subContainer);
        container.Print();
    }
}

```

**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 - מחסנית $\text{Stack}<T>$

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). <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 $Queue<T>$

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. <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.

## שאלון 97105 מבני נתונים ותכנות מונחה עצמיים – מועד א' אביב 2022

שאלה	סעיף	ניקוד	הערות
1	א	8	חריגה – להוריד 2 נקודות לולאה אין סופית – להוריד 2 נק'
	ב	7	חריגה – להוריד 2 נקודות לולאה אין סופית – להוריד 2 נק'
2	א	12	הערה: אין חשיבות לסדר איברים בתוך חלקים
	ב	3	בלי הסבר – לא לתת נקודות
3	א	9	• כל סעיף – 1.5 נקודות
	ב	6	
4	א	5	• כל סעיף – נקודה אחת
	ב	2	
	ג	4	• כל סעיף – נקודה אחת
	ד	4	
5	א	5	אם לא ציין ממשק – להוריד 0.5 נקודות עבור כל מחלקה
	ב	6	• כל סעיף – נקודה אחת
	ג	4	• כותרת פעולה – 1 נקודה
			• מעבר – 1 נקודה
			• בדיקה והמרה – 1 נקודה
			• מניה והחזרת ערך – 1 נקודה
6	א	9	אם בנה שרשרת מספרים שלמים – להוריד נקודה אחת אם כלל 1- בחישוב – להוריד נקודה אחת
	ב	6	אם לא בדק שיש יותר מציון אחד ברצף – להוריד נקודה אחת
7	א	6	אם השתמש בפעולות נוספות – להוריד 4 נקודות
	ב	6	אם השתמש בפעולות נוספות – להוריד 4 נקודות
8	ג	3	בלי הסבר – לא לתת נקודות
	א	8	בלי מעקב – לא לתת נקודות
	ב	7	הערה: לא חובה להראות דרך לבנייה
9	א	6	• כל כותרת – נקודה אחת אם לא ציון ירושה – לא לתת נקודה
	ב	6	כל פעולה – 2 נקודות
10	א	1	
	ב	6	כל בנאי – 2 נקודות
	ג	1	
	ד	4	
11	א	2	אם הפך כיוון חצים – להוריד נקודה אחת
	ב	10	הדפסה – 4 נקודות בנייה – 6 נקודות
			בלי מעקב – לתת 4 נקודות לכל היותר