

מדעי המחשב

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים.

שים לב: בבחינה זו חלוקת הנקודות שונה מן החלוקה בבחינה הרגילה.

פרק ראשון	—	$(15 \times 1) + (20 \times 1)$	—	35	נקודות
פרק שני	—	(20×2)	—	40	נקודות
פרק שלישי	—	(25×1)	—	25	נקודות
סך הכול	—			100	נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב שיש בו אפשרות תכנות.

ד. הוראות מיוחדות:

(1) את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני

כתוב בשפה אחת בלבד — Java או C#.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב — Java או C#.

(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.

המסלול הוא אחד מארבעת המסלולים האלה:

מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתוכניות שאתה כותב לא יורדו לך נקודות אם תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד. רשום "טיוטה" בראש כל עמוד המשמש טיוטה.

כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

השאלות

בשאלון זה שלושה פרקים.

עליך לענות על שאלות משלושת הפרקים, לפי ההוראות בכל פרק.

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הנח שבתוכנית כתובה ההוראה:

```
Scanner input = new Scanner (System.in);
```

פרק ראשון (35 נקודות)

ענה על שאלה 1 – חובה. (15 נקודות)

שים לב: לשאלה זו שני נוסחים: בשפת Java (עמודים 2-3) ובשפת C# (עמוד 4).

1. לפותרים בשפת Java

לפניך המחלקה עיפרון **Pencil**.

Pencil	
private int length	תכונה של אורך העיפרון
private boolean sharpened	תכונה שהערך שלה true אם העיפרון מחודד, אחרת הערך שלה false.
public Pencil (int length, boolean sharpened)	פעולה בונה המקבלת ערכים בעבור כל התכונות של העיפרון.
public boolean isLonger (Pencil other)	פעולה המקבלת עצם אחר מטיפוס Pencil ומחזירה true אם העיפרון הנוכחי ארוך יותר מן העיפרון האחר – other, אחרת היא מחזירה false.
	הנח ש־ other אינו null.
public boolean isSharpened ()	פעולה המחזירה true אם העיפרון מחודד, אחרת היא מחזירה false.

(שים לב: המשך השאלה בעמוד הבא.)

א. ממש את הפעולות isLonger ו־ isSharpened.

ב. לפניך קטע תוכנית הכתוב בפעולה main של המחלקה הראשית.

```
Pencil pencil1 = new Pencil (12, true);
Pencil pencil2 = new Pencil (13, false);
if (pencil1.isLonger (pencil2) && pencil1.isSharpened ())
    System.out.println ("I choose pencil1");
else {
    if (pencil2.isSharpened ())
        System.out.println ("I choose pencil2");
    else
        System.out.println ("pencil2 needs to be sharpened");
}
```

עקוב אחר ביצוע קטע התוכנית, הצג את העצמים שנוצרו, ורשום את הפלט.

C# לפותרים בשפת

לפניך המחלקה עיפרון **Pencil**.

Pencil	
private int length	תכונה של אורך העיפרון
private bool sharpened	תכונה שהערך שלה true אם העיפרון מחודד, אחרת הערך שלה false.
public Pencil (int length, bool sharpened)	פעולה בונה המקבלת ערכים בעבור כל התכונות של העיפרון.
public bool IsLonger (Pencil other)	פעולה המקבלת עצם אחר מטיפוס Pencil ומחזירה true אם העיפרון הנוכחי ארוך יותר מן העיפרון האחר – other, אחרת היא מחזירה false. הנחש other אינו null.
public bool IsSharpened ()	פעולה המחזירה true אם העיפרון מחודד, אחרת היא מחזירה false.

א. ממש את הפעולות IsLonger ו- IsSharpened.

ב. לפניך קטע תוכנית הכתוב בפעולה Main של המחלקה הראשית.

```

Pencil pencil1 = new Pencil (12, true);
Pencil pencil2 = new Pencil (13, false);
if (pencil1.IsLonger (pencil2) && pencil1.IsSharpened ())
    Console.WriteLine ("I choose pencil1");
else {
    if (pencil2.IsSharpened ())
        Console.WriteLine ("I choose pencil2");
    else
        Console.WriteLine ("pencil2 need to be sharpened");
}
    
```

עקוב אחר ביצוע קטע התוכנית, הצג את העצמים שנוצרו, ורשום את הפלט.

ענה על אחת מן השאלות 2-3. (20 נקודות)

2. נתונה המחלקה **Doctor** – רופא, ולה שלוש תכונות:

- name – שם הרופא, מטיפוס מחרוזת.
- specialization – תחום ההתמחות, מטיפוס מחרוזת.
- rate – דירוג הרופא, מטיפוס מספר ממשי, שמקבל ערכים מ-0 עד 10.

הנח שלכל תכונה במחלקה **Doctor** הוגדרו בשפת Java הפעולות get ו-set ובשפת C# הפעולות Get ו-Set.

א. כתוב פעולה בונה המקבלת ערכים בעבור שם הרופא ותחום ההתמחות ומאתחלת את התכונה rate ל-0.

דירוג (rate) של רופא מחושב לפי ממוצע ההערכות של המטופלים שלו. הערכה היא מספר שלם בין 0 ל-10 (כולל).

ב. כתוב פעולה void חיצונית בשפת Java בשם doctorRating, או בשפת C# בשם DoctorRating, שתקבל

מערך בשם doctors מטיפוס **Doctor**. הנח שערך הדירוג – rate של איברי המערך הוא 0.

הפעולה תקלוט, לפי סדר האיברים במערך, את כל ההערכות שנתנו המטופלים לכל רופא.

הקלט בעבור כל רופא מסתיים בקליטת הערך (-1), ולאחר הערך (-1) מתחיל הקלט של הרופא הבא במערך,

עד אחרון הרופאים במערך (המסתיים גם הוא בקליטת הערך (-1)).

הפעולה תעדכן את הדירוג של כל רופא לפי חישוב ממוצע ההערכות של המטופלים שלו.

לדוגמה: בעבור מערך – doctors בגודל 3 התקבל הקלט הזה:

10, 7, 9, 8, -1, 2, -1, 6, 7, -1 (משמאל לימין).

הפעולה תעדכן את הדירוג של הרופא הראשון במערך (באינדקס 0) ל-8.5, את הדירוג של הרופא השני ל-2

ואת הדירוג של הרופא השלישי ל-6.5.

הנח שיש לפחות הערכה אחת בעבור כל רופא במערך.

הנח שכל האיברים במערך doctors שונים מ-null.

אין צורך לבדוק את תקינות הקלט.

3. איבר במערך חד־ממדי ייקרא **איבר שוויון** אם סכום האיברים במערך מצד ימין שלו שווה לסכום האיברים במערך מצד שמאל שלו.

לאיבר שנמצא בקצה המערך אין איברים מצד ימין או מצד שמאל שלו, ובמקרה זה הסכום שבצד שבו אין איברים נחשב ל־0.

לדוגמה: במערך החד־ממדי בגודל 6 שלפניו האיבר המסומן הוא **איבר שוויון**, כי סכום האיברים משני צדדיו שווה ל־6.

-1	7	4	2	3	1
----	---	---	---	---	---

דוגמה נוספת: במערך החד־ממדי בגודל 7 שלפניו האיבר המסומן הוא **איבר שוויון**, כי סכום האיברים משני צדדיו שווה ל־0.

6	-3	4	2	0	-6	3
---	----	---	---	---	----	---

א. כתוב בשפת Java או בשפת C# פעולה חיצונית שתקבל:

— מערך חד־ממדי מטיפוס שלם.

— מספר המציין מקום של איבר במערך (אינדקס), מטיפוס שלם.

הנח שהמספר המתקבל תקין.

הפעולה תבדוק אם איבר זה הוא **איבר שוויון**.

אם כן — תחזיר הפעולה true, אחרת היא תחזיר false.

ב. מערך חד־ממדי ייקרא **מערך שוויוני** אם במערך יש לפחות איבר אחד שהוא **איבר שוויון**.

כתוב בשפת Java או בשפת C# פעולה חיצונית שתקבל מערך חד־ממדי מטיפוס שלם.

הפעולה תבדוק אם מערך זה הוא **מערך שוויוני**.

אם כן — תחזיר הפעולה true, אחרת היא תחזיר false.

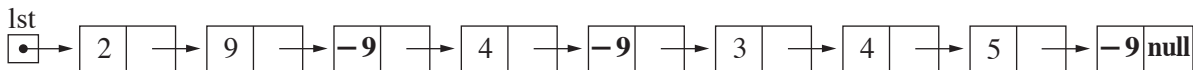
הערה: אפשר להשתמש בפעולה שכתבת בסעיף א.

פרק שני (40 נקודות)

שים לב: בכל שאלה שנדרש בה מימוש, תוכל להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם תשתמש בפעולות נוספות, יהיה עליך לממש אותן.

ענה על שתיים מן השאלות 4-6. (לכל שאלה – 20 נקודות)

- 4. • **"שרשרת מספרים שלמים חיוביים"** היא שרשרת חוליות שכל חוליה בה מכילה מספר שלם הגדול מ-0.
 - **"שרשרת ספרות"** היא שרשרת חוליות שכל חוליה בה מכילה ספרה בין 0 ל-9 (כולל) או את המספר (-9). כל רצף ספרות בשרשרת מייצג מספר: הספרה הראשונה מייצגת את האחדות, הספרה השנייה את העשרות וכן הלאה. לאחר כל רצף של ספרות מופיע המספר (-9), והוא מסמן סוף של מספר בשרשרת.
- לפניך דוגמה ל"שרשרת ספרות" המייצגת את המספרים: 92, 4, 543.



כתוב פעולה חיצונית בשפת Java בשם buildDigit או בשפת C# בשם BuildDigit, המקבלת הפניה lst שאינה null ל"שרשרת מספרים שלמים חיוביים". הפעולה תחזיר "שרשרת ספרות" המייצגת את המספרים שב"שרשרת מספרים שלמים חיוביים" לפי הסדר.

5.

נגדיר "בלוק" במחסנית כרצף של לפחות שני איברים זהים.

כתוב בשפת Java או בשפת C# פעולה חיצונית המקבלת מחסנית stk – מטיפוס שלם, ומחזירה מחסנית חדשה.

המחסנית המוחזרת תכיל את כל האיברים מהמחסנית stk שאינם ב"בלוק".

אין חשיבות לסדר האיברים במחסנית המוחזרת.

הערות: – אם מספר מסוים מופיע לא ב"בלוק", הוא יהיה במחסנית המוחזרת גם אם נוסף על כך אותו המספר

מופיע ב"בלוק".

– אם מספר מסוים מופיע כמה פעמים, לא ב"בלוק", הוא יופיע אותה כמות פעמים במחסנית המוחזרת.

– אם המחסנית stk ריקה, תוחזר מחסנית ריקה.

דוגמה:

המחסנית המוחזרת

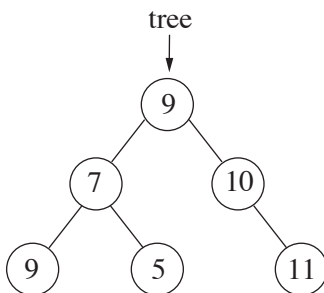
15
3
5
15
14

מחסנית stk

14
15
5
5
0
0
0
5
3
-4
-4
15

ראש המחסנית ←

6. עץ בינרי מטיפוס שלם של מספרים שאינם שליליים הוא "עץ שאריות שוויוני" במקרה הזה:
 כמות האיברים שמספריהם מתחלקים ב-3 עם שארית 1 שווה לכמות האיברים שמספריהם מתחלקים ב-3 עם שארית 2, ושווה לכמות האיברים שמספריהם מתחלקים ב-3 ללא שארית.
 דוגמה של "עץ שאריות שוויוני":



עץ בינרי זה הוא "עץ שאריות שוויוני" משום שיש בו שני מספרים שמתחלקים ב-3 ללא שארית (9, 9), שני מספרים שמתחלקים ב-3 עם שארית 1 (10, 7) ושני מספרים שמתחלקים ב-3 עם שארית 2 (11, 5).

כתוב פעולה חיצונית בוליאנית בשפת Java בשם `treeEqual` או בשפת C# בשם `TreeEqual` המקבלת עץ בינרי מטיפוס שלם, לא ריק, של מספרים שאינם שליליים ובודקת אם הוא "עץ שאריות שוויוני".
 אם כן – תחזיר הפעולה `true`, אחרת היא תחזיר `false`.

פרק שלישי (25 נקודות)

בפרק זה שאלות בארבעה מסלולים:
 מערכות מחשב ואסמבלי, עמודים 10-11.
 מבוא לחקר ביצועים, עמודים 12-14.
 מודלים חישוביים, עמוד 15.
 תכנות מונחה עצמים בשפת Java, עמודים 16-19; תכנות מונחה עצמים בשפת C#, עמודים 20-23.
ענה על שאלה אחת במסלול שלמדת.

מערכות מחשב ואסמבלי

אם למדת מסלול זה, ענה על אחת מן השאלות 7-8 (25 נקודות).

7. בשאלה זו שלושה סעיפים, א-ג. אין קשר בין הסעיפים. עליך לענות על כולם.

א. לפניך קטע תוכנית באסמבלי:

```
MOV BX, 100H
MOV SI, 0
MOV DI, 7
MOV AL, 4
MOV AH, 8
AGAIN: MOV [ BX +SI ], AL
        MOV [ BX +DI ], AH
        INC SI
        DEC DI
        CMP DI, 1
        JA AGAIN
```

עקוב בעזרת טבלת מעקב אחר ביצוע קטע התוכנית. בטבלת המעקב יש לכלול עמודה בעבור כל אחד מן הגורמים האלה: AL, AH, BX, SI, DI, BX+DI, BX+SI. כמו כן יש לצייר את מפת הזיכרון בתחום הרלוונטי.

ב. כתוב קטע קוד הבודק אם בשני האוגרים, AL ו־AH, מאוחסנים מספרים אי־זוגיים. אם כן, יציב קטע הקוד את הספרה 1 באוגר DL, אחרת יציב קטע הקוד את הספרה 0 באוגר DL.

ג. מה יהיו ערכי הדגלים SF, ZF, CF והאוגר AX לאחר ביצוע כל הפקודות שלפניך?

```
MOV AL, 20h
MOV AH, 6h
SUB AX, 0F6h
```

8. מערך "סדרתי ריבועי" הוא מערך שבו הערך של כל איבר שווה לריבוע של האינדקס שלו במערך.

לדוגמה: המערך שלפניך הוא מערך "סדרתי ריבועי":

0	1	2	3	4	5	6	7	8	9	אינדקס
0	1	4	9	16	25	36	49	64	81	ערך

במקטע הנתונים הוגדרו הנתונים שלפניך:

ARR DB 10 DUP (?)

RANK DB ?

א. כתוב פרוצדורה בשם POW המקבלת מספר שלם בין 0 ל-9 (כולל) באמצעות מחסנית, ומאחסנת באוגר AL את הריבוע שלו. אין צורך לבדוק את תקינות הקלט.

ב. כתוב תוכנית הבודקת אם המערך ARR הוא "סדרתי ריבועי".

אם כן, תציב התוכנית במשתנה RANK את המספר 10, אחרת, תציב התוכנית במשתנה RANK את האינדקס של האיבר הראשון שערכו אינו ריבוע של האינדקס שלו. עליך להיעזר בפרוצדורה שכתבת בסעיף א.

לדוגמה: המערך ARR שלפניך אינו "סדרתי ריבועי", מכיוון שיש בו איברים שערכיהם אינם שווים לריבוע של האינדקסים שלהם. במקרה שבדוגמה זו תציב התוכנית במשתנה RANK את האינדקס 4, משום שבו האיבר הראשון במערך שאינו שווה לריבוע האינדקס שלו.

0	1	2	3	4	5	6	7	8	9	אינדקס
0	1	4	9	20	22	36	49	60	70	ערך

מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על אחת מן השאלות 9-10. (25 נקודות)

9. בשאלה זו חמישה סעיפים, א-ה. אין קשר בין הסעיפים. עליך לענות על כולם.

נתונה בעיית תכנון לינארי:

$$\max \{z = x_1 + 3x_2\}$$

בכפוף לאילוצים האלה:

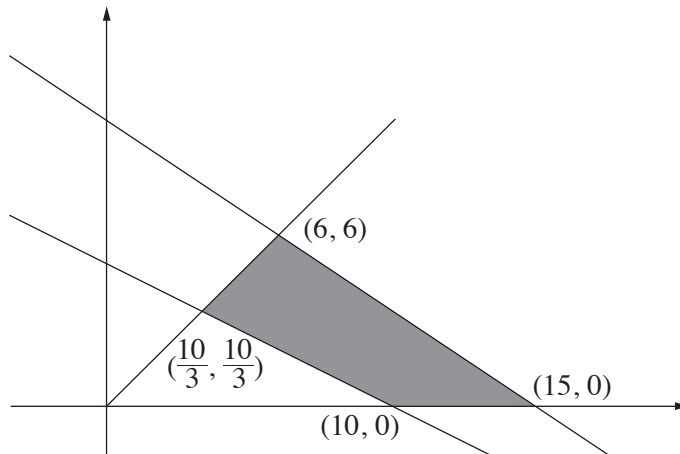
(1) $-10x_1 - 15x_2 \geq -150$

(2) $5x_1 + 10x_2 \geq 50$

(3) $x_1 - x_2 \geq 0$

(4) $x_2 \geq 0$

לפניך סרטוט של תחום הפתרונות האפשריים לבעיה הנתונה.



כל אחד מן הסעיפים, א-ה, שבעמוד הבא מתייחס לבעיית התכנון הלינארי הנתונה.

נתונים ארבעה היגדים, i-iv. בעבור כל סעיף מן הסעיפים, א-ה, שבעמוד הבא יש רק היגד אחד שהוא נכון.

i יש רק פתרון אופטימלי יחיד.

ii יש אין-סוף פתרונות אופטימליים.

iii הפתרון האופטימלי אינו חסום.

iv אין פתרון אופטימלי.

בעבור כל אחד מן הסעיפים א-ה קבע איזה מן ההיגדים i-iv הוא הנכון. ציין את הסעיף, העתק את ההיגד הנכון למחברתך, ונמק את קביעתך.

- אם בחרת בהיגד i בעבור סעיף כלשהו, עליך למצוא את הפתרון האופטימלי היחיד, ואת הערך של פונקציית המטרה בפתרון זה.
- אם בחרת בהיגד ii בעבור סעיף כלשהו, עליך לרשום את הפתרון האופטימלי הכללי לבעיה, ואת הערך של פונקציית המטרה בתחום הפתרונות האופטימליים.
- א.** איזה היגד הוא הנכון בעבור בעיית התכנון הלינארי הנתונה בתחילת השאלה? נמק את תשובתך.
- ב.** מְשַׁנִּים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\min \{z = 3x_1 + 3x_2\}$. איזה היגד הוא הנכון בעבור הבעיה לאחר השינוי? נמק את תשובתך.
- ג.** מְשַׁנִּים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\max \{z = -(3x_1 + 3x_2)\}$. איזה היגד הוא הנכון בעבור הבעיה לאחר השינוי? נמק את תשובתך.
- ד.** מוסיפים את האילוץ $x_1 \geq 15$ לבעיה הנתונה בתחילת השאלה. איזה היגד הוא הנכון בעבור הבעיה לאחר הוספת האילוץ? נמק את תשובתך.
- ה.** מְשַׁנִּים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\max \{z = ax_1 + 3x_2\}$. בעבור אילו ערכים של a יהיה $(15, 0)$ הפתרון האופטימלי לבעיית התכנון הנתונה? נמק את תשובתך.

10. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

א. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכים של $u_1, u_2, u_3, v_1, v_2, v_3$. האם הפתרון הוא אופטימלי? נמק את תשובתך.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	14	15	17	180	0
		150	30		
2	10	8	14	100	-7
	100	0			
3	15	20	18	100	1
			100		
ביקוש	100	150	130		
v_j	17	15	17		

ב. בטבלה שלפניך נתונה בעיית תובלה וחלק מפתרון בסיסי אפשרי: $x_{11} = 100$.

מקורות	יעדים				היצע
	1	2	3	4	
1	5	3	8	7	300
	100				
2	9	5	10	12	500
3	8	6	7	3	200
ביקוש	100	300	200	400	

- (1) העתק את הטבלה למחברתך, והשלם בה את הערכים לפי שיטת הפינה הצפונית-מערבית.
- (2) אם משנים את המחיר בתא (3, 3) מ-7 ל-3, האם ישתנה הפתרון הבסיסי האפשרי שמצאת בתת-סעיף (1)? נמק את תשובתך.
- (3) נקבע $u_2 = 0$. השלם בטבלה שכתבת בתת-סעיף (1) את ערכי $u_1, u_3, v_1, v_2, v_3, v_4$.
- (4) האם הפתרון הבסיסי האפשרי הוא הפתרון האופטימלי?

מודלים חישוביים

אם למדת מסלול זה, ענה על אחת מן השאלות 11-12. (25 נקודות)

11. לפניך השפה L מעל הא"ב $\{a, b\}$.

$$L = \{a^n b^m a^k \mid n > 0, m > n, n + k > m\}$$

בנה אוטומט מחסנית שיקבל את השפה L .

דוגמאות למילים ששייכות לשפה L :

abbaa, aabbbaa

דוגמאות למילים שאינן שייכות לשפה L :

baa, aba, abba

12. נתונה פעולה הכתובה בשפת Java ובשפת C#.

הפעולה מקבלת שני מספרים שלמים הגדולים מ-0.

Java	C#
public static int foo (int x, int y)	public static int Foo (int x, int y)
{	{
return (x+y)/3;	return (x+y)/3;
}	}

כתוב מכונת טיורינג שתממש את הפעולה הנתונה.

הנח ש- $x + y$ מתחלק ב-3 ללא שארית.

הקלט של המכונה הוא שני המספרים x, y שהפעולה מקבלת, והוא כתוב על הסרט מתחילתו.

כל מספר כתוב בצורה אונרית. הסימן # מפריד בין מספר למספר.

לדוגמה: אם הפעולה מקבלת 3 ל- x ו-3 ל- y , הסרט ייראה כך:

⊢	1	1	1	#	1	1	1	Δ
---	---	---	---	---	---	---	---	---	------

הפלט הוא הערך שהפעולה מחזירה והוא ייכתב על הסרט במקום כלשהו כערך אונרי בין שני סימני \$.

פלט אפשרי לדוגמה שלעיל:

....	Δ	Δ	\$	1	1	\$	Δ	Δ
------	---	---	----	---	---	----	---	---	------

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב בשפת Java, ענה על אחת מן השאלות 13-14. (25 נקודות)

13. במועדון כושר "השלום" אפשר להתאמן בכל מתקני הכושר. נוסף על כך אפשר להתאמן אימון מיוחד שבו אפשר

להשתמש בכל מתקני הכושר בליווי של מאמן אישי לזמן מוגדר.

כדי לשמור מידע על המתרחש במועדון נבנו המחלקות שלפניך:

מאמן — **Trainer**, אימון — **Training**, אימון מיוחד — **SpecialT**, לקוח — **Client**, הנהלה — **Management**.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים על מאמן:

- name — שם המאמן, מטיפוס מחרוזת.
- wid — מספר עובד, מטיפוס שלם.

המידע ששומרים על אימון:

- num — מספר המתקנים שבהם השתמש הלקוח באימון, מטיפוס שלם.

המידע ששומרים על אימון מיוחד:

- num — מספר המתקנים שבהם השתמש הלקוח באימון, מטיפוס שלם.
- trainer — המאמן שליווה את הלקוח באימון, מטיפוס **Trainer**.
- time — מספר הדקות של האימון, מטיפוס שלם. המספר גדול מ-0 ואינו מוגבל ל-60.

המידע ששומרים על לקוחות המועדון:

- id — מספר תעודת זהות, מטיפוס מחרוזת.
- name — שם הלקוח, מטיפוס מחרוזת.
- visits — כל האימונים שהיו ללקוח, במערך מטיפוס **Training**.

המידע ששומרים בעבור ההנהלה:

- staff — כל המאמנים העובדים במועדון, במערך מטיפוס **Trainer**.
- clients — כל הלקוחות של המועדון, במערך מטיפוס **Client**.

(שים לב: המשך השאלה בעמוד הבא.)

על הסעיפים שלפניך יש לענות באופן המתאים ביותר לעקרונות תכנות מונחה עצמים.

א. סרטט תרשים הייררכייה בין המחלקות.

יש לסמן ירושה בעזרת החץ: \longrightarrow

יש לסמן הכלה באמצעות הסימן: \blacklozenge

ב. כתוב את כותרות ותכונות המחלקה.

הנח שיש פעולה בונה ופעולות `get` ו-`set` בכל המחלקות.

ג. כתוב פעולה פנימית במחלקה **Client**, המחזירה את סכום כל הדקות שלקוח התאמן אימון מיוחד.

הפעולה תחזיר 0 אם הלוקוח אף פעם לא התאמן אימון מיוחד.

ד. כתוב פעולה פנימית במחלקה **Management**, המחזירה את סך כל הלוקוחות שהתאמנו אימון מיוחד

לפחות פעם אחת.

14. בחנות הכולבו "הכול לבית" מוכרים מוצרי מזון שאינם דורשים אחסון בקירור, מוצרי מזון הדורשים אחסון בקירור ומוצרי אלקטרוניקה.

כדי לנהל את מלאי המוצרים שבחנות נבנו המחלקות האלה:

Item – מוצר, **FoodItem** – מוצר מזון, **FoodRefrigerated** – מוצר מזון הדורש אחסון בקירור, **ElectronicItem** – מוצר אלקטרוניקה, **ItemDate**.

ItemDate היא מחלקה שמייצגת תאריך ויש בה 3 תכונות מטיפוס מספר שלם: יום day, חודש month, שנה year.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים בנוגע לכל אחד מן המוצרים בחנות:

- name – שם המוצר, מטיפוס מחרוזת.
- catalogNumber – מספר קטלוגי, מטיפוס מחרוזת.
- quantity – מספר פריטים מן המוצר במלאי, מטיפוס שלם.
- minQuantity – המספר המינימלי של הפריטים מן המוצר שיש להחזיק במלאי, מטיפוס שלם.

המידע הנוסף ששומרים בנוגע לכל מוצרי המזון:

- expiryDate – תאריך התפוגה של המוצר, מטיפוס **ItemDate**.

המידע הנוסף ששומרים בנוגע למוצרי המזון הדורשים אחסון בקירור:

- minTemperature – הטמפרטורה המינימלית הנדרשת לאחסון מוצר המזון, מטיפוס שלם.
- maxTemperature – הטמפרטורה המקסימלית האפשרית לאחסון מוצר המזון, מטיפוס שלם.

המידע הנוסף ששומרים בנוגע למוצרי האלקטרוניקה:

- guaranteeDate – תאריך סוף האחריות של היצרן על המוצר, מטיפוס **ItemDate**.

הערה: מוצרי המזון שאינם דורשים אחסון בקירור ומוצרי האלקטרוניקה נשמרים בכל טמפרטורה.


אפשר להניח שבכל המחלקות יש פעולות בונות ופעולות get ו set.

(שים לב: המשך השאלה בעמוד הבא.)

א. סרטט תרשים הייררכייה בין המחלקות:

Item, ElectronicItem, FoodItem, FoodRefrigerated, ItemDate.

יש לסמן ירושה באמצעות החץ: 

יש לסמן הכלה באמצעות הסימן: 

ב. מְמַש את הפעולה הפנימית () isMissing . הפעולה תחזיר true בעבור המוצר אם מספר הפריטים במלאי

נמוך ממספר הפריטים המינימלי הנדרש במלאי, אחרת היא תחזיר false .

יש לציין באיזו מחלקה נמצאת הפעולה.

ג. במחלקה **Tester** הוגדרה הפעולה:

```
public class Tester {
    public static boolean canBeStored (Item item, int temp) {
        return item.canBeStored (temp); }
}
```

הפעולה מחזירה true אם אפשר לשמור את המוצר item בטמפרטורה temp , אחרת היא מחזירה false .

הוסף פעולות נדרשות למחלקות כדי שהפעולה תבצע את הנדרש.

ציין בעבור כל פעולה באיזו מחלקה היא מוספת.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב בשפת C#, ענה על אחת מן השאלות 15-16. (25 נקודות)

15. במועדון כושר "השלום" אפשר להתאמן בכל מתקני הכושר. נוסף על כך אפשר להתאמן אימון מיוחד שבו אפשר

להשתמש בכל מתקני הכושר בליווי של מאמן אישי לזמן מוגדר.

כדי לשמור מידע על המתרחש במועדון נבנו המחלקות שלפניך:

מאמן — **Trainer**, אימון — **Training**, אימון מיוחד — **SpecialT**, לקוח — **Client**, הנהלה — **Management**.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים על מאמן:

- name — שם המאמן, מטיפוס מחרוזת.
- wid — מספר עובד, מטיפוס שלם.

המידע ששומרים על אימון:

- num — מספר המתקנים שבהם השתמש הלקוח באימון, מטיפוס שלם.

המידע ששומרים על אימון מיוחד:

- num — מספר המתקנים שבהם השתמש הלקוח באימון, מטיפוס שלם.
- trainer — המאמן שליווה את הלקוח באימון, מטיפוס **Trainer**.
- time — מספר הזקות של האימון, מטיפוס שלם. המספר גדול מ־0 ואינו מוגבל ל־60.

המידע ששומרים על לקוחות המועדון:

- id — מספר תעודת זהות, מטיפוס מחרוזת.
- name — שם הלקוח, מטיפוס מחרוזת.
- visits — כל האימונים שהיו ללקוח, במערך מטיפוס **Training**.

המידע ששומרים בעבור ההנהלה:

- staff — כל המאמנים העובדים במועדון, במערך מטיפוס **Trainer**.
- clients — כל הלקוחות של המועדון, במערך מטיפוס **Client**.

(שים לב: המשך השאלה בעמוד הבא.)

על הסעיפים שלפניך יש לענות באופן המתאים ביותר לעקרונות תכנות מונחה עצמים.

א. סרטט תרשים הייררכייה בין המחלקות.

יש לסמן ירושה בעזרת החץ: 

יש לסמן הכלה באמצעות הסימן: 

ב. כתוב את כותרות ותכונות המחלקה.

הנח שיש פעולה בונה ופעולות Get ו- Set בכל המחלקות.

ג. כתוב פעולה פנימית במחלקה **Client**, המחזירה את סכום כל הדקות שלקוח התאמן אימון מיוחד.

הפעולה תחזיר 0 אם הלוקוח אף פעם לא התאמן אימון מיוחד.

ד. כתוב פעולה פנימית במחלקה **Management**, המחזירה את סך כל הלקוחות שהתאמנו אימון מיוחד

לפחות פעם אחת.

16. בחנות הכולבו "הכול לבית" מוכרים מוצרי מזון שאינם דורשים אחסון בקירור, מוצרי מזון הדורשים אחסון בקירור ומוצרי אלקטרוניקה.

כדי לנהל את מלאי המוצרים שבחנות נבנו המחלקות האלה:

Item – מוצר, **FoodItem** – מוצר מזון, **FoodRefrigerated** – מוצר מזון הדורש אחסון בקירור, **ElectronicItem** – מוצר אלקטרוניקה, **ItemDate**.

ItemDate היא מחלקה שמייצגת תאריך ויש בה 3 תכונות מטיפוס מספר שלם: יום day, חודש month, שנה year.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים בנוגע לכל אחד מן המוצרים בחנות:

- name – שם המוצר, מטיפוס מחרוזת.
- catalogNumber – מספר קטלוגי, מטיפוס מחרוזת.
- quantity – מספר פריטים מן המוצר במלאי, מטיפוס שלם.
- minQuantity – המספר המינימלי של הפריטים מן המוצר שיש להחזיק במלאי, מטיפוס שלם.

המידע הנוסף ששומרים בנוגע לכל מוצרי המזון:

- expiryDate – תאריך התפוגה של המוצר, מטיפוס **ItemDate**.

המידע הנוסף ששומרים בנוגע למוצרי המזון הדורשים אחסון בקירור:

- minTemperature – הטמפרטורה המינימלית הנדרשת לאחסון מוצר המזון, מטיפוס שלם.
- maxTemperature – הטמפרטורה המקסימלית האפשרית לאחסון מוצר המזון, מטיפוס שלם.

המידע הנוסף ששומרים בנוגע למוצרי האלקטרוניקה:

- guaranteeDate – תאריך סוף האחריות של היצרן על המוצר, מטיפוס **ItemDate**.

הערה: מוצרי המזון שאינם דורשים אחסון בקירור ומוצרי האלקטרוניקה נשמרים בכל טמפרטורה.

אפשר להניח שבכל המחלקות יש פעולות בונות ופעולות Get ו-Set.

(שים לב: המשך השאלה בעמוד הבא.)

א. סרטט תרשים הייררכייה בין המחלקות:

Item, ElectronicItem, FoodItem, FoodRefrigerated, ItemDate.

יש לסמן ירושה באמצעות החץ: 

יש לסמן הכלה באמצעות הסימן: 

ב. ממש את הפעולה הפנימית () `IsMissing`. הפעולה תחזיר `true` בעבור המוצר אם מספר הפריטים במלאי

נמוך ממספר הפריטים המינימלי הנדרש במלאי, אחרת היא תחזיר `false`.

יש לציין באיזו מחלקה נמצאת הפעולה.

ג. במחלקה **Tester** הוגדרה הפעולה:

```
public class Tester {
    public static bool CanBeStored (Item item, int temp) {
        return item.CanBeStored (temp); }
}
```

הפעולה מחזירה `true` אם אפשר לשמור את המוצר `item` בטמפרטורה `temp`, אחרת היא מחזירה `false`.

הוסף פעולות נדרשות למחלקות כדי שהפעולה תבצע את הנדרש.

ציין בעבור כל פעולה באיזו מחלקה היא מוספת.

בהצלחה!