

## מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

### הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.  
פרק ראשון – בפרק זה ארבע שאלות, — (25x2) — 50 נקודות  
ומהן יש לענות על שתיים.  
פרק שני – בפרק זה שאלות בארבעה מסלולים שונים — (25x2) — 50 נקודות  
ענה על שאלות רק במסלול שלמדת,  
לפי ההוראות בקבוצת השאלות במסלול זה.  
סה"כ — 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.
- ד. הוראות מיוחדות:  
(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון  
כתוב בשפה אחת בלבד – C# או Java.  
(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – C# או Java.  
(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת,  
אחד מארבעת המסלולים: מערכות מחשב ואסמבלר, מבוא לחקר ביצועים,  
מודלים חישוביים, תכנות מונחה עצמים.  
הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום  
אות קטנה או להפך.  
כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).  
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!
- ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

**בהצלחה!**

/המשך מעבר לדף/

## השאלות

בשאלון זה שני פרקים: פרק ראשון ופרק שני.  
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. רשימה L תיקרא **משולשת** אם היא מקיימת את התנאים האלה:

- \* הרשימה אינה ריקה.
  - \* מספר האיברים בה מתחלק ב-3 בלי שארית.
  - \* האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים בשליש השני של הרשימה ואותם ערכים שמכילים האיברים בשליש השלישי של הרשימה. הערכים מסודרים באותו סדר בכל אחד מהשלישים.
- לדוגמה: הרשימה L1 שלפניך היא רשימה **משולשת** באורך 12.

L1:  $\boxed{2} \rightarrow \boxed{5} \rightarrow \boxed{3} \rightarrow \boxed{7} \rightarrow \boxed{2} \rightarrow \boxed{5} \rightarrow \boxed{3} \rightarrow \boxed{7} \rightarrow \boxed{2} \rightarrow \boxed{5} \rightarrow \boxed{3} \rightarrow \boxed{7} \rightarrow \text{null}$

כתוב ב-C# או ב-Java פעולה חיצונית שתקבל רשימה L שהאיברים שלה הם מטיפוס שלם.

אם L היא רשימה **משולשת**, הפעולה תחזיר true.

אחרת – הפעולה תחזיר false.

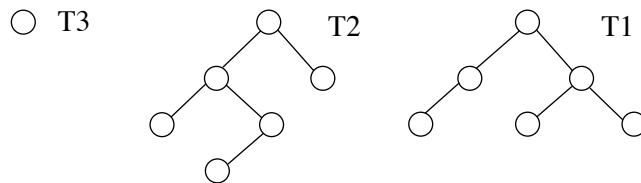
אתה יכול להשתמש בפעולות של המחלקות `Node < T >`, `List < T >`, בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

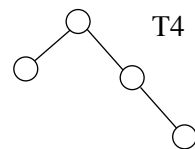
2. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. עץ בינרי ייקרא עץ **ימין-שמאל** אם לכל צומת בעץ אשר יש לו בן ימני, יש גם בן שמאלי.  
לדוגמה:

העצים T1, T2, T3 הם עצי **ימין-שמאל**



והעץ T4 אינו עץ **ימין-שמאל**



כתוב ב- C# או ב- Java פעולה חיצונית שתקבל עץ בינרי T שהצמתים שלו הם מטיפוס שלם. הפעולה תחזיר true אם הוא עץ **ימין-שמאל**, אחרת – הפעולה תחזיר false.

אתה יכול להשתמש בפעולות של המחלקה `BinTreeNode <T>` בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ב. (אין קשר לסעיף א.)

נתון עץ בינרי T שהצמתים שלו מכילים ערכים מטיפוס תו. אם סורקים את העץ T בסדר תחילי (preorder), אזי סדר הערכים המתקבל מביקור בצמתים (משמאל לימין) הוא:

X A I O N Y T D S

אם סורקים את העץ T בסדר תוכי (inorder), אזי סדר הערכים המתקבל מביקור בצמתים (משמאל לימין) הוא:

I N O A X D T S Y

צייר את העץ T, ורשום את סדר הערכים המתקבל מביקור בצמתים אם סורקים את העץ בסדר סופי (postorder).

/המשך בעמוד 4/

.3

**שים לב:** לשאלה זו שני נוסחים, האחד ב-Java (עמודים 4-5) ואחד ב-C# (עמודים 6-7). עבוד על פי השפה שלמדת.

לפתרים ב-Java

לפניך 2 פעולות חיצוניות הכתובות ב-Java .

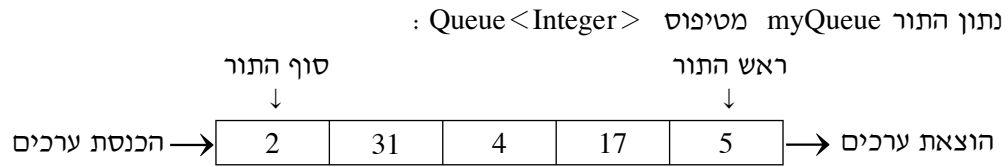
```
/** הפעולה מקבלת תור לא ריק, המכיל מספרים שלמים. **  
*/ הפעולה מחזירה...
```

```
public static int sod1 (Queue<Integer> q)  
{  
    int i = q.remove();  
    int result = i;  
  
    if (!q.isEmpty())  
    {  
        int j = sod1 (q);  
        if (result > j)  
            result = j;  
    }  
    q.insert(i);  
    return result;  
}
```

```
/** הפעולה מקבלת מספר שלם גדול מ-0 או שווה לו **  
*/ הפעולה מחזירה...
```

```
public static int sod2 (int i)  
{  
    if (i == 0)  
        return 0;  
    int a = i % 10;  
    int b = sod2(i / 10);  
    if (a > b)  
        return a;  
    return b;  
}
```

/המשך בעמוד 5/



- א.** מה יחזיר הזימון `sod1(myQueue)`? רשום את המעקב.  
 רשום את התור המתקבל בתום המעקב. ציין את ראש התור ואת סוף התור.
- ב.** מה מבצעת הפעולה `sod1(queue)` בעבור תור `queue` לא ריק מטיפוס `Queue<Integer>`?  
 מה יחזיר הזימון `sod2(17852)`? רשום את המעקב.
- ד.** מה מבצעת הפעולה `sod2(k)` בעבור מספר `k` גדול מ-0 מטיפוס שלם?
- ה.** מה מבצעת הפעולה `sod2(sod1(queue))` בעבור תור `queue` לא ריק מטיפוס `Queue<Integer>` המכיל רק מספרים שלמים וגדולים מ-0?

לפותרים ב- C#

לפניך 2 פעולות חיזוניות הכתובות ב- C#.

/\*  
 הפעולה מקבלת תור לא ריק, המכיל מספרים שלמים. \*\*  
 \*/  
 הפעולה מחזירה...

```
public static int Sod1 (Queue<int> q)
{
    int i = q.Remove();
    int result = i;

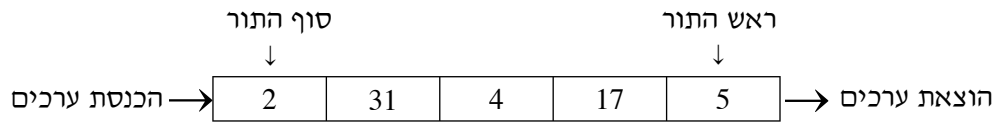
    if (!q.IsEmpty())
    {
        int j = Sod1 (q);
        if (result > j)
            result = j;
    }
    q.Insert(i);
    return result;
}
```

/\*  
 הפעולה מקבלת מספר שלם גדול מ- 0 או שווה לו \*\*  
 \*/  
 הפעולה מחזירה...

```
public static int Sod2 (int i)
{
    if (i == 0)
        return 0;
    int a = i % 10;
    int b = Sod2(i / 10);
    if (a > b)
        return a;
    return b;
}
```

/המשך בעמוד 7/

נתון התור `myQueue` מטיפוס `Queue<int>`:



- א. מה יחזיר הזימון `Sod1(myQueue)`? רשום את המעקב.  
 רשום את התור המתקבל בתום המעקב. ציין את ראש התור ואת סוף התור.
- ב. מה מבצעת הפעולה `Sod1(queue)` בעבור תור `queue` לא ריק מטיפוס `Queue<int>`?  
 מה יחזיר הזימון `Sod2(17852)`? רשום את המעקב.
- ד. מה מבצעת הפעולה `Sod2(k)` בעבור מספר `k` גדול מ-0 מטיפוס שלם?
- ה. מה מבצעת הפעולה `Sod2(Sod1(queue))` בעבור תור `queue` לא ריק מטיפוס `Queue<int>` המכיל רק מספרים שלמים וגדולים מ-0?

.4

**שים לב:** לשאלה זו שני נוסחים, האחד ב-Java (עמודים 8-9) ואחד ב-C# (עמודים 10-11). עבוד על פי השפה שלמדת.

לפותרים ב-Java

יומן אלקטרוני לניהול פגישות מכיל את הימים של שנה אחת.

כל יום מיוצג על ידי:

\* תאריך, הכולל חודש ויום בחודש.

\* רשימת הפגישות באותו יום. בעבור כל פגישה נשמרים: שעת תחילת הפגישה,

תוכן הפגישה, ומשך הפגישה בדקות. משך הפגישה לא חייב להיות שעות שלמות.

בכל יום יכולות להיות פגישות בין 8:00 בבוקר ל- 20:00 בערב. כל פגישה מתחילה

בשעה עגולה. הפגישות מסתיימות לכל המאוחר ב- 20:00 בערב.

לפניך תיאור חלקי ב-Java של המחלקה **פגישה – Meeting**:

<b>Meeting</b>	
private int startHour;	// שעת התחלת הפגישה
private int minutes;	// משך זמן הפגישה בדקות
private String content;	// תוכן הפגישה
public Meeting (int startHour, int minutes, String content)	
public int getStartHour()	
public int getMinutes()	
public String getContent()	



לפניך חלק מממשק המחלקה **יום ביומן** – **DayInSchedule**:

<b>DayInSchedule</b>	
public int getDay()	
public int getMonth()	
public List<Integer> getFreeHours()	הפעולה מחזירה רשימה של כל השעות הפנויות ביום בין 08:00 ל- 19:00 (שעות שבהן לא מתקיימות פגישות). כל איבר ברשימה מכיל שעה עגולה אחת. הרשימה ממוינת בסדר עולה.
public boolean canStart (int startHour, int minutes)	הפעולה מקבלת שעת התחלה ומשך זמן הפגישה. הפעולה מחזירה true אם אפשר להתחיל את הפגישה בשעה המבוקשת, אחרת – הפעולה מחזירה false. אפשר להתחיל פגישה בשעה פנויה, בתנאי שיש אחריה די שעות פנויות כדי לסיים את הפגישה (בהתאם למשך הפגישה).

א. כתוב ב-Java את כותרת המחלקה **DayInSchedule**, ואת התכונות שלה.

ב. ממש ב-Java את הפעולה **canStart** המוצגת בממשק של המחלקה **DayInSchedule**.

אתה יכול להשתמש בפעולות האחרות של המחלקות **Meeting** ו-**DayInSchedule** בלי לממש אותן.

כמו כן אתה יכול להשתמש בפעולות של המחלקות **List<T>**, **Node<T>** בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ג. ממש ב-Java פעולה חיצונית שתקבל רשימה של ימים ביומן, ופגישה. הפעולה תדפיס את החודש והיום בחודש של כל אחד מהימים ברשימה, שבהם אפשר לשבץ את הפגישה.

כותרת הפעולה:

```
public static void printAvailableDay (List<DayInSchedule> lst , Meeting m)
```

אתה יכול להשתמש בפעולות הממשק של המחלקות **Meeting** ו-**DayInSchedule** בלי לממש אותן.

כמו כן אתה יכול להשתמש בפעולות של המחלקות **List<T>**, **Node<T>** בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

לפותרים ב- C#

יומן אלקטרוני לניהול פגישות מכיל את הימים של שנה אחת.

כל יום מיוצג על ידי:

\* תאריך, הכולל חודש ויום בחודש.

\* רשימת הפגישות באותו יום. בעבור כל פגישה נשמרים: שעת תחילת הפגישה,

תוכן הפגישה, ומשך הפגישה בדקות. משך הפגישה לא חייב להיות שעות שלמות.

בכל יום יכולות להיות פגישות בין 8:00 בבוקר ל- 20:00 בערב. כל פגישה מתחילה

בשעה עגולה. הפגישות מסתיימות לכל המאוחר ב- 20:00 בערב.

לפניך תיאור חלקי ב- C# של המחלקה **פגישה - Meeting**:

<b>Meeting</b>	
private int startHour;	// שעת התחלת הפגישה
private int minutes;	// משך זמן הפגישה בדקות
private string content;	// תוכן הפגישה
public Meeting (int startHour, int minutes, string content)	
public int GetStartHour()	
public int GetMinutes()	
public string GetContent()	

לפניך חלק מממשק המחלקה **יום ביומן** – **DayInSchedule**:

<b>DayInSchedule</b>	
public int GetDay()	
public int GetMonth()	
public List<int> GetFreeHours()	הפעולה מחזירה רשימה של כל השעות הפנויות ביום בין 08:00 ל- 19:00 (שעות שבהן לא מתקיימות פגישות). כל איבר ברשימה מכיל שעה עגולה אחת. הרשימה ממוינת בסדר עולה.
public bool CanStart (int startHour, int minutes)	הפעולה מקבלת שעת התחלה ומשך זמן הפגישה. הפעולה מחזירה true אם אפשר להתחיל את הפגישה בשעה המבוקשת, אחרת – הפעולה מחזירה false. אפשר להתחיל פגישה בשעה פנויה, בתנאי שיש אחריה די שעות פנויות כדי לסיים את הפגישה (בהתאם למשך הפגישה).

א. כתוב ב- C# את כותרת המחלקה **DayInSchedule** ואת התכונות שלה.

ב. ממש ב- C# את הפעולה CanStart המוצגת בממשק של המחלקה **DayInSchedule**.

אתה יכול להשתמש בפעולות האחרות של המחלקות **Meeting** ו- **DayInSchedule** בלי לממש אותן.

כמו כן אתה יכול להשתמש בפעולות של המחלקות **Node<T>** , **List<T>** בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ג. ממש ב- C# פעולה חיצונית שתקבל רשימה של ימים ביומן, ופגישה. הפעולה תדפיס את החודש והיום בחודש של כל אחד מהימים ברשימה, שבהם אפשר לשבץ את הפגישה.

כותרת הפעולה:

```
public static void PrintAvailableDay (List<DayInSchedule> lst , Meeting m)
```

אתה יכול להשתמש בפעולות הממשק של המחלקות **Meeting** ו- **DayInSchedule** בלי לממש אותן.

כמו כן אתה יכול להשתמש בפעולות של המחלקות **Node<T>** , **List<T>** בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

**פרק שני** (50 נקודות)

בפרק זה שאלות בארבעה מסלולים שונים:

מערכות מחשב ואסמבלר, עמ' 12-17

מבוא לחקר ביצועים, עמ' 18-27

מודלים חישוביים, עמ' 28-31

תכנות מונחה עצמים ב-Java, עמ' 32-39; תכנות מונחה עצמים ב-C#, עמ' 40-47

**ענה רק על שאלות במסלול שלמדת.**

**מערכות מחשב ואסמבלר**

אם למדת מסלול זה, ענה על שתיים מהשאלות 5-8 (לכל שאלה – 25 נקודות).

5. בשאלה זו שני סעיפים א-ב, שאין קשר ביניהם. ענה על שניהם.

א. במקטע הנתונים הוגדרו נתונים בצורה זו:

```
NUM1    DB    2
NUM2    DB    5
ARR     DW    255 DUP (?)
```

לפניך קטע תכנית באסמבלר.

```
MOV     SI, 0
MOV     DL, NUM1
XOR     AH, AH
MOV     AL, NUM2
SUB     AL, DL
MOV     CX, AX
MOV     AL, NUM1
NEXT:   MOV     DX, AX
        MOV     BX, DX
        DEC     DX
AGAIN:  ADD     BX, AX
        DEC     DX
        JNZ    AGAIN
        INC     AX
        MOV     ARR[SI], BX
        INC     SI
        LOOP   NEXT
FINISH: NOP
```

- i עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית.  
 בטבלת המעקב פרט בכל שלב את התוכן של:  
 . ARR[SI] , SI , DX , CX , BX , AX
- ii בעבור  $NUM1 \geq 2$  ו-  $NUM2 > NUM1$  , מה מבצע קטע התכנית?

ב. (אין קשר לסעיף א.)

באוגר AL הוצב המספר X שהוא מספר שלם בלי סימן.  
 נסמן ב-Z את החלק השלם של  $X/4$  .  
 יש לחשב את ערך הביטוי:  $Y = 3 * X - Z$   
 ולהציב אותו ב-AH.

עקוב בעזרת טבלת מעקב אחר ביצוע כל אחד מהקטעים i-iv שלפניך, וקבע אם הוא מבצע את הנדרש או אינו מבצע את הנדרש.

הנח ש-  $3 * X \leq 255$

i	MOV	DL , AL
	MOV	AH , AL
	ADD	AL , AL
	ADD	AH , AL
	MOV	CL , 3
	SHR	AL , CL
	SUB	AH , DL
ii	MOV	DL , AL
	MOV	AH , AL
	MOV	AL , 3
	MUL	AH
	MOV	CL , 2
	SHR	AL , CL
	SUB	AH , AL

(שים לב: המשך השאלה בעמוד הבא.)

<b>iii</b>	MOV	DL , AL
	MOV	AH , AL
	ADD	AH , AH
	ADD	AH , DL
	MOV	CL , 2
	SHR	DL , CL
	SUB	AH , DL
<b>iv</b>	MOV	DL , AL
	MOV	AH , AL
	ADD	AH , AH
	ADD	AH , AL
	MOV	CL , 4
	SHR	AL , CL
	SUB	AH , AL

6. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך שיטה לחישוב ריבוע של מספר שלם וגדול מ-0 :

$$1^2 = 1$$

$$2^2 = 1 + 3$$

$$3^2 = 1 + 3 + 5$$

$$4^2 = 1 + 3 + 5 + 7$$

$$5^2 = 1 + 3 + 5 + 7 + 9$$

וכן הלאה.

ב- AL מאוחסן מספר שלם, גדול מ-0 וקטן מ-16.

לפניך קטע תכנית באסמבלר שמחשב, לפי השיטה המתוארת, את הריבוע של המספר

המאוחסן ב- AL, ומאחסן את התוצאה ב- AX .

מקטע התכנית הושמטו שורות שלמות או חלקי שורות במקומות המסומנים

במספרים i-iv.

העתק למחברתך את מספרי השורות המסומנות, וכתוב ליד כל אחד מהמספרים את

שורת הפקודה כולל החלקים החסרים, כדי שקטע התכנית יבצע את הנדרש.

		MOV	BX , 0
<b>i</b>		-----	-----
		MOV	DX , 1
		MOV	CX , AX
		DEC	CX
<b>ii</b>		-----	-----
	A1:	ADD	BX , DX
<b>iii</b>		ADD	DX , -----
		LOOP	A1
	A2:	ADD	BX , DX
<b>iv</b>		MOV	AX , -----
	A3:	NOP	

**ב.** (אין קשר לסעיף א.)

באוגרים AX ו- BX מאוחסנים מספרים כלשהם.  
 יש להחליף את תוכן האוגרים AX ו- BX, כך שהמספר שאוחסן ב- AX יהיה  
 ב- BX, והמספר שאוחסן ב- BX יהיה ב- AX.  
 עקוב בעזרת טבלת מעקב אחר ביצוע כל אחד מהקטעים i - iv שלפניך, וקבע אם  
 הוא מבצע את הנדרש א או אינו מבצע את הנדרש.  
 הנח שאין גלישות בביצוע הוראות חיבור וחיסור.

**i**        MOV     CX , BX  
           SUB     CX , AX  
           MOV     AX , CX  
           SUB     BX , AX  
           ADD     AX , BX

**ii**        XOR     AX , BX  
           XOR     BX , AX  
           XOR     AX , BX

**iii**       XOR     CX , CX  
           XOR     BX , AX  
           XOR     AX , BX  
           XOR     BX , CX

**iv**        ADD     AX , BX  
           MOV     CX , AX  
           SUB     CX , BX  
           MOV     BX , CX  
           SUB     CX , AX



7. במקטע הנתונים הוגדרו הנתונים בצורה זו:

ARR1 DB 36 DUP (?)  
ARR2 DB 36 DUP (?)

המערך ARR1 מכיל מספרים שלמים וגדולים מ-0. כתוב קטע תכנית באסמבלר, שיאחסן במערך ARR2 את המספרים הנמצאים ב-ARR1 ללא חזרות. כלומר אם מספר כלשהו מופיע כמה פעמים ב-ARR1, הוא יופיע רק פעם אחת ב-ARR2.

לדוגמה בעבור ARR1 בגודל 10 המכיל מספרים:

ARR1	3	1	3	6	9	4	4	3	5	6
------	---	---	---	---	---	---	---	---	---	---

ייראה ARR2 כך:

ARR2	3	1	6	9	4	5	?	?	?	?
------	---	---	---	---	---	---	---	---	---	---

8. במקטע הנתונים הוגדרו מערך ARR בגודל 58 ומשתנים K ו-ANS.

ARR DB 58 DUP (?)  
K DB ?  
ANS DB ?

המערך מכיל מספרים שלמים וגדולים מ-0, ו-K הוא מספר שלם וגדול מ-0. כתוב קטע תכנית באסמבלר, שיבדוק אם במערך ARR המספר 1 מופיע בדיוק פעם אחת, המספר 2 מופיע בדיוק שתי פעמים, המספר 3 מופיע בדיוק שלוש פעמים, ..., והמספר K מופיע בדיוק K פעמים. אם כן – קטע התכנית יאחסן את הערך 1 במשתנה ANS, אחרת – קטע התכנית יאחסן 0 במשתנה ANS.

לדוגמה, בעבור המערך בגודל 12 שלפניך ו- $K = 4$ :

2	3	1	2	4	3	3	4	5	4	4	8
---	---	---	---	---	---	---	---	---	---	---	---

יאחסן 1 ב-ANS.

**מבוא לחקר ביצועים**

אם למדת מסלול זה, ענה על שתיים מהשאלות 9-12 (לכל שאלה – 25 נקודות).

9. נתונה בעיית תכנון לינארי:

$$\text{Max } \{z = 5x_1 - x_2\}$$

בכפוף לאילוצים האלה:

$$2x_1 + 3x_2 \leq 18$$

$$x_1 - x_2 \leq 2$$

$$x_1 \leq 3$$

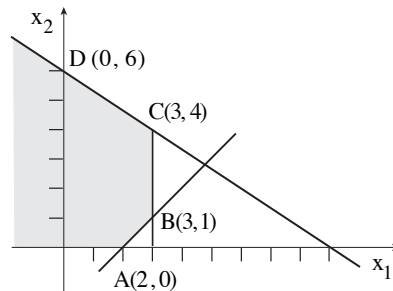
$$x_2 \geq 0$$

$x_1$  אינו מוגבל בסימן.

נתון גם כי הפתרון  $(3, 1)$  הוא פתרון אופטימלי של הבעיה.

כל אחד מהסעיפים א-ה שלפניך מתייחס לבעיית התכנון הלינארי הנתונה. הסעיפים אינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



העתק למחברתך את הסרטוט, ובצע את הצעדים האלה:

צעד 1: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה בעבור  $z = 5$ .

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים  $x_1$  ו- $x_2$ , וסמן אותן על הסרטוט.

צעד 2: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה בעבור  $z = 15$ .

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים  $x_1$  ו- $x_2$ , וסמן אותן על הסרטוט.

צעד 3: סמן בסרטוט שבמחברתך, באמצעות חץ, את כיוון העלייה של פונקציית המטרה.

האם הצעדים 1-3 מראים שהפתרון  $(3, 1)$  הוא הפתרון האופטימלי היחיד? נמק את תשובתך.

ב. לבעיה הנתונה בתחילת השאלה מוסיפים את האילוץ:

$$x_1 - x_2 \geq 2$$

סרטט במחברתך מערכת צירים חדשה, וסמן עליה רק את תחום הפתרונות האפשריים שיתקבל לאחר הוספת אילוץ זה.

ג. לבעיה הנתונה בתחילת השאלה מוסיפים אילוץ חדש כלשהו, כך שלאחר הוספתו

תחום הפתרונות האפשריים הוא הקטע שבין שתי הנקודות (3, 4) ו-(3, 1).

האם הפתרון האופטימלי הנתון (3, 1) ישתנה? נמק את תשובתך.

ד. לבעיה הנתונה בתחילת השאלה מוסיפים אילוץ חדש:

$$3x_1 + 2x_2 \leq 14$$

תלמיד טען שהפתרון האופטימלי הנתון (3, 1) לא ישתנה.

האם התלמיד צדק? נמק את תשובתך בלי לסרטט מחדש את תחום הפתרונות האפשריים לאחר הוספת אילוץ זה.

ה. משנים רק את פונקציית המטרה של הבעיה שבתחילת השאלה ל-  $z = 3x_1$ .

לפניך ההיגדים iv-i שרק אחד מהם נכון. העתק למחברתך את ההיגד הנכון, ונמק את בחירתך.

i הפתרון האופטימלי היחיד יהיה (3, 1).

ii הפתרון האופטימלי היחיד יהיה (3, 4).

iii הפתרון האופטימלי החדש יהיה:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} 3 \\ 1 \end{pmatrix} + (1 - \lambda) \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

כאשר  $0 \leq \lambda \leq 1$

iv הפתרון האופטימלי יהיה לא חסום.

10. בשאלה זו שישנה סעיפים, א-ו, שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפניך בעיית תובלה ובה 2 מקורות ו-4 יעדים. העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

מקורות	יעדים				היצע
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
ביקוש	10	40	30	70	

על פי הטבלה הנתונה לא ניתן להפעיל את שיטת הפינה הצפון-מערבית כדי למצוא פתרון אפשרי לבעיית התובלה. הסבר מדוע.

ב. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה:  $x_{11} = 10$ ,  $x_{12} = 2$ .

מקורות	יעדים			היצע
	1	2	3	
1	4 10	5 2	10	12
2	6	3	6	12
3	6	2	6	6
ביקוש	10	10	10	

העתק את הטבלה למחברתך, והשלם אותה לפי שיטת הפינה הצפון-מערבית.

(שים לב: המשך השאלה בעמוד הבא.)

- ג. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה. הערך שנקבע ל- $u_1$  הוא 0. העתק את הטבלה למחברתך, חשב את הערך של  $v_1, v_2, v_3, u_2, u_3$ , ורשום את הערכים שחישבת במקומות המתאימים בטבלה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	4 10	5 2	10	12	0
2	6	3 2	6 10	12	
3	6	2 6	6	6	
ביקוש	10	10	10		
$v_j$					

- ד. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכים של  $v_1, v_2, v_3, u_1, u_2, u_3$ .

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	4 20	5 4	10	24	0
2	2	6 16	3 4	20	1
3	6	2	6 8	8	4
ביקוש	20	20	12		
$v_j$	4	5	2		

האם הפתרון הוא אופטימלי? נמק את תשובתך.

ה. בטבלה שלפניך נתון פתרון לבעיית תובלה, ונתונים ערכים של  $u_1, v_1, v_2, v_3, u_1, u_2, u_3$ , שמתאימים לפתרון זה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	10 20	25	30	20	10
2	10 30	22	14 20	50	10
3	16	20 40	20 20	60	16
ביקוש	50	40	40		
$v_j$	0	4	4		

לפניך ההיגדים iv-i שרק אחד מהם נכון. העתק למחברתך את ההיגד הנכון, ונמק את בחירתך.

- i הפתרון הנתון איננו פתרון אפשרי.
- ii הפתרון הנתון הוא פתרון בסיסי אפשרי אך לא אופטימלי.
- iii הפתרון הנתון הוא פתרון אופטימלי יחיד.
- iv הפתרון הנתון הוא פתרון אופטימלי אך איננו פתרון אופטימלי יחיד.

1. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה שחושב על פי שיטת הפינה הצפון מערבית, ונתונים ערכים של  $u_3, u_2, u_1, v_3, v_2, v_1$  שחושבו על פי פתרון זה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	12 20	15	17	20	2
2	10	18 10	14	10	0
3	20	10 5	18 10	15	-8
ביקוש	20	15	10		
$v_j$	10	18	26		

מה צריך להוסיף לטבלה כדי לקבל פתרון בסיסי אפשרי המתאים לכל הנתונים שבטבלה?

העתק למחברתך את התשובה הנכונה מבין האפשרויות i-iv, ונמק את תשובתך.

.  $X_{12} = 0$     **i**

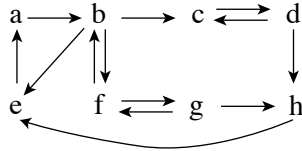
.  $X_{21} = 0$     **ii**

.  $X_{13} = 0$     **iii**

$X_{31} = 0$     **iv**

11. בשאלה זו שלושה סעיפים, א-ג, שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפיך גרף מכוון  $G$ .



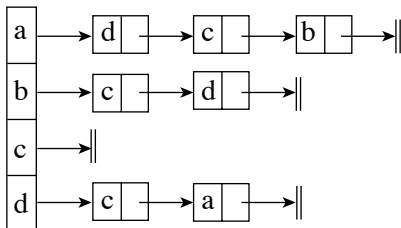
i מצא את רכיב/רכיבי הקשירות החזקה (רק"ח/רק"חים) שבגרף הנתון.

בעבור כל רק"ח שמצאת, רשום את קבוצת הקדקודים שלו.

ii האם הגרף הנתון הוא גרף קשיר מכוון בחוזקה (קשיר היטב)? נמק את תשובתך.

ב.  $G = (V, E)$  הוא גרף מכוון, כאשר  $V$  היא קבוצת הקדקודים בגרף ו- $E$  היא קבוצת

הקשתות בגרף.  $G$  מיוצג על ידי רשימת הסמיכויות שלפניך:



i הפעל אלגוריתם לחיפוש לעומק (בשיטת DFS) על הגרף הנתון החל מקדקוד a.

סרטט במחברתך את העץ הפורש (DFS) / היער הפורש (DFS) שהתקבל.

ii הפעל אלגוריתם לחיפוש לעומק (בשיטת DFS) על הגרף הנתון החל מקדקוד c.

סרטט במחברתך את העץ הפורש (DFS) / היער הפורש (DFS) שהתקבל.

iii הפעל אלגוריתם לחיפוש לרוחב (בשיטת BFS) על הגרף הנתון החל מקדקוד d.

סרטט במחברתך את העץ הפורש (BFS) / היער הפורש (BFS) שהתקבל.



ג.  $G = (V, E)$  הוא גרף מכוון, כאשר  $V$  היא קבוצת הקדקודים בגרף ו- $E$  היא קבוצת הקשתות בגרף.  $G$  מיוצג על ידי רשימת סמיכויות. לפניך תיאור של אלגוריתם אשר בודק אם בגרף  $G$  יש מעגל, ומדפיס הודעה מתאימה. באלגוריתם חסרים שני ביטויים המסומנים (1) ו- (2).

### תיאור האלגוריתם

צעד 1: הפעל את האלגוריתם למציאת רכיבי קשירות חזקה (SCC) על הגרף  $G$ .  
(האלגוריתם מניב את הרק"חים הקיימים בגרף.)

צעד 2: אם (1) , אז  
הפלט הוא: בגרף  $G$  אין מעגל.

צעד 3: אחרת  
הפלט הוא: בגרף  $G$  יש מעגל.

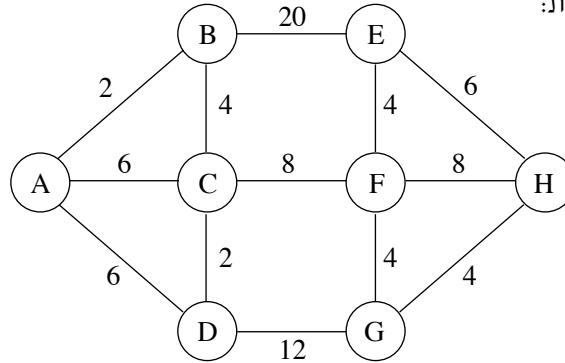
לפניך שלוש אפשרויות להשלמת ביטוי (1) ושלוש אפשרויות להשלמת ביטוי (2).  
האפשרויות:

- (1) הוא אחד מאלה:
- אין רק"ח המכיל
  - רק"ח אחד לפחות מכיל
  - כל אחד מהרק"חים מכיל
- (2) הוא אחד מאלה:
- קדקוד אחד בלבד
  - שני קדקודים לפחות
  - את כל קדקודי הגרף

בחר באחת מהאפשרויות להשלמת ביטוי (1) ובאחת מהאפשרויות להשלמת ביטוי (2), כך שהאלגוריתם יבצע את הנדרש. כתוב במחברתך את שתי בחירותיך.

12. בשאלה זו שני סעיפים, א-ב, שאינם תלויים זה בזה. ענה על שניהם.

א. לפניך רשת:



רשום במחברתך את חמשת המסלולים הקצרים ביותר מקדקוד A לקדקוד H ברשת הנתונה. תאר כל מסלול כזה בנפרד.

ב. הגרף  $G$  הוא קשיר ולא מכוון, ומוגדר על ידי  $G = (V, E)$ , כאשר  $V$  היא קבוצת הקדקודים בגרף ו- $E$  היא קבוצת הקשתות בגרף. פונקציית המשקל  $W: E \rightarrow \mathbb{R}^+$  קובעת משקל (מספר) לכל קשת בגרף  $G$ .

יהיו  $X, Y$  ו- $Z$  קדקודים בגרף  $G$ . נניח כי כל קשת בגרף צבועה בצבע כחול או בצבע אדום. לפניך תיאור של אלגוריתם, אשר בודק אם מבין המסלולים המורכבים מקשתות כחולות בלבד, כל המסלולים הקצרים ביותר מ- $X$  ל- $Y$  עוברים דרך  $Z$ . אם כן – האלגוריתם מחזיר את הערך "אמת" (TRUE). אחרת – הוא מחזיר את הערך "שקר" (FALSE). באלגוריתם חסרים 5 ביטויים המסומנים (1)-(5).

### תיאור האלגוריתם

צעד 1:

נסיר מהגרף את כל הקשתות (1) ונקבל גרף חדש  $G_1 = (V, E_1)$ .

צעד 2:

נחשב, בעזרת האלגוריתם של דיקסטרה, את אורך המסלול הקצר ביותר מ- $X$  ל- $Y$  בגרף (2). נסמן את אורך המסלול הזה ב- $M_1$ .

צעד 3:

נסיר מהגרף  $G_1$  את הקדקוד (3) ואת כל הקשתות המחוברות אליו, ונקבל גרף חדש  $G_2 = (V_2, E_2)$ .

צעד 4:

נחשב, בעזרת האלגוריתם של דיקסטר, את אורך המסלול הקצר ביותר מ- $X$  ל- $Y$  בגרף (4). נסמן את אורך המסלול הזה ב- $M_2$ .

צעד 5:

אם (5) אזי החזר את הערך "אמת".  
אחרת – החזר את הערך "שקר".

העתק למחברתך את הטבלה שלפניך, ורשום בה את הביטויים החסרים.

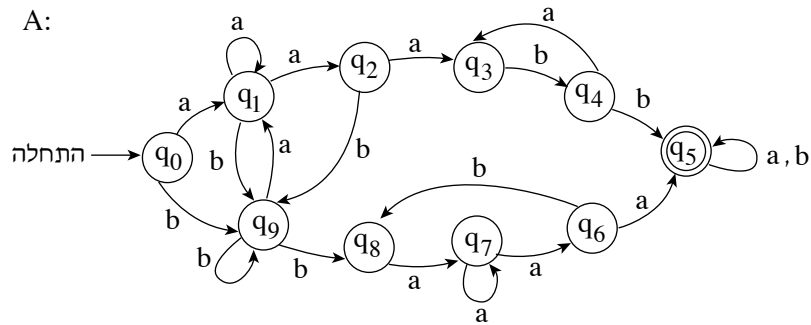
	ביטוי (1)
	ביטוי (2)
	ביטוי (3)
	ביטוי (4)
	ביטוי (5)

**מודלים חישוביים**

אם למדת מסלול זה, ענה על שתיים מהשאלות 13-16 (לכל שאלה – 25 נקודות).

**13.** בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

**א.** לפניך אוטומט סופי לא דטרמיניסטי A:



**i** קבע לכל אחת מהמילים (1)-(3) שלפניך אם היא מתקבלת

על ידי האוטומט A.

אם המילה מתקבלת על ידי האוטומט, רשום מסלול מקבל בעבור מילה זו.

babbaaa (1)

aababaaa (2)

aaaabbba (3)

**ii** כתוב את כל המילים הקצרות ביותר, המתקבלות על ידי האוטומט A.

**ב.** (אין קשר לסעיף א.)

נתונות השפות  $L_1, L_2$  מעל הא"ב  $\{a, b\}$ .

$$L_1 = \{a^n b^k \mid n \neq k, n \geq 0, k \geq 0\}$$

$$L_2 = \{b^i a^j \mid i \geq 0, j \geq 1\}$$

האם  $L_1 \cap L_2$  רגולרית? נמק את תשובתך.

14. א. מילה באורך 3 תיקרא **פלינדרום באורך 3**, אם התו הראשון במילה זהה לתו האחרון במילה.

לדוגמה: המילה aba היא **פלינדרום באורך 3**.

לפניך השפה L מעל הא"ב {a, b}.

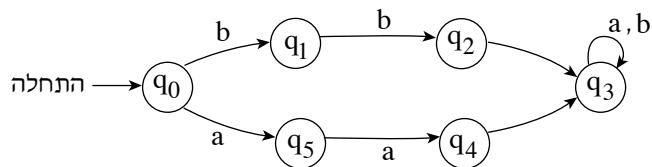
$L = \{w \mid w \text{ היא שרשור של } 0 \text{ או יותר פלינדרומים באורך } 3\}$ .

לדוגמה:

המילה babbbbaba שייכת ל-L.

המילה bababb לא שייכת ל-L.

לפניך ציור חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L.



הציור מכיל את כל המצבים של האוטומט.

העתק למחברתך את הציור, והשלם אותו כך שהאוטומט יקבל את השפה L.

עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים,

ולסמן את המצב המקבל / המצבים המקבלים.

שים לב: אין להוסיף מצבים לאוטומט או להוריד ממנו מצבים.

ב. מהי השפה  $L \cap \{(aab)^n \mid n \geq 0\}$  ?

15. נגדיר  $\Sigma^*$  – אוסף כל המילים מעל א"ב נתון, כולל המילה הריקה.

בעבור שפה כלשהי  $L$  נגדיר:

$$\text{Init}(L) = \{u \mid uv \in L, u, v \in \Sigma^*\}$$

$$\text{Fin}(L) = \{v \mid uv \in L, u, v \in \Sigma^*\}$$

$$\text{Min}(L) = \left\{ w \mid \begin{array}{l} w \in L \\ \text{ובעבור כל } w_1, w_2 \text{ המקיימות } w = w_1 w_2 \\ \text{ו- } w_2 \text{ אינה מילה ריקה, מתקיים } w_1 \notin L \end{array} \right\}$$

לפניך חמש שפות  $L_1$ - $L_5$  מעל הא"ב  $\{0, 1\}$

$$L_1 = \{0^n 1^n 0^k 1^k \mid n \geq 1, k \geq 0\}$$

$$L_2 = \{0^n 1^k \mid n \geq 0, k \geq 0\}$$

$$L_3 = \{0^i 1^i \mid i \geq 0\}$$

$$L_4 = \{0^i 1^k \mid 0 \leq i \leq k\}$$

$$L_5 = \{0^i 1^k \mid 0 \leq k \leq i\}$$

א. מהי השפה  $L_1 \cap L_2$  ?

ב. מהי השפה  $\text{Init}(L_3)$  ?

ג. מהי השפה  $\text{Fin}(L_3)$  ?

ד. האם  $0011 \in \text{Min}(L_4)$  ? נמק את תשובתך.

ה. האם  $0011 \in \text{Min}(L_5)$  ? נמק את תשובתך.

ו. האם  $L_4 \cap L_5$  היא רגולרית? נמק את תשובתך.

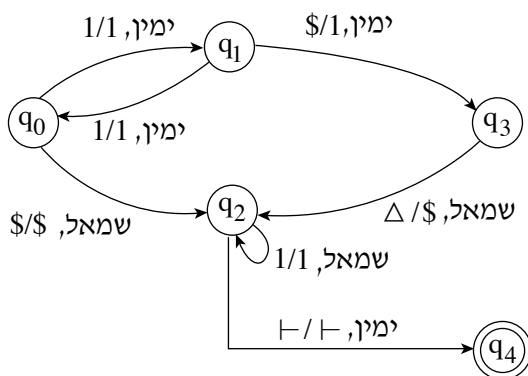
16. לפניך מכונת טיורינג המחשבת פונקציה  $f(x)$ .

המכונה מקבלת כקלט מספר  $x$  גדול מ-0, הרשום על הסרט כמספר אונרי על ידי  $x$  תווים של 1 ואחריהם הסימן \$.

לדוגמה: בעבור  $x = 4$  יהיה סרט הזיכרון לפני תחילת החישוב:

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

המכונה רושמת את תוצאת החישוב של  $f(x)$  על הסרט כמספר אונרי, מיד אחרי הסימן ⊢.



- א. מה יכיל הסרט לאחר חישוב  $f(5)$  ?
- ב. מה יכיל הסרט לאחר חישוב  $f(6)$  ?
- ג. מהי הפונקציה  $f(x)$  שהמכונה מחשבת?

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב-Java, ענה על שתיים מהשאלות 17-20.  
(לכל שאלה – 25 נקודות)

**17. לפניך המחלקה מצולע – Polygon.**

חלק מתכונות המצולע והפעולות שלו מתועדים בגוף המחלקה.

```
public class Polygon
{
    public static int maxSides = 100;           // מספר צלעות מקסימלי
    private int[] values = new int[maxSides];  // מערך אורכי הצלעות
    private int numSides = 0;                  // מספר צלעות במצולע
    /** פעולה בונה המחזירה מצולע ריק */
    public Polygon()
    {}
    /** פעולה המחזירה את מספר הצלעות במצולע */
    public int getNumSides()
    {
        return this.numSides;
    }
    /** פעולה המחזירה את מערך אורכי הצלעות של המצולע */
    public int[] getValues()
    {
        return this.values;
    }
    /** פעולה המאתחלת את maxSides */
    public static void setMaxSides (int maxSides)
    {
        Polygon.maxSides = maxSides;
    }
    /** פעולה המקבלת מספר שלם וגדול מ-0 ומוסיפה למצולע צלע באורך זה */
    /** הנחה: יש מקום להוסיף צלע למערך אורכי הצלעות */
    public void addSide(int x)
    {
        this.values[this.numSides] = x;
        this.numSides ++;
    }
    /** פעולה הקולטת מהמשתמש את מספר הצלעות שיש להוסיף למצולע
        ואת האורך של כל אחת מהן, ומוסיפה את הצלעות למצולע.
        /** הנחה: יש מקום להוסיף את הצלעות למערך אורכי הצלעות.
    public void readToPolygon()
    {.....}
}
/המשך בעמוד 33/
```



- א. מהי המשמעות של הגדרת התכונה `maxSides` כ- `static` ? הסבר.
- ב. מה תהיה המשמעות של הגדרת התכונה `maxSides` אם היא תוגדר כ- `static final` ?
- ג. עליך לממש ב- Java, פעולה `expand` המוסיפה למצולע צלעות של מצולע אחר, בכל אחד מהמקרים ii-i שלפניך:

- i הפעולה `expand` תוגדר כפעולה פנימית במחלקה `Polygon`.
- ii הפעולה `expand` תוגדר כפעולה סטטית חיצונית (לא במחלקה `Polygon`).  
הערה: הנח כי במערך אורכי הצלעות של המצולע יש די מקום להוספת כל הצלעות של המצולע האחר.  
ד. לפניך שלד של פעולה ראשית:

```
public static void main(String[] args)
{
    Polygon p1 = new Polygon();
    p1.readToPolygon();
    Polygon p2 = new Polygon();
    p2.readToPolygon();
    (*) _____ ;
}
```

יש לרשום בשורה המסומנת ב-(\*), הוראה שתביא לכך שהצלעות של מצולע `p2` יתווספו לצלעות של מצולע `p1`.

רשום במחברתך את ההוראה בעבור כל אחד מהמקרים ii-i :

- i תוך כדי שימוש בפעולה הפנימית `expand` כפי שהוגדרה במחלקה `Polygon` לפי סעיף ג i.
- ii תוך כדי שימוש בפעולה הסטטית `expand` כפי שהוגדרה בסעיף ג ii אם היא מוגדרת בתוך המחלקה שבה נמצאת הפעולה `main`.
- iii תוך כדי שימוש בפעולה הסטטית `expand` כפי שהוגדרה בסעיף ג ii אם היא מוגדרת במחלקה אחרת הנקראת `PolygonOperations`.

**18.** בחנות למכירת אופניים אפשר לשלם באמצעי התשלום האלה: מזומן, המחאה (צ'ק), כרטיס אשראי. את הסכום המשולם בהמחאות אפשר לחלק לכמה תשלומים, התשלומים לא חייבים להיות שווים. תשלום בעבור כל קנייה יכול להתבצע באמצעי תשלום אחד או יותר, כך ששילוב אמצעי התשלום ייתן את הסכום הנדרש. הנח כי הסכום לתשלום וכל אחד מהתשלומים הם מספרים שלמים.

לדוגמה:

- תפניך כמה שילובים אפשריים של אמצעי תשלום בעבור קנייה בסכום של 1000 ש"ח.
- לשלם את כל הסכום באמצעי תשלום אחד: מזומן או המחאה או כרטיס אשראי.
  - לשלם 200 ש"ח במזומן ו- 800 ש"ח בכרטיס אשראי.
  - לשלם 100 ש"ח במזומן, 500 ש"ח באמצעות שתי המחאות: הראשונה על סך 200 ש"ח והשנייה על סך 300 ש"ח, ו- 400 ש"ח בכרטיס אשראי.

החנות זקוקה לתכנה כדי לנהל את תשלומי הקונים.

בעבור כל קנייה, המידע המתקבל בחנות הוא:

תאריך הקנייה, הסכום לתשלום בעבור הקנייה ופירוט השילוב של אמצעי התשלום.

בעבור תשלום במזומן – הסכום לתשלום.

בעבור תשלום בהמחאה – הסכום לתשלום, מספר ההמחאה, שם הבנק

והתאריך הרשום על ההמחאה.

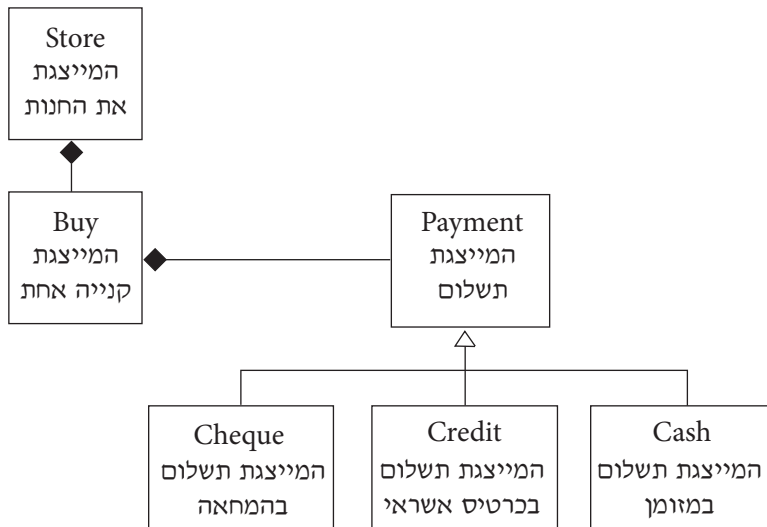
בעבור תשלום בכרטיס אשראי – הסכום לתשלום, מספר כרטיס האשראי, תוקף הכרטיס

והתאריך שבו יחויב בעל כרטיס האשראי.

הפעולות הנדרשות מהתוכנה הן:

- קליטה ושמירה של המידע המתקבל בעבור כל קנייה.
- בדיקה שסכום כל התשלומים בעבור קנייה אחת שווה לסכום הקנייה.
- הדפסת קבלה בעבור הקנייה.
- בעבור תאריך מסוים, חישוב של סכום מחירי כל האופניים שנמכרו באותו תאריך.

לפניך תרשים של היררכיית המחלקות הנדרשות בעבור כתיבת התוכנה.



בתרשים מופיעים הסימנים האלה:



בעבור כל מחלקה בתרשים, הגדר את התכונות ואת הפעולות שלה באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism).

יש לכלול רק את הפעולות הנחוצות כדי לענות על הדרישות מהתוכנה שתוארו בשאלה.

אין צורך לרשום פעולות בונות, פעולות קובעות (פעולות set) ופעולות מאחזרות (פעולות get).

הנח שקיימת מחלקה Date, המייצגת תאריך.

בעבור כל תכונה, רשום את ההגדרה שלה ב-Java, ורשום את התיעוד שלה.

בעבור כל פעולה, רשום את הכותרת שלה ב-Java, ורשום תיעוד הכולל מה היא מקבלת ומה היא מחזירה. אין צורך לממש את הפעולה.

19. לפניך הממשקים: `IPrintBinary`, `IPrintHtml`  
והמחלקות: `Page`, `Page1`, `Page2`, `CreateReport`.

```
public interface IPrintHtml
{
    public void createHtml();
}

public interface IPrintBinary
{
    public void createBinary();
}

public class Page implements IPrintBinary
{
    protected int num;
    public Page()
    {}
    public Page(int n)
    {
        this.num = n;
    }
    public void print()
    {
        System.out.println(this.num);
    }
    public void createBinary()
    {}
}

public class Page1 extends Page
{
    protected int num1;
    public Page1 (int n, int n1)
    {
        super(n);
        this.num1 = n1;
    }
}
```

```
public void print()
{
    super.print();
    System.out.println(this.num1);
}

public void createBinary()
{
    System.out.println("Binary Data: num = " + this.num + ", num1 = " + this.num1);
}
}

public class Page2 extends Page implements IPrintHtml
{
    protected int num2;

    public Page2(int n, int n2)
    {
        super(n);
        this.num2 = n2;
    }

    public void print()
    {
        super.print();
        System.out.println(this.num2);
    }

    public void createHtml()
    {
        System.out.println("Html Data: num = " + this.num + ", num2 = " + this.num2);
    }
}

public class CreateReport
{
    public static void createBinaryDoc(IPrintBinary doc)
    {
        System.out.println("*** Binary Doc ***");
        doc.createBinary();
    }

    public static void createHtmlDoc(IPrintHtml doc)
    {
        System.out.println("*** Html Doc ***");
        doc.createHtml();
    }
}
```

א. לפניך שלושה קטעים i-iii הכתובים ב-Java.

לכל אחד מהקטעים, קבע אם הוא חוקי או אינו חוקי. נמק את קביעותיך.

- i Page1 doc1 = new Page1 (10 , 20);  
CreateReport.createHtmlDoc (doc1);
- ii Page doc2 = new Page2 (30 , 40);  
CreateReport.createHtmlDoc (doc2);
- iii IPrintBinary doc3 = new Page1 (50 , 60);  
CreateReport.createBinaryDoc (doc3);

ב. הפעולה writeHtmlDoc שלפניך הוספה למחלקה CreateReport .

לאחר ההוספה הקומפילר הודיע על שגיאה. הסבר מהי השגיאה, ותקן אותה.

```
public static void writeHtmlDoc(Page doc)
{
    if (doc instanceof Page2)
        doc.createHtml();
}
```

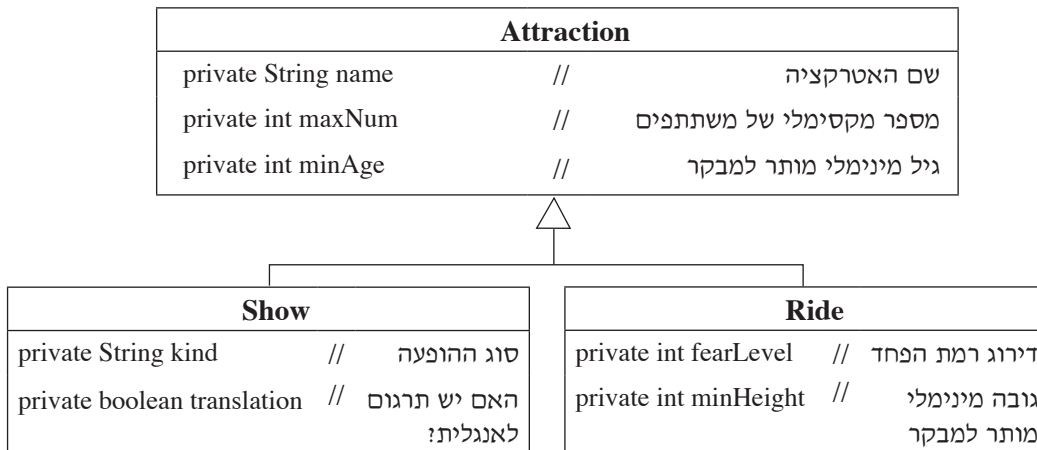
ג. כתוב את הפלט של קטע התכנית שלפניך:

```
Page2 doc1 = new Page2 (11 , 22);
CreateReport.createHtmlDoc (doc1);
doc1.print();
```

```
Page doc2 = new Page1 (33 , 44);
CreateReport.createBinaryDoc (doc2);
doc2.print();
```

```
Page doc3 = new Page2 (55 , 66);
CreateReport.createBinaryDoc (doc3);
doc3.print();
```

20. פארק שעשועים מציע למבקריו אטרקציות מסוגים שונים. חלק מהאטרקציות הן מתקנים (כמו קרוסלה או רכבת הרים) וחלק מהאטרקציות הן הופעות. נתונה דיאגרמת UML של חלק מהמחלקות במערכת הממוחשבת של הפארק. המחלקה **Attraction** בעבור אטרקציה, המחלקה **Ride** בעבור מתקן, והמחלקה **Show** בעבור הופעה. המחלקות **Ride** ו-**Show** יורשות מהמחלקה **Attraction**.



הנח שבכל אחת משלוש המחלקות מוגדרות:

- פעולות set ו- get בעבור כל התכונות.
- פעולה בונה המקבלת פרמטרים לכל תכונה, ומאתחלת את תכונות העצם בהתאם.
- א. יש להוסיף למערכת הממוחשבת מחלקה **Park** ובה התכונות:
  - שם הפארק, מערך האטרקציות בפארק.
  - ממש ב-Java את הסעיפים i-iii במחלקה **Park**:
    - i כותרת המחלקה והתכונות שלה.
    - ii פעולה בונה, המקבלת את שם הפארק ואת מספר האטרקציות שהפארק יכול להכיל. הפעולה תאתחל את שם הפארק ותאתחל את מערך האטרקציות בפארק להיות בגודל שהתקבל.
    - iii פעולה המדפיסה את שמות ההופעות שיש להן תרגום לאנגלית.
- ב. במחלקה **Park** נתונה הפעולה addAttraction המקבלת אטרקציה a ומוסיפה אותה למערך האטרקציות בפארק. כותרת הפעולה היא:

public void addAttraction (Attraction a)

כתוב ב-Java, פעולה ראשית שתיצור עצם p מטיפוס **Park** ששמו lunafun ויש בו מקום ל-30 אטרקציות. כמו כן, הפעולה הראשית תיצור 2 עצמים: אחד מטיפוס **Ride** ואחד מטיפוס **Show**, ותוסיף אותם למערך האטרקציות של p. בחר ערכים כרצונך לאתחול 2 העצמים.

/המשך בעמוד 40/

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- **C#**, ענה על שתיים מהשאלות 21-24.  
(לכל שאלה – 25 נקודות)

**21. לפניך המחלקה מצולע – Polygon.**

חלק מתכונות המצולע והפעולות שלו מתועדים בגוף המחלקה.

```
public class Polygon
{
    public static int maxSides = 100;           // מספר צלעות מקסימלי
    private int[] values = new int[maxSides];   // מערך אורכי הצלעות
    private int numSides = 0;                   // מספר צלעות במצולע
    /** פעולה בונה המחזירה מצולע ריק */
    public Polygon()
    {
    }
    /** פעולה המחזירה את מספר הצלעות במצולע */
    public int GetNumSides()
    {
        return this.numSides;
    }
    /** פעולה המחזירה את מערך אורכי הצלעות של המצולע */
    public int[] GetValues()
    {
        return this.values;
    }
    /** פעולה המאתחלת את maxSides */
    public static void SetMaxSides (int maxSides)
    {
        Polygon.maxSides = maxSides;
    }
    /** פעולה המקבלת מספר שלם וגדול מ-0 ומוסיפה למצולע צלע באורך זה */
    /** הנחה: יש מקום להוסיף צלע למערך אורכי הצלעות */
    public void AddSide(int x)
    {
        this.values[this.numSides] = x;
        this.numSides ++;
    }
    /** פעולה הקולטת מהמשתמש את מספר הצלעות שיש להוסיף למצולע */
    /** ואת האורך של כל אחת מהן ומוסיפה את הצלעות למצולע. */
    /** הנחה: יש מקום להוסיף את הצלעות למערך אורכי הצלעות. */
    public void ReadToPolygon()
    {.....}
}
/המשך בעמוד 41/
```



- א.** מהי המשמעות של הגדרת התכונה `maxSides` כ- `static`? הסבר.
- ב.** מה תהיה המשמעות של הגדרת התכונה `maxSides` אם היא תוגדר כ- `static const`?
- ג.** עליך לממש ב- `C#`, פעולה `Expand` המוסיפה למצולע צלעות של מצולע אחר, בכל אחד מהמקרים i-ii שלפניך:
- i** הפעולה `Expand` תוגדר כפעולה פנימית במחלקה `Polygon`.
- ii** הפעולה `Expand` תוגדר כפעולה סטטית חיצונית (לא במחלקה `Polygon`).
- הערה: הנח כי במערך אורכי הצלעות של המצולע יש די מקום להוספת כל הצלעות של המצולע האחר.
- ד.** לפניך שלד של פעולה ראשית:

```
public static void Main(string[] args)
{
    Polygon p1 = new Polygon();
    p1.ReadToPolygon();
    Polygon p2 = new Polygon();
    p2.ReadToPolygon();
    (*) _____ ;
}
```

יש לרשום בשורה המסומנת ב-(\*), הוראה שתביא לכך שהצלעות של מצולע `p2` יתווספו לצלעות של מצולע `p1`.

- רשום במחברתך את ההוראה בעבור כל אחד מהמקרים i-iii:
- i** תוך כדי שימוש בפעולה הפנימית `Expand` כפי שהוגדרה במחלקה `Polygon` לפי סעיף **ג i**.
- ii** תוך כדי שימוש בפעולה הסטטית `Expand` כפי שהוגדרה בסעיף **ג ii** אם היא מוגדרת בתוך המחלקה שבה נמצאת הפעולה `Main`.
- iii** תוך כדי שימוש בפעולה הסטטית `Expand` כפי שהוגדרה בסעיף **ג ii** אם היא מוגדרת במחלקה אחרת הנקראת `PolygonOperations`.

22. בחנות למכירת אופניים אפשר לשלם באמצעי התשלום האלה: מזומן, המחאה (צ'ק), כרטיס אשראי. את הסכום המשולם בהמחאות אפשר לחלק לכמה תשלומים, התשלומים לא חייבים להיות שווים. תשלום בעבור כל קנייה יכול להתבצע באמצעי תשלום אחד או יותר, כך ששילוב אמצעי התשלום ייתן את הסכום הנדרש. הנח כי הסכום לתשלום וכל אחד מהתשלומים הם מספרים שלמים.

לדוגמה:

- לפניך כמה שילובים אפשריים של אמצעי תשלום בעבור קנייה בסכום של 1000 ש"ח.
- לשלם את כל הסכום באמצעי תשלום אחד: מזומן או המחאה או כרטיס אשראי.
  - לשלם 200 ש"ח במזומן ו- 800 ש"ח בכרטיס אשראי.
  - לשלם 100 ש"ח במזומן, 500 ש"ח באמצעות שתי המחאות: הראשונה על סך 200 ש"ח והשנייה על סך 300 ש"ח, ו- 400 ש"ח בכרטיס אשראי.

החנות זקוקה לתכנה כדי לנהל את תשלומי הקונים.

בעבור כל קנייה, המידע המתקבל בחנות הוא:

תאריך הקנייה, הסכום לתשלום בעבור הקנייה ופירוט השילוב של אמצעי התשלום.

בעבור תשלום במזומן – הסכום לתשלום.

בעבור תשלום בהמחאה – הסכום לתשלום, מספר המחאה, שם הבנק

והתאריך הרשום על המחאה.

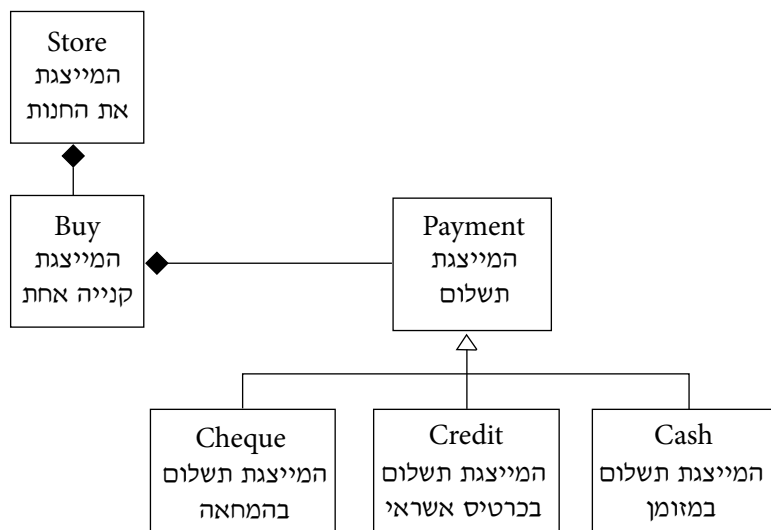
בעבור תשלום בכרטיס אשראי – הסכום לתשלום, מספר כרטיס האשראי, תוקף הכרטיס

והתאריך שבו יחויב בעל כרטיס האשראי.

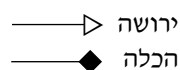
הפעולות הנדרשות מהתוכנה הן:

- קליטה ושמירה של המידע המתקבל בעבור כל קנייה.
- בדיקה שסכום כל התשלומים בעבור קנייה אחת שווה לסכום הקנייה.
- הדפסת קבלה בעבור הקנייה.
- בעבור תאריך מסוים, חישוב של סכום מחירי כל האופניים שנמכרו באותו תאריך.

לפניך תרשים של היררכיית המחלקות הנדרשות בעבור כתיבת התוכנה.



בתרשים מופיעים הסימנים האלה:



בעבור כל מחלקה בתרשים, הגדר את התכונות ואת הפעולות שלה באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism).

יש לכלול רק את הפעולות הנחוצות כדי לענות על הדרישות מהתוכנה שתוארו בשאלה.

אין צורך לרשום פעולות בונות, פעולות קובעות (פעולות Set) ופעולות מאחזרות (פעולות Get).

הנח שקיימת מחלקה Date, המייצגת תאריך.

בעבור כל תכונה, רשום את ההגדרה שלה ב-C#, ורשום את התיעוד שלה.

בעבור כל פעולה, רשום את הכותרת שלה ב-C#, ורשום תיעוד הכולל מה היא מקבלת ומה היא מחזירה. אין צורך לממש את הפעולה.

23. לפניך הממשקים: `IPrintBinary`, `IPrintHtml`

והמחלקות: `Page`, `Page1`, `Page2`, `CreateReport`.

```
public interface IPrintHtml
{
    public void CreateHtml();
}

public interface IPrintBinary
{
    public void CreateBinary();
}

public class Page: IPrintBinary
{
    protected int num;
    public Page()
    {}
    public Page(int n)
    {
        this.num = n;
    }
    public virtual void Print()
    {
        Console.WriteLine (this.num);
    }
    public virtual void CreateBinary()
    {}
}

public class Page1 : Page
{
    protected int num1;
    public Page1 (int n, int n1) : base(n)
    {
        this.num1 = n1;
    }
}
```

```
public override void Print()
{
    base.Print();
    Console.WriteLine (this.num1);
}

public override void CreateBinary()
{
    Console.WriteLine ("Binary Data: num = " + this.num + ", num1 = " + this.num1);
}
}

public class Page2 : Page, IPrintHtml
{
    protected int num2;

    public Page2(int n, int n2): base(n)
    {
        this.num2 = n2;
    }

    public override void Print()
    {
        base.Print();
        Console.WriteLine (this.num2);
    }

    public void CreateHtml()
    {
        Console.WriteLine ("Html Data: num = " + this.num + ", num2 = " + this.num2);
    }
}

public class CreateReport
{
    public static void CreateBinaryDoc(IPrintBinary doc)
    {
        Console.WriteLine("*** Binary Doc ***");
        doc.CreateBinary();
    }

    public static void CreateHtmlDoc(IPrintHtml doc)
    {
        Console.WriteLine("*** Html Doc ***");
        doc.CreateHtml();
    }
}
```

(שים לב: סעיפי השאלה בעמוד הבא.)

א. לפניך שלושה קטעים i-iii הכתובים ב- C#.

לכל אחד מהקטעים, קבע אם הוא חוקי או אינו חוקי. נמק את קביעותיך.

- i Page1 doc1 = new Page1 (10 , 20);  
CreateReport.CreateHtmlDoc (doc1);
- ii Page doc2 = new Page2 (30 , 40);  
CreateReport.CreateHtmlDoc (doc2);
- iii IPrintBinary doc3 = new Page1 (50 , 60);  
CreateReport.CreateBinaryDoc (doc3);

ב. הפעולה WriteHtmlDoc שלפניך הוספה למחלקה CreateReport .

לאחר ההוספה הקומפילר הודיע על שגיאה. הסבר מהי השגיאה, ותקן אותה.

```
public static void WriteHtmlDoc(Page doc)
{
    if (doc is Page2)
        doc.CreateHtml();
}
```

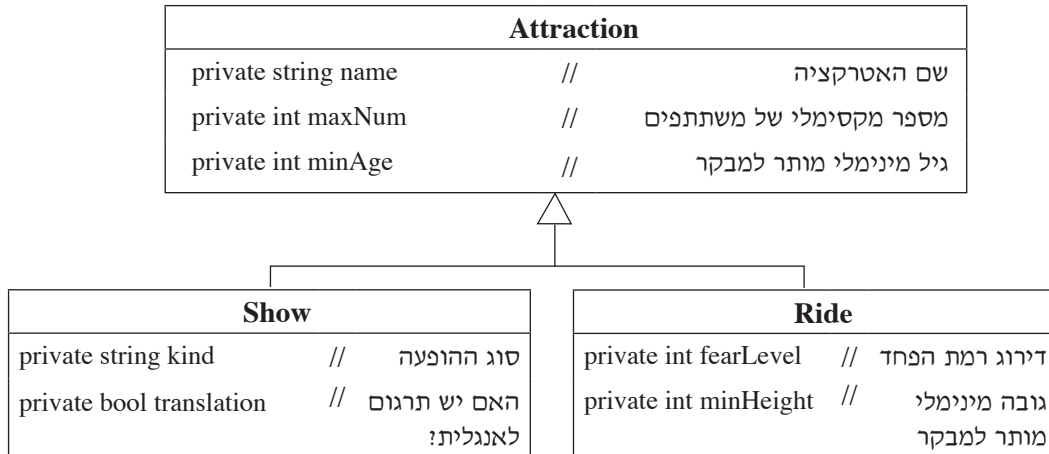
ג. כתוב את הפלט של קטע התכנית שלפניך:

```
Page2 doc1 = new Page2 (11 , 22);
CreateReport.CreateHtmlDoc (doc1);
doc1.Print();
```

```
Page doc2 = new Page1 (33 , 44);
CreateReport.CreateBinaryDoc (doc2);
doc2.Print();
```

```
Page doc3 = new Page2 (55 , 66);
CreateReport.CreateBinaryDoc (doc3);
doc3.Print();
```

24. פארק שעשועים מציע למבקרו אטרקציות מסוגים שונים. חלק מהאטרקציות הן מתקנים (כמו קרוסלה או רכבת הרים) וחלק מהאטרקציות הן הופעות. נתונה דיאגרמת UML של חלק מהמחלקות במערכת הממוחשבת של הפארק. המחלקה **Attraction** בעבור אטרקציה, המחלקה **Ride** בעבור מתקן, והמחלקה **Show** בעבור הופעה. המחלקות **Ride** ו- **Show** יורשות מהמחלקה **Attraction**.



הנח שבכל אחת משלוש המחלקות מוגדרות:

- פעולות Set ו- Get בעבור כל התכונות.
  - פעולה בונה המקבלת פרמטרים לכל תכונה, ומאתחלת את תכונות העצם בהתאם.
- א. יש להוסיף למערכת הממוחשבת מחלקה **Park** ובה התכונות:
- שם הפארק, מערך האטרקציות בפארק.
  - ממש ב- C# את הסעיפים i-iii במחלקה **Park**:
    - i כותרת המחלקה והתכונות שלה.
    - ii פעולה בונה, המקבלת את שם הפארק ואת מספר האטרקציות שהפארק יכול להכיל. הפעולה תאתחל את שם הפארק ותאתחל את מערך האטרקציות בפארק להיות בגודל שהתקבל.
    - iii פעולה המדפיסה את שמות ההופעות שיש להן תרגום לאנגלית.
- ב. במחלקה **Park** נתונה הפעולה **AddAttraction** המקבלת אטרקציה **a** ומוסיפה אותה למערך האטרקציות בפארק. כותרת הפעולה היא:

`public void AddAttraction (Attraction a)`

כתוב ב- C#, פעולה ראשית שתיצור עצם **p** מטיפוס **Park** ששמו **lunafun** ויש בו מקום ל- 30 אטרקציות. כמו כן, הפעולה הראשית תיצור 2 עצמים: אחד מטיפוס **Ride** ואחד מטיפוס **Show**, ותוסיף אותם למערך האטרקציות של **p**.  
בחר ערכים כרצונך לאתחול 2 העצמים.

## בהצלחה!