מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: דוגמה
מספר השאלון: 899271

מדעי המחשב

הוראות

א. משך הבחינה: שעתיים וחצי.

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

פרק ראשון	–	(25x2)	–	50 נקודות
פרק שני	–	(25x2)	–	50 נקודות
		סך הכול	–	100 נקודות

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

ד. הוראות מיוחדות:

(1) **רשמו על הכריכה החיצונית** של המחברת את **שם המסלול שלמדתם**.
המסלול הוא אחד משלושת המסלולים האלה: אלגוריתמים, מודלים חישוביים, תכנות מונחה עצמים.

(2) את כל התוכניות שיש לכתוב בשפת מחשב בפרקים הראשון והשני כתבו בשפה אחת בלבד – Java או C# .

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

השאלות בשאלון זה מנוסחות בלשון רבים, אך אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

השאלות

בשאלון זה שני פרקים.

יש לענות על שאלות משני 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, הניחו שבתוכנית כתובה ההוראה:

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

פרק ראשון (50 נקודות)

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, ערימה בינרית וחוליה, בלי לממש אותן.
אם משתמשים בפעולות נוספות, יש לממש אותן.

ענו על שתיים מן השאלות 1–3 (לכל שאלה – 25 נקודות).

1. בשאלה זו נוספה למחלקה **Queue** הפעולה size/Size שלפניכם. אפשר להשתמש בפעולה בלי לממש אותה.

כותרת הפעולה	תיאור הפעולה
Java – public int size() **C#** – public int Size()	הפעולה מחזירה את מספר האיברים בתור.

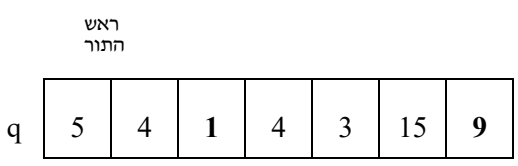
ממשו את הפעולה החיצונית שלפניכם:

Java – public static boolean twoSum (Queue<Integer> q, int x)

C# – public static bool TwoSum (Queue<int> q, int x)

הפעולה מחזירה true אם בתור q שהתקבל יש שני מספרים שסכומם שווה לערך הפרמטר x . אחרת הפעולה מחזירה false .

דוגמה: עבור התור q שלפניכם ו־ x=10 הפעולה תחזיר true , כי יש בתור שני מספרים (9 , 1) שסכומם שווה ל־ 10 .



הערות:

- הניחו שבתור q יש שני איברים לפחות.
- אין צורך לשמור על התור q .
- **אין** להשתמש בשאלה זו במערך וברשימה מקושרת.

.2

נתונה המחלקה **NumCount** – מספר ערכים, ולה שתי תכונות:

- num – ערך מספרי, מטיפוס שלם.
- count – מספר המופעים של הערך (num), מטיפוס שלם. המספר גדול או שווה ל- 0.

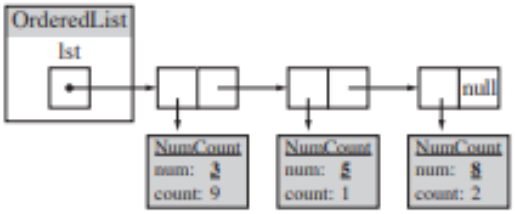
הניחו שקיימות פעולות get/Get ו- set/Set לכל אחת מן התכונות במחלקה, ופעולה בונה המקבלת ערכים עבור תכונות המחלקה.

נתונה המחלקה **OrderedList** – שרשרת ממוינת, ולה תכונה אחת:

- lst – מצביע על ראש של שרשרת חוליות מטיפוס **NumCount**.

שרשרת החוליות ממוינת לפי סדר עולה של ערך התכונה – num. ערך התכונה num שונה בכל חוליה.

<u>דוגמה</u>: השרשרת שלפניכם מקיימות את תנאי המחלקה (השרשרת ממוינת בסדר עולה לפי ערך התכונה num, וערך התכונה num שונה בכל חוליה).



א. (1) ממשו במחלקה OrderedList את הפעולה הפנימית שלפניכם:

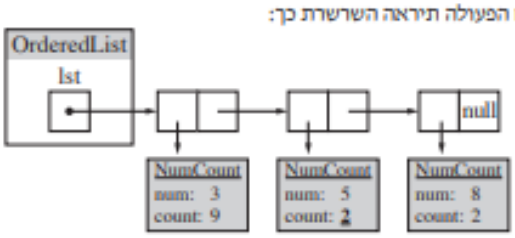
Java – public void insertNum (int x)

C# – public void InsertNum (int x)

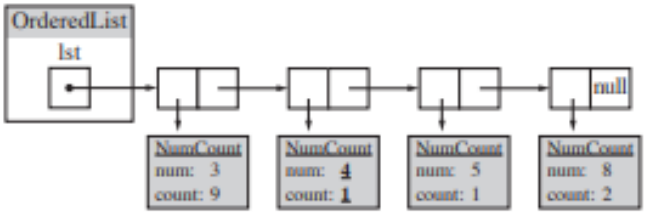
הפעולה מוסיפה את הערך של x לשרשרת באופן שלפניכם:

- אם קיימת בשרשרת חוליה שהתכונה num שלה שווה ל- x, הפעולה תגדיל ב- 1 את התכונה count (כמות המופעים) באותה החוליה.
- אם השרשרת ריקה או שלא קיימת בשרשרת חוליה שהתכונה num שלה שווה ל- x, הפעולה תכניס חוליה חדשה, שבה התכונה num תהיה שווה ל- x והתכונה count תהיה שווה ל- 1, במיקום השומר את הסדר העולה של השרשרת.

<u>דוגמה</u>: עבור השרשרת המוצגת לעיל ו- x = 5, בתום הפעולה תיראה השרשרת כך:



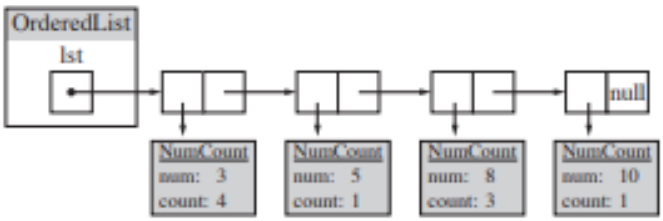
<u>דוגמה נוספת</u>: עבור אותה השרשרת המוצגת לעיל (בדוגמה הראשונה) ו- x = 4, בתום הפעולה תיראה השרשרת כך:



(2) מהי סיבוכיות זמן הריצה של הפעולה שכתבתם בסעיף א(1)? נמקו את תשובתכם.

ב. "ערך המופע ה־ n " הוא הערך שמופיע בְּמקום ה־ n לפי הסדר מתחילת השרשרת (בשקלול כמות המופעים – count של כל ערך).

לדוגמה: עבור השרשרת שלפניכם ו־ n = 7 הפעולה תחזיר את הערך 8 .



הסבר: סדר הערכים של השרשרת ברצף, בהתאם לכמות המופעים שלהם, הוא: 10 , 8 , **8** , 8 , 5 , 3 , 3 , 3 , 3 .

n = 7 ובמקום השביעי ברצף מופיע הערך 8 . לכן הפעולה תחזיר את הערך 8 .

ממשו במחלקה OrderedList את הפעולה הפנימית שלפניכם:

Java – public int valueN (int n)

C# – public int ValueN (int n)

הפעולה מקבלת את המספר n , ומחזירה את "ערך המופע ה־ n ".

הניחו ש"ערך המופע ה־ n " קיים בשרשרת.

.3

"מספר ראשוני" הוא מספר המתחלק רק בעצמו וב־ 1 (גם המספרים 1 ו־ 2 הם ראשוניים).

לפניכם הפעולה החיצונית isPrime/IsPrime . אפשר להשתמש בפעולה בלי לממש אותה.

כותרת הפעולה	תיאור הפעולה
Java – public static boolean isPrime (int num) **C#** – public static bool IsPrime (int num)	הפעולה מחזירה true אם הערך num שהתקבל הוא מספר ראשוני, אחרת היא מחזירה false .

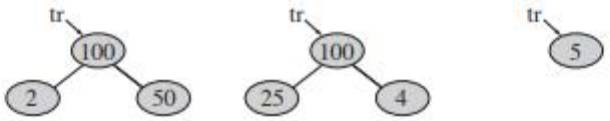
א. ממשו את הפעולה החיצונית שלפניכם:

Java – public static boolean addNodes (BinNode<Integer> tr)

C# – public static bool AddNodes (BinNode<int> tr)

הפעולה מקבלת צומת ללא בנים (עלֶה) שערכו גדול מ־ 0 . אם ערך הצומת הוא מספר ראשוני, הפעולה מחזירה false .
אחרת, הפעולה מוסיפה לצומת שני בנים שערך המכפלה שלהם שווה לערך הצומת, והערך של כל אחד מהם גדול מ־ 1 .
לאחר ההוספה הפעולה מחזירה true .

<u>דוגמאות</u>: בתום הפעולה הצמתים יכולים להיראות כך (עבור הערכים 5 , 100 , 100):



ב. נתונה הפעולה what/What שלפניכם, המקבלת צומת ללא בנים שערכו גדול מ־ 0 .

בשפת **Java**	בשפת **C#**

public static void what (BinNode<Integer> tr)
{
if (addNodes (tr))
{
what (tr.getLeft());
what (tr.getRight());
}
}
	public static void What (BinNode<int> tr)
	{
	if (AddNodes (tr))
	{
	What (tr.GetLeft());
	What (tr.GetRight());
	}
	}

(1) סרטטו את העץ כפי שהוא ייראה בתום הפעולה what/What עבור צומת ללא בנים – tr שערכו 150 .
יש להראות מעקב.

(2) הסבירו מה מבצעת הפעולה what/What .

פרק שני (50 נקודות)

בפרק זה שאלות בשלושה מסלולים:

אלגוריתמים, עמודים 6–9.

מודלים חישוביים, עמודים 10–12

תכנות מונחה עצמים בשפת Java , עמודים 13 –18; תכנות מונחה עצמים בשפת C# , עמודים 19–24.

יש לענות על שתי שאלות במסלול שלמדתם.

אלגוריתמים

אם למדתם מסלול זה, ענו על שתיים מן השאלות 4–6 (לכל שאלה - 25 נקודות).

4. בשאלה זו שני סעיפים, א–ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. לפניכם חמש טענות. בחרו בארבע מהן.

עבור כל אחת מן הטענות שבחרתם, כתבו את מספר הטענה במחברתכם, וציינו אם היא נכונה או לא נכונה.

אם ציינתם שהטענה נכונה – נמקו מדוע, ואם ציינתם שהטענה אינה נכונה, הביאו דוגמה נגדית או נמקו מדוע.

1. לכל עץ פורש DFS של אותו גרף G לא מכוון יש תמיד אותו מספר עלים.

2. גובה עץ פורש BFS של גרף G לא מכוון, תמיד קטן או שווה לגובה עץ פורש DFS של אותו הגרף.

3. גרף G לא מכוון בעל n צמתים ו־ 2–n קשתות הוא תמיד לא קשיר.

4. בכל גרף G מכוון שבו n צמתים ו־ m קשתות, ו־ , קיים מעגל.

5. גרף G לא מכוון שבו הדרגות של כל הקודקודים גדולות מ־ 0 הוא תמיד קשיר.

ב. "עץ פורש מקסימלי" הוא עץ פורש של גרף G לא מכוון שבו סכום ערכי הקשתות הוא מקסימלי.

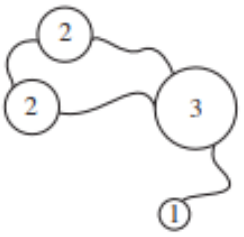
כתבו אלגוריתם המוצא בגרף G "עץ פורש מקסימלי".

.5

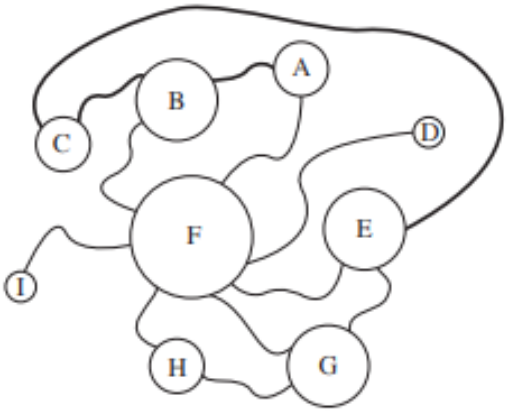
ב"רשת רחובות" המעבר מרחוב לרחוב הוא תמיד דרך כיכר.

גודלו (במטרים) של רדיוס של כיכר הוא כמספר הרחובות המחוברים אל הכיכר. לדוגמה, הרדיוס של כיכר המחֻברת 3 רחובות הוא 3 מטרים, והרדיוס של כיכר המחֻברת 6 רחובות הוא 6 מטרים.

<u>דוגמה</u>: לפניכם סרטוט של רשת רחובות. בתוך כל כיכר מצוין הרדיוס של אותה הכיכר.



לפניכם רשת הרחובות NET בעיר מסוימת:



הולך רגל נדרש ללכת מכיכר אחת לאחרת <u>במסלול הקצר ביותר</u>. המסלול הקצר ביותר הוא המסלול שבו סכום הרדיוסים של כל הכיכרות שבהן הוא עובר הוא הקטן ביותר. סכום הרדיוסים <u>אינו כולל</u> את הרדיוס של הכיכר שבה הוא מתחיל את מסלולו אך הוא <u>כולל</u> את הרדיוס של הכיכר שבה הוא מסיים את מסלולו.

<u>דוגמה</u>: בסרטוט שלעיל של רשת הרחובות NET מודגש המסלול הקצר ביותר מן הכיכר A לכיכר E.

גודל הרדיוסים במסלול זה הוא 3+2+3 וסכומם הוא 8. סכום זה הוא הקטן ביותר מבין כל האפשרויות.

א. ברשת הרחובות NET, מהו המסלול הקצר ביותר מכיכר H לכיכר C? כתבו את שמות הכיכרות במסלול זה, לפי הסדר, משמאל לימין (אין צורך לבצע מעקב).

ב. **(1)** כתבו אלגוריתם המוצא עבור רשת רחובות כלשהי את המסלול הקצר ביותר מן הכיכר K_1 לכיכר K_2.

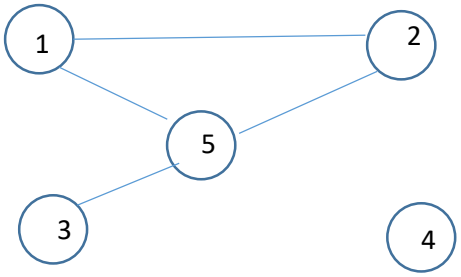
<u>הערה</u>: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

(2) סרטטו את הגרף המייצג את רשת הרחובות NET הנתונה לעיל, באופן שיתאים לאלגוריתם שכתבתם.

6. בשאלה זו שני סעיפים, א–ב, שאין ביניהם קשר. יש לענות על שני הסעיפים.

א. להלן גרף $G = (V, E)$ שאינו מכוון:

<u>גרף G</u>



לפניכם 4 סעיפים <u>שאין</u> קשר ביניהם. בכל סעיף <u>אין</u> לשנות את מספר הצמתים של הגרף.

(1) הגרף G אינו קשיר. הוסיפו מינימום קשתות כך שיהיה קשיר.

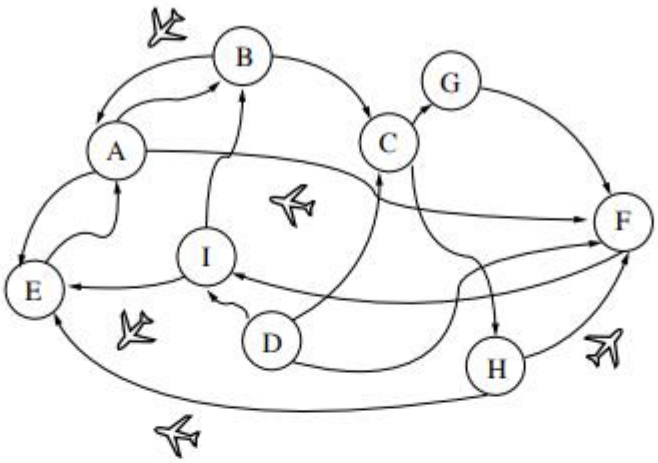
(2) שנו את הגרף G המקורי כך שיהיה דו צדדי (מבלי להוריד או להוסיף צמתים). הציגו את שתי הקבוצות שנוצרו.

(3) הגרף G המקורי אינו מלא. הוסיפו מינימום קשתות כך שיהיה מלא.

(4) הגרף G המקורי אינו עץ. כתבו <u>שתי</u> סיבות מדוע הוא איננו עץ.

(שימו לב: המשך השאלה בעמוד הבא.)

ב. חברת תעופה מפעילה בשגרה קווי תעופה בין המדינות A ,B ,C ,D ,E ,F ,G ,H ,I על פי הסרטוט שלהלן:



(1) סרטטו את רשימת הסמיכות המייצגת את קווי התעופה בין המדינות.

בעקבות שינויים בחברת התעופה הוחלט לבטל כמה מקווי התעופה במדינות שלהלן:

מדינה G – ביטול הטיסות הנכנסות והטיסות היוצאות

מדינה H – ביטול הטיסות היוצאות למדינה F

מדינה D – ביטול הטיסות היוצאות למדינה C

מדינה A – ביטול הטיסות היוצאות למדינה E

(2) סרטטו את רשימת הסמיכות המתקבלת לאחר הביטולים המצוינים לעיל.

(3) **I.** אביטל צריכה להגיע ממדינה מסוימת למדינה אחרת.

לשם כך היא רוצה לבדוק אם היא יכולה להגיע לאותה מדינה, ואם היא אכן יכולה, היא רוצה למצוא מסלול טיסה כלשהו המגיע לאותה מדינה.

מהו האלגוריתם שעליה להשתמש בו למטרות אלה? הסבירו.

<u>הערה</u>: אם יש כמה אלגוריתמים מתאימים, בחרו באחד מהם.

II. הפעילו את האלגוריתם שכתבתם עליו בסעיף I(3) כדי למצוא את מסלול הטיסה ממדינה A למדינה E. כִּתבו את מסלול הטיסה שמצאתם.

יש לבצע מעקב מפורט בכל שלב בהתאם לאלגוריתם שהפעלתם.

מודלים חישוביים

אם למדתם מסלול זה, ענו על שתיים מן השאלות 7–9 (לכל שאלה - 25 נקודות).

.7

נתונות השפות $L_1 - L_4$ מעל הא"ב $\{a,b\}$:

$L_1 =$ שפת כל המילים שבהן מספר המופעים של האות a שווה למספר המופעים של האות b.

$L_2 =$ שפת כל המילים שבהן מספר המופעים של האות a גדול ממספר המופעים של האות b.

$L_3 =$ שפת כל המילים שיש בהן יותר משלוש אותיות.

$L_4 =$ שפת כל המילים שמתחילות באות a ומסתיימות באות b או שמתחילות באות b ומסתיימות באות a.

ענו על כל הסעיפים א–ז שלפניכם:

א. בנו אוטומט סופי דטרמיניסטי המקבל את השפה L_4.

ב. הוכיחו שהשפה $L_3 \cap L_4$ רגולרית.

ג. כתבו את השפה המתקבלת מן הפעולה $L_1 \cup L_2$. האם השפה המתקבלת רגולרית? נמקו את תשובתכם.

ד. כתבו את השפה המתקבלת מן הפעולות $L_1 \cup L_2 \cup \overline{L_2}$.

ה. האם השפה $L_1 \cap L_2$ רגולרית? נמקו את תשובתכם.

ו. בנו אוטומט סופי דטרמיניסטי שאינו מלא, המקבל את השפה $L_2 \cap \overline{L_3}$.

ז. כתבו את השפה המתקבלת מן הפעולה $L_4 \cap R(L_4)$.

.8

הפעולה **Generate** מבוצעת על שני מספרים שאורכם זהה, והם מורכבים אך ורק מן הספרות 0 ו־ 1 . הפעולה יוצרת מילה חדשה באותו האורך, באופן שלהלן:

אם בשני המספרים הספרה במקום ה־ n היא 0 , האות במקום ה־ n היא F , אחרת האות במקום ה־ n היא T .

דוגמה: פעולת **Generate** על המספרים $x = 11001$ ו־ $y = 10010$, יוצרת את המילה TTFTT כמתואר לפניכם:

	0	1	2	3	4
x	1	1	0	0	1
y	1	0	0	1	0
	Generate				
המילה המתקבלת	T	T	F	T	T

כתבו מכונת טיורינג המקבלת בתחילת הסרט קלט של שני מספרים שאורכם זהה, המורכבים אך ורק מן הספרות 0 ו־ 1 .

המכונה מחזירה את המילה המתקבלת מפעולת **Generate** על שניהם.

הנחיות:

- שני המספרים בקלט מופרדים ב־ # .
- אין צורך לשמור על הקלט (אפשר לשנות את הקלט לסימנים שונים).
- המילה החדשה תופיע במקום כלשהו בסרט בין שני סימני $.
- אין צורך לבדוק את תקינות הקלט.

דוגמה:

הסרט לפני ההרצה:

	0	1	2	3	4	5		0	1	2	3	4	5			
⊢	1	0	1	0	1	0	#	1	0	1	1	0	0	△	△	...

הסרט לאחר ההרצה:

...	$	T	F	T	T	T	F	$	...

.9

נתונה השפה L_1 מעל הא"ב $\{a, b, c\}$:

$$L_1 = \{c^{1+k+n} b^k a^{2n} \mid n, k \geq 1\}$$

א. **(1)** מהי המילה הקצרה ביותר בשפה L_1 ?

(2) בְּנו אוטומט מחסנית המקבל את השפה L_1 .

ב. נגדיר את השפה L_2 מעל הא"ב $\{a, b, c, d\}$:

$$L_2 = L_1 \cdot d \cdot R(L_1)$$

בעבור כל אחת מן המילים שלפניכם, ציינו אם המילה שייכת לשפה L_2 , ונמקו.

cccbaaddaabccc

cccbaadcccbaa

cccbaadaaaabcccc

<u>הערה</u>: אין צורך לבנות אוטומט בעבור השפה L_2 .

תכנות מונחה עצמים בשפת Java

אם למדתם מסלול זה ואתם כותבים בשפת Java , ענו על שתיים מן השאלות 10–12 (לכל שאלה - 25 נקודות).

10. במסעדת "הרמה" אפשר להזמין מקום עבור סועד אחד או יותר. בסוף הארוחה, כל אחד מן הסועדים משלם בנפרד על מה שאכל. כל סועד יכול לבחור אם לשלם במזומן או באמצעות אפליקציה או בכרטיס אשראי. למשלמים בכרטיס אשראי יש אפשרות לפצל את החשבון לכמה תשלומים שווים.

כדי לנהל מערכת תשלומים, מפתחים עבור המסעדה פרויקט ובו הממשק (Interface) **IPayment** , והמחלקות **Restaurant, Reservation, Credit, App, Cash** , כמפורט להלן:

- **IPayment** – תשלום

  בממשק קיימת הפעולה: double getPrice()

- **Cash** – תשלום במזומן

  תכונות המחלקה:

  - sumCash – הסכום לתשלום במזומן, מטיפוס ממשי.
  - name – שם הלקוח, מטיפוס מחרוזת.

- **App** – תשלום באמצעות אפליקציה

  תכונות המחלקה:

  - sumApp – הסכום לתשלום באפליקציה, מטיפוס ממשי.
  - phoneNumber – מספר טלפון, מטיפוס מחרוזת.

- **Credit** – תשלום בכרטיס אשראי

  תכונות המחלקה:

  - num – מספר התשלומים, מטיפוס שלם.
  - part – הסכום בכל תשלום, מטיפוס ממשי.
  - creditNumber – מספר כרטיס האשראי, מטיפוס מחרוזת.

- **Reservation** – הזמנה

  תכונות המחלקה:

  - date – תאריך ההזמנה, מטיפוס מחרוזת.
  - total – החשבון סך הכול של כל הסועדים יחד באותה הזמנה, מטיפוס ממשי.
  - payments – מערך השומר את התשלום ששילם כל אחד מן הסועדים באותה הזמנה, מטיפוס **IPayment** (המערך ללא ערכי null).

- **Restaurant** – המסעדה

  תכונות המחלקה:

  - array – מערך בגודל 10,000 , מטיפוס **Reservation** .
  - current – כמות ההזמנות השמורות, מטיפוס שלם.

  הניחו שההזמנות נשמרות ברצף במערך ואין ביניהן ערכי null .

הערה: הניחו שבכל מחלקה הוגדרו פעולות get ו־ set ופעולות בונות.

(שימו לב: המשך השאלה בעמוד הבא.)

א. **(1)** סרטטו תרשים היירכייה המתאר את הקשרים של המחלקות והממשק של הפרויקט.

יש לסמן מימוש ממשק באמצעות החץ ⇢ והכלה באמצעות הסימן ⬟——— .

(2) המסעדה עושה הנחות לסועדים בהתאם לצורת התשלום: סועד שמשלם במזומן והסכום לתשלום הוא מעל 200 שקלים זכאי ל־ 10% הנחה, סועד שמשלם באמצעות האפליקציה זכאי תמיד ל־ 5% הנחה, וסועד שמשלם בכרטיס אשראי משלם מחיר מלא.

פעולת הממשק getPrice מחזירה את סך התשלום שכל סועד משלם לאחר שקלול ההנחה (כאשר אין הנחה, הפעולה מחזירה מחיר מלא). ממשו את פעולת הממשק getPrice בכל אחת מן המחלקות הנדרשות.

ב. כתבו במחלקה **Reservation** פעולה ששמה cashTotal . הפעולה מחזירה את סכום הכסף <u>המזומן</u> שהתקבל בהזמנה (לאחר שקלול ההנחה).

ג. לפניכם פעולה במחלקה **Reservation** :

```
public void printDetails () {
    for (int i = 0; i < payments.length; i ++) {
        System.out.println (payments[i].getDetails());
    }
}
```

הפעולה מדפיסה עבור כל סועד פרטים מסוימים על פי הקריטריונים שלהלן:

- אם שילם הסועד במזומן, הפעולה מדפיסה את השם שלו.
- אם שילם הסועד באמצעות האפליקציה, הפעולה מדפיסה את מספר הטלפון שלו.
- אם שילם הסועד בכרטיס אשראי, הפעולה מדפיסה את מספר כרטיס האשראי שלו.

הוסיפו את הפעולות הנחוצות לפרויקט כדי שהפעולה תבצע את הנדרש. ציינו עבור כל פעולה שהוספתם לאיזו מחלקה או לאיזה ממשק היא שייכת.

<u>הערה</u>: <u>אין</u> לשנות את הפעולה printDetails .

11. לפניכם המחלקות First ו־ Second :

```
public class First {

 private static int count = 0;
 protected int x;
 protected int y;
 public First (int num) {
        this.x = num;
        this.y = num;
        count ++;
        System.out.println ("First 1");
 }
 public First (int num1, int num2) {
        this.x = num1;
        this.y = num2;
        count ++;
        System.out.println ("First 2");
 }
 public static int getCount() {
        return count;
 }
 public int getX() {
             return x;
 }
 public int getY() {
        return y; }
 public int sum() {
        return this.x + this.y; }
 public void add(First other) {
        this.x += other.x;
        this.y += other.y;
        System.out.println("x = "+ this.x +      "y = "+
this.y);}}
```

```
public class Second extends First{

 private int z;
 public Second (int num) {
        super (num);
        this.z = num;
        System.out.println ("Second");
 }
 public int sum() {
        return super.sum() + this.z;
 }
 public void add (First other) {
        this.x += other.getX();
        this.y += other.getY();
        if (other instanceof Second)
              this.z += ((Second)other).z;
        System.out.println("x = "+ this.x +
                                  " y = "+   this.y+ " z = " + this.z);
 }}
```

נתונה המחלקה Tester :

```
public class Tester
{
    public static void main(String[] args)
    {
        First f1 = new First (40);
        First f2 = new First (40, 50);
        First f3 = new Second (100);
        Second s1 = new Second (100);
        Second s2 = new Second (100);
        // ***
    }
}
```

א. ציירו את העצמים שנוצרו בפעולה main , וכתבו את הפלט של הפעולה.

ב. הציבו כל אחת מן הפקודות 1–10 שלהלן בפעולה main במקום המצוין לעיל ב- *** .

כתבו במחברת את מספר הפקודה וציינו אם הקוד תקין או לא תקין.

אם הקוד תקין – כתבו את הפלט, ואם הוא אינו תקין, הסבירו מדוע.

<u>הערה</u>: אין קשר בין הפקודות. כלומר, <u>יש להתייחס לכל פקודה כאילו היא היחידה בפעולה main</u> .

1. System.out.println ("Total = " + First.getCount());
2. System.out.println ("Total = " + Second.getCount());
3. System.out.println ("sum = " + s1.sum());
4. System.out.println ("sum = " + f3.sum());
5. s1=new First (100);
6. f1.add (s2);
7. s1.add (s2);
8. s2.add (f3);
9. ((First)s1).add (f1);
10. s1=new Second (100, 100);

.12

לפניך פרויקט שהוגדרו בו המחלקות האלה: A , B , C , D , E . בפרויקט מוּמשָׁה פעולה f() בשתי מחלקות.

קטע הקוד שלפניך **תקין**.

```
A a1 = new A();
A e1 = new E();
E c1 = new C();
C b1 = new B();
C d1 = new D();
```

נתון:

ההוראה B d2 = new D(); גורמת לשגיאת הידור (קומפילציה).

ההוראה a1.f(); גורמת לשגיאת הידור (קומפילציה).

ההוראה: ((E)e1).f(); תקינה ומדפיסה "bye-bye" .

ההוראה ((B)b1).f(); תקינה ומדפיסה "hello" .

ההוראה ((D)b1).f(); גורמת לשגיאת זמן ריצה.

א. צייר עץ ירושה של כל המחלקות, וציין באֵילו שתי מחלקות מומשה הפעולה f() .

ב. נתונה המחלקה Z :

```
public class Z
{
    public void g(){}
}
```

(1) הוסף את המחלקה Z לעץ הירושה שציירת בסעיף א כך שההוראה שלפניך תהיה תקינה:

```
Z x = new A();
```

(2) קטע הקוד שלפניך **תקין**.

```
A a2 = new A();
Z z1 = new Z();
Z a3 = new A();
```

לכל אחת מן ההוראות v-i שלפניך, ציין אם היא תקינה או אינה תקינה.

אם ההוראה אינה תקינה, נמק מדוע.

הסעיפים v-i **אינם** תלויים זה בזה.

```
i    a2 = z1;
ii   a3 = z1;
iii  ((A) z1) .g();
iv   a2.g();
v    ((A)a3).g();
```

המשך השאלה בעמוד הבא.

ג. נתונה המחלקה Y :

```
public class Y
{
    public void m(){}
}
```

(1) הוסף את המחלקה Y לעץ הירושה שציירת בסעיף א כך שההוראה שלפניך תהיה תקינה (ללא התייחסות לנתונים בסעיף ב).

```
C f = new Y();
```

(2) קטע הקוד שלפניך **תקין**.

```
A a2 = new A();
Y y1 = new Y();
C y2 = new Y();
```

עבור <u>כל אחת</u> מן ההוראות v-i שלפניך, ציין אם היא תקינה <u>או</u> אינה תקינה.

אם ההוראה אינה תקינה, נמק מדוע.

הסעיפים v-i **אינם** תלויים זה בזה.

```
i    a2 = y1;
ii   y2 = y1;
iii  ((A)y1).m();
iv   y2.m();
v    ((A)y2).m();
```

תכנות מונחה עצמים בשפת C#

אם למדתם מסלול זה ואתם כותבים בשפת C# , ענו על שתיים מן השאלות 13–15 (לכל שאלה - 25 נקודות).

13. במסעדת "הרמה" אפשר להזמין מקום עבור סועד אחד או יותר. בסוף הארוחה, כל אחד מן הסועדים משלם בנפרד על מה שאכל. כל סועד יכול לבחור אם לשלם במזומן או באמצעות אפליקציה או בכרטיס אשראי. למשלמים בכרטיס אשראי יש אפשרות לפצל את החשבון לכמה תשלומים שווים.

כדי לנהל מערכת תשלומים, מפתחים עבור המסעדה פרויקט ובו הממשק (Interface) **IPayment** , והמחלקות **Restaurant, Reservation, Credit, App, Cash** , כמפורט להלן:

- **IPayment** – תשלום

  בממשק קיימת הפעולה: double GetPrice()

- **Cash** – תשלום במזומן

  תכונות המחלקה:

  - sumCash – הסכום לתשלום במזומן, מטיפוס ממשי.
  - name – שם הלקוח, מטיפוס מחרוזת.

- **App** – תשלום באמצעות אפליקציה

  תכונות המחלקה:

  - sumApp – הסכום לתשלום באפליקציה, מטיפוס ממשי.
  - phoneNumber – מספר טלפון, מטיפוס מחרוזת.

- **Credit** – תשלום בכרטיס אשראי

  תכונות המחלקה:

  - num – מספר התשלומים, מטיפוס שלם.
  - part – הסכום בכל תשלום, מטיפוס ממשי.
  - creditNumber – מספר כרטיס האשראי, מטיפוס מחרוזת.

- **Reservation** – הזמנה

  תכונות המחלקה:

  - date – תאריך ההזמנה, מטיפוס מחרוזת.
  - total – החשבון סך הכול של כל הסועדים יחד באותה הזמנה, מטיפוס ממשי.
  - payments – מערך השומר את התשלום ששילם כל אחד מן הסועדים באותה הזמנה, מטיפוס **IPayment** (המערך ללא ערכי null).

- **Restaurant** – המסעדה

  תכונות המחלקה:

  - array – מערך בגודל 10,000 , מטיפוס **Reservation** .
  - current – כמות ההזמנות השמורות, מטיפוס שלם.

  הניחו שההזמנות נשמרות ברצף במערך ואין ביניהן ערכי null .

הערה: הניחו שבכל מחלקה הוגדרו פעולות Get ו־ Set ופעולות בונות.

(שימו לב: המשך השאלה בעמוד הבא.)

א. **(1)** סרטטו תרשים היירכייה המתאר את הקשרים של המחלקות והממשק של הפרויקט.

יש לסמן מימוש ממשק באמצעות החץ ⇢◁ והכלה באמצעות הסימן ⬟——— .

(2) המסעדה עושה הנחות לסועדים בהתאם לצורת התשלום: סועד שמשלם במזומן והסכום לתשלום הוא מעל 200 שקלים זכאי ל־ 10% הנחה, סועד שמשלם באמצעות האפליקציה זכאי תמיד ל־ 5% הנחה, וסועד שמשלם בכרטיס אשראי משלם מחיר מלא.

פעולת הממשק GetPrice מחזירה את סך התשלום שכל סועד משלם לאחר שקלול ההנחה (כאשר אין הנחה, הפעולה מחזירה מחיר מלא). ממשו את פעולת הממשק GetPrice בכל אחת מן המחלקות הנדרשות.

ב. כתבו במחלקה **Reservation** פעולה ששמה CashTotal . הפעולה מחזירה את סכום הכסף <u>המזומן</u> שהתקבל בהזמנה (לאחר שקלול ההנחה).

ג. לפניכם פעולה במחלקה **Reservation** .

```
public void PrintDetails () {
    for (int i = 0; i< payments.Length; i ++) {
        Console.WriteLine (payments[i].GetDetails());
    }
}
```

הפעולה מדפיסה עבור כל סועד פרטים מסוימים על פי הקריטריונים שלהלן:

- אם שילם הסועד במזומן, הפעולה מדפיסה את השם שלו.
- אם שילם הסועד באמצעות האפליקציה, הפעולה מדפיסה את מספר הטלפון שלו.
- אם שילם הסועד בכרטיס אשראי, הפעולה מדפיסה את מספר כרטיס האשראי שלו.

הוסיפו את הפעולות הנחוצות לפרויקט כדי שהפעולה תבצע את הנדרש. ציינו עבור כל פעולה שהוספתם לאיזו מחלקה או לאיזה ממשק היא שייכת.

<u>הערה</u>: אין לשנות את הפעולה PrintDetails .

14. לפניכם המחלקות First ו־ Second :

```
public class First
{
    private static int count = 0;
    protected int x;
    protected int y;
    public First (int num) {
        this.x = num;
        this.y = num;
        count ++;
        Console.WriteLine ("First 1");
    }
    public First (int num1, int num2) {
        this.x = num1;
        this.y = num2;
        count ++;
        Console.WriteLine ("First 2");
    }
    public static int GetCount() {
        return count;
    }
    public int GetX() {
        return x; }
    public int GetY() {
        return y; }
    public  virtual int Sum() {
        return this.x + this.y;
    }
    public virtual void Add (First other) {
        this.x += other.x;
        this.y += other.y;
        Console.WriteLine ("x = "+this.x +" y = "
+this.y);}}
```

```
public class Second : First
{
    private int z;
    public Second (int num) : base (num) {
        this.z = num;
        Console.WriteLine (“Second”);
    }
    public override int Sum() {
        return base.Sum() + this.z;
    }
    public override void Add (First other) {
        this.x += other.GetX();
        this.y += other.GetY();
        if (other is Second)
            this.z += ((Second)other).z;
        Console.WriteLine("x = "+ this.x +
                    " y =" + this.y + " z =" + this.z);
    }
}
```

נתונה המחלקה Tester :

```
public class Tester
{
    public static void Main(string[] args)
    {
        First f1 = new First (40);
        First f2 = new First (40, 50);
        First f3 = new Second (100);
        Second s1 = new Second (100);
        Second s2 = new Second (100);
        // ***
    }
}
```

ציירו את העצמים שנוצרו בפעולה Main , וכתבו את הפלט של הפעולה.

הציבו כל אחת מן הפקודות 1–10 שלהלן בפעולה Main במקום המצוין לעיל ב- *** .

כתבו במחברת את מספר הפקודה וציינו אם הקוד תקין או לא תקין.

אם הקוד תקין – כתבו את הפלט, ואם הוא אינו תקין, הסבירו מדוע.

<u>הערה</u>: אין קשר בין הפקודות. כלומר, <u>יש להתייחס לכל פקודה כאילו היא היחידה בפעולה Main</u> .

```
1.  Console.WriteLine ("Total = " + First.GetCount());
2.  Console.WriteLine ("Total = " + Second.GetCount());
3.  Console.WriteLine ("sum = " + s1.Sum());
4.  Console.WriteLine ("sum = " + f3.Sum());
5.  s1 = new First (100);
6.  f1.Add (s2);
7.  s1.Add (s2);
8.  s2.Add (f3);
9.  ((First)s1).Add (f1);
10.     s1 = new Second (100, 100);
```

15. לפניך פרויקט שהוגדרו בו המחלקות האלה: A , B , C , D , E . בפרויקט מומשה פעולה F() בשתי מחלקות.
קטע קוד שלפניך **תקין**.

```
A a1 = new A();
A e1 = new E();
E c1 = new C();
C b1 = new B();
C d1=  new D();
```

נתון:
ההוראה B d2=new D(); גורמת לשגיאת הידור (קומפילציה).
ההוראה a1.F(); גורמת לשגיאת הידור (קומפילציה).
ההוראה ((E)e1).F(); תקינה ומדפיסה "bye-bye" .
ההוראה ((B)b1).F(); תקינה ומדפיסה "hello" .
ההוראה ((D)b1).F(); גורמת לשגיאת זמן ריצה.

א. צייר עץ ירושה של כל המחלקות, וציין באילו שתי מחלקות מומשה הפעולה F() .

ב. נתונה המחלקה Z :

```
public class Z
{
        public void G(){}
}
```

(1) הוסף את המחלקה Z לעץ הירושה שציירת בסעיף א כך שההוראה שלפניך תהיה תקינה:

```
Z x = new A();
```

(2) קטע הקוד שלפניך **תקין**.

```
A a2 = new A();
Z z1 = new Z();
Z a3 = new A();
```

לכל אחת מן ההוראות v-i שלפניך, ציין אם היא תקינה או אינה תקינה.
אם ההוראה אינה תקינה, נמק מדוע.
הסעיפים v-i **אינם** תלויים זה בזה:

```
i     a2 = z1;
ii    a3 = z1;
iii   ((A)z1).G();
iv    a2.G();
v     ((A)a3).G();
```

המשך השאלה בעמוד הבא.

ג. נתונה המחלקה Y :

```
public class Y
{
    public void M(){}
}
```

(1) הוסף את המחלקה Y לעץ הירושה שציירת בסעיף א כך שההוראה שלפניך תהיה תקינה (ללא התייחסות לנתונים בסעיף ב).

```
C f = new Y();
```

(2) קטע הקוד שלפניך **תקין**.

```
A a2 = new A();
Y y1 = new Y();
C y2 = new Y();
```

עבור כל אחת מן ההוראות v-i שלפניך, ציין אם היא תקינה או אינה תקינה.
אם ההוראה אינה תקינה, נמק מדוע.
הסעיפים v-i **אינם** תלויים זה בזה.

```
i    a2 = y1;
ii   y2 = y1;
iii  ((A)y1).M();
iv   y2.M();
v    ((A)y2).M();
```