

מדעי המחשב

הוראות

א. משך הבחינה : שלוש שעות.

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

פרק ראשון – (25 X 2) – 50 נקודות

פרק שני – (25 X 2) – 50 נקודות

סך הכול – 100 נקודות

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

ד. הוראות מיוחדות :

את כל התוכניות שיש לכתוב בשפת מחשב 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, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

/ המשך בעמוד הבא /

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

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

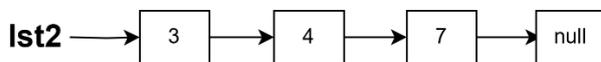
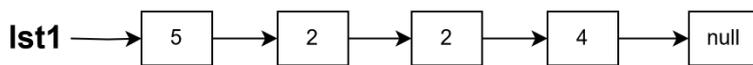
פרק ראשון (50 נקודות)

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

1. נתונות 2 רשימות של מספרים שלמים lst1 ו-lst2.

א. כתבו פעולה חיצונית בשם createUnion בשפת Java או CreateUnion בשפת C#. הפעולה מקבלת כקלט את שתי הרשימות ומחזירה רשימה חדשה המכילה את כל המספרים המופיעים לפחות באחת מהרשימות. סדר המספרים ברשימה המוחזרת אינו חשוב וכל מספר מופיע פעם אחת בלבד.

דוגמה: עבור זימון הפעולה עם הרשימות lst1, lst2 שלפניכם:



תוחזר הרשימה הבאה (אין חשיבות לסדר המשתנים):

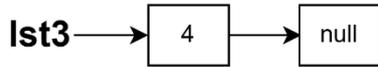


הסבר: כל מספר שהופיע לפחות באחת הרשימות, מופיע ברשימה החדשה. כמו כן, ברשימה החדשה כל מספר מופיע פעם אחת בלבד (ללא כפילות).

(המשך השאלה בעמוד הבא.)

ב. כתבו פעולה חיצונית בשם intersection בשפת Java או Intersection בשפת C#. הפעולה מקבלת כקלט את שתי הרשימות ומחזירה רשימה חדשה המכילה רק את המספרים המופיעים גם ב-lst1 וגם ב-lst2. סדר המספרים ברשימה המוחזרת אינו חשוב וכל מספר מופיע פעם אחת בלבד. אם אין אף מספר משותף, תוחזר רשימה ריקה.

דוגמה: עבור זימון הפעולה עם הרשימות lst1, lst2, lst1 (בסעיף א), תוחזר הרשימה הבאה:



הסבר: רק המספר 4 מופיע בשתי הרשימות.

2. שני תורים הם "M-מחוברים" אם ערכם של M האיברים האחרונים של התור הראשון זהים (לפי אותו סדר) לערכם של M האיברים הראשונים של התור השני.

א. כתבו פעולה ששמה isMConnect בשפת Java או IsMConnect בשפת C#, המקבלת תור ראשון q_1 , ותור שני q_2 (שניהם מטיפוס שלם) ומשתנה מטיפוס שלם M. הפעולה תחזיר true אם התור q_1 הוא "מחובר" לתור q_2 , אחרת היא תחזיר false.

- הערות:
- בסיום הפעולה חובה לשמור על מבנה שני התורים כפי שהתקבלו.
 - אין להשתמש בסעיף זה במערך או ברשימה מקושרת. פתרון הכולל שימוש בהם לא יזוכה בנקודות.
 - הניחו ש-M קטן או שווה מאורך 2 התורים.

לדוגמה: עבור התורים שלפניכם:

	ראש התור						סוף התור
q_1	2	5	4	4	7	<u>2</u>	<u>6</u>

	ראש התור	סוף התור							
q_2	<u>2</u>	<u>6</u>	7	1	4	2	1	8	1

עבור $M = 1$ יוחזר false (ערך האיבר האחרון של q_1 - 6, שונה מערך האיבר הראשון של q_2 - 2).
 עבור $M = 2$ יוחזר true (ערכי שני האיברים האחרונים של q_1 שווים לפי אותו סדר לערכי שני האיברים הראשונים של q_2).
 עבור $M = 3$ יוחזר false (ערכי שלושת האיברים האחרונים של q_1 - 7, 2, 6, אינם שווים לפי אותו סדר לערכי שלושת האיברים הראשונים של q_2 - 2, 6, 7).

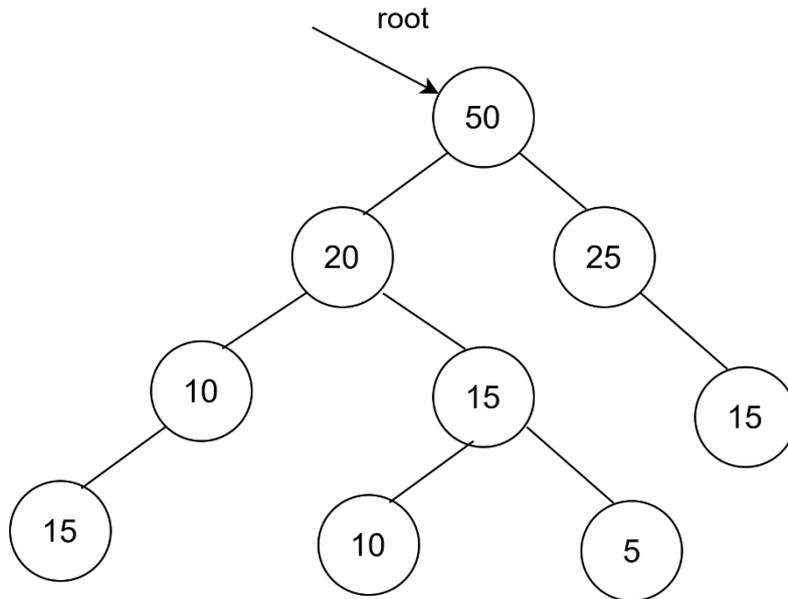
ב. כתבו פעולה בשם maxConnect בשפת Java או MaxConnect בשפת C#, המקבלת תור ראשון q_1 ותור שני q_2 (שניהם מטיפוס שלם). הפעולה תחזיר את ה-M הגדול ביותר שעבורו מתקיים ש- q_1 "מחובר" ל- q_2 . אם הם לא מחוברים בכלל, יוחזר 0.
הערה: ניתן להשתמש בפעולה מסעיף א.

דוגמאות:

עבור שני התורים מהדוגמה בסעיף א, והזימון $\text{maxConnect}(q_1, q_2)$ יוחזר המספר 2 (יש שני איברים בסוף q_1 שזהים לתחילת q_2 . כאשר מגדילים את M הם אינם זהים).

עבור אותם שני התורים והזימון $\text{maxConnect}(q_2, q_1)$ יוחזר המספר 0 (אין איברים בסוף q_2 שמתאימים לתחילת q_1 עבור אף M).

3. לפניכם עץ בינארי ששורשו root :



א. נתונה הפעולה sod בשפת Java ו-Sod בשפת C# :

Java	<pre> public static boolean sod(BinNode<Integer> root, int num) { if (root == null) return false; if (root.getValue() == num) return true; return sod(root.getLeft(), num) sod(root.getRight(), num); } </pre>
C#	<pre> public static bool Sod(BinNode<int> root, int num) { if (root == null) return false; if (root.GetValue() == num) return true; return Sod(root.GetLeft(), num) Sod(root.GetRight(), num); } </pre>

- 1) מהו הערך שיוחזר מהפעולה עבור הזימון : sod(root, 25)? יש לבצע מעקב על הקוד.
- 2) כתבו זימון נוסף שעבורו יוחזר ערך שונה מהערך שהוחזר בסעיף 1.
- 3) הסבירו מה מבצעת הפעולה.
(שימו לב: המשך השאלה בעמוד הבא.)

ב. נתונה הפעולה secret בשפת Java ו-Secret בשפת C# :

Java	<pre>public static int secret(BinNode<Integer> root, int x) { if (!sod(root,x)) return 0; if ((root.getLeft() != null && root.getLeft().getValue() == x) (root.getRight() != null && root.getRight().getValue() == x)) { return root.getValue()+ secret(root.getLeft(),x)+ secret(root.getRight(),x); } return secret(root.getLeft(),x)+secret(root.getRight(),x); }</pre>
C#	<pre>public static int Secret(BinNode<int> root, int x) { if (!Sod(root, x)) return 0; if ((root.GetLeft() != null && root.GetLeft().GetValue() == x) (root.GetRight() != null && root.GetRight().GetValue() == x)) { return root.GetValue() + Secret(root.GetLeft(), x) + Secret(root.GetRight(), x); } return Secret(root.GetLeft(), x) + Secret(root.GetRight(), x); }</pre>

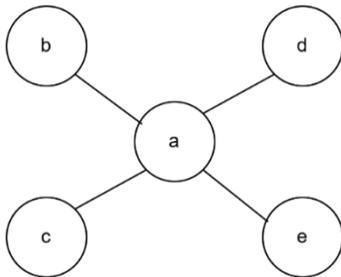
- 1) מהו הערך שיוחזר מהפעולה עבור הזימון: Secret(root, 10) ? יש לבצע מעקב. אין צורך לבצע מעקב על הקריאה לפעולה sod/Sod אלא רק לציין את הערך שחוזר מהפעולה.
- 2) מהו הערך שיוחזר מהפעולה עבור הזימון: Secret(root, 5) ? אין צורך לבצע מעקב.
- 3) כתבו זימון נוסף שעבורו יוחזר הערך 50. אין צורך לבצע מעקב.
- 4) הסבירו מה מבצעת הפעולה.

פרק שני (50 נקודות)

בפרק זה שאלות בשלושה מסלולים :
 אלגוריתמים, שאלות 4-6
 מודלים חישוביים, שאלות 7-9
 תכנות מונחה עצמים בשפת Java ובשפת C#, שאלות 10-12 (3 השאלות בשפת C# מופיעות לאחר 3 השאלות בשפת Java)
 ענו על שתי שאלות מתוך מסלול אחד בלבד (לכל שאלה – 25 נקודות).

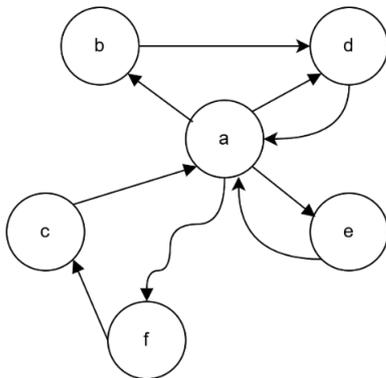
אלגוריתמים:

4. בשאלה זו שני סעיפים, א-ב, שאין ביניהם קשר. יש לענות על שני הסעיפים.
 א. להלן גרף $G=(V,E)$ שאינו מכוון :



- i. האם הגרף קשיר? נמקו.
- ii. האם הגרף דו-צדדי? נמקו.
- iii. האם הגרף מלא (שלם)? נמקו.
- iv. האם הגרף הוא עץ? נמקו.

ב. מהנדס העיר הגדיר את המעבר בין השכונות a,b,c,d,e,f על פי הגרף שלהלן :



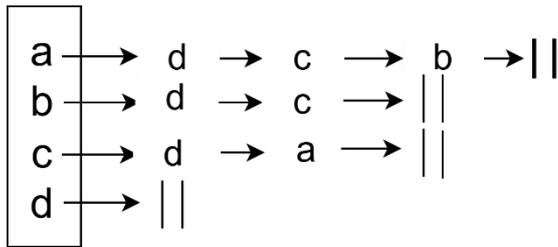
- i. שרטטו מטריצת סמיכויות.
 - ii. האם הגרף הוא גרף קשיר חזק? נמקו.
 - iii. תושב העיר צריך להגיע משכונה אחת לשכונה אחרת. לשם כך עליו לבדוק אם קיים מסלול בין שתי השכונות ואם קיים, הוא רוצה למצוא מסלול כלשהו המגיע לאותה שכונה. מהו האלגוריתם שעליו להשתמש בו למטרות אלה? הסבירו.
 - iv. הפעילו את האלגוריתם שכתבתם עליו בסעיף הקודם כדי למצוא את המסלול משכונה b לשכונה c. כתבו את המסלול שמצאתם.
- יש לבצע מעקב מפורט בכל שלב בהתאם לאלגוריתם שהפעלתם.

5. לפניכם שבע טענות. בחרו בחמש 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) בגרף לא מכוון, אם יש קשתות שמחברות בין כל זוג קודקודים, אז הוא תמיד קשיר.
- (2) גרף מכוון שבו n קודקודים ויותר מ- $n-1$ קשתות, תמיד מכיל מעגל.
- (3) בגרף מכוון G שבו n קודקודים ו- m קשתות, אם $m < n-1$ אז הגרף תמיד קשיר.
- (4) אלגוריתם דייקסטרה הוא הבחירה המועדפת כאשר הגרף מכיל קשתות בעלות משקל שלילי, מכיוון שהוא מהיר יותר מהאלגוריתם של בלמן-פורד.
- (5) יער הוא גרף שאין בו מעגלים, והוא בהכרח לא קשיר.
- (6) ניתן להשתמש באלגוריתם בלמן-פורד למציאת מסלול קצר בגרף מכוון המכיל קשתות עם משקל שלילי, אך לא עם מעגל שלילי.
- (7) גרף G מכוון שבו הדרגות של כל הקודקודים גדולות מ-0 הוא תמיד קשיר.

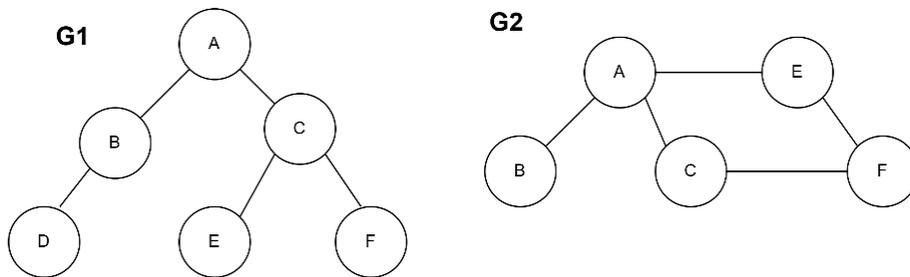
6. בשאלה זו שני סעיפים, א-ב, שאין ביניהם קשר. יש לענות על שני הסעיפים.

א. נתון גרף $G = (V, E)$ מכוון, המיוצג על ידי רשימת הסמיכויות הבאה:

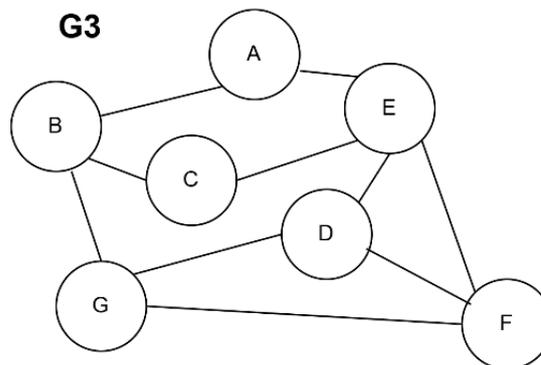


- 1) שרטטו את הגרף G המיוצג על ידי רשימת הסמיכויות.
 - 2) הפעילו אלגוריתם סריקה לעומק (DFS) על הגרף G החל בצומת a . שרטטו את העץ הפורש DFS.
 - 3) הפעילו אלגוריתם סריקה לרוחב (BFS) על הגרף G החל בצומת c . שרטטו את העץ הפורש BFS.
- ב. גרף דו-צדדי הוא גרף בו ניתן לחלק את הצמתים שבו לשתי קבוצות זרות, כך שלא קיימת קשת בין שני צמתים השייכים לאותה הקבוצה.

1) לפניכם 2 גרפים דו צדדיים. הראו את שתי קבוצות הצמתים עבור כל גרף.



2) לפניכם גרף שאינו דו צדדי. מהו מספר הקשתות המינימלי שיש להסיר כדי שהגרף יהיה דו צדדי? כתבו אילו קשתות יש להסיר והציגו את שתי הקבוצות.



מודלים חישוביים

7. לפניכם השפות L_1 ו- L_2 מעל הא"ב $\{0,1\}$:

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

$$L_2 = \{0^k 1^n \mid n, k > 0, n \bmod 2 = k \bmod 2\}$$

- א. כתבו מילה באורך 6 השייכת לשפה L_1 ומילה באורך 6 השייכת לשפה L_2 .
- ב. אם השפה L_1 רגולרית, יש לבנות אוטומט סופי דטרמיניסטי לא מלא שיקבל את השפה. אם השפה אינה רגולרית, יש לבנות אוטומט מחסנית דטרמיניסטי שיקבל את השפה.
- ג. אם השפה L_2 רגולרית, יש לבנות אוטומט סופי דטרמיניסטי לא מלא שיקבל את השפה. אם השפה אינה רגולרית, יש לבנות אוטומט מחסנית דטרמיניסטי שיקבל את השפה.

8. לפניכם שני סעיפים, א-ב, שאין קשר ביניהם. עליכם לענות על שניהם.

א. נתונות השפות L_1, L_2, L_3 מעל הא"ב $\{a,b\}$:

$$L_1 = \{\text{כל המילים בהן מספר המופעים של האות } a \text{ והאות } b \text{ שווה}\}$$

$$L_2 = \{\text{כל המילים בהן מספר הפעמים שהאות } a \text{ מופיעה לפחות שתי פעמים}\}$$

$$L_3 = L_1 \cap L_2$$

- (1) כתבו מילה באורך 6 או יותר השייכת לשפה L_1 .
- (2) כתבו מילה באורך 6 או יותר השייכת לשפה L_2 .
- (3) הגדירו במילים את השפה L_3 .
- (4) מהו אורך המילה המינימלי בכל אחת מן השפות?

ב. נתונה השפה L מעל הא"ב $\{0,1\}$ בה המילים מקיימות את שני התנאים הבאים:

- אורך כל המילים בשפה הוא לפחות 2.
- הרצף 000 מופיע לכל היותר פעם אחת בלבד במילה.

דוגמאות למילים בשפה:

00, 01010, 101000100

דוגמאות למילים שאינן בשפה:

0000, 1, 1000110001

הוכיחו שהשפה L רגולרית. מומלץ להשתמש בתכונות הסגירות.

9. נתונה הפעולה compare בשפת java ו- Compare בשפת C#:

Java	C#
<pre>public static int compare(int n1, int n2) { if (n1 >= n2) return 1; return 2; }</pre>	<pre>public static int Compare(int n1, int n2) { if (n1 >= n2) return 1; return 2; }</pre>

בנו מכונת טיורינג המממשת את הפעולה.

המכונה מקבלת 2 מספרים שלמים גדולים מ-0 הכתובים בתחילת הסרט ומיוצגים בשפה אונירית (ע"י הספרה 1 בלבד). בין שני המספרים מופיע הסימן #. את הערך המוחזר מהפעולה יש לרשום בין 2 סימני \$ במיקום כלשהו על הסרט.

דוגמה, לפני הרצת

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

המכונה:

אחרי הרצת המכונה:

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

הסבר: המספר הראשון $n_1 = 2$ והמספר השני $n_2 = 3$ ולכן המספר שהמכונה מחזירה הוא 2 (11) באונירית).

תכנות מונחה עצמים בשפת Java:

10. לפניכם 3 מחלקות Y, Z, Tester :

```
public class Y{
    public Y(){
        System.out.println("ctor Y");
    }
    public void do1(int x){
        System.out.println("Y do1 int "+x);
    }
    public void do1(String str){
        System.out.println("Y do1 String "+str);
    }
    public void do2(int x){
        System.out.println("Y do2 "+x);
        do1(x);
    }
}
```

```
public class Z extends Y{
    public Z(){
        System.out.println("ctor Z");
    }
    public void do1(int x){
        System.out.println("Z do1 int "+x);
    }
    public void do2(){
        System.out.println("Z do2");
        do1(2);
    }
    public void do3(int x){
        System.out.println("Z do3 "+x);
        do1(3);
    }
}
```

```
public class Tester{
    public static void main(String[] args)
    {

    }
}
```

לפניכם 12 קטעי קוד. בחרו 10 מהם.

הציבו כל אחד מקטעי הקוד שבחרתם בפעולה main במחלקת Tester. יש לכתוב במחברת הבחינה את מספר הקטע שבחרתם ולציין האם הקוד תקין או לא.

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

הערה: אין קשר בין קטעי הקוד. כלומר, יש להתייחס לכל קטע קוד כאילו הוא היחיד בפעולה.

1. Y y = new Y0; y.do1(2);	2. Z z = new Z0; z.do1("str");	3. Z z = new Z0; Y y = z; y.do1("str");
4. Y y = new Y0; Z z = y; z.do1(2);	5. Y y = new Y0; Z z = (Z)y; z.do1(2);	6. Y yz = new Z0; yz.do1("str");
7. Y yz = new Z0; yz.do3(4);	8. Z z = new Z0; z.do20;	9. Z yz = new Y0; yz.do20;
10. Y yz = new Z0; yz.do20;	11. Y yz = new Z0; ((Z)yz).do20;	12. Y yz = new Z0; ((Z)yz).do1("hh");

11. בחנות כסאות פיתחו מערכת ממוחשבת שבה המחלקות הבאות:
Date, Chair, ComputerChair, GardenChair, BarStool כמפורט להלן:
 להלן פירוט תכונות המחלקות:

❖ **Date** - תאריך

תכונות המחלקה:

- month - חודש, מטיפוס שלם
- year - שנה, מטיפוס שלם

❖ **Chair** - כסא

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- createDate - תאריך ייצור, מטיפוס Date

❖ **ComputerChair** - כסא מחשב

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- wheels - מספר גלגלים, מטיפוס שלם
- adjustHeight - גובה הרמת מושב, מטיפוס שלם
- createDate - תאריך ייצור, מטיפוס Date

❖ **GardenChair** - כסא גינה

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- isFoldable - האם ניתן לקפל, מטיפוס בוליאני
- createDate - תאריך ייצור מטיפוס Date

❖ **BarStool** - כסא בר

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- adjustHeight - גובה הרמת מושב, מטיפוס שלם
- createDate - תאריך ייצור, מטיפוס Date

א. (1) סרטו תרשים היררכיה המתאר את הקשר בין המחלקות של המערכת הממוחשבת.



יש לסמן ירושה באמצעות החץ

הכלה באמצעות הסימן

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שהפעולות get ו-set קיימות בכל התכונות של המחלקות ואין צורך למשש אותן.

נתונה הפעולה הבונה של מחלקת Chair :

public Chair(int price, Date createDate)

אין צורך לממש את הפעולה.

(3) לפניכם כותרת הפעולה הבונה של המחלקה ComputerChair. הפעולה מקבלת מחיר, מספר גלגלים, גובה הרמת מושב ותאריך ייצור.

public ComputerChair(int price, int wheels, int height, Date date)

ממשו את הפעולה הבונה.

ב. נתונה המחלקה Order – הזמנה.

תכונות המחלקה :

- chairs – מערך של כסאות, מטיפוס Chair. כמות כסאות מקסימלית בהזמנה – 100.
- current - כמות הכסאות בהזמנה הנוכחית.

החנות מעניקה הנחה להזמנה בהתאם לכמות הכסאות בהזמנה, באופן הבא :

❖ אם מספר כסאות המחשב בהזמנה גדול מ 10, תינתן הנחה של 20% על כל כסאות המחשב שבהזמנה.

❖ אם מספר כסאות הגינה בהזמנה גדול מ 15, תינתן הנחה של 25% על כל כסאות הגינה שבהזמנה. כיתבו פעולה בשם totalPrice במחלקה Order המחזירה את סכום ההזמנה. בנוסף, הפעולה מדפיסה את פרטי הכסאות בהזמנה (בהתאם לתכונות שלהם).

12. לפניכם המחלקות One ו-Two :

<pre> public class One { private static int num = 0; protected int x; public One (int n) { this.x = n; num += n; System.out.println ("One 1"); } public One (int n1, int n2) { this.x = n1+n2; num += n1; System.out.println ("One 2"); } public int getNum() { return num; } public int getX() { return x; } public int mul(int n) { return this.x * n; } public void add(One other) { this.x += other.x; System.out.println("x = "+ this.x); } } </pre>	<pre> public class Two extends One { private int y; public Two (){ super(10); this.y = 12; } public Two (int num) { super (num/2); this.y = num; System.out.println ("Second"); } public int mul() { return this.x * this.y; } public void add (One other) { if (other instanceof Two) this.y += ((Two)other).y; System.out.println("y = "+ this.y); } } </pre>
---	---

נתונה המחלקה Tester :

```
public class Tester
{
    public static void main(String[] args)
    {
        One one1 = new One (1);
        One one2 = new One (2, 3);
        One oneTwo = new Two (4);
        Two two1 = new Two ();
        Two two2 = new Two (6);
        // ***
    }
}
```

- 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.
הערה: אין קשר בין הפקודות. כלומר, יש להתייחס לכל פקודה כאילו היא היחידה בפעולה.

1. System.out.println ("Total = " + one1.getNum());
2. System.out.println ("Total = " + two2.getNum());
3. System.out.println ("mul = " + one1.mul(1));
4. System.out.println ("mul = " + oneTwo.mul());
5. System.out.println ("mul = " + two1.mul());
6. one1.add(oneTwo);
7. two1.add(oneTwo);
8. ((Two)one1).mul();
9. two1 = new One(1,2);
10. ((One)two1).mul(1);

תכנות מונחה עצמים בשפת C#:

10. לפניכם 3 מחלקות Y, Z, Tester :

```

public class Y
{
    public Y()
    {
        Console.WriteLine("ctor Y");
    }
    public virtual void Do1(int x)
    {
        Console.WriteLine("Y Do1 int " + x);
    }
    public void Do1(string str)
    {
        Console.WriteLine("Y Do1 String " + str);
    }
    public void Do2(int x)
    {
        Console.WriteLine("Y Do2 " + x);
        Do1(x);
    }
}

public class Z: Y
{
    public Z()
    {
        Console.WriteLine("ctor Z");
    }
    public override void Do1(int x)
    {
        Console.WriteLine("Z Do1 int " + x);
    }
    public void Do2()
    {
        Console.WriteLine("Z Do2");
        Do1(2);
    }
    public void Do3(int x)
    {
        Console.WriteLine("Z Do3 " + x);
        Do1(3);
    }
}

public class Tester{
    public static void Main(string[] args)
    {

    }
}

```

לפניכם 12 קטעי קוד. בחרו 10 מהם.

הציבו כל אחד מקטעי הקוד שבחרתם בפעולה main במחלקת Tester. יש לכתוב במחברת הבחינה את מספר הקטע שבחרתם ולציין האם הקוד תקין או לא.

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

הערה: אין קשר בין קטעי הקוד. כלומר, יש להתייחס לכל קטע קוד כאילו הוא היחיד בפעולה.

1. Y y = new Y0; y.Do1(2);	2. Z z = new Z0; z.Do1("str");	3. Z z = new Z0; Y y = z; y.Do1("str");
4. Y y = new Y0; Z z = y; z.Do1(2);	5. Y y = new Y0; Z z = (Z)y; z.Do1(2);	6. Y yz = new Z0; yz.Do1("str");
7. Y yz = new Z0; yz.Do3(4);	8. Z z = new Z0; z.Do20;	9. Z yz = new Y0; yz.Do20;
10. Y yz = new Z0; yz.Do20;	11. Y yz = new Z0; ((Z)yz).Do20;	12. Y yz = new Z0; ((Z)yz).Do1("hh");

11. בחנות כסאות פיתחו מערכת ממוחשבת שבה המחלקות הבאות:
Date, Chair, ComputerChair, GardenChair, BarStool כמפורט להלן:
 להלן פירוט תכונות המחלקות:

❖ **Date** - תאריך

תכונות המחלקה:

- month - חודש, מטיפוס שלם
- year - שנה, מטיפוס שלם

❖ **Chair** - כסא

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- createDate - תאריך ייצור, מטיפוס Date

❖ **ComputerChair** - כסא מחשב

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- wheels - מספר גלגלים, מטיפוס שלם
- adjustHeight - גובה הרמת מושב, מטיפוס שלם
- createDate תאריך ייצור, מטיפוס Date

❖ **GardenChair** - כסא גינה

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- isFoldable - האם ניתן לקפל, מטיפוס בוליאני
- createDate - תאריך ייצור מטיפוס Date

❖ **BarStool** - כסא בר

תכונות המחלקה:

- price – מחיר, מטיפוס שלם
- adjustHeight - גובה הרמת מושב, מטיפוס שלם
- createDate - תאריך ייצור, מטיפוס Date

א. (1) סרטוטו תרשים היררכיה המתאר את הקשר בין המחלקות של המערכת הממוחשבת.



יש לסמן ירושה באמצעות החץ

הכלה באמצעות הסימן

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שהפעולות Get ו-Set קיימות בכל התכונות של המחלקות ואין צורך לממש אותן.

נתונה הפעולה הבונה של מחלקת Chair :

public Chair(int price, Date createDate)

אין צורך לממש את הפעולה.

(3) לפניכם כותרת הפעולה הבונה של המחלקה ComputerChair. הפעולה מקבלת מחיר, מספר גלגלים, גובה הרמת מושב ותאריך ייצור.

public ComputerChair(int price, int wheels, int height, Date date)

ממשו את הפעולה הבונה.

ב. נתונה המחלקה Order – הזמנה.

תכונות המחלקה :

- chairs – מערך של כסאות, מטיפוס Chair. כמות כסאות מקסימלית בהזמנה – 100.
- current - כמות הכסאות בהזמנה הנוכחית.

החנות מעניקה הנחה להזמנה בהתאם לכמות הכסאות בהזמנה באופן הבא :

❖ אם מספר כסאות המחשב בהזמנה גדול מ 10, תינתן הנחה של 20% על כל כסאות המחשב שבהזמנה.

❖ אם מספר כסאות הגינה בהזמנה גדול מ 15, תינתן הנחה של 25% על כל כסאות הגינה שבהזמנה. כיתבו פעולה בשם TotalPrice במחלקה Order המחזירה את סכום ההזמנה. בנוסף, הפעולה מדפיסה את פרטי הכסאות בהזמנה (בהתאם לתכונות שלהם).

12. לפניכם המחלקות One ו-One :

<pre> public class One { private static int num = 0; protected int x; public One(int n) { this.x = n; num += n; Console.WriteLine("One 1"); } public One(int n1, int n2) { this.x = n1 + n2; num += n1; Console.WriteLine("One 2"); } public int GetNum() { return num; } public int GetX() { return x; } public int Mul(int n) { return this.x * n; } public virtual void Add(One other) { this.x += other.x; Console.WriteLine("x = " + this.x); } } </pre>	<pre> public class Two : One { private int y; public Two(): base(10) { this.y = 12; } public Two(int num): base(num/2) { this.y = num; Console.WriteLine("Second"); } public int Mul() { return this.x * this.y; } public override void Add(One other) { if (other is Two) this.y += ((Two)other).y; Console.WriteLine("y = " + this.y); } } </pre>
---	---

נתונה המחלקה Tester :

```
public class Tester
{
    public static void Main(string[] args)
    {
        One one1 = new One (1);
        One one2 = new One (2, 3);
        One oneTwo = new Two (4);
        Two two1 = new Two ();
        Two two2 = new Two (6);
        // ***
    }
}
```

- 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.
הערה : אין קשר בין הפקודות. כלומר, יש להתייחס לכל פקודה כאילו היא היחידה בפעולה.

1. Console.WriteLine ("Total = " + one1.GetNum());
2. Console.WriteLine ("Total = " + two2.GetNum());
3. Console.WriteLine ("Mul = " + one1.Mul(1));
4. Console.WriteLine ("Mul = " + oneTwo.Mul());
5. Console.WriteLine ("Mul = " + two1.Mul());
6. one1.Add(oneTwo);
7. two1.Add(oneTwo);
8. ((Two)one1).Mul();
9. two1 = new One(1,2);
10. ((One)two1).Mul(1);

בהצלחה!