

סוג הבחינה: א. בגרות לבתי ספר על-יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: קיץ תשע"ד, 2014
מספר השאלון: 603,899205

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 בק 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:
- את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון כתוב בשפה אחת בלבד – Java או C#.
 - רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – Java או C#.
 - רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.
המסלול הוא אחד מארבעת המסלולים האלה:
מערכות מחשב ואסמבלר, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמיים.
- 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 כטיוטה (ראשי פרקים, חישובים וכדומה).
רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות 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

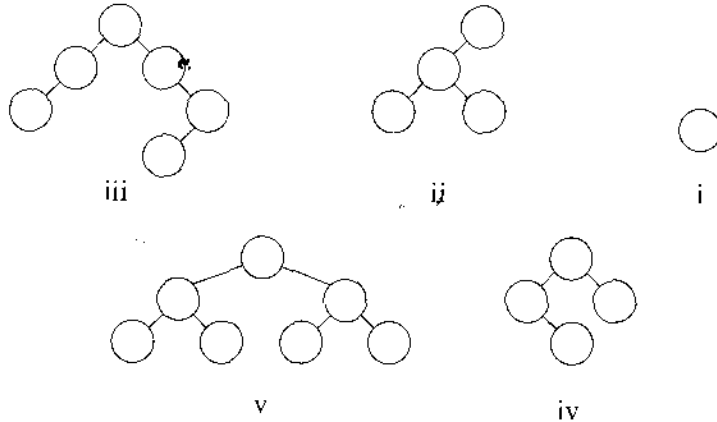
בשאלון זה שני פרקים: פרק ראשון ופרק שני.
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

פרק ראשון (50 נקודות)

שים לב: בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות רשימה, תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

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

1. עץ בינארי מאוזן הוא עץ בינארי שבכל צומת שלו הערך המוחלט של הפרש הגבהים בין התת-עץ הימני לתת-עץ השמאלי הוא לכל היותר 1.
- א. לפניך 5 עצים בינאריים i-v. לכל אחד מן העצים, קבע אם הוא עץ בינארי מאוזן או אינו עץ בינארי מאוזן. אם העץ אינו עץ בינארי מאוזן, העתק אותו למחברתך וסמן x בצומת שמכר את האיזון.



- ב. כתוב ב-Java או ב-C# פעולה שתקבל עץ בינארי כלשהו, ותחזיר true אם הוא עץ בינארי מאוזן, אחרת – הפעולה תחזיר false.
- אם הפרמטר שמועבר לפעולה הוא null – הפעולה תחזיר true.

2.

שים לב: לשאלה זו שני נוסחים:
נוסח אחד ב־ Java (עמוד 3),
ונוסח אחד ב־ C# (עמוד 4).
עבוד על פי השפה שלמדת.

לפותרים ב־ Java

המחלקה **Collec** היא אוסף של מספרים שלמים וגדולים מ־ 0. לאוסף זה אפשר להוסיף איבר רק אם אין באוסף איבר אחר גדול ממנו.

א. כתוב ב־ Java את כותרת המחלקה **Collec** ואת התכונות שלה.

ב. כתוב ב־ Java שתי פעולות בונות למחלקה **Collec** :

— פעולה בונה, בלי פרמטרים, היוצרת אוסף ריק.

— פעולה בונה שתקבל מספר שלם n גדול מ־ 0, ותיצור אוסף שיש בו איבר אחד שערכו n .

ג. ממש ב־ Java במחלקה **Collec** פעולה **add**, שתקבל מספר שלם וגדול מ־ 0

ותוסיף אותו לאוסף, אם אפשר. אם המספר צורך לאוסף, הפעולה תחזיר **true**. אחרת — הפעולה תחזיר **false**.

ד. ממש ב־ Java במחלקה **Collec** פעולה **small**, שתחזיר את המספר הקטן ביותר באוסף.

אם האוסף ריק, הפעולה תחזיר **-1**.

האוסף אינו משתנה בעקבות הפעלת הפעולה **small**.

ה. ממש ב־ Java במחלקה **Collec** פעולה בשם **smallest**, שתקבל עצם **c** מטיפוס **Collec**,

ותחזיר את המספר הקטן ביותר מבין שני האוספים — האוסף הנוכחי **c**.

הנח כי שני האוספים אינם ריקים.

לפותרים ב- C#

המחלקה **Collec** היא אוסף של מספרים שלמים וגדולים מ-0. לאוסף זה אפשר להוסיף איבר רק אם אין באוסף איבר אחר גדול ממנו.

- א. כתוב ב- C# את כותרת המחלקה **Collec** ואת התכונות שלה.
- ב. כתוב ב- C# שתי פעולות בונות למחלקה **Collec** :
 - פעולה בונה, בלי כרמטרים, היוצרת אוסף ריק.
 - פעולה בונה שתקבל מספר שלם n גדול מ-0, ותיצור אוסף שיש בו איבר אחד שערכו n.
- ג. ממש ב- C# במחלקה **Collec** פעולה **Add**, שתקבל מספר שלם וגדול מ-0 ותוסיף אותו לאוסף, אם אפשר. אם המספר צורף לאוסף, הפעולה תחזיר true. אחרת – הפעולה תחזיר false.
- ד. ממש ב- C# במחלקה **Collec** פעולה **Small**, שתחזיר את המספר הקטן ביותר באוסף. אם האוסף ריק, הפעולה תחזיר -1.
האוסף אינו משתנה בעקבות הפעלת הפעולה **Small**.
- ה. ממש ב- C# במחלקה **Collec** פעולה בשם **Smallest**, שתקבל עצם c מטיפוס **Collec**, ותחזיר את המספר הקטן ביותר מבין שני האוספים – האוסף הנוכחי ו- c.
הנח כי שני האוספים אינם ריקים.

/המשך בעמוד 5/

3.

שים לב: לשאלה זו שני נוסחים:
 נוסח אחד לפותרים על פי התכנית הנוכחית (ב־ Java וב־ C# – עמוד 5),
 ונוסח אחד לפותרים על פי התכנית החדשה (ב־ Java וב־ C# – עמוד 6).
 עבוד על פי התכנית שלמדת.

לפותרים על פי התכנית הנוכחית:

לפניך שלוש מחלקות:

המחלקה **Birth** – תאריך לידה, שתכונותיה הן: יום בחודש – day, חודש – month ושנה – year. כל אחת מן התכונות היא מספר שלם.

המחלקה **Student** – תלמיד, שתכונותיה הן: שם התלמיד – name מטיפוס מחרוזת, ותאריך הלידה של התלמיד – birthDay מטיפוס **Birth**.

המחלקה **School** – בית ספר, שתכונתה היא: מערך חד־ממדי ar בגודל 6, מטיפוס `List<Student>`. כל תא במערך ar מייצג שכבת גיל בבית הספר: תא 0 מייצג את שכבה ז', תא 1 מייצג את שכבה ח' וכך הלאה, עד תא 5 שמייצג את שכבה י"ב. כל תא מכיל רשימה של כל תלמידי השכבה.

הנח שבכל מחלקה יש: פעולה בונה בִּרְרַת מחדל ופעולה בונה שמקבלת פרמטר לכל תכונה, ב־ Java פעולות get ו־ set לכל תכונה, וב־ C# פעולות Get ו־ Set לכל תכונה.

כתוב ב־ Java או ב־ C# פעולה חיזונית, שתקבל עצם מטיפוס **School** ותחזיר מערך חד־ממדי בגודל 12 מטיפוס `List<Student>`. כל תא במערך מייצג חודש בשנה: תא 0 מייצג את ינואר, תא 1 מייצג את פברואר וכך הלאה, עד תא 11 שמייצג את דצמבר. כל תא יכיל רשימה של כל התלמידים מכל השכבות שנולדו בחודש זה, בלי חשיבות לסדר.

/המשך בעמוד 6/

לפותרים על פי התכנית החדשה

לפניך שלוש מחלקות:

המחלקה **Birth** – תאריך לידה, שתכונותיה הן: יום בחודש – day, חודש – month ושנה – year. כל אחת מן התכונות היא מספר שלם.

המחלקה **Student** – תלמיד, שתכונותיה הן: שם התלמיד – name מטיפוס מחרוזת, ותאריך הלידה של התלמיד – birthDay מטיפוס **Birth**.

המחלקה **School** – בית ספר שתכונתה היא: מערך חד־ממדי ar בגודל 6, מטיפוס `<Student>Node`. כל תא במערך ar מייצג שכבת גיל בבית הספר: תא 0 מייצג את שכבה ז', תא 1 מייצג את שכבה ח' וכך הלאה, עד תא 5 שמייצג את שכבה י"ב. כל תא מכיל רשימה של כל תלמידי השכבה.

הנה שבכל מחלקה יש: פעולה בונה בררת מחדל ופעולה בונה שמקבלת פרמטר לכל תכונה, ב־Java פעולות get ו־set לכל תכונה, וב־C# פעולות Get ו־Set לכל תכונה.

כתוב ב־Java או ב־C# פעולה חיצונית, שתקבל עצם מטיפוס **School** ותחזיר מערך חד־ממדי בגודל 12 מטיפוס `<Student>Node`. כל תא במערך מייצג חודש בשנה: תא 0 מייצג את ינואר, תא 1 מייצג את פברואר וכך הלאה, עד תא 11 שמייצג את דצמבר. כל תא יכיל רשימה של כל התלמידים מכל השכבות שנולדו בחודש זה, בלי חשיבות לסדר.

/המשך בעמוד 7/

4. שים לב: לשאלה זו שני נוסחים:
 נוסח אחד לפותרים על פי התכנית הנוכחית (ב' Java וב' C# – עמודים 7-11),
 ונוסח אחד לפותרים על פי התכנית החדשה (ב' Java וב' C# – עמודים 12-16).
 עבוד על פי התכנית שלמדת.

לפותרים על פי התכנית הנוכחית

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

א. נתונה פעולה המקבלת רשימה lst של מספרים שלמים:

C#	Java
<pre>public static int What(List<int> lst) { if (lst.IsEmpty()) return 0; int x = lst.GetFirst().GetInfo(); lst.Remove(lst.GetFirst()); if(x >= 0) return What(lst); //(*) return 1 + What(lst); }</pre>	<pre>public static int what(List<Integer> lst) { if (lst.isEmpty()) return 0; int x = lst.getFirst().getInfo(); lst.remove(lst.getFirst()); if(x >= 0) return what(lst); //(*) return 1 + what(lst); }</pre>

נתונה הרשימה $lst: -2 \longrightarrow -9087 \longrightarrow 16 \longrightarrow -43 \longrightarrow 5 \longrightarrow 0$

- (1) עקוב אחר הפעולה הנתונה בעבור הרשימה lst.
- (2) מה הפעולה מבצעת?
- (3) החליפו את ההוראה המסומנת ב' (*) להוראה return 0;
 - i מה תבצע הפעולה לאחר ההחלפה?
 - ii מה תכיל הרשימה lst לאחר ביצוע הפעולה?

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

שים לב: לסעיף זה שני נוסחים:
 נוסח אחד ב־ Java (עמודים 8-9),
 ונוסח אחד ב־ C# (עמודים 10-11). עבוד על פי השפה שלמדת.

לפותרים ב־ Java

לפניך הפעולה sum הכתובה במחלקה ראשית.

```
public static int sum (Stack<Integer> s)
{
    if (s.isEmpty())
        return 0;
    int x = s.pop();
    if (x%6 == 0)
        return x + sum(s);
    return sum(s);
}
```

נתונה המחסנית s .

→ ראש המחסנית	12
	4
	33
	6
	30
	0

(שים לב: המשך סעיף ב בעמוד הבא.)

/המשך בעמוד 9/

(1) עקוב אחר הפעולה sum בעבור המחסנית s, ורשום את הערך שיוחזר.

במעקב הראה את המעבר על המחסנית s.

(2) לפניך הפעולה sod.

```
public static void sod(Queue<Stack<Integer>> qq, Queue<Integer> qm)
{
    if (!qq.isEmpty())
    {
        int x = sum(qq.remove());
        qm.insert(x);
        sod(qq, qm);
    }
}
```

לפניך קטע תכנית המשתמש בפעולה sod.

סרטט במחברתך כל אחד משני התורים q1 ו-qr, לפני הקריאה לפעולה sod ואחרי ביצוע הפעולה sod.

```
public static void main(String[] args)
{
    Stack<Integer> s1 = new Stack<Integer>();
    Stack<Integer> s2 = new Stack<Integer>();

    Queue<Stack<Integer>> q1 = new Queue<Stack<Integer>>();
    s1.push(0);
    s1.push(30);
    s1.push(6);
    s1.push(33);
    s1.push(4);
    s1.push(12);
    s2.push(23);
    s2.push(36);
    s2.push(1);
    q1.insert(s1);
    q1.insert(s2);
    Queue<Integer> qr = new Queue<Integer>();
    sod(q1, qr);
}
```

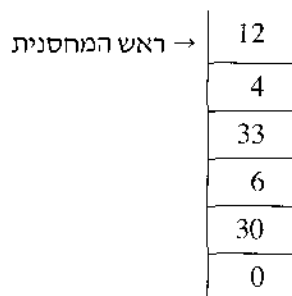
/המשך בעמוד 10/

לפותרים ב-C#

לפניך הפעולה Sum הכתובה במחלקה ראשית.

```
public static int Sum (Stack<int> s)
{
    if(s.IsEmpty())
        return 0;
    int x = s.Pop();
    if(x%6 == 0)
        return x + Sum(s);
    return Sum(s);
}
```

נתונה המחסנית s.



(1) עקוב אחר הפעולה Sum בעבוד המחסנית s, ורשום את הערך שיוחזר.

במעקב הראה את המעבר על המחסנית s.

(2) לפניך הפעולה Sod.

```
public static void Sod(Queue<Stack<int>> qq, Queue<int> qm)
{
    if (!qq.IsEmpty())
    {
        int x = Sum(qq.Remove());
        qm.Insert(x);
        Sod(qq, qm);
    }
}
```

/המשך בעמוד 11/

(שים לב: המשך תת-סעיף (2) בעמוד הבא.)

לפניך קטע תכנית המשתמש בפעולה Sod.

סרטט במחברתך את כל אחד משני התורים q1 ו־qr, לפני הקריאה לפעולה Sod

ואחרי ביצוע הפעולה Sod.

```
public static void Main(string[] args)
{
    Stack<int> s1 = new Stack<int>();
    Stack<int> s2 = new Stack<int>();

    Queue<Stack<int>> q1 = new Queue<Stack<int>>();
    s1.Push(0);
    s1.Push(30);
    s1.Push(6);
    s1.Push(33);
    s1.Push(4);
    s1.Push(12);
    s2.Push(23);
    s2.Push(36);
    s2.Push(1);
    q1.Insert(s1);
    q1.Insert(s2);
    Queue<int> qr = new Queue<int>();
    Sod(q1 , qr);
}
```

/המשך בעמוד 12/

לפותרים על פי התכנית החדשה

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

א. נתונה פעולה המקבלת רשימה lst של מספרים שלמים:

C#	Java
<pre> public static int What(Node<int> lst) { if (lst == null) return 0; int x = lst.GetValue(); lst = lst.GetNext(); if (x >= 0) return What(lst); //(*) return 1 + What(lst); } </pre>	<pre> public static int what(Node<Integer> lst) { if (lst == null) return 0; int x = lst.getValuc(); lst = lst.getNext(); if (x >= 0) return what(lst); //(*) return 1 + what(lst); } </pre>

נתונה הרשימה || → 5 → -43 → 16 → -9087 → -2 lst:

- (1) עקוב אחר הפעולה הנתונה בעבור הרשימה lst.
- (2) רשום את הערך שיוחזר, וסרטט את הרשימה lst לאחר ביצוע הפעולה. מה הפעולה מבצעת?
- (3) החליפו את ההוראה המסומנת ב- (*) להוראה return 0;
 - i מה תבצע הפעולה לאחר ההחלפה?
 - ii מה תכיל הרשימה lst לאחר ביצוע הפעולה?

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

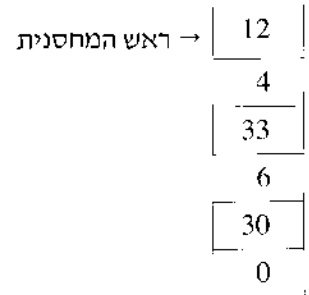
שים לב: לסעיף זה שני נוסחים:
 נוסח אחד ב־ Java (עמודים 13-14),
 ונוסח אחד ב־ C# (עמודים 15-16). עבוד על פי השפה שלמדת.

לפותרים ב־ Java

לפניך הפעולה sum הכתובה במחלקה ראשית.

```
public static int sum (Stack<Integer> s)
{
    if (s.isEmpty())
        return 0;
    int x = s.pop();
    if (x%6 == 0)
        return x + sum(s);
    return sum(s);
}
```

נתונה המחסנית s.



(שים לב: המשך סעיף ב בעמוד הבא.)

/המשך בעמוד 14/

(1) עקוב אחר הפעולה sum בעבור המחסנית s, ורשום את הערך שיוחזר.

במעקב הראה את המעבר על המחסנית s.

(2) לפניך הפעולה sod.

```
public static void sod(Queue<Stack<Integer>> qq, Qucuc<Integer> qm)
{
    if (!qq.isEmpty())
    {
        int x = sum(qq.remove());
        qm.inser(x);
        sod(qq , qm);
    }
}
```

לפניך קטע תכנית המשתמש בפעולה sod.

סרטט במחברתך כל אחד משני התורים q1 ו־ q2, לפני הקריאה לפעולה sod ואחרי ביצוע הפעולה sod.

```
public static void main(String[] args)
{
    Stack<Integer> s1 = new Stack<Integer>();
    Stack<Integer> s2 = new Stack<Integer>();

    Queue<Stack<Integer>> q1 = new Qucuc<Stack<Integer>>();
    s1.push(0);
    s1.push(30);
    s1.push(6);
    s1.push(33);
    s1.push(4);
    s1.push(12);
    s2.push(23);
    s2.push(36);
    s2.push(1);
    q1.insert(s1);
    q1.insert(s2);
    Queue<Integer> qr = new Qucuc<Integer>();
    sod(q1 , qr);
}
```

/המשך בעמוד 15/

לפותרים ב־ C#

לפניך הפעולה Sum הכתובה במחלקה ראשית.

```
public static int Sum (Stack<int> s)
{
    if(s.IsEmpty())
        return 0;
    int x = s.Pop();
    if(x%6 == 0)
        return x + Sum(s);
    return Sum(s);
}
```

נתונה המחסנית s .

ראש המחסנית →	12
	4
	33
	6
	30
	0

(1) עקוב אחר הפעולה Sum בעבור המחסנית s, ורשום את הערך שיוחזר.

במעקב הראה את המעבר על המחסנית s.

(2) לפניך הפעולה Sod .

```
public static void Sod(Queue<Stack<int>> qq, Queue<int> qm)
{
    if (!qq.IsEmpty())
    {
        int x = Sum(qq.Remove());
        qm.Insert(x);
        Sod(qq, qm);
    }
}
```

/המשך בעמוד 16/

(שים לב: המשך תת־סעיף (2) בעמוד הבא.)

לפניך קטע תכנית המשתמש בפעולה Sod .

סרטט במחברתך את כל אחד משני התורים q1 ו־q2, לפני הקריאה לפעולה Sod

ואחרי ביצוע הפעולה Sod.

```
public static void Main(string[] args)
{
    Stack<int> s1 = new Stack<int>();
    Stack<int> s2 = new Stack<int>();

    Queue<Stack<int>> q1 = new Queue<Stack<int>>();
    s1.Push(0);
    s1.Push(30);
    s1.Push(6);
    s1.Push(33);
    s1.Push(4);
    s1.Push(12);
    s2.Push(23);
    s2.Push(36);
    s2.Push(1);
    q1.Insert(s1);
    q1.Insert(s2);
    Queue<int> qr = new Queue<int>();
    Sod(q1 , qr);
}
```

/המשך בעמוד 17/

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלר, עמודים 17-23.

מבוא לחקר ביצועים, עמודים 24-33.

מודלים חישוביים, עמודים 34-37.

תכנות מונחה עצמים ב-Java, עמ' 38-45; תכנות מונחה עצמים ב-C#, עמודים 46-54.

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. לפניך קטע תכנית באסמבלר.

לפני ביצוע הקטע, תוכן התא שכתובתו 200H הוא 10100001B.

```

MOV     SI, 200H
MOV     AL, |SI|
MOV     BL, 0
MOV     CL, 8
AA:    ROR     AL, 1
      JNC     BB
      ADD     BL, 1
BB:    DEC     CL
      JNZ     AA
      MOV     SI, 200H
      MOV     [SI + 1], BL
    
```

(1) עקוב אחר ביצוע קטע התכנית בעזרת טבלת מעקב. בטבלת המעקב פרט בכל שלב את התוכן של AL, BL, ו-CL.

רשום את תוכן התא שכתובתו 201H לאחר ביצוע קטע התכנית.

(2) מה מבצע קטע התכנית?

(3) שינו את קטע התכנית באופן הזה:

במקום ההוראה השישית JNC BB, נכתבה ההוראה JC BB.

מה יהיה תוכן התא שכתובתו 201H עם סיום הביצוע של

קטע התכנית לאחר השינוי?

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

נתון קטע תכנית באסמבלר.

לפני ביצוע קטע התכנית, תוכנו של האוגר AX היה 1A2B11.

MOV CX, 8

ROR AX, CL

(1) מה יכיל האוגר AX לאחר הביצוע של קטע התכנית? רשום את תשובתך

במספרים הקסדצימליים.

(2) רשום הוראה אחת שתבצע את מה שמבצע קטע התכנית הנתון.

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

א. לפניך קטע תכנית באסמבלר.

לפני הביצוע של קטע התכנית, התוכן של האוגר AX הוא 42H.

```

MOV     CL, 2
PUSH   AX
SHL    AX, CL
POP    BX
ADD    AX, BX

```

(1) עקוב בעזרת טבלת מעקב אחר הביצוע של קטע התכנית, ורשום את התוכן של

האוגרים AX ו־ BX לאחר הביצוע של קטע התכנית.

(2) רשום הוראה אחת שתבצע את מה שמבצע קטע התכנית שלפניך. הנח שהתוכן של

האוגר DI הוא 05H.

ב. (אין קשר לסעיף א.)

במקטע הנתונים הוגדר המשתנה

```

X DB ?

```

שבו מאוחסן מספר שערכו בין 1 ל־18 (כולל).

רוצים לכתוב תכנית שתאחסן באוגר AL את מספר המספרים הדו־ספרתיים שסכום

הספרות של כל אחד מהם שווה לערך המאוחסן במשתנה X.

לדוגמה:

אם במשתנה X מאוחסן הערך 15, באוגר AL יאוחסן הערך 4, כי יש 4 מספרים

דו־ספרתיים שסכום הספרות של כל אחד מהם הוא 15. מספרים אלה הם:

69, 78, 87, 96.

(שים לב: המשך השאלה בעמוד הבא.)

קטע התכנית שלפניך, הכתוב ב־ Java וב־ C#, מבצע את הנדרש, בלי לפרק מספר דורספרתי לספרות.

```
int a = 0;
for (int i = 1 ; i <= 9 ; i++)
    for (int j = 0 ; j <= 9 ; j++)
        if (i + j == x)
            a++;
```

קטע התכנית תורגם לקטע תכנית באסמבלר.

בקטע התכנית באסמבלר הושמטו 4 הוראות המסומנות iv-i.

רשום במחברתך את מספרי ההוראות החסרות, וליד כל אחד מן המספרים כתוב באסמבלר את ההוראה החסרה כך שקטע התכנית יבצע את הנדרש.

```
MOV     AL, 0
i
-----
MOV     DH, 1
A1: MOV  BX, CX
ii
-----
MOV     DL, 0
A2: MOV  AH, DH
iii
-----
CMP     AH, X
JNE     A3
INC     AL
A3: INC  DL
LOOP   A2
iv
-----
INC     DH
LOOP   A1
```

/המשך בעמוד 21/

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

א. השגרה (פרוצדורה) CHECK בודקת אם מילה (WORD), המועברת כפרמטר דרך המחסנית, מהווה שני מספרים עוקבים כך שהבית הנמוך של המילה קטן ב-1 מהבית הגבוה של המילה. למשל: 1, 2, או 203, 204. אם כן, השגרה מאחסנת 1 באוגר AL, אחרת – היא מאחסנת בו 0. לאחר מכן השגרה חוזרת למקום שממנו נקראה. לפניך ארבעה קטעי תכנית, i-iv, הכתובים באסמבלר. עקוב בעזרת טבלת מעקב אחר כל אחד מן הקטעים, וקבע אם הוא מבצע את הנדרש או אינו מבצע את הנדרש.

- i CHECK: POP CX
 POP BX
 XOR AL , AL
 ADD BH , 1
 CMP BH , BL
 JZ A1
 INC AL
 A1: PUSH CX
 RET
- ii CHECK: POP BX
 POP CX
 MOV AL , 0
 DEC CL
 SUB CH , CL
 JNZ A1
 INC AL
 A1: PUSH BX
 RET

(שים לב: המשך סעיף א וסעיף ב בעמוד הבא.)

```

iii  CHECK: POP  CX
      POP      BX
      MOV     AL, 0
      SUB     CL, 1
      CMP     BH, BL
      JNE    AL
      MOV     AL, 1
AI:   PUSH   CX
      RET
    
```

```

iv   CHECK: POP  BX
      POP      CX
      MOV     AL, 1
      ADD     CL, 1
      CMP     CH, CL
      JE     AL
      DEC     AL
AI:   PUSH   BX
      RET
    
```

ב. (אין קשר לסעיף א.)

לפניך קטע תכנית הכתוב ב־ Java וב־ C#. א ר־ y הם משתנים מטיפוס שלם.

```

while ((x > 0) && (y > 0))
{
    if(x > 10)
        x--;
    else
        y--;
}
    
```

כתוב באסמבלר קטע תכנית שיבצע את קטע התכנית הכתוב ב־ Java וב־ C#.

הנח שהמשתנה x מאוחסן באוגר BX, והמשתנה y מאוחסן באוגר CX.

/המשך בעמוד 23/

8. מערך חד-ממדי ARR בגודל N של מספרים שלמים – כאשר N זוגי – ייקרא **מאוזן** אם: הסכום של הערך המאוחסן בתא הראשון של המערך ושל הערך המאוחסן בתא האחרון של המערך שווה לסכום של הערך המאוחסן בתא השני של המערך ושל הערך המאוחסן בתא שלפני האחרון במערך, ושווה לסכום של הערך המאוחסן בתא השלישי של המערך ושל הערך המאוחסן בתא השלישי מסוף המערך, וכך הלאה. וסכום זה שווה לערך המאוחסן במשתנה x. **לדוגמה:**

בעבור מערך ARR בגודל 6 ומשתנה x שהערך המאוחסן בו הוא 10,

	0	1	2	3	4	5
ARR	3	1	4	6	9	7

המערך ARR **מאוזן** כי מתקיים: $3 + 7 = 1 + 9 = 4 + 6 = 10$.

א. כתוב באסמבלר שגרה (פרוצדורה) בשם CHECK, שתקבל כפרמטרים שלוש כתובות של תאים בזיכרון. השגרה תבדוק אם סכום הערכים המאוחסנים בתאים שבשתי הכתובות הראשונות שווה לערך המאוחסן בתא שבכתובת השלישית. אם כן – השגרה תאחסן I באוגר DI, אחרת – היא תאחסן בו 0.

ב. במקטע הנתונים מוגדרים מערך ARR ומשתנה x:

```
ARR    DB 100    DUP (?)
X      DB ?
```

כתוב באסמבלר קטע תכנית שיבדוק אם המערך ARR הוא **מאוזן**.

אם כן – קטע התכנית יאחסן I באוגר DH, אחרת – הוא יאחסן בו 0.

הנח שסכום הערכים של שני תאים אינו עולה על 255.

עליך להשתמש בשגרה CHECK שכתבת בסעיף א.

מבוא לחקר ביצועים

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

9. בשאלה זו שלושה סעיפים, א-ג, שאין קשר ביניהם. ענה על שלושת הסעיפים.
 א. נתונה בעיית תכנון ליניארי, ונתונים האילוצים:

$$2x_1 - x_2 \leq 6$$

$$x_2 \leq 4$$

$$x_1 \geq 4$$

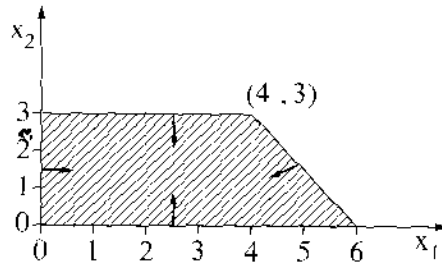
$$x_2 \geq 0$$

סרטט את תחום הפתרונות האפשריים הנקבע על סמך האילוצים הנתונים.

סמן את תחום הפתרונות האפשריים של האילוצים,

ורשום בגרף שמתקבל את הערכים של x_1 ו- x_2 בעבור כל אחד מן הקדקודים של תחום הפתרונות האפשריים.

- ב. נתון סרטוט של תחום הפתרונות האפשריים של בעיית תכנון ליניארי מסוימת.



(1) נתונה פונקציית המטרה: $\text{Max } \{Z = 3x_1 + 2x_2\}$.

אם לבעיה הנתונה יש אינסוף פתרונות בתחום הפתרונות האפשריים הנתון, רשום את הפתרון הכללי, ואת ערכה של פונקציית המטרה בפתרון האופטימלי.

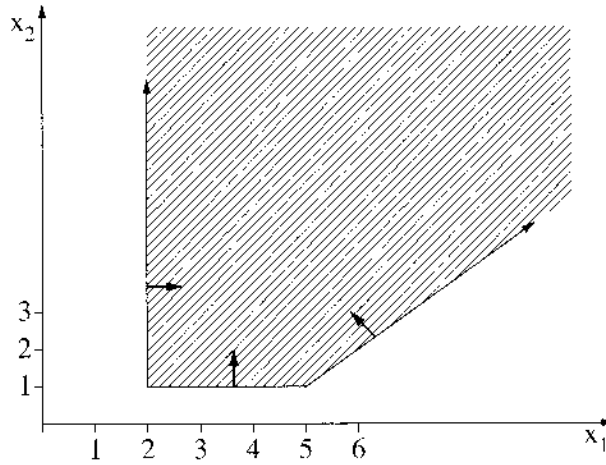
אחרת – רשום את הפתרון היחיד, ואת ערכה של פונקציית המטרה בפתרון זה.

(2) נתונה פונקציית המטרה: $\text{Max } \{Z = kx_1 + x_2\}$ ו- $k > 0$.

קבע בעבור אילו ערכים של k יהיו לבעיית התכנון הליניארי הנתונה בתחום הפתרונות האפשריים הנתון אינסוף פתרונות אופטימליים. כתוב את הערך של פונקציית המטרה בעבור כל אחד מערכי ה- k האלה. נמק את קביעותיך.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. נתון סרטוט של תחום הפתרונות האפשריים של בעיית תכנון לינארי, שאחד האילוצים שלה הוא: $x_1 - x_2 \leq 4$.



- (1) האם בתחום הפתרונות האפשריים הנתון יש פתרון אופטימלי לפונקציית המטרה $\text{Max}\{Z = x_1 - x_2\}$? נמק את תשובתך.
- (2) האם בתחום הפתרונות האפשריים הנתון יש פתרון אופטימלי לפונקציית המטרה $\text{Min}\{Z = x_1 - x_2\}$? נמק את תשובתך.

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

א. $G = (V, E)$ הוא גרף לא מכוון המיוצג על ידי מטריצת הסמיכויות שלפניך:

	a	b	c	d	e
a	0	1	0	1	0
b	1	0	1	0	1
c	0	1	0	1	0
d	1	0	1	0	1
e	0	1	0	1	0

- (1) סרטט את הגרף G המיוצג על ידי המטריצה הנתונה.
- (2) האם הגרף G הוא גרף שלם? נמק את תשובתך.
- (3) כמה רכיבי קשירות (Connected Components) יש בגרף הנתון, ומה הם?
- (4) מצא שני מסלולים קצרים ביותר מקדקוד a לקדקוד e .
- תאר כל אחד מן המסלולים תיאור סכמתי.
- (5) מצא בגרף G מעגל המתחיל בקדקוד a ומכיל 4 קשתות בלבד.
- תאר מעגל זה תיאור סכמתי.

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

$G = (V, E)$ הוא גרף מכוון המיוצג על ידי רשימת הסמיכויות שלפניך:

a	→ c → d →
b	→ a >
c	→ d →
d	> c >
e	→ b →

(1) סרטט את הגרף G המיוצג על ידי הרשימה הנתונה.

(2) כמה רכיבי קשירות חזקה (רק"חים – Strong Connected Components) יש בגרף G ,

ומה הם?

(3) i. העתק למחברתך את הטבלה שלפניך.

הפעל אלגוריתם סריקה לעומק (DFS) על הגרף G , החל מקדקוד a .

בהפעלת האלגוריתם DFS חשב בעבור כל קדקוד u את $d[u]$ ואת $f[u]$.

$d[u]$ מציין את מועד הגילוי של הקדקוד u בעת הסריקה,

ו- $f[u]$ מציין את המועד של סיום הטיפול בקדקוד u .

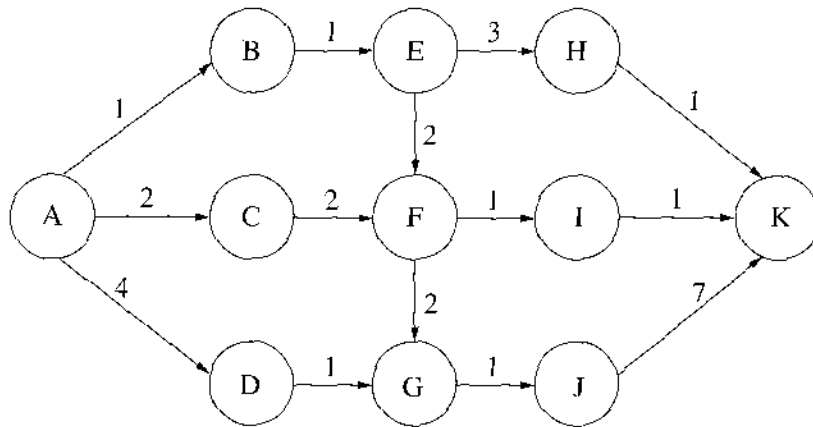
כתוב את תוצאות החישובים בטבלה שבמחברתך.

קדקוד u	a	b	c	d	e
$d[u]$					
$f[u]$					

ii. סרטט במחברתך רק את העץ הפורש (DFS)/היער הפורש (DFS) שמתקבל.

התבסס על הייצוג הנתון ברשימת הסמיכויות.

11. א. לפניך רשת $G = (V, E)$.



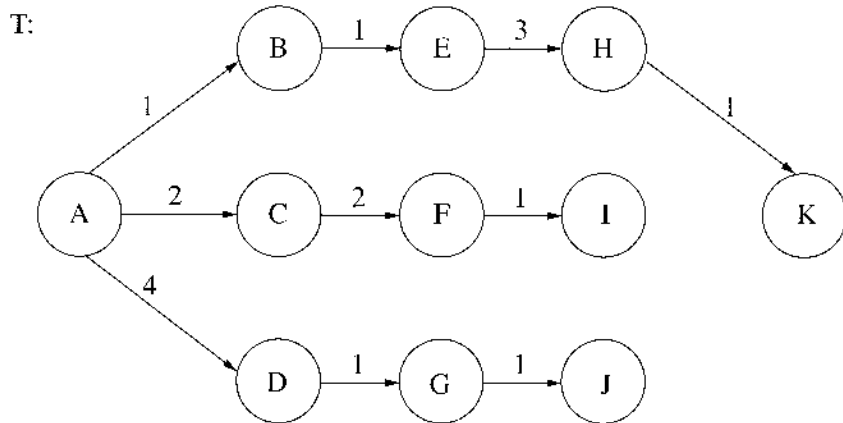
מצא את כל המסלולים הקצרים ביותר מקדקוד A לקדקוד K ברשת הנתונה.
תאר כל אחד מן המסלולים תיאור סכמתי.

ב. נוסף לרשת G , נתון עץ פורש מכון T שמחבר את כל קדקודי הרשת ואינו מכיל מעגלים.
לפניך אלגוריתם:

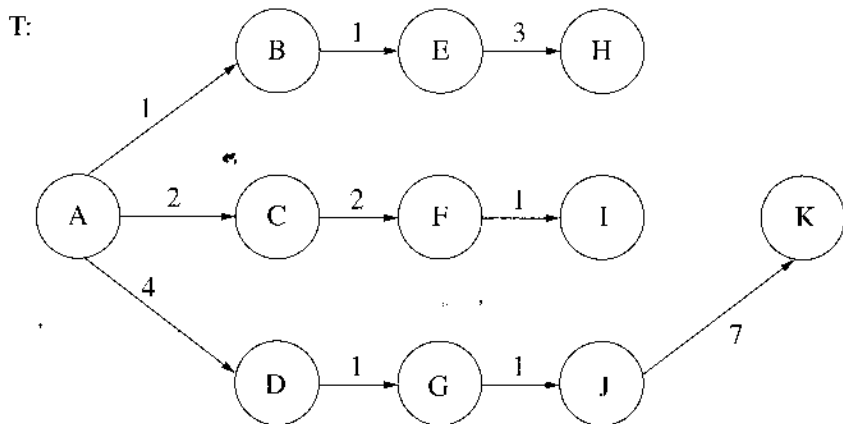
- צעד 1: הפעל את האלגוריתם של דיקסטר מקדקוד נתון בעץ T .
צעד 2: בעבור כל קשת מכוונת $e = (u, v)$, השייכת ל- E ברשת G ואינה שייכת לעץ T , בצע:
 אם $d[v] > d[u] + w(e)$ – החזר "שקר".
 $d[v]$ מציינ את אורך המסלול הקצר ביותר מקדקוד מקור לקדקוד v .
צעד 3: החזר "אמת".

(שים לב: המשך סעיף ב' בעמוד הבא.)

- (1) מהו הערך המוחזר לאחר הפעלת האלגוריתם על הרשת הנתונה G ועל העץ T שלפניך, החל מקדקוד A? כתוב את כל הצעדים של האלגוריתם ואת התוצאה המתקבלת מכל צעד.



- (2) מהו הערך המוחזר לאחר הפעלת האלגוריתם על הרשת הנתונה G ועל העץ T שלפניך, החל מקדקוד A? כתוב את כל הצעדים של האלגוריתם ואת התוצאה המתקבלת מכל צעד.



12. בשאלה זו חמישה סעיפים, א-ה, שאין קשר ביניהם. ענה על כל הסעיפים.
 א. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית התובלה: $x_{22} = 0$.

מקורות	יעדים			היצע
	1	2	3	
1	10	0	20	20
2	12	7 0	9	120
3	0	14	16	60
ביקוש	20	0	180	

- (1) העתק למחברתך את הטבלה, והשלם בה את הפתרון הבסיסי האפשרי לפי שיטת הפינה הצפונית-מערבית.
 (2) משניים את המחיר בתא (3, 1) מ-0 ל-15. האם ישתנה הפתרון הבסיסי האפשרי שמצאת בתת-סעיף (1)? נמק.

ב. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתון הערך של u_1 .

מקורות	יעדים			היצע	u_i
	1	2	3		
1	5	8 5	10 0	5	0
2	5 8	20	12	8	
3	1 4	13	14 12	16	
ביקוש	12	5	12		
v_j					

- (1) העתק למחברתך את הטבלה, והשלם בה את הערכים של u_2, u_3, v_1, v_2, v_3 .

(2) הסבר מדוע הפתרון הנתון אינו פתרון אופטימלי.

(שים לב: המשך השאלה בעמוד הבא.)

ג. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה, ונתונים הערכים של $u_1, u_2, u_3, v_1, v_2, v_3, v_4$. שמתאימים לפתרון זה.

מקורות	יעדים				היצע	u_i
	1	2	3	4		
1	8	9	4	6	280	0
2	4	12	13	8	160	-10
3	15	10	40	12	270	1
ביקוש	200	320	160	30		
v_j	14	9	4	6		

העתק למחברתך את הטבלה, והשלם בה את הפתרון, בהתחשב בערכים של $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ כך שיתקבל פתרון בסיסי אפשרי.

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 32/

ד. בטבלה שלפניך נתון פתרון לא אופטימלי שהתקבל לאחר k איטרציות בעבור בעיית תובלה.

מסוימת, ונתונים הערכים של $u_1, u_2, u_3, v_1, v_2, v_3$.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	22	15	17	200	0
		100	100		
2	10	18	4	100	-15
	100				
3	20	10	8	150	-5
	100	50			
ביקוש	200	150	100		
v_j	25	15	17		

עליך לבצע איטרציה נוספת, כלומר איטרציה $k + 1$.

- (1) מהו המשתנה שיוצא מן הבסיס באיטרציה זו?
- (2) סרטט במחברתך טבלה חדשה, ורשום בה את הפתרון שיתקבל לאחר איטרציה זו.

(שים לב: סעיף ה של השאלה בעמוד הבא.)

ה. לחברה בין-לאומית גדולה יש 3 מפעלים A, B, C, המסוגלים לייצר 120 מיליון, 80 מיליון, 50 מיליון פרוסות סיליקון בשנה בהתאמה. כדי לצרוב שבבים מפרוסות הסיליקון מועברות הפרוסות ל-3 מפעלים אחרים 1, 2, 3. כושר העיבוד של מפעלים אלה הוא 150 מיליון, 80 מיליון ו-70 מיליון פרוסות סיליקון בשנה בהתאמה. מפעל 1 חייב לקבל את כל הכמות המתאימה לכושר העיבוד שלו. הבעיה הנתונה אינה "מאוזנת", לכן הציע סטודנט להוסיף "מקור דמיוני" D אשר יספק 50 מיליון פרוסות סיליקון בשנה. העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

מקורות	יעדים			היצע
	1	2	3	
A	25	18	18	120
B	34	15	32	80
C	58	24	18	50
D	x	y	z	50
ביקוש	150	80	70	

ערכי העלויות של המקור D נתונים בטבלה כפרמטרים x, y, z .
 מה צריך להיות הערך של כל אחד מן הפרמטרים x, y, z כדי שיהיה אפשר למצוא פתרון בסיסי אפשרי לבעיית התובלה הנתונה?
 בחר בתשובה הנכונה מארבע התשובות שלפניך, העתק אותה למחברתך, ונמק את בחירתך.
 (M מוגדר מספר גדול מאוד.)

1. $x = 0, y = 0, z = 0$

2. $x = M, y = M, z = M$

3. $x = M, y = 0, z = 0$

4. $x = 0, y = M, z = M$

מודלים חישוביים

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

13. בנה מכונת טיורינג שעל סרט הזיכרון שלה כתובה מילה w מעל הא"ב $\{a\}$.

w כתובה מתחילת הסרט.

המכונה תכתוב על הסרט מתחילתו את המילה $w^?w$.

אם w היא המילה הריקה, המכונה תכתוב על הסרט מתחילתו את המילה $?$.

לדוגמה:

לפני פעולת המכונה הסרט נראה כך:



לאחר פעולת המכונה ייראה הסרט כך:



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

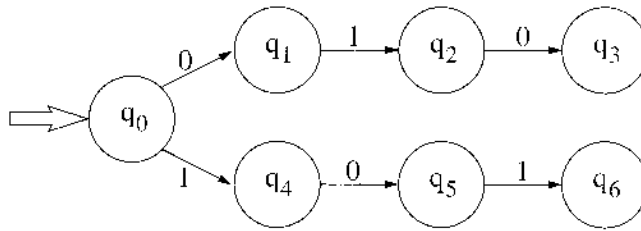
א. לפניך השפה L מעל הא"ב $\{0, 1\}$. L היא אוסף המילים שבהן רצף התווים 01 מופיע מספר פעמים זהה למספר הפעמים שמופיע רצף התווים 10, דוגמה למילים השייכות לשפה L :

111, 0, 010, 0111010, 1110001000110101

דוגמה למילים שאינן שייכות לשפה L :

011, 0101, 100

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L . בסרטוט חסרים מעברים, סימני קלט ומצבים מקבלים.



הסרטוט כולל את כל המצבים של האוטומט.

העתק למחברתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L . עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים, ולסמן את כל המצבים המקבלים.

שים לב: אין להוסיף לאוטומט מצבים, ואין להוריד ממנו מצבים.

ב. (אין קשר לסעיף א.)

(1) הבא דוגמה לשלוש שפות L_1, L_2, L_3 כך ש: $L_1 \cap L_2 = L_3$ הן שפות רגולריות.

ר L_3 היא שפה לא רגולרית, ומתקיים $L_3 \cup L_2 = L_1$.

(2) הבא דוגמה לשלוש שפות L_1, L_2, L_3 כך ש: $L_1 \cap L_2 = L_3$ הן שפות רגולריות.

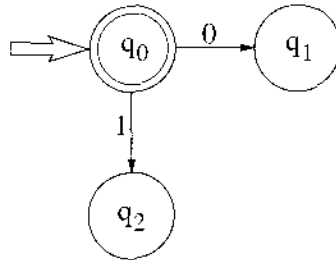
ר L_3 היא שפה לא רגולרית, ומתקיים $L_3 \cap L_2 = L_1$.

15. בשאלה זו שלושה סעיפים, א-ג, שאין קשר ביניהם. ענה על שלושתם.

א. לפניך השפה L מעל הא"ב $\{0, 1\}$:

$$L = \left\{ w \mid \begin{array}{l} \text{שארית החלוקה ב-3 של מספר ה-0ים ב-} w \\ \text{שווה לשארית החלוקה ב-3 של מספר ה-1ים ב-} w \end{array} \right\}$$

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L . בסרטוט חסרים מעברים וסימני קלט.



הסרטוט מכיל את כל המצבים של האוטומט ואת כל המצבים המקבלים. העתק למחברתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L . עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים. שים לב: אין להוסיף לאוטומט מצבים ואין להוריד ממנו מצבים.

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

$$L_1 = \{0^n \cdot x \mid \{0, 1\} \text{ מעל הא"ב } x, 1 \leq |x| \leq n\}$$

$$L_2 = \{0^m \mid m \geq 1\}$$

נתונות שתי מילים ב- L_2 : $w_1 = 0^i$ ו- $w_2 = 0^j$ כך ש: $j < i$.

מצא מילה w כך שיתקיים:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. לפניך השפות L_1 ו- L_2 מעל הא"ב $\{a,b,c\}$:

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

$$L_2 = \{a^s \mid s \geq 1\}$$

נתונות שתי מילים ב- L_2 : $w_1 = a^i$ ו- $w_2 = a^j$ כך ש: $j \neq i$.

מצא מילה w כך שיתקיים:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

16. לפניך השפה L מעל הא"ב $\{a,b,c\}$.

$$L = \{a^i b^j c^j \mid j \geq 1, i - j \geq 0\}$$

א. כתוב את המילה הקצרה ביותר בשפה L .

ב. בנה אוטומט מחסנית שיקבל את השפה L .

תכנות מונחה עצמים

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

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

א. לפניך שתי מחלקות A, B :

```
public class A
{
    public A() {
        System.out.println( "A" );
    }
    public void one( ) {
        System.out.println( "one of A" );
    }
    public void two( ) {
        one( );
    }
}
public class B extends A
{
    public B() {
        System.out.println( "B" );
    }
    public void one( ) {
        System.out.println( "one of B" );
    }
    public void two( ) {
        System.out.println( "two of B" );
    }
    public void three( ) {
        super.two( );
    }
}
```

(שים לב: המשך סעיף א בעמוד הבא.)

לפניך שבעה קטעי תכנית, i-vii. לכל אחד מהם קבע אם הוא תקין או אינו תקין.
אם הקטע תקין – כתוב מהו הפלט שיתקבל מהרצת קטע התכנית.
אם הקטע אינו תקין – נמק את קביעתך, וקבע אם יש שגיאת הידור (קומפילציה)
או שגיאת זמן ריצה.

i A a1 = new A();
a1.two();
(A a1).two();

ii A a2 = new A();
(B a2).two();

iii A a3 = new B();
a3.two();
(B a3).three();

iv B b1 = new A();
b1.two();

v B b2 = new B();
b2.two();
(A b2).two();

vi B b3 = new B();
(A b3).three();

vii B b5 = new B();
(B b5).three();

(שים לב: סעיף ב של השאלה בעמוד הבא.)

/המשך בעמוד 40/

ב. (אין קשר לסעיף א.)

לפניך חתימה של המחלקות AAA, BBB ותכונותיהן:

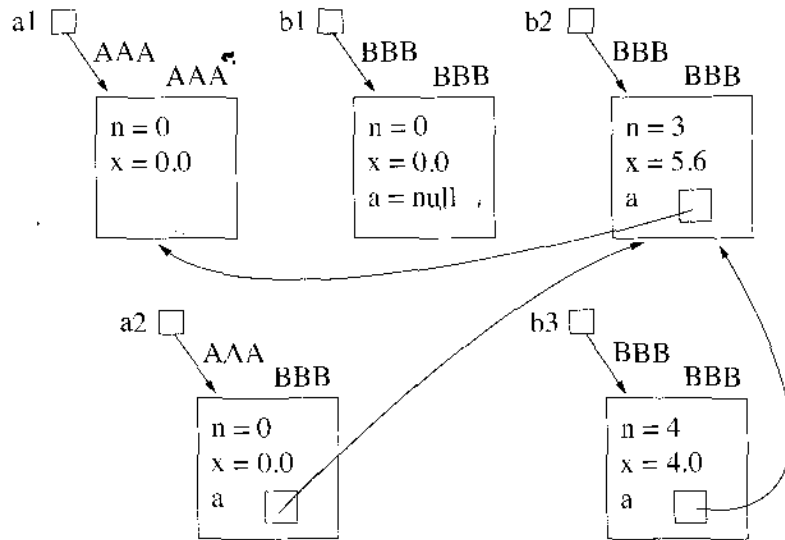
```
public class AAA
{
    private int n;
    private double x;
}

public class BBB extends AAA
{
    private AAA a;
}
```

לפניך המחלקה Test, ובה פעולה ראשית:

```
public class Test
{
    public static void main (String[] args )
    {
        AAA a1 = new AAA( );
        BBB b1 = new BBB( );
        BBB b2 = new BBB(3 ,5.6 ,a1);
        AAA a2 = new BBB(b2);
        BBB b3 = new BBB(4 ,b2);
    }
}
```

הרצת הפעולה הראשית main יצרה את העצמים האלה:



ממש ב־ Java במחלקות AAA, BBB את הפעולות הבונות הנדרשות להרצת הפעולה main כך שיתקבלו העצמים המתוארים.

/המשך בעמוד 41/

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

א. לפניך המחלקות C, D, UseCD. רשום את הפלט של הפעולה הראשית.

```
public class C {
    public C() {
        System.out.println("C");
    }
    public C(int n) {
        for (int i = 0; i < n; i++)
            new C();
        System.out.println("CC");
    }
}
public class D extends C {
    public D() {
        super();
        System.out.println("D");
    }
    public D(int n) {
        super(n);
        for (int i = 0; i < n; i++)
            new D();
    }
}
public class UseCD {
    public static void main(String[] args) {
        System.out.println("***");
        C c1 = new C(3);
        System.out.println("---");
        D d1 = new D(1);
        System.out.println("###");
        D d2 = new D(4);
    }
}
```

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך המחלקות AA, BB, ופעולה ראשית במחלקה Run. עקוב אחר הפעולה הראשית, ורשום את הפלט המתקבל. במעקב יש לכתוב את ערכי המשתנים, את ערכי תכונות המחלקה (תכונות סטטיות), ובעבור כל עצם יש לכתוב את ערכי התכונות שלו.

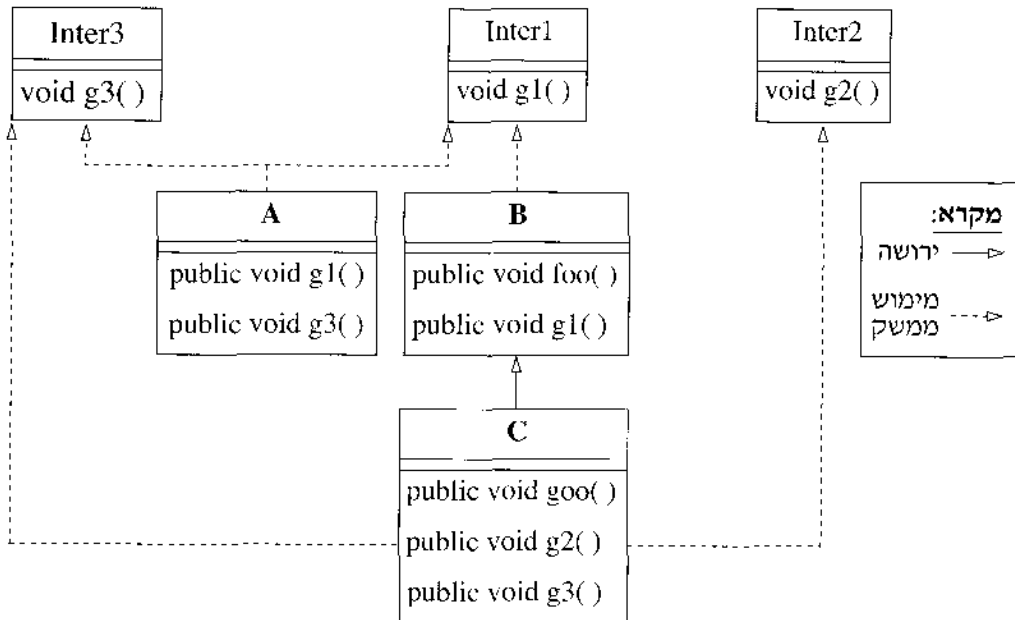
```
public class AA
{
    private int num1 = 0;
    private int num2 = 0;
    private static int count = 0;
    public AA(int num)
    {
        num1 = num;
        num2 = num;
        count++;
        System.out.println("AA ctor1");
    }
    public AA(int num1, int num2)
    {
        this.num1 = num1;
        this.num2 = num2;
        count++;
        System.out.println("AA ctor2");
    }
    public int sum()
    {
        return num1 + num2;
    }
    public static int getCount()
    {
        return count;
    }
}
```

```
public class BB extends AA
{
    private int num3 = 0;
    public BB(int num)
    {
        super (num);
        num3 = num;
        System.out.println("BB ctor1");
    }
    public BB(int num1, int num2, int num3)
    {
        super (num1, num2);
        this.num3 = num3;
        System.out.println("BB ctor2");
    }
    public int sum()
    {
        return super.sum() + num3;
    }
    public void setNum3(int num)
    {
        num3 = num;
    }
}
```

(שים לב: המשך סעיף ב בעמוד הבא.)

```
public class Run
{
    public static void main(String[] args)
    {
        AA f1 = new AA(10);
        System.out.println("sum1 = " + f1.sum());
        System.out.println("count = " + AA.getCount());
        AA f2 = new AA(10, 20);
        System.out.println("count = " + AA.getCount());
        BB s1 = new BB(1);
        AA f3 = new BB(2);
        System.out.println("count = " + AA.getCount());
        f2 = s1;
        System.out.println("sum = " + f2.sum());
        s1.setNum3(2);
        System.out.println("sum = " + s1.sum());
        System.out.println("sum = " + f2.sum());
    }
}
```

19. לפניך תרשים UML חלקי.



א. השלם את שלוש השורות החסרות כך שקטע התכנית שלפניך יעבוד בלי שגיאות הידור (קומפילציה) ו/או זמן ריצה. העתק למחברתך את השורות שהשלמת.

```

_____ x = new A( ) ;
_____ y = new B( ) ;
_____ z = new C( ) ;

x = y ;
y = new C( ) ;
y.foo( ) ;
z.g3( ) ;
( (C) z ). goo( ) ;
z = new A( ) ;
    
```

ב. כתוב ב־ Java פעולה שתקבל מערך חד־ממדי מטיפוס Inter1.

הפעולה תעבור על איברי המערך, ותקרא לפעולה g3() אם היא קיימת בעבור העצם. אם הפעולה g3() אינה קיימת – הפעולה שכתבת תקרא לפעולה g1().

20. חברה מסוימת מייצרת שלושה סוגים של טלפונים סלולריים: type1, type2, type3.

בטלפון מסוג type1 אפשר לבצע את הפעולות האלה:

— חיוג — dial

— קבלת שיחה — recCall

— שליחת הודעה — sendMsg

— קבלת הודעה — recMsg

בטלפון מסוג type2 אפשר לבצע את הפעולות האלה:

— חיוג — dial

— קבלת שיחה — recCall

— שליחת הודעה — sendMsg

— קבלת הודעה — recMsg

— צילום תמונה — takePic

— הצגת תמונה — showPic

בטלפון מסוג type3 אפשר לבצע את הפעולות האלה:

— חיוג — dial

— קבלת שיחה — recCall

— צילום תמונה — takePic

— הצגת תמונה — showPic

— התקנת אפליקציות — installApp

בכל אחד מסוגי הטלפונים:

נשמר מספר דקות השיחה שנעשו מן הטלפון.

אפשר לאפס את מספר דקות השיחה — reset.

אפשר להחזיר חשבון טלפון עדכני — bill, המחושב כך: מספר דקות השיחה $\times 1$ ש"ח.

בטלפונים שיש בהם אפשרות לשלוח הודעות, החזרת חשבון הטלפון העדכני היא:

מספר דקות השיחה $\times 1$ ש"ח + מספר ההודעות שנשלחו $\times 0.5$ ש"ח.

בטלפונים האלה פעולת האיפוס מאפסת את מספר דקות השיחה וכן את מספר ההודעות

שנשלחו.

א. בעבור הטלפונים הסלולריים המתוארים למעלה, סרטט את היררכיית המחלקות

והממשקים, באופן המתאים ביותר לעקרונות של תכונות מונחה עצמים (הכמסה —

encapsulation, הורשה — inheritance, פולימורפיזם — polymorphism).

סמן בסרטוט את הקשרים בין המחלקות, ואת הקשרים בין המחלקות ובין הממשקים.

אין צורך לכלול בסרטוט תכונות ופעולות.

ב. לכל אחת מן המחלקות שבסרטוטך כתוב ב־ Java את כותרת המחלקה ואת התכונות שלה,

וכותרות לכל הפעולות.

לכל אחד מן הממשקים שבסרטוטך כתוב ב־ Java את הכותרת שלו, וכותרות לכל הפעולות.

אינך צריך לכתוב את הפעולות get ו־ set ופעולות בונות.

ג. ממש ב־ Java את הפעולה המחזירה את החשבון העדכני בכל אחת מן המחלקות שהיא

מופיעה בהן.

תכנות מונחה עצמים

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

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

א. לפניך שתי מחלקות, A, B :

```
public class A
{
    public A() {
        Console.WriteLine("A");
    }
    public virtual void One() {
        Console.WriteLine("One of A");
    }
    public virtual void Two() {
        One();
    }
}
public class B : A
{
    public B() {
        Console.WriteLine("B");
    }
    public override void One() {
        Console.WriteLine("One of B");
    }
    public override void Two() {
        Console.WriteLine("Two of B");
    }
    public void Three() {
        base.Two();
    }
}
```

(שים לב: המשך סעיף א בעמוד הבא.)

לפניך שבעה קטעי תכנית, vii-i. לכל אחד מהם קבע אם הוא תקין או אינו תקין.
אם הקטע תקין – כתוב מה הפלט שיתקבל מהרצת קטע התכנית.
אם הקטע אינו תקין – נמק את קביעתך, וכתוב אם יש שגיאת הידור (קומפילציה)
או שגיאת זמן ריצה.

- i A a1 = new A() ;
 a1 . Two() ;
 ((A) a1) . Two() ;

- ii A a2 = new A() ;
 ((B) a2) . Two() ;

- iii A a3 = new B() ;
 a3 . Two() ;
 ((B) a3) . Threc() ;

- iv B b1 = new A() ;
 b1 . Two() ;

- v B b2 = new B() ;
 b2 . Two() ;
 ((A) b2) . Two() ;

- vi B b3 = new B() ;
 ((A) b3) . Three() ;

- vii B b5 = new B() ;
 ((B) b5) . Three() ;

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך חתימה של המחלקות AAA, BBB ותכונותיהן:

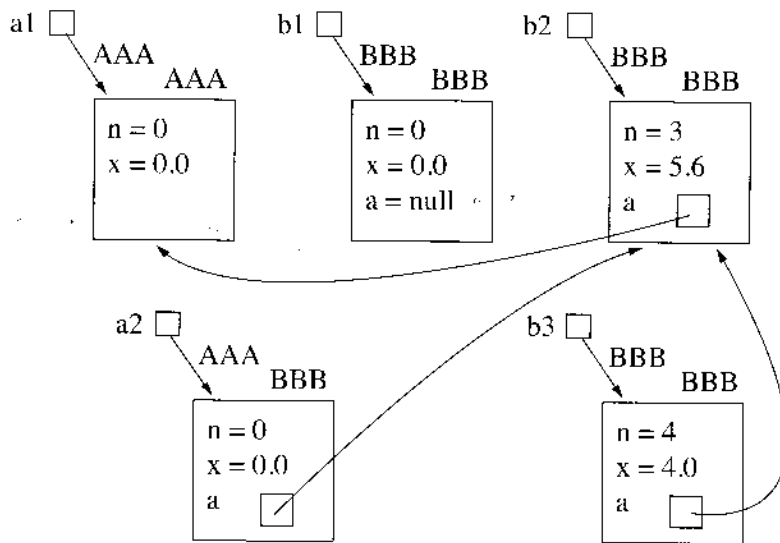
```
public class AAA
{
    private int n ;
    private double x ;
}

public class BBB : AAA
{
    private AAA a;
}
```

לפניך המחלקה Test, ובה פעולה ראשית:

```
public class Test
{
    public static void Main()
    {
        AAA a1 = new AAA();
        BBB b1 = new BBB();
        BBB b2 = new BBB(3, 5.6, a1);
        AAA a2 = new BBB(b2);
        BBB b3 = new BBB(4, b2);
    }
}
```

הרצת הפעולה הראשית Main יצרה את העצמים האלה:



ממש ב- C# במחלקות AAA, BBB את הפעולות הבונות הנדרשות להרצת הפעולה Main כך שיתקבלו העצמים המתוארים.

/המשך בעמוד 49/

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

א. לפניך המחלקות UseCD, D, C. רשום את הפלט של הפעולה הראשית.

```
public class C {
    public C() {
        Console.WriteLine("C");
    }
    public C(int n) {
        for (int i = 0; i < n; i++)
            new C();
        Console.WriteLine("CC");
    }
}

public class D : C {
    public D() : base() {
        Console.WriteLine("D");
    }
    public D(int n) : base(n) {
        for (int i = 0; i < n; i++)
            new D();
    }
}

public class UseCD {
    public static void Main() {
        Console.WriteLine("****");
        C c1 = new C(3);
        Console.WriteLine("---");
        D d1 = new D(1);
        Console.WriteLine("###");
        D d2 = new D(4);
    }
}
```

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ג. (אין קשר לסעיף א.)

לפניך המחלקות AA, BB, ופעולה ראשית במחלקה Run.
עקוב אחר הפעולה הראשית, ורשום את הפלט המתקבל.

במעקב יש לכתוב את ערכי המשתנים, את ערכי תכונות המחלקה (תכונות סטטיות),
ובעבור כל עצם יש לכתוב את ערכי התכונות שלו.

```

public class AA
{
    private int num1 = 0;
    private int num2 = 0;
    private static int count = 0;
    public AA(int num)
    {
        num1 = num;
        num2 = num;
        count++;
        Console.WriteLine("AA ctor1");
    }
    public AA(int num1, int num2)
    {
        this.num1 = num1;
        this.num2 = num2;
        count++;
        Console.WriteLine("AA ctor2");
    }
    public virtual int Sum()
    {
        return num1 + num2;
    }
    public static int GetCount()
    {
        return count;
    }
}

public class BB : AA
{
    private int num3 = 0;
    public BB(int num)
        :base (num)
    {
        num3 = num;
        Console.WriteLine("BB ctor1");
    }
    public BB (int num1, int num2, int num3)
        :base(num1, num2)
    {
        this.num3 = num3;
        Console.WriteLine("BB ctor2");
    }
    public override int Sum()
    {
        return base.Sum() + num3;
    }
    public void SetNum3(int num)
    {
        num3 = num;
    }
}

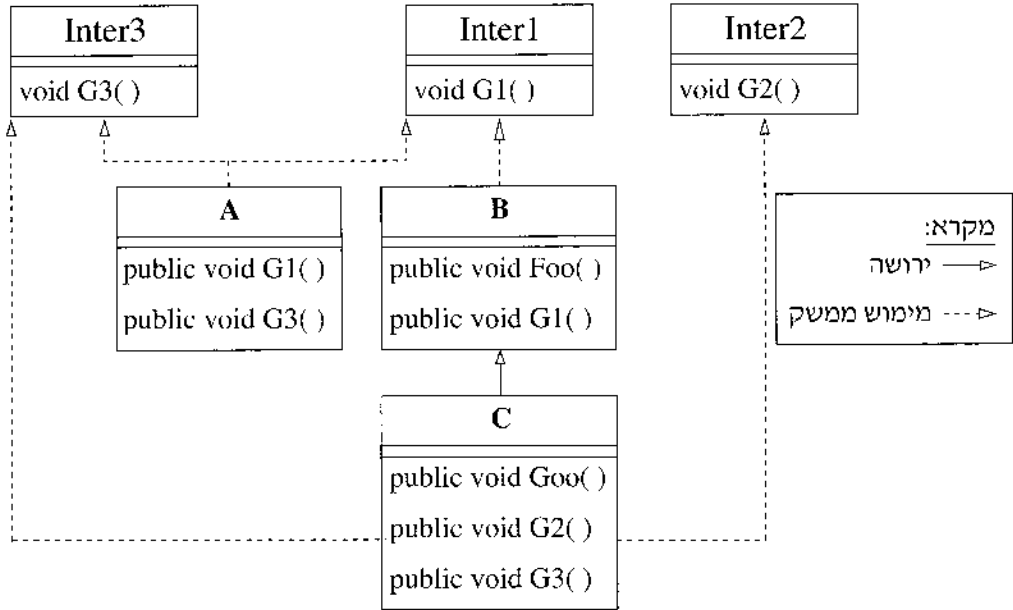
```

(שים לב: המשך סעיף ב בעמוד הבא.)

/המשך בעמוד 51/

```
public class Run
{
    public static void Main()
    {
        AA f1 = new AA(10);
        Console.WriteLine("sum1 = " + f1.Sum());
        Console.WriteLine("count = " + AA.GetCount());
        AA f2 = new AA(10, 20);
        Console.WriteLine("count = " + AA.GetCount());
        BB s1 = new BB(1);
        AA f3 = new BB(2);
        Console.WriteLine("count = " + AA.GetCount());
        f2 = s1;
        Console.WriteLine("sum = " + f2.Sum());
        s1.SetNum3(2);
        Console.WriteLine("sum = " + s1.Sum());
        Console.WriteLine("sum = " + f2.Sum());
    }
}
```

23. לפניך תרשים UML חלקי.



- א. השלם את שלוש השורות החסרות כך שקטע התכנית שלפניך יעבוד בלי שגיאות הידור (קומפילציה) ו/או זמן ריצה.
העתק למחברתך את השורות שהשלמת.

```

_____ x = new A() ;
_____ y = new B() ;
_____ z = new C() ;
x = y ;
y = new C() ;
y.Foo() ;
z.G3() ;
((C) z).Goo() ;
z = new A() ;

```

- ב. כתוב ב-C# פעולה שתקבל מערך חדר-מדי מטיפוס Inter1. הפעולה תעבור על איברי המערך, ותקרא לפעולה G3() אם היא קיימת בעבור העצם. אם הפעולה G3() אינה קיימת – הפעולה שכתבת תקרא לפעולה G1().

/המשך בעמוד 53/

24. חברה מסוימת מייצרת שלושה סוגים של טלפונים סלולריים: type1, type2, type3.

בטלפון מסוג type1 אפשר לבצע את הפעולות האלה:

– חיוג – Dial

– קבלת שיחה – RecCall

– שליחת הודעה – SendMsg

– קבלת הודעה – RecMsg

בטלפון מסוג type2 אפשר לבצע את הפעולות האלה:

– חיוג – Dial

– קבלת שיחה – RecCall

– שליחת הודעה – SendMsg

– קבלת הודעה – RecMsg

– צילום תמונה – TakePic

– הצגת תמונה – ShowPic

בטלפון מסוג type3 אפשר לבצע את הפעולות האלה:

– חיוג – Dial

– קבלת שיחה – RecCall

– צילום תמונה – TakePic

– הצגת תמונה – ShowPic

– התקנת אפליקציות – InstallApp

בכל אחד מסוגי הטלפונים:

נשמר מספר דקות השיחה שנעשו מן הטלפון.

אפשר לאפס את מספר דקות השיחה – Reset.

אפשר להחזיר חשבון טלפון עדכני – Bill, המחושב כך: מספר דקות השיחה $\times 1$ ש"ח.

בטלפונים שיש בהם אפשרות לשלוח הודעות, החזרת חשבון הטלפון העדכני היא:

מספר דקות השיחה $\times 1$ ש"ח + מספר ההודעות שנשלחו $\times 0.5$ ש"ח.

בטלפונים האלה פעולת האיפוס מאפסת את מספר דקות השיחה וכן את מספר ההודעות שנשלחו.

(שים לב: המשך השאלה בעמוד הבא.)

- א. בעבור הטלפונים הסלולריים המתוארים למעלה, סרטט את היררכיית המחלקות והממשקים, באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism).
- סמן בסרטוט את הקשרים בין המחלקות, ואת הקשרים בין המחלקות ובין הממשקים. אין צורך לכלול בסרטוט תכונות ופעולות.
- ב. לכל אחת מן המחלקות שבסרטוטך כתוב ב- C# את כותרת המחלקה ואת התכונות שלה, וכותרות לכל הפעולות.
- לכל אחד מן הממשקים שבסרטוטך כתוב ב- C# את הכותרת שלו, וכותרות לכל הפעולות. אינך צריך לכתוב את הפעולות Get ו- Set ופעולות בונות.
- ג. ממש ב- C# את הפעולה המחזירה את החשבון העדכני בכל אחת מן המחלקות שהיא מופיעה בהן.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל
אין להעתיק או לפרסם אלא ברשות משרד החינוך