מדעי המחשב ג'

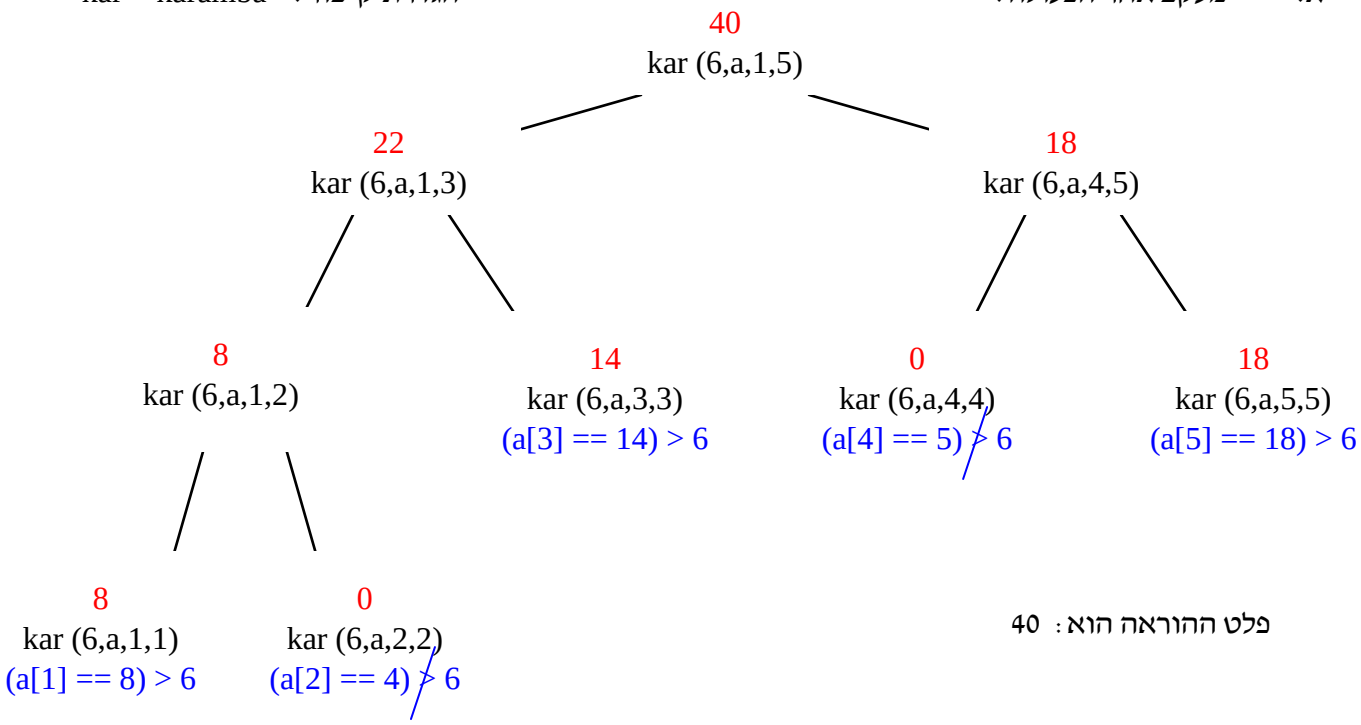
פתרון בחינת הבגרות

פרק א - עיצוב תכנה

שאלה 1:

	0	1	2	3	4	5
a	2	8	4	14	5	18

א. מעקב אחר הפעולה: הגדרת קיצור: kar = karamba



פלט ההוראה הוא: 40

ב. עבור מערך בגודל 6 המכיל מספרים שלמים שערכם 2,
כל ההוראות הבאות יחזירו ערך 6: (ערך k יכול להיות כל מספר הקטן מ- 2, כולל מספר שלילי)

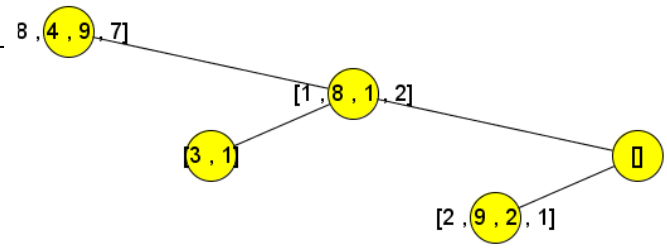
karamba(1, a, 0, 2) , karamba(1, a, 1, 3) , karamba(1, a, 2, 4) , karamba(1, a, 3, 5)

ג. הפעולה מחזירה את סכום כל איברי המערך, בין המקומות s ו- e (כולל), הגדולים מ- k.

ד. סיבוכיות הפעולה הוא O(n)
נימוק: פונקצית זמן הריצה של הפעולה הוא: $f_n = \log_2 n + n \Rightarrow O(n)$
הפעולה מחלקת שוב ושוב את המערך עד לקבלת איבר אחד $(\log_2 n)$
ואחר כך מסכמת - בחזרה מהרקורסיה - את כל האיברים.
בסה"כ עוברת הפעולה פעם אחת על כל אחד מ- n איברי המערך.

שאלה 2: Java:

```
//---                                          פעולה המקבלת עץ בינארי של מחסניות,
//--- ומחזירה מחסנית שכל איבר בה הוא סכום עד 3 האיברים הראשונים שבכל צומת בעץ ---
public static Stack<Integer> treeStkSum (BinTreeNode<Stack<Integer>> t)
{
    Stack<Integer> s = new Stack<Integer>();
    treeStkSum (t, s);
    return s;
}

private static void treeStkSum (BinTreeNode<Stack<Integer>> t, Stack<Integer> s)
{
    if (t != null)
    {
        treeStkSum (t.getLeft(), s);        //  פניה לבן שמאלי

        //--- טיפול בצומת נוכחי ---
        Stack<Integer> s1 = t.getInfo();
        int n = size(s1);                   //  ספירת האיברים שבמחסנית
        int num = 0;

        if (n >= 1) num += s1.pop();        //    איבר ראשון
        if (n >= 2) num += s1.pop();        //    איבר שני
        if (n >= 3) num += s1.pop();        //    איבר שלישי
        s.push(num);

        treeStkSum (t.getRight(), s);       //  פניה לבן ימני
    }
}
```




```
//--- פעולה המחזירה את מספר האיברים במחסנית ---
private static int size (Stack<Integer> s)
{
    //--- מספר האיברים הכללי במחסנית  -  I אפשרות ---
    int count = 0;
    Stack<Integer> sTemp = new Stack<Integer>();

    while (! s.isEmpty())
    {
        count ++;
        sTemp.push(s.pop());
    }

    while (! sTemp.isEmpty())
        s.push(sTemp.pop());

    return count;
}
```

```
//--- פעולה המחזירה את מספר האיברים במחסנית ---
private static int size (Stack<Integer> s)
{
    //---  לספור רק עד 3 איברים ראשונים  -  II אפשרות
    int num = 3;
    int count = 0;
    Stack<Integer> sTemp = new Stack<Integer>();

    while (! s.isEmpty() &&  num > 0)
    {
        count ++;
        num -- ;
        sTemp.push(s.pop());
    }

    while (! sTemp.isEmpty())
        s.push(sTemp.pop());

    return count;
}
```

ב. סיבוכיות הפעולה:

הפעולה עוברת על כל אחד מ- n הצמתים בעץ.

לפי אפשרות I - ספירת n האיברים במחסנית, סיבוכיות הפעולה $O(n^2)$

לפי אפשרות II - ספירה של עד 3 איברים במחסנית $O(1)$ ולכן סיבוכיות הפעולה $O(n)$

```
//---                                          פעולה המקבלת עץ בינארי של מחסניות, ---
//--- ומחזירה מחסנית שכל איבר בה הוא סכום עד 3 האיברים הראשונים שבכל צומת בעץ ---
public static Stack<Integer> treeStkSum (BinTreeNode<Stack<Integer>> t)
{
    Stack<Integer> s = new Stack<Integer>();
    treeStkSum (t, s);
    return s;
}

private static void treeStkSum (BinTreeNode<Stack<Integer>> t, Stack<Integer> s)
{
    if (t != null)
    {
        treeStkSum (t.getLeft(), s);        //  פניה לבן שמאלי

        //--- טיפול בצומת נוכחי ---
        Stack<Integer> sTree = t.getInfo();
        sumStack (s, sTree);

        treeStkSum (t.getRight(), s);       //  פניה לבן ימני
    }
}



        //--- פעולה המחשבת את סכום ** עד ** 3 האיברים שבראש המחסנית ---
        //---                              s ומכניסה אותו למחסנית sTree ---
        public static void sumStack(Stack<Integer>s, Stack <Integer> sTree)
        {
            int sum = 0;
            for (int i = 0 ; i < 3 ; i++)
            {
                if (! sTree.isEmpty())
                    sum += sTree.pop();
            }
        }
```

C#:

```
//--- פעולה המחזירה מחסנית של סכום עד 3 איברים שבצמתי העץ ---
public static Stack<int> TreeSumStack (BinTreeNode<Stack<int>> t)
{
      Stack<int> s = new Stack<int>();
      Sum (t, s);
      return s;
}

//--- דרך 1 ---
public static void TreeSumStack(BinTreeNode<Stack<int>> t, Stack<int> s)
{
      if (t != null)
      {
            Sum(t.GetLeft(), s);

            int n = 3, sum = 0;
            Stack<int> s2 = t.GetInfo();

            while (! s2.IsEmpty() && n > 0)
            {
                  sum = sum + s2.Pop();
                  n -- ;
            }

            Sum(t.GetRight(), s);
      }
}


//--- דרך 2 ---
public static void TreeSumStack(BinTreeNode<Stack<int>> t, Stack<int> s)
{
      if (t != null)
      {
            Sum(t.GetLeft(), s);

            int sum = SumStack (t.GetInfo();
            s.Push(sum);

            Sum(t.GetRight(), s);
      }
}
//--- פעולה המחזירה את סכום עד 3 האיברים שבראש המחסנית ---
public static int SumStack(Stack<int> s)
{
      int sum = 0;
      for (int i = 0 ; I < 3 ; i++)
            if (! s.IsEmpty())
                  sum += s.Pop();
      return sum;
}
```

שאלה 3: **Java :**

א.

```
//--- פעולה המחזירה עותק של הקבוצה ---
public static RealSet clone (RealSet  rs)
{
        RealSet rs1 = new RealSet();
        RealSet rsTemp = new RealSet();

        int n = sr.size();
        double x;

        for (int i = 0 ; i < n ; i ++)
        {
                x = rs.findBiggest();
                rs1.insert (x);
                rsTemp.insert (x);
                rs.remove (x);
        }

        for (int i = 0 ; i < n ; i ++)
        {
                x = rsTemp.findBiggest();
                rs.insert (x);
                rsTemp.remove (x);
        }
        return rs1;
}


//--- פעולה המחזירה קבוצה חדשה המכירה רק את האיברים השליליים מהקבוצה הנתונה ---
public static RealSet buildNeg (RealSet rs)
{
        RealSet rs1 = clone (rs);

        double x;
        int n = rs1.size();
        while (n > 0)
        {
                x = rs1.findBiggest();
                if (x < 0)
                        return rs1;
                rs1.remove();
                n -- ;
        }

        return rs1;             //      אם אין איברים שליליים, תוחזר קבוצה ריקה
}
```

C# : נכתב ע"י **ראמי ג'באלי**

```
//--- פעולה המחזירה עותק של הקבוצה ---
public static RealSet Clone(RealSet rs)
{
      RealSet s1 = new RealSet();
      RealSet s2 = new RealSet();
      int n = rs.Size();
      double num;
      for (int i = 1; i <= n; i++)
      {
            num = rs.FinBiggest();
            rs.Remove(num);
            s1.Insert(num);
            s2.Insert(num);
      }
      for (int i = 1; i <= n; i++)
      {
            num = s2.FinBiggest();
            s2.Remove(num);
            rs.Insert(num);
      }
      return s1;
}

//--- פעולה המחזירה קבוצה חדשה המכילה רק את האיברים השליליים מהקבוצה הנתונה ---
public static RealSet BuidNeg(RealSet rs)
{
      RealSet s = Clone(rs);

      double num;
      bool finish = false;
      int n = s.Size();
      while (n > 0)
      {
            num = s.FinBiggest();
            if (num >= 0)
            {
                  s.Remove(num);
                  n -- ;
            }
            else
                  finish = true;
      }
      return s2;
}
```

שאלה 4:

Java :

א.

```
//--- מחלקה המייצגת את 5 ערימות הקלפים ---
public class Deck
{
    private Stack<Card> [] heaps;    //  ערימות הקלפים
    private int sum;                 //  (0 ערימה) סכום הקלפים שבערימה החמישית

    //---  פעולה בונה ---
    public Deck()

    //--- פעולה המקבלת קלף ומכניסה אותו לראש הערימה הנכונה ---
    //--- ערימה נכונה היא ערימה 1 - 4, לפי צורת הקלף הנתון ---
    public void insert (Card c)

    //--- פעולה המגרילה מספר ערימה, ומחזירה 'אמת' אם הצליחה להעביר ---
    //---     קלף מראש הערימה שהוגרלה לערימה החמישית, ו-'שקר' אחרת ---
    public boolean move ()

    //--- פעולה המחזירה את סכום הקלפים שבערימה החמישית  ---
    public int getSum()

}
```

ב.

```
//---  פעולה בונה ---
public Deck()
{
    this.heaps = new Stack[5];

    for (int i = 0 ; i < heaps.length ; i++)
        this.heaps[i] = new Stack<Card>();

    this.sum = 0;
}
```

ג.

```
//--- פעולה המחזירה את סכום הקלפים שבערימה החמישית  ---
public int getSum()
{
    return this.sum;
}
```

ד.

```
//--- פעולה המחזירה 'אמת' אם המשחק הסתיים בניצחון ---
//---   על פי הכללים שמוגדרים בשאלה, ו-'שקר' אחרת ---
public static boolean game (Card [] card)
{

    Deck deck = new Deck ();

    //--- שלב ראשון של המשחק - מילוי ערימות הקלפים ---
    for (int i = 0 ; i < card.length ; i++)
        deck.insert(card[i]);

    //--- שלב שני של המשחק - העברת הקלפים לערימה החמישית ---
    boolean gameOver = false;
    while (! gameOver)
        gameOver = ! deck.move();

    //--- ? האם יש ניצחון ---
    if (deck.getSum() % 100 == 0)
        return true;
    return false;
}
```

דרך כתיבה נוספת לניהול השלב השני של המשחק:

```
boolean gameOver = false;
while (! gameOver)
{
      if (! deck.move())
            gameOver = true;

}
```

פרק ב'

מערכות מחשב ואסמבלר

הפתרון לפרק זה נכתב ע"י: **סוזי גנזיה – מקיף ט' ראשון לציון**

תרגיל 5:

א.

תא זיכרון A	AX		CX	הוראה
69H= 01101001B	AL	AH		
	00	00	0002	MOV CX,2
	00	00		MOV AX,0
	01101001B (69h)			MOV AL,A
		01101001B (69h)		ADD AH,A
	10100100b (A4h)	10100101b (A5h)		Next: SHL AX,CL
	10100101b (A5h)			OR AL,AH
		00101001b (29h)		SHR AH,CL
			0001	LOOP NEXT
	01001001B (4Ah)	01010011b (53h)		Next: SHL AX,CL
	01011011b (5Bh)			OR AL,AH
	01011011b (5Bh)	00101001b (29h)		SHR AH,CL
			0000	LOOP NEXT
01011011b (5Bh)				MOV A,AL

ב. 1. MOV AX, PLACE השלמה 1

MOV BX,1 השלמה 2

2 אם ההוראה SHR AX,1 אשר שייכת למספרים לא מכוונים תוחלף בהוראה SAR AX,1 השייכת למספרים מכוונים ישתנה ביצוע הקטע בשל השפעתו על הסיבית המשמעותית ביותר, סיבית הסימן (מריחת סימן) והתוצאה שתתקבל תהיה: 0FFFFH.

תרגיל 6:

א.

SI	CX	AX	
20A		AH	AL
	0007		
			0C
209			
	0006		
			11
208			
	0005		
			1A
207			
	0004		
			A8
206			
	0003		
			21
205			
	0002		
			06
204			
	0001		
			02
204			
	0000		
			00

מפת זיכרון של המערך לפני ביצוע התוכנית

204	205	206	207	208	209	20A	20B
2	6	21	A8	1A	11	C	F1

מפת זיכרון של המערך לאחר ביצוע התוכנית:

204	205	206	207	208	209	20A	20B
0	2	6	21	A8	1A	11	C

2. קטע התכנית מבצע הזזה שמאלה של תאי המערך ומכניס אפס לתא הראשון.

ב.

1. MOV AL, 0A1h (0A1h =10100001b)
SUB AL,11111101b

10100001-11111101=10100100b (borrow בפעולת החיסור התבצע)

AL= 10100100b (A4h) ,לאחר הפעולה

cf	zf	sf	pf	of
1	0	1	0	0

2. MOV AL, 01110111b (77h)
ADD AL,29D (29D = 1Dh)

77h+1dh= 94h (94h מספר שלילי, התבצעה גלישה)

AL= 10010100b (94h) לאחר הפעולה

cf	zf	sf	pf	of
0	0	1	0	1

תרגיל 7:

	AX	DX	CX		Cmp	מחסנית
			CH	CL		בתוכנית הראשית:
תוכנית ראשית	8888					sp 88 88
						SP 11 11 88 88 אחרי פקודת CALL UPLIL:
שגרה		A305				כתובת חזרה SP 11 11 88 88
				05		
	1111					כתובת חזרה BP=SP 11 BP+2 11 88 BP+4 88
					F	כתובת חזרה BP=SP 22 BP+2 22 88 BP+4 88
	Shl -> 2222			04		
		8888			F	כתובת חזרה BP=SP 44 BP+2 44 88 BP+4 88
	4444 Shl ->					

תמונת מחסנית במהלך הרצת השגרה:

BP=SP	כתובת חזרה
BP+2	11
	11
BP+4	88
	88

BP=SP	כתובת חזרה
BP+2	22
	22
BP+4	88
	88

BP=SP	כתובת חזרה
BP+2	44
	44
BP+4	88
	88

BP=SP	כתובת חזרה
BP+2	88
	88
BP+4	88
	88

תמונת זיכרון:

CHECK	SUM		NUMBER	
0004	0003	0002	0001	0000
?	11	11	A3	05
1	11	11	A3	05

ב. SUM יישאר ללא שינוי, ערכו 1111 בשל מחיקת ערכי המחסנית ע"י הפקודה RET 4 ואי החזרתם לתכנית הראשית.

שאלה 8:

```
.CODE
      MOV AX, @DATA
      MOV DS,AX
      MOV CX,100
      MOV DH,0
      MOV SI,OFFSET NUMBERS
AGAIN: MOV BL,[SI]
      MOV BH,0
      MOV DH,APP [BX]
      INC DH
      MOV APP[BX],DH
      INC SI
      LOOP AGAIN
SOF:  MOV AX, 4C00h
      INT 21h
      END
```

פרק ב'

מבוא לחקר ביצועים

שאלה 9:

שאלה 10:

שאלה 11:

שאלה 12:

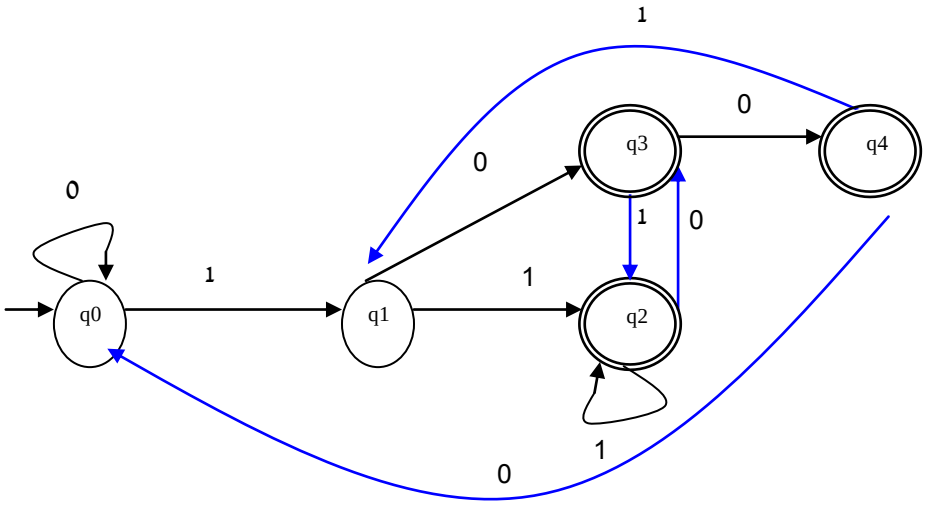
פרק ב'

מודלים חישוביים

הפתרון לפרק זה נכתב ע"י **רחל לודמר**.

תרגיל 13:

א.



ב. (1) מילה מינימאלית המתקבלת ע"י האוטומט: ab,aa,bb

(2) מילה המתחילה ב – a וארכה גדול מ 3: abbb

(3) מילה המתחילה ב – b וארכה גדול מ 3: bbbbbb

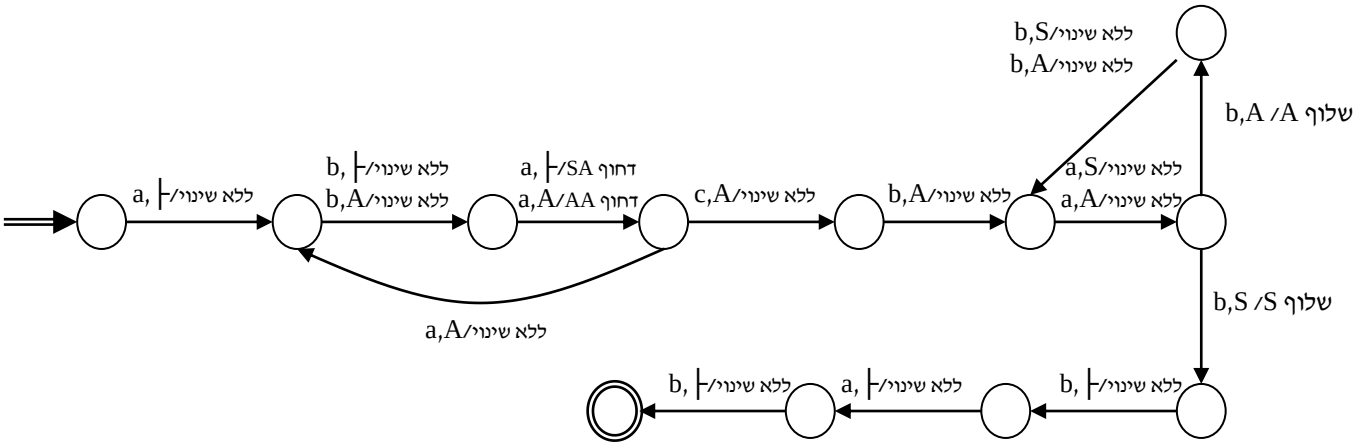
(4) $L = \{aw \text{ או } b^{2n} \mid n > 0,\ w \in \{a,b\}, |w| \%2 = 1\}$

המילים ב- L הן מילים המתחילות ב- a ואורכן זוגי או רצף של b באורך זוגי חיובי .

שאלה 14:

אוטומט מחסנית עבור השפה:

$$L = \{(aba)^n c\,(bab)^{2n+1} \mid n > 0\}$$



שאלה 15:

א.

$$w = b^i c^i \mid i \geq 1$$
$$w_1 w = a^i b^i c^i \in L_1$$
$$w_2 w = a^j b^i c^i \notin L_1$$

ב.

$$w = a^i b \mid i \geq 1$$
$$w_1 w = a^i b a^i b \in L_1$$
$$w_2 w = a^j b a^i b \notin L_1$$

$w_2 w$ אינו מהצורה $v \cdot v$.

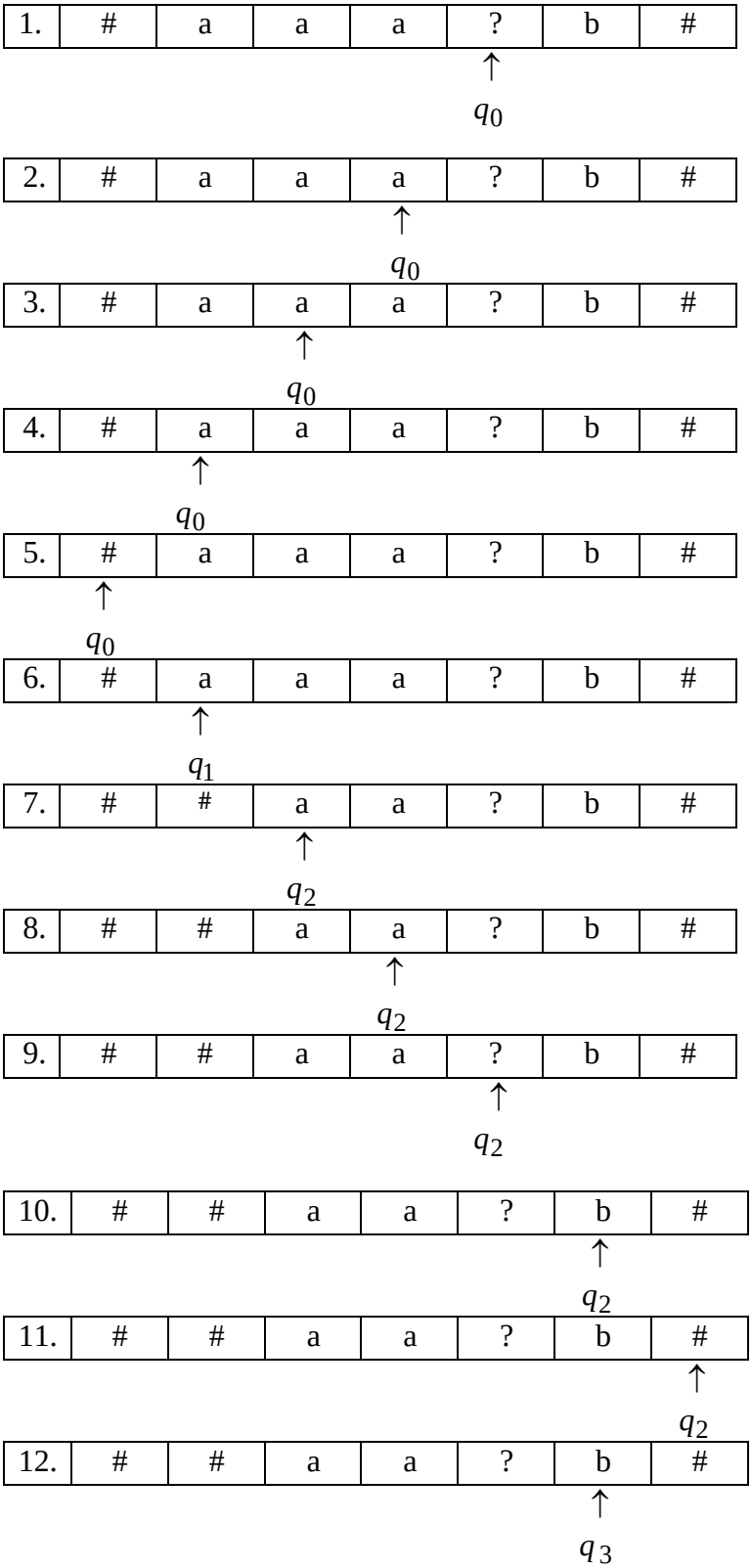
ג. (i) הטענה לא נכונה. המילה ab שייכת לשפה ואין a אחרי ה- b.

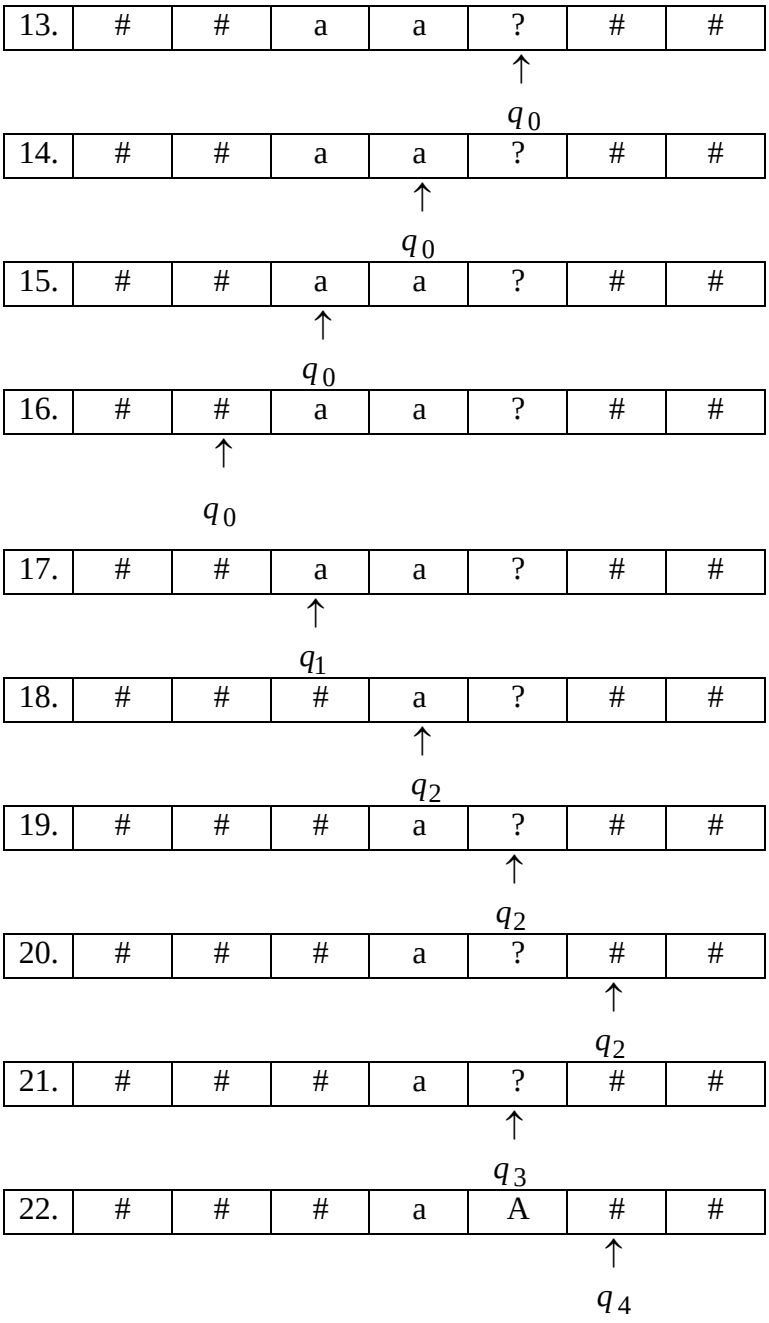
(ii) הטענה לא נכונה. המילה abb מסתיימת ב- b ולא שייכת לשפה.

(iii) הטענה נכונה. שפת האוטומט היא אוסף כל המילים מעל הא"ב {a,b} שאין בהם 2 - b רצופים.

שאלה 16:

א. מסלול חישוב עבור הקלט: #aaa?b#





הפלט: ###aA###

ב. התו שירשם במקום ? הוא B. הפלט במקרה זה הוא: ###B###.

ג. מטרת המכונה:
כאשר מספר ה- a גדול ממספר ה- b, המכונה תרשום A במקום התו ?, וכן m התווים הראשוניים של ה- a (השמאליים ביותר) יהפכו ל- #, וגם כל תווי ה- b יהפכו ל- #.

הפלט יהיה מהצורה: $\#\#^m a^{n-m} A\#^m \# \mid n > m, m > 0$

וכאשר מספר ה- b גדול או שווה למספר ה- a, התו ? יהפוך ל- B, כל ה- a יהפכו ל- #, וכן n ה- b הימניים ביותר יהפכו ל- #.

הפלט יהיה מהצורה: $\#\#^n Bb^{m-n}\#^n \# \mid m > n, n > 0$

פרק ב'

תכנות מונחה עצמים Java

תרגיל 17:

א.
(i) לא נכון. A היא מחלקת העל. הפעולה validCode מוגדרת בתת המחלקה שלה.
(ii) נכון. B יורשת מ- A ולכן היא גם יורשת את כל התכונות ואת כל הפעולות.
(iii) לא נכון. התכונה myVal שב- A היא תכונה פרטית (private) ולא מוגנת (protected).
(iv) לא נכון. A אינה מכירה את התכונות של B ולכן אינה יכולה להתייחס אליהן (גם אם הן לא היו מוגדרות כתכונות פרטיות).

ב.

```
public B (int val, double x)
{
      super(val);
      this.x = x;
}
```

ג.
(i) שגיאת הידור. code הוא אובייקט מטיפוס B שבעקבות ההוראה עבר המרה כלפי מעלה לטיפוס A. ולכן הניסיון לבקש ממנו לבצע פעולה של B אינה חוקית.

(ii) שגיאת הידור. num הוא אובייקט מטיפוס A, ולכן validCode אינה שייכת לפעולות שלו.

(iii) שגיאת הידור. ההמרה לטיפוס B מתבצעת על התוצאה של הפעולה.
הפעולה מביאה לשגיאת הידור (סעיף i). התוצאה היא ערך בוליאני שאינו יכול להיות מומר לטיפוס בוליאני.
שינוי ההוראה ל- **boolean** myBool = ((B)code).validCode() הייתה הופכת את ההוראה לתקינה.

(iv) שגיאת הידור. num אינו מטיפוס A ולכן אי אפשר לבצע לו המרה לטיפוס B.
(שינוי ההוראה ל- **boolean** myBool = ((B)num).validCode() תיתן שגיאת זמן ריצה).

ד.

דרך א: הפעולה f שבמחלקה A אינה עוברת שינוי.

במחלקה של הפעולה הראשית נכתוב:

קטע הקוד ב- main

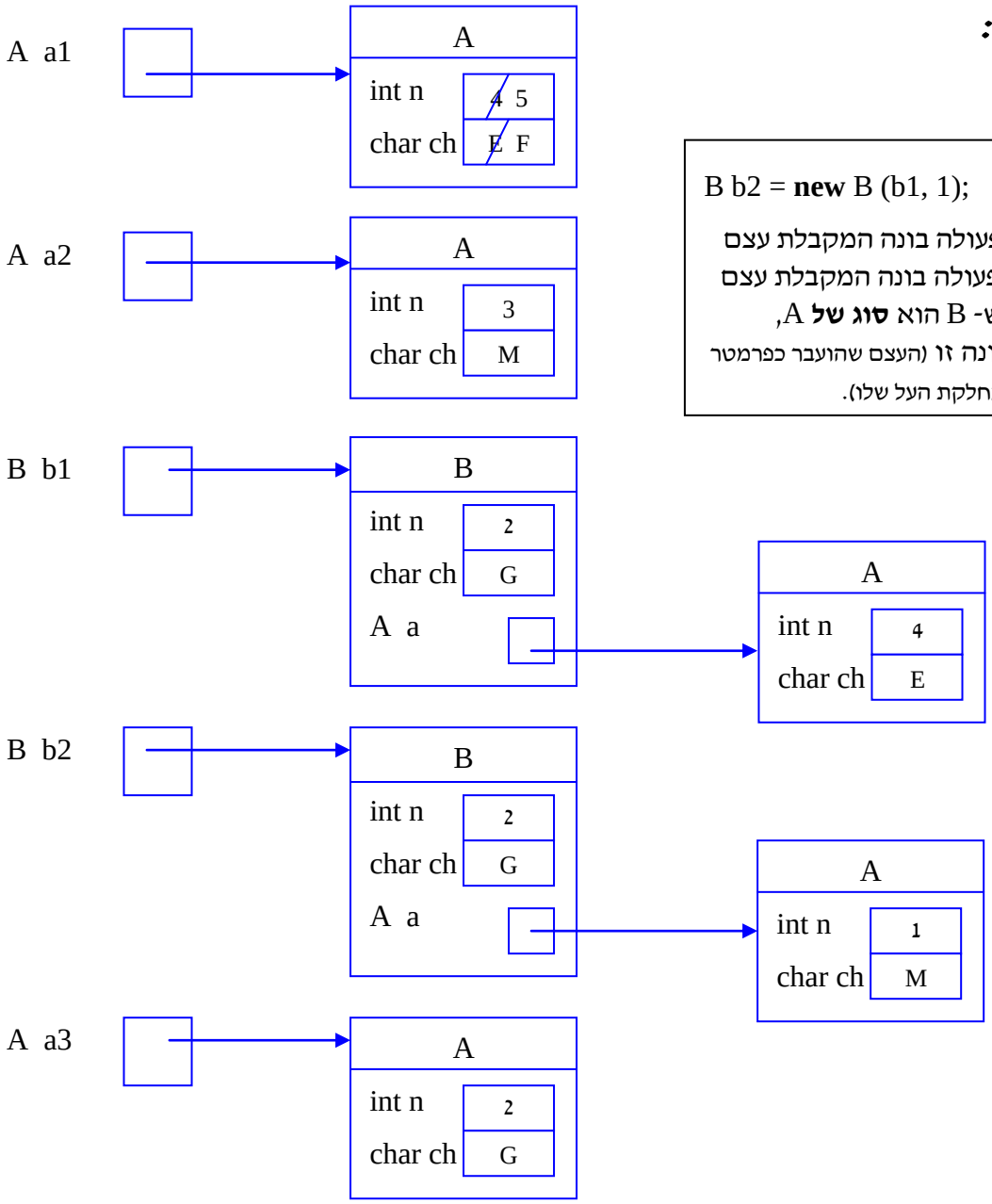
```
A [] arr = new A[5];
arr[0] = new A (1);
arr[1] = new B (2, 2.2);
arr[2] = new B (3, 3.3);
arr[3] = new A(4);
arr[4] = new B (5, 5.5);

int countB = 0;
for (int i=0 ; i<arr.length ; i++)
{
     if (arr[i] instanceof B)
          countB ++;
}
int countA = arr.length - countB;
```

דרך ב: שינוי הפעולה f במחלקה A:

קטע הקוד ב- main	שינוי הפעולה f במחלקה A

`A [] arr = new A[5];`
`arr[0] = new A (1);`
`arr[1] = new B (2, 2.2);`
`arr[2] = new B (3, 3.3);`
`arr[3] = new A(4);`
`arr[4] = new B (5, 5.5);`
 	
`int countA = 0, countB = 0;`
`for (int i=0 ; i<arr.length ; i++)`
`{`
`  if (arr[i].f() == 1)`
`  countB ++;`
`  else`
`  countA ++;`
`}`
	`public int f()`
	`{`
	`  if (this instanceof B)`
	`  return 1;`
	`  return 0;`
	
	`}`

תרגיל 18:



B b2 = **new** B (b1, 1);

במחלקה B אין פעולה בונה המקבלת עצם מסוג B אבל יש פעולה בונה המקבלת עצם מסוג A. מכיוון ש- B הוא **סוג של** A, מופעלת פעולה בונה זו (העצם שהועבר כפרמטר מקבל הפניה מסוג מחלקת העל שלו).

פלט התכנית:

F F F F F

M M M

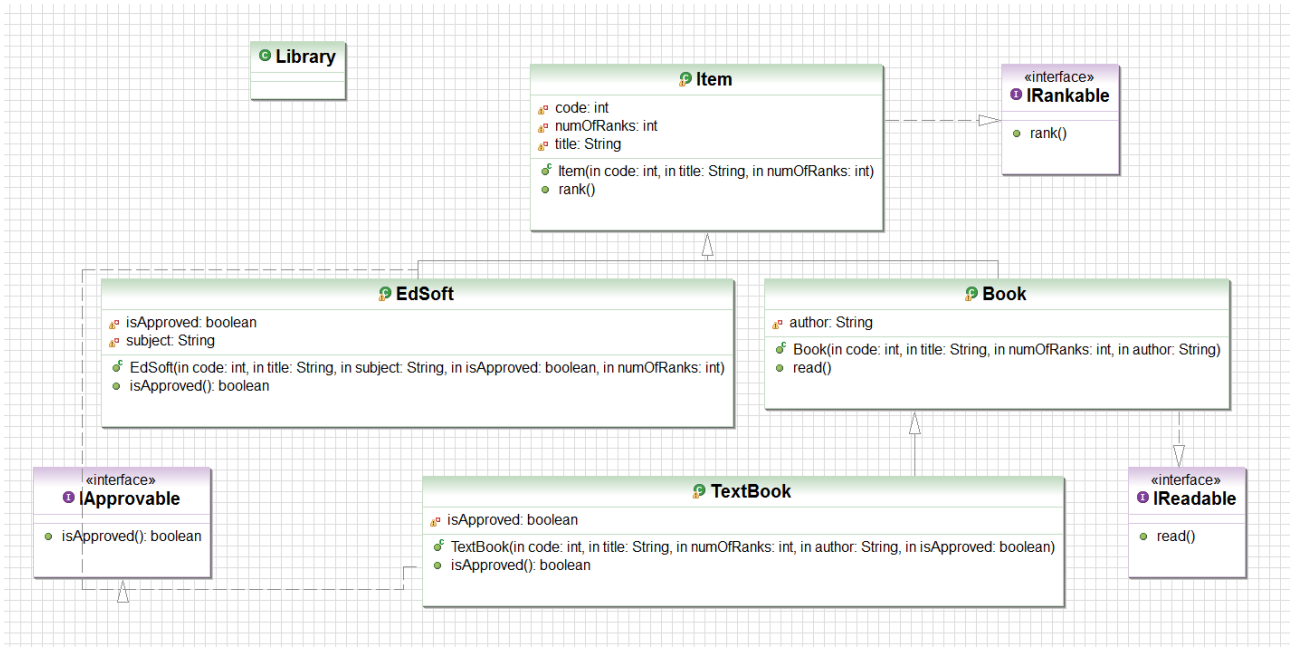
E E E E

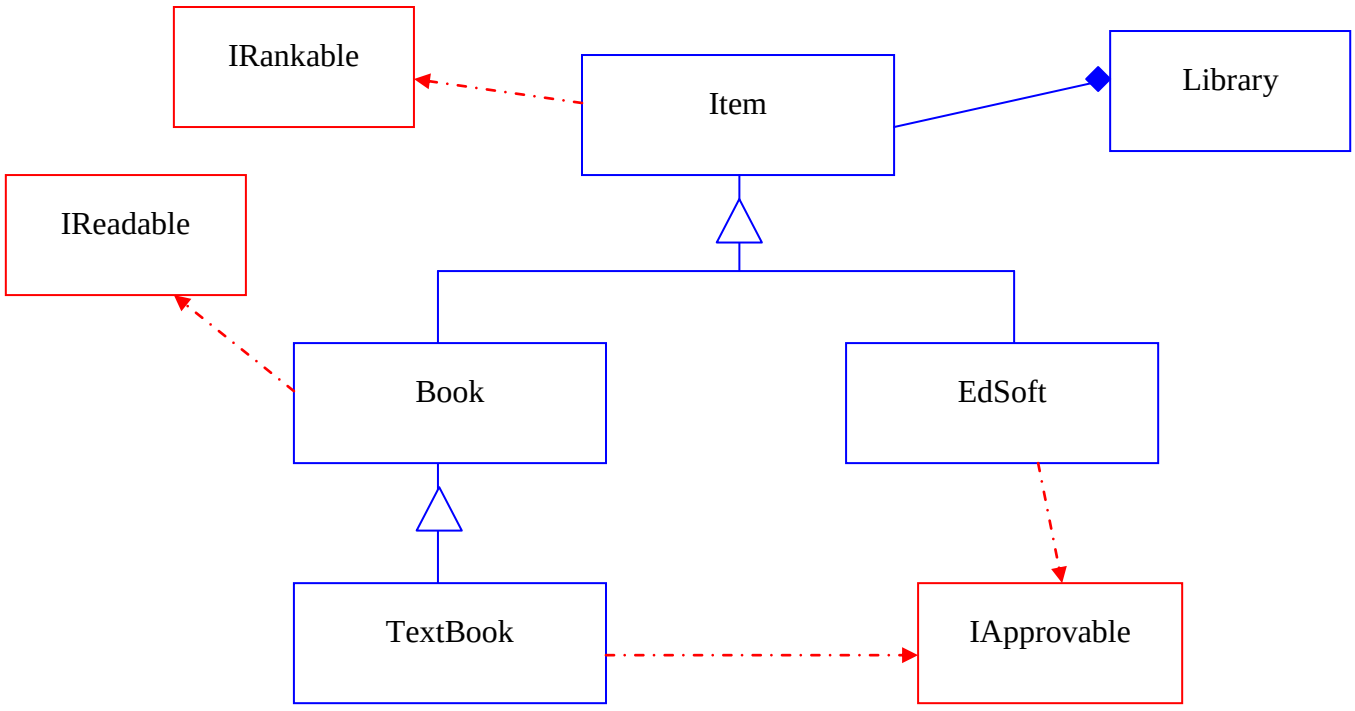
M

G G

תרגיל 19:

א.


Library
Item
code: int
numOfRanks: int
title: String
Item(in code: int, in title: String, in numOfRanks: int)
rank()
«interface»
IRankable
rank()
EdSoft
isApproved: boolean
subject: String
EdSoft(in code: int, in title: String, in subject: String, in isApproved: boolean, in numOfRanks: int)
isApproved(): boolean
Book
author: String
Book(in code: int, in title: String, in numOfRanks: int, in author: String)
read()
«interface»
IApprovable
isApproved(): boolean
TextBook
isApproved: boolean
TextBook(in code: int, in title: String, in numOfRanks: int, in author: String, in isApproved: boolean)
isApproved(): boolean
«interface»
IReadable
read()


IRankable
Item
Library
IReadable
Book
EdSoft
TextBook
IApprovable

ב.

```
public class Item implements IRankable
{
    private int code;        //  קוד פריט
    private String title;    //  שם הכותר
    private int numOfRanks; //  מספר הממליצים

    public Item(int code, String title, int numOfRanks)
    {
        this.code = code;
        this.title = title;
        this.numOfRanks = numOfRanks;
    }

    public void rank() { }

}
```

```
public class Library
{
    Item [] arr = new Item[5];
}
```

```
public class EdSoft extends Item implements IApprovable
{
    private String subject;     //  שם מקצוע
    private boolean isApproved; //  האם מאושר ע"י משרה"ח

    public EdSoft(int code, String title, String subject, boolean isApproved, int numOfRanks)
    {
        super(code, title, numOfRanks);
        this.subject = subject;
        this.isApproved = isApproved;
    }

    public boolean isApproved() {  return true; }
}
```

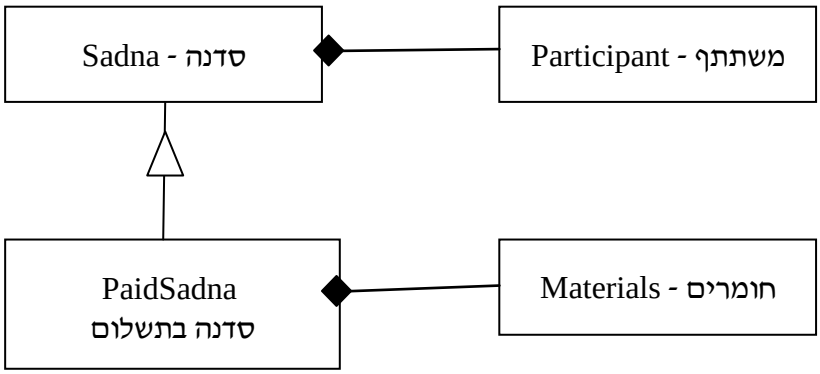
```
public class Book   extends Item implements IReadable
{
    private String author;  //  שם המחבר

    public Book(int code, String title, int numOfRanks, String author)
    {
        super(code, title, numOfRanks);
        this.author = author;
    }

    public void read() {  }
}
```

```
public class TextBook extends Book implements IApprovable
{
    private boolean isApproved; //  האם מאושר ע"י משרה"ח

    public TextBook(int code, String title, int numOfRanks, String author, boolean isApproved)
    {
        super(code, title, numOfRanks, author);
        this.isApproved = isApproved;
    }

    public boolean isApproved() {return true; }
}
```

Hugim – חוגים
התכנית הראשית
כוללת מערך של סדנאות

שאלה 20:

א.



ב.

```
//---  המחלקה סדנה  ---
public class Sadna
{
    private int code;           //  קוד סדנה
    private String name;        //  שם סדנה
    private int sug;            //  סוג סדנה: 1 – חרוזים, 2 – ציור על בד, 3 – ציור על עץ
    private int duration;       //  משך סדנה: 1 – חד פעמי, 2 – 10 פגישות, 3 – שנתי
    private char time;          //  מועד הסדנה : 'm' – סדנת בוקר , 'e' – ערב
    protected double price;     //  מחיר הסדנה
    private int max;            //  מספר משתתפים מקסימלי (לכל סדנה מספר משתתפים שונה)
    private Participant [] arr; //  מערך המשתתפים (אפשר גם רשימה של משתתפים)
    private int lastPosition;   //  מספר הרשומים בפועל / המקום הראשון הפנוי במערך

    //---  הפעולה הבונה  ---
    public Sadna(int code, String name, int sug, int duration, char time, double price, int max)

    //---  הוספת משתתף לסדנה ---
    public void add(Participant p)

    //--- פעולה המחזירה 'אמת' אם יש מקום בסדנה, ו-'שקר' אחרת ---
    public boolean isFull()

    //--- החזרת מחיר הסדנה ---
    public double getPrice()

    //--- החזרת מספר המשתתפים בסדנה ---
    public int numOfParticipant()

    //--- החזרת שם הסדנה ---
    public String getName()
```

```
//---  סדנה בתשלום  :תת-מחלקה ---
public class PaidSadna  extends Sadna
{
    private List<Materials> lst;            //  רשימת חומרים לסדנה
    private double TotalMatirialPrice;      //  סך מחיר החומרים לסדנה

    //---   פעולה בונה  ---
    public PaidSadna(int code, String name, int sug, int duration, char time, int price, int max)

    //--- הוספת חומר גלם לסדנה ---
    public void add (Materials mat)

    //--- החזרת מחיר הסדנה ---
    public double getPrice()
}
```

```
//---  המחלקה משתתף בסדנה ---
public class Participant
{
    private String name;         //  שם המשתתף
    private String phone;    //  מס' טלפון
```

```
//---  המחלקה חומרים לסדנה ---
public class Materials
{
    private String name;         //  שם חומר הגלם
    private double price;            //  מחיר חומר הגלם
```

התכנית הראשית - **חוגים**

כולל יצירת חוג אחד מטיפוס סדנה שתשלום (לא נדרש בבחינה.)

```
public class T20_Hugim
{

    /**
     *       20 שאלה - 2012 '899205 בגרות
     *
     *              רישום לחוגים
     */
    public static void main(String[] args)
    {
        Sadna [] sd = new Sadna [3];

        sd[0] = new PaidSadna (123, "Beads - Adults", 1, 2, 'e', 800, 4 );

        //---  הוספת חומרים לסדנה ---
        ((PaidSadna)sd[0]).add(new Materials ("Cristal", 34.75));
        ((PaidSadna)sd[0]).add(new Materials ("Svarovski", 52.50));
        ((PaidSadna)sd[0]).add(new Materials ("FireLine", 34));

        //--- הוספת משתתפים בסדנה ---
        sd[0].add(new Participant ("Aviva", "050-5556667"));
        sd[0].add(new Participant ("Batya", "052-5552223"));


        sd[1] = new Sadna (234, "Drawing on wood - childs", 2, 1, 'm', 50, 5);
        sd[1].add(new Participant ("Abner", "050-5557788"));
        sd[1].add(new Participant ("Beni", "052-5553344"));
        sd[1].add(new Participant ("Gali", "054-6663344"));
        sd[1].add(new Participant ("Dalit", "052-5557788"));
```

פרק ב'

תכנות מונחה עצמים C#

הפתרון לפרק זה נכתב ע"י .....

תרגיל 21:

תרגיל 22:

תרגיל 23:

שאלה 24: