

מבחן השלמה למגמת הנדסת תוכנה על-תכונות

הוראות לנבחנים

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים ובהם חמש שאלות. עליכם לענות על כל השאלות.
לכל שאלה – 20 נקודות. סך-הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: מחשבון פשוט שאינו ניתן לתכנות.
כל חומר עזר כתוב בכתב-יד או מודפס על נייר.

ד. הוראות מיוחדות:

- כתבו בעט בלבד.
- ענו על השאלות במחברת הבחינה.
- הדביקו את מדבקות הנבחן במקומות המיועדים לכך.

כתבו במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונכם לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).
כתבו "טייטה" בראש כל עמוד טייטה. כתיבת טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלולה לגרום לפסילת הבחינה!

בשאלון זה 9 עמודים.

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

בהצלחה!

השאלות

בשאלון זה שני פרקים ובהם חמש שאלות (לכל שאלה – 20 נקודות).
עליכם לענות על כל השאלות.
שימו לב, כל קטעי הקוד בשאלון כתובים בשפת Java.

פרק ראשון (60 נקודות)

ענו על השאלות 1–3.

שאלה 1 (20 נקודות)

קו אוטובוס עירוני אמור לצאת מתחנת המוצא שלו במרווחים של 15 דקות, ברבעי שעות שלמות (כגון בשעות 8:00, 8:15, 8:30 וכו').
עקב שיבושים בתחבורה ביום מסוים, הקו לא יצא בכל הזמנים. נתון מערך שמייצג את זמני היציאה באותו יום. הזמנים מופיעים בפורמט של מספר בן שלוש או ארבע ספרות (לדוגמה, המספר 945 מייצג את השעה 9:45).

לדוגמה, במערך שלהלן:

900	915	1000	1015	1030	1045	1100	1130
-----	-----	------	------	------	------	------	------

חסרים הזמנים: 930, 945, 1115.

הניחו שזמן היציאה המאוחר ביותר הוא 2345.

(8 נק') א. לפניכם פעולת עזר: incTime

הפעולה מקבלת מספר טבעי שלם, בן שלוש או ארבע ספרות, המייצג שעת יציאה של אוטובוס בפורמט שהוצג לעיל, ומחזירה את זמן היציאה של האוטובוס הבא, באותו פורמט.

בפעולה שלפניכם חסרים שני ביטויים, המסומנים בין סוגריים עגולים.
כתבו במחברת הבחינה את הביטויים החסרים (1)–(2), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static int incTime(int t)
{
    if ( _____ (1) _____ )
        return t + 55;
    else return t _____ (2) _____;
}
```

12 נק') **ב.** לפניכם קטע תוכנית שעובר על המערך ומדפיס את כל הזמנים שבהם האוטובוס היה אמור לצאת ולא יצא. בקטע חסרים **ארבעה** ביטויים, המסומנים בין סוגריים עגולים. כתבו במחברת הבחינה את הביטויים החסרים (3)–(6), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
int t;  
for (int i = 0; i < arr.length - 1; i++)  
{  
    t = _____ (3);  
    while (_____ (4))  
    {  
        _____ (5);  
        t = _____ (6);  
    }  
}
```

שאלה 2 (20 נקודות)

בעלים של קונדיטוריה, שאופים בה מאפים מסוגים שונים, החליט למחשב את העסק שלו. להלן מחלקת "מאפה" (**Pastry**) שהוגדרה:

```
public class Pastry{  
    protected String name;  
    protected int price;  
  
    public Pastry (String name, int price) {  
        this.name = name;  
        this.price = price;  
    }  
}
```

ענו על הסעיפים שלפניכם. **שימו לב:** בסעיפים א–ג יש להשתמש בעקרון ההורשה.

(3 נק') **א.** "עוגה" היא סוג של מאפה. במחלקה "עוגה" יימצא, נוסף למידע של מאפה כללי, גם: זמן ההכנה בדקות, ועוגה טבעונית היכולה להחליף אותה (בתכונה זו יהיה ערך null אם העוגה עצמה היא טבעונית, או שאין לה תחליף טבעוני).

הגדירו את המחלקה "עוגה" (**Cake**): כותרת, תכונות, ופעולה בונה סטנדרטית.

(4 נק') **ב.** "בורקס" גם הוא סוג של מאפה, ובמחלקה "בורקס" יימצא, נוסף למידע של מאפה כללי, גם: אם טבעוני (כן או לא), וסוג המילוי (מחרוזת).

הגדירו את המחלקה "בורקס" (**Burekas**): כותרת, תכונות, ופעולה בונה סטנדרטית.

(3 נק') **ג.** ציירו תרשים מחלקות UML מלא, הכולל את מחלקת "מאפייה" (**Bakery**) שבתוכה מאגר של **שישים** סוגי מאפים.

10 נק') ד. כתבו במחלקת "מאפייה" פעולה המקבלת מערך של הזמנות מאפים לאירוע הנערך על טהרת הטבעונות. הפעולה תבצע:

- הוצאה של בורקסים שאינם טבעוניים מההזמנה.
- החלפת עוגות לא טבעוניות בגרסה הטבעונית שלהן. עוגות שאין להן גרסה טבעונית - יוצאו מההזמנה (יוגדרו null).

לרשותכם פעולות get בכל המחלקות, ואין צורך לממשן.

שאלה 3 (20 נקודות)

נתונה פעולה:

```
public static int make (int num)
{
    int help=0;
    while (num>0)
    {
        if (num%10==Math.pow (num/10%10,2) )
            help = help*100 + num/10%10*10 + num%10;
        num=num/100;
    }
    return help;
}
```

4 נק') א. עקבו אחרי הפעולה עם זימון המספר num=241739 וציינו מה יהיה הערך המוחזר.

4 נק') ב. כתבו בקצרה מה מחזירה הפעולה.

12 נק') ג. לפניכם קוד של פעולה המקבלת מטריצה ריבועית של מספרים חיוביים ושלמים. על הפעולה להחזיר "true" אם כל איברי האלכסון הראשי, אך לא איברים אחרים במטריצה, מקיימים את התנאי: זימון פעולת make עבורם לא תשנה את ערכם.

בקוד חסרים שישה ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (1)–(6), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static boolean take (int [][]mat)
{
    boolean sign = true;
    for(int x=0;x<mat.length;x++)
    {
        for (int y=0;y<mat.length;y++)
        {
            if(_____(1)_____)
            {
                if(mat[x][y]_____(2)_____)
                    sign = _____(3)_____;
            }
            else
            {
                if_____ (4) _____)
                    sign = _____(5)_____;
            }
        }
        _____(6)_____;
    }
}
```

פרק שני (40 נקודות)
ענו על השאלות 4 ו-5.

שאלה 4 (20 נקודות)

נתון ממשק המחלקה Queue המייצג תור:

פונקצייה	תיאור הפעולה של הפונקצייה
Queue()	בניית תור ריק
remove()	הוצאה מראש התור (בהנחה שהתור איננו ריק)
insert(Queue,x)	הכנסה של הערך x לסוף התור
boolean isEmpty()	בדיקה אם התור ריק (אם כן – מחזירה "true", אחרת – מחזירה "false")

(8 נק') א. בפעולה חסרים **ארבעה** ביטויים, המסומנים בין סוגריים עגולים. לפניהם פעולה המקבלת תור של מספרים חיוביים שלמים. על הפעולה להדפיס את כל המספרים בתור שגדולים מהממוצע של המספרים הנמצאים לפניהם בתור.

לדוגמה:

עבור התור: 6, 3, 1, 2, 4 (יש לקרוא מימין לשמאל) יודפסו המספרים 4, 5
עבור התור: 1, 2, 3, 4, 6 (יש לקרוא מימין לשמאל) יודפסו המספרים 2, 3, 4, 6
שימו לב: אין צורך לשמור על ערכי התור עם סיום הפעולה.

כתבו במחברת הבחינה את הביטויים החסרים (1)–(4), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
1. public static void overAverage (Queue <int>q)
2. {
3.     int temp, counter=1;
4.     _____ (1)
5.     while(!q.isEmpty() )
6.     {
7.         _____ (2)
8.         if (_____ (3) )
9.             System.out.println(_____ (4) );
10.        sum+=temp;
11.    }
12. }
```

(8 נק') ב. הדרישה המופיעה בסעיף א' השתנתה, וכעת יש לשמור על התור המקורי. ענו על הסעיפים הבאים לפי הקוד שהוצג לעיל בסעיף א':

1. אם נדרש שינוי בפקודה מסוימת – מה מספר השורה, ומהו השינוי?

2. אם נדרשת מחיקה – אילו שורות יש למחוק?

3. אם נדרשת הוספה – כתבו את השורה הנדרשת, וציינו את מספר השורה שאחריה תתבצע ההוספה.

(4 נק') ג. לפניכם פעולה המקבלת תור של מספרים חיוביים שלמים. על הפעולה להדפיס את כל המספרים בתור שגדולים מהממוצע של המספרים **הבאים אחריהם** בתור. הפעולה משתמשת בפעולות על מחסנית (Stack) סטנדרטית של Java.

שימו לב: אין צורך לשמור על ערכי התור עם סיום הפעולה.

בפעולה חסרים **שני** ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (5)–(6), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static void overAverage (Queue <int>q)
{
    int temp, sum, counter=1;
    Stack<int> s1=new stack();
    while(!q.isEmpty() )
    {
        _____ (5) _____;
    }
    sum=_____ (6) _____;
    while(!s1.isEmpty() )
    {
        temp=s1.pop()
        if( temp>(sum/counter++) )
            System.out.println(temp);
        sum+=temp;
    }
}
```

שאלה 5 (20 נקודות)

נתונות שתי מחלקות –

מחלקה המתארת צומת ברשימה:

```
public class Node {  
    int data;  
    Node next;  
  
    Node(int data) {  
        this.data = data;  
        this.next = null;  
    }  
}
```

ומחלקה המתארת רשימה:

```
public class MyLinkedList {  
    Node head = null;  
    public void add(int data) {  
        Node newNode = new Node(data);  
        Node tail = null;  
        if (head == null) {  
            head = newNode;  
            tail = newNode;  
        } else {  
            tail.next = newNode;  
            tail = newNode;  
        }  
    }  
}
```

(5 נק') א. להלן פעולה המקבלת רשימה של מספרים שלמים ומחזירה "true" אם היא מעגלית (כלומר: הצומת האחרון מצביע לראשון).

בפעולה חסרים **חמישה** ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (1)–(5), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.


```
public static boolean isCircle(MyLinkedList my)
{
    Node help = _____ (1) _____;
    if (_____ (2) _____)
        return false;
    while(_____ (3) _____)
    {
        if(_____ (4) _____)
            return true;
        help=_____ (5) _____;
    }
    return false;
}
```

(5 נק')**ב.**

להלן פעולה המקבלת רשימה לא ריקה של מספרים שלמים, בודקת אם היא מעגלית, ואם כן - הפעולה מנתקת את המעגל.

בפעולה חסרים **חמישה** ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (6)-(10), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static void depart(MyLinkedList my)
{
    Node first = _____ (6) _____;
    if(_____ (7) _____)
    {
        while(_____ (8) _____)
            first=_____ (9) _____;
        _____ (10) _____=null;
    }
}
```

(10 נק')**ג.**

הסבירו במילים איזה שינוי יחיד ניתן לעשות במחלקת **MyLinkedList** כדי שלשתי הפעולות בסעיפים א-ב לא תידרשנה לולאות, **וכתבו מחדש** את פעולה א' (**isCircle**) במלואה לאור מה שהסברתם.

בהצלחה!