

קרא בעיון את ההנחיות שלהלן:

- בבחינה יש שלוש שאלות. עליכם לענות על כולן.
 - כל התכניות צריכות להיות מתועדות היטב. יש לכתוב תחילה **בקצרה** את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית. יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית. תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85% מהניקוד.
 - יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה. **תכנית לא יעילה לא תקבל את מלוא הנקודות.**
 - **אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את המחלקה למחברת הבחינה. מספיק להפנות למקום הנכון, ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').**
 - אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.
 - יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהציון.
 - בכל השאלות ניתן להניח כי הקלט תקין, אלא אם כן מצוין אחרת.
 - בכתיבת התכניות יש להשתמש רק במרכיבי השפה שנלמדו בקורס זה.
- **שימו לב, התשובות לכל השאלות צריכות להיכתב על גבי שאלון הבחינה. תשובות שייכתבו במקום אחר לא ייבדקו!**

חומר עזר המותר בשימוש הוא:

1. חוברת השקפים של הקורס

2. ספר הלימוד Java Software Solutions

אסור להשתמש במחשב מכל סוג שהוא!

שאלה 1 - 60 נקודות

בשאלה זו נספק לכם מספר מחלקות שלמות ומספר מחלקות שאתם תצטרכו להשלימן לפי המתואר להלן.

השיטות הכתובות בתיאור המחלקה ממומשות כבר. כלומר, כשכתוב {...} הכוונה היא שגוף השיטה כתוב ואינכם צריכים להשלימו. אתם יכולים להניח שהשיטה נכונה ומתבצעת כמתואר. שימו לב, הפתרונות לשאלות אינם תלויים זה בזה, וניתן לפתור כל אחת מהשאלות, גם אם לא פתרנו את קודמותיה.

מערך הבחינות באוניברסיטה הפתוחה הוא מורכב מאוד. ההנהלה החליטה למחשב את מרכזי הבחינות.

לצורך כך נגדיר שלוש מחלקות:

המחלקה Date שתייצג תאריך;

המחלקה Exam שתייצג בחינת גמר בקורס;

המחלקה ExamsCenter שתייצג מרכז בחינות.

המחלקה Date מייצגת תאריך לפי יום, חודש ושנה.

את המחלקה Date כבר כתבנו, ואתם יכולים להשתמש בה לפי הממשק והתיאור שאנו מביאים:

למחלקה Date יש את התכונות הפרטיות (instance variables) הבאות:

- int _day – שמייצגת את היום;

- int _month – שמייצגת את החודש;

- int _year – שמייצגת את השנה;

למחלקה Date הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שלושה פרמטרים (יום, חודש ושנה) של התאריך.

```
public Date(int d, int m, int y)
```

אפשר להניח שהפרמטרים נכונים והתאריך שמתקבל הוא חוקי.

- השני - בנאי העתקה המקבל תאריך אחר, ומעתיק את ערכיו.

```
public Date (Date d)
```

כמו כן, הוגדרו פעולות האחזור:

```
getDay(), getMonth(), getYear().
```

והפעולות הקובעות :

`setDay(int x), setMonth(int x), setYear(int x).`

שיטות נוספות במחלקה :

- השיטה `equals` המקבלת כפרמטר תאריך מסוים ובודקת אם הוא זהה לתאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה. אם התאריכים זהים השיטה מחזירה `true` ואחרת `false`.
- השיטה `before` המקבלת כפרמטר תאריך מסוים ובודקת אם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **קודם** לתאריך שהתקבל כפרמטר. אם הוא קודם השיטה מחזירה `true` ואחרת `false`.
- השיטה `after` המקבלת כפרמטר תאריך מסוים ובודקת אם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **מאוחר** מהתאריך שהתקבל כפרמטר. אם הוא מאוחר השיטה מחזירה `true` ואחרת `false`.
- השיטה `toString` מחזירה מחרוזת תווים המייצגת את התאריך כך : `day / month / year`

המחלקה Exam מייצגת בחינה.

למחלקה Exam התכונות הפרטיות (instance variables) הבאות :

- `String _course` – שם הקורס
- `Date _date` – תאריך הבחינה
- `int _noOfStudents` – מספר הסטודנטים שנרשמו להיבחן בבחינה זו

למחלקה Exam יש שני בנאים :

- בנאי אחד שמקבל את הפרמטרים הבאים : שם הקורס, שלושה מספרים שלמים המהווים את תאריך הבחינה (יום, חודש ושנה) ומספר שלם נוסף המייצג את מספר הסטודנטים שנבחנו בה.
- בנאי העתקה המקבל בחינה אחרת, ומעתיק את ערכיה.

במחלקה הוגדרו פעולות האחזור לפי השמות המקובלים.

כמו כן הוגדרו השיטות הבוליאניות הבאות :

- `sameCourse` המקבלת כפרמטר שם של קורס ומחזירה `true` אם השם זהה לשם של הקורס של הבחינה עליה מופעלת השיטה, ו-`false` אחרת ;
- `sameDate` המקבלת כפרמטר תאריך ומחזירה `true` אם הוא זהה לתאריך של הבחינה עליה מופעלת השיטה, ו-`false` אחרת ;
- `isValidName` המחזירה `true` אם שם הקורס מורכב מאותיות קטנות אנגליות בלבד, ו-`false` אחרת.

סעיף א (30 נקודות) :

לפניכם המימוש ב- Java של המחלקה Exam. השלימו את הדרוש, במקומות שמסומנים בקווים.
מספר השורות הריקות לא זהה בהכרח למספר השורות בפתרון.

```
public class Exam
{
    private _____
    private _____
    private _____

    public Exam(_____)
    {
        _____
        _____
        _____
    }

    public Exam (_____ other)
    {
        _____
        _____
        _____
    }

    public _____ getDate()
    {
        _____
    }

    public String _____
    {
        _____
        _____
    }

    public _____
    {
        return _noOfStudents;
    }
}
```

```

public boolean sameCourse (String name)
{
    _____
}

public boolean sameDate (Date date)
{
    _____
}

public boolean isValidName()
{
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
    _____
}
}

```

המחלקה ExamsCenter מייצגת מרכז בחינות ובו מתקיימות בחינות שונות.

הייצוג נעשה על-ידי מערך ששומר את רשימת הבחינות. התכונות במחלקה הן:

- מערך של הבחינות
Exam [] _center
- מספר הבחינות במרכז
int _noOfExams

כמו כן קיים במחלקה קבוע שלם MAX_EXAMS המציין את המספר המקסימלי של בחינות במרכז – 100.

הבחינות (כלומר האובייקטים מהמחלקה Exams) נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

סעיף ב (30 נקודות):

הנה המימוש ב-Java של המחלקה ExamsCenter. מימוש המחלקה כולל רק את הסעיפים שלהלן:

1. הגדרת הקבועים של המחלקה.
2. הגדרת התכונות של המחלקה.
3. בנאי שמאתחל את תכונות המחלקה כך שהמערך יהיה בגודל מקסימלי.
4. שיטה (addExam) בוליאנית המוסיפה בחינה למרכז. השיטה מקבלת את הבחינה כפרמטר. השיטה מחזירה ערך true אם ההוספה התבצעה כשורה, אם לא, השיטה תחזיר false.
5. שיטה (howManyExamsBetweenDates) שמקבלת כפרמטרים שני תאריכים ומחזירה מספר האומר כמה בחינות יתקיימו במרכז הבחינה בין התאריכים האלו (כולל). אפשר להניח שהתאריכים תקינים, אך לא בהכרח התאריך הראשון (הפרמטר הראשון) קודם לשני.
6. שיטה (mostPopularDate) שמחזירה את התאריך המבוקש ביותר על ידי הסטודנטים. כלומר, התאריך בו נבחנים הכי הרבה סטודנטים.
7. שיטה (deleteExams) המקבלת כפרמטר שם של קורס, ומוחקת ממערך הבחינות את כל הבחינות שיש בקורס הזה.

להלן המימוש ב-Java של המחלקה ExamsCenter. השלימו את הדרוש, במקומות שמסומנים בקווים. **מספר השורות הריקות לא זהה בהכרח למספר השורות בפתרון.**

```
public class ExamsCenter
{
    private final _____

    private _____
    private _____

    public ExamsCenter()
    {
        _____
        _____
    }
}
```

[illegible]

שאלה 2 - 20 נקודות

נתונה השיטות הבאות:

```
public int fun1(String s, int f)
{
    int ans = f;
    for(int i =f+1; i<s.length(); i++)
        if (s.charAt(ans) > s.charAt(i))
            ans = i;
    return ans;
}

public String fun2(String s, int i)
{
    char value = s.charAt(i);
    while (i > 0 && s.charAt(i-1) > value) {
        s=s.substring(0,i)+s.charAt(i-1)+s.substring(i+1);
        i = i-1;
    }
    s=s.substring(0,i)+value+s.substring(i+1);
    return s;
}

public String something(String s)
{
    for(int i=0; i<s.length()-1; i++) {
        int m = fun1(s, i);
        s=fun2(s, m);
    }
    return s;
}
```

להזכירכם:

- השיטה `length()` מחזירה את אורך המחרוזת עליה היא פועלת. לדוגמא, אם `s = "abcdef"` אז אורכה הוא 6
- השיטה `charAt(i)` מחזירה את התו ה-`i` במחרוזת עליה היא פועלת. לדוגמא, אם `s = "abcdef"` ו-`i=2`, אז התו שיוחזר הוא 'c'
- השיטה `substring(i)` מחזירה את התת-מחרוזת החל במקום ה-`i` ועד לסוף המחרוזת עליה היא פועלת. לדוגמא, אם `s = "abcdef"` ו-`i=2`, אז התת-מחרוזת שתוחזר היא "cdef".
- השיטה `substring(i, j)` מחזירה את התת-מחרוזת החל במקום ה-`i` ועד למקום `j` (לא כולל) במחרוזת עליה היא פועלת. לדוגמא, אם `s = "abcdef"` ו-`i=2`, `j=4` אז התת-מחרוזת שתוחזר היא "cde".

ענו על שלושת הסעיפים הבאים:

(4 נק') (i) מה מבצעת השיטה fun1 באופן כללי בהינתן לה מחרוזת תווים s ומספר שלם f כלשהו? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות, וכן שהפרמטר f מקיים $0 \leq f < s.length()$.

התשובה היא:

(6 נק') (ii) מה מבצעת השיטה fun2 באופן כללי בהינתן לה מחרוזת תווים s ומספר שלם i כלשהו? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות, וכן שהפרמטר i מקיים $0 \leq i < s.length()$.

התשובה היא:

(10 נק') (iii) מה מבצעת השיטה something באופן כללי בהינתן לה מחרוזת תווים s? שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. אתם יכולים להניח שהמחרוזת מכילה אך ורק אותיות אנגליות קטנות.

התשובה היא:

שאלה 3 - 20 נקודות -

במחלקה A מוגדרות השיטות הבאות:

```
public class A
{
    public void f (int n)
    {
        for (int i=n; i>0; i--)
            g(i);
    }

    public void g (int n)
    {
        for (int i=0; i<n; i++)
            h(i);
    }

    public void h (int n)
    {
        for (int i=0; i<n; i++)
            System.out.print ( " ? " );
        System.out.println();
    }
}
```

ענו על השאלות הבאות בהנחה שהאובייקט a הוא מהמחלקה A. כל תשובה 5 נקודות.

1. כמה סימני שאלה, וכמה שורות (כולל שורות ריקות) יודפסו על המסך בעקבות הפעלת הפקודה `a.g(4)` ?

2. כמה סימני שאלה, וכמה שורות (כולל שורות ריקות) יודפסו על המסך בעקבות הפעלת הפקודה `a.g(3)` ?

3. כמה סימני שאלה, וכמה שורות (כולל שורות ריקות) יודפסו על המסך בעקבות הפעלת הפקודה `a.f(1)` ?

4. כמה סימני שאלה, וכמה שורות (כולל שורות ריקות) יודפסו על המסך בעקבות הפעלת הפקודה `a.f(5)` ?

ב ה צ ל ח ה