

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, כל התשובות לשאלות צריכות להיכתב על גבי שאלון הבחינה. תשובה שתיכתב במקום אחר לא תיבדק!

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 שאתם תצטרכו להשלימן לפי המתואר להלן.

חלק מהשיטות 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 פתרם את קודמותיה.

ברצוננו לבנות מערכת עזרה לניהול מערכת שעות במכללה.

לשם כך כתבנו מחלקות לפי המתואר:

המחלקה Clsrm שתייצג כיתת לימוד;

המחלקה Course שתייצג קורס במכללה;

המחלקה College שתייצג את מערכת השעות של המכללה.

המחלקה Clsrm מייצגת כיתת לימוד;

למחלקה Clsrm התכונות (instance variables) הבאות:

- מספר הכיתה - שדה מסוג int ומאוחסן בו מספר תלת-ספרתי, כאשר ספרת המאות מציינת את הקומה בה הכיתה נמצאת. לדוגמא, אם מספר הכיתה הוא 317, היא נמצאת בקומה 3. אם המספר קטן מ-100, הכיתה נמצאת בקומת קרקע.
- מספר המקומות בכיתה – שדה מסוג int.
- האם הכיתה פנויה – שדה מסוג boolean.

למחלקה Clsrm הוגדרו שני בנאים (constructors):

1. בנאי אחד המקבל את השדות ומציב אותם לתכונות המחלקה. אפשר להניח שהפרמטרים נכונים וחוקיים. חתימתו היא:

```
public Clsrm(int num, int maxPlaces, boolean isAvailable)
```

2. בנאי שני שהוא בנאי העתקה (copy constructor) המקבל כיתה אחרת, ומעתיק את תכונותיה. חתימתו היא:

```
public Clsrm(Clsrm c)
```

כמו כן, הוגדרו פעולות אחזור (get) על כל אחת מהתכונות:
getClassNum(), getClassPlaces(), getAvailable()

הפעולה הקובעת (set) הוגדרה עבור התכונה הבוליאנית בלבד (שימו לב – את מספר הכיתה ואת מספר המקומות לא ניתן לשנות!).

- setAvailable(boolean avail)

שיטות בוליאניות נוספות שהוגדרו במחלקה:

- handicappedSuitable()

השיטה מחזירה true אם הכיתה מתאימה לנכים, ו- false אחרת. נכה יכול ללמוד בכיתה אם היא בקומת קרקע.

- equals(Clsrcm other)

השיטה מחזירה true אם הכיתה שעליה מופעלת השיטה זהה בגודלה לכיתה שהועברה כפרמטר לשיטה, ו- false אחרת. גודל מבחינת מספר המקומות בכיתה.

- isBigger(Clsrcm other)

השיטה מחזירה true אם הכיתה שעליה מופעלת השיטה גדולה יותר מהכיתה שהועברה כפרמטר לשיטה, ו- false אחרת. גדולה מבחינת מספר המקומות בכיתה.

- isSmaller(Clsrcm other)

השיטה מחזירה true אם הכיתה שעליה מופעלת השיטה קטנה יותר מהכיתה שהועברה כפרמטר לשיטה, ו- false אחרת. השיטה isSmaller חייבת להשתמש אך ורק בשיטה isBigger (היא לא יכולה לחשב את התשובה לפי ערכי התכונות, ולא יכולה להשתמש בשיטה equals).

- isHigher(Clsrcm other)

השיטה מחזירה true אם הכיתה שעליה מופעלת השיטה גבוהה יותר מהכיתה שהועברה כפרמטר לשיטה, ו- false אחרת. גבוהה מבחינת הקומה.

סעיף א (20 נקודות):

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

```
public class Clsrcm
{
    private int _classNum;
    private int _classPlaces;
    private boolean _isAvailable;
```

```

public Clsrm (int num,int maxPlaces, boolean isAvail)
{
    _____
    _____
    _____
}

public Clsrm (Clsrm c)
{
    _____
    _____
    _____
}

public int getClassNum()
{
    _____
}

public int getClassPlaces() {...}
public boolean getAvailable() {...}

public void setAvailable (boolean avail)
{
    _____
}

public boolean handicappedSuitable()
{
    _____
    _____
    _____
    _____
}

public boolean equals (Clsrm other) {...}

```

```
public boolean isBigger (Clsrm other)
{
    _____
    _____
    _____
    _____
}

public boolean isSmaller (Clsrm other)
{
    _____
    _____
    _____
    _____
}

public boolean isHigher (Clsrm other)
{
    _____
    _____
    _____
    _____
}

} //end of class Clsrm
```

המחלקה Course מייצגת קורס במכללה;

לכל קורס יש את התכונות הבאות:

- שם הקורס – `_courseName` שדה מסוג `.String`.
- מספר הסטודנטים שלומדים בקורס – `_numOfStudents` שדה מסוג `int` שמציין את מספר הסטודנטים שרשומים לקורס.
- הכיתה בה נלמד הקורס – `_courseClsrm` שדה מסוג `.Clsrm`.

במחלקה Course הוגדר **בנאי** שמקבל שני נתונים על הקורס (שם ומספר סטודנטים) ומציב אותן בתכונות המחלקה. שימו לב שהבנאי צריך לבדוק שמספר הסטודנטים לא שלילי. אם לא, יוצב הערך 1- בשדה זה כדי לציין ערך לא חוקי. **הבנאי לא קובע באיזו כיתה הקורס יילמד, כלומר, יציב null בשדה זה.**

המחלקה מכילה שיטות get ו-set כמקובל.

- השיטה getClsrcm תחזיר אובייקט מסוג Clsrcm שמייצג את הכיתה בה נלמד הקורס. אם לקורס לא קיימת כיתה השיטה תחזיר null.
- שימו לב שהשיטה setClsrcm תקבל אובייקט מסוג Clsrcm שמייצג את הכיתה בה נלמד הקורס, תבדוק אם הכיתה מתאימה לקורס (לפי מספר הסטודנטים בקורס ומספר המקומות בכיתה) ותעדכן את שדה המחלקה בהתאם. השיטה צריכה לעדכן גם את האובייקט Clsrcm כך שהכיתה לא תהיה פנויה יותר. אם הכיתה לא פנויה, או אם היא לא מתאימה לא ייעשה כלום.

שיטות בוליאניות נוספות במחלקה Course:

- `clsrcmSuitable(boolean handi)`
השיטה מקבלת פרמטר בוליאני שערכו true אם יש סטודנטים נכים הרשומים לקורס, ו-false אם אין. השיטה מחזירה true אם הכיתה שהוקצתה לקורס מתאימה לנכים (אם יש כאלו הרשומים לקורס), ו-false אחרת. אם אין סטודנטים נכים הרשומים לקורס יוחזר true; אם לא הוקצתה בכלל כיתה לקורס, יוחזר false.
- `morePopular(Course c)`
השיטה מקבלת כפרמטר קורס אחר. השיטה מחזירה true אם הקורס שעליו מופעלת השיטה פופולרי יותר (מבחינת מספר סטודנטים) מהקורס שהועבר כפרמטר, ו-false אחרת.

סעיף ב (20 נקודות):

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

```
public class Course
{
    private String _courseName;
    private int _numOfStudents;
    private Clsrcm _courseClsrcm;
```

```

public Course(_____)
{
    _courseName = name;
    if _____
        _____
    else
        _numOfStudents = -1;
    _courseClsrm = _____
}

public int getNumOfStudents () {...}
public String getCourseName() {...}

public Clsrm getClsrm() {
    return _____
}

public void setClsrm (Clsrm cls) {
    _____
    _____
    _____
    _____
    _____
}

public boolean morePopular (Course c) {
    _____
    _____
    _____
    _____
    _____
}

```

[illegible]

המחלקה College מייצגת את מערכת השעות של המכללה;

למחלקה College יהיו התכונות הבאות –

- מערך של הכיתות הקיימות במכללה (ניתן להניח שבמכללה יש בדיוק 20 כיתות).
- מערך של הקורסים הקיימים במכללה (ניתן להניח שבמכללה יש בדיוק 20 קורסים).

למחלקה הוגדר בנאי שמאתחל את המערכים בכיתות ובקורסים המתאימים. **אינכם צריכים לממש בנאי זה. הניחו שהוא כבר ממומש.**

השיטות שהוגדרו במחלקה הן:

- `public Course mostPopular()`

השיטה צריכה להחזיר את הקורס הכי פופולרי שיש במערכת. אם יש כמה קורסים כאלו, יוחזר הראשון מהם במערך.

- `public Clsrm biggestClass()`

השיטה צריכה להחזיר את הכיתה הכי גדולה שיש במערכת. אם יש כמה כיתות כאלו, תוחזר הכיתה הראשונה מהן במערך.

- `public int howManyHandi()`

השיטה צריכה להחזיר את מספר הכיתות במערכת שהן נגישות לנכים, והן פנויות.

- `public int howManyOnTopFloor()`

השיטה צריכה להחזיר את מספר הכיתות ששוכנות בקומה הגבוהה ביותר במכללה. שימו לב שעליכם לבדוק גם מהי הקומה הגבוהה ביותר.

סעיף ג (20 נקודות):

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

```
public class College
{
    public final _____
    private Course [] _courses;
    private Clsrm [] _clsrms;

    public College()      {...}

    public Course mostPopular()
    {
        int index = 0;
        for (_____)
        {
            if _____
                _____
                _____
        }
        return _____
    }
}
```

```
public Clsrm biggestClsrm()  
{
```

```
}
```

```
public int howManyHandi()  
{
```

```
}
```

```
public int howManyOnTopFloor()  
{
```

```
}
```

```
} //end of class College
```

שאלה 2 - 20 נקודות

כתבו שיטה שמקבלת כפרמטרים מערך המלא במספרים שלמים ומספר שלם x ומחזירה `true` אם יש במערך זוג איברים שההפרש ביניהם גדול מ- x , ו-`false` אחרת.

חתימת השיטה היא:

```
public boolean difference (int [] a, int x)
```

שימו לב, שיטה שתבצע את הנדרש תוך כדי שימוש ביותר מלולאה אחת, תקבל לכל היותר 10 נקודות.

כתבו כאן את התשובה:

שאלה 3 - 20 נקודות -

לפניכם השיטות הבאות, כולן במחלקה אחת בשם Test :

```
public static boolean f (int[] a, int x)
{
    for (int i=0; i< a.length; i++)
        if (a[i] == x)
            return true;
    return false;
}

public static int g (int [] a, int num)
{
    int s = 0;
    boolean flag = true;
    for (int i=1; (i< a.length) && flag; i++)
    {
        if (f(a, num*i))
            s++;
        else
            flag = false;
    }
    return s;
}

public static int what (int []a)
{
    int m=0;
    int num = a[0];
    for (int i=0; i< a.length; i++)
    {
        int c = g(a, a[i]);
        if (c > m)
        {
            m = c;
            num = a[i];
        }
    }
    return num;
}
```

אתם יכולים להניח שהמערך a מלא במספרים שלמים.

הקיפו בעיגול את כל הטענות הנכונות שכתובות להלן:

- א. אם נריץ את השיטה f על המערך $\{6, 18, 25, 9, 8, 10, 4, 20, 1, 15\}$ והמספר $x = 4$, נקבל את התשובה `true`.
- ב. אם נריץ את השיטה g על המערך $\{9, 8, 7, 6, 5, 4, 3, 2\}$ והמספר $num = 4$, נקבל את התשובה 2.
- ג. אם נריץ את השיטה g פעם אחת על המערך $\{6, 18, 25, 9, 8, 10, 4, 20, 2, 1, 5\}$ והמספר $num = 5$ ופעם שניה על המערך $\{6, 18, 25, 9, 12, 10, 4, 20, 3, 15, 2\}$ והמספר $num = 4$ נקבל אותה תשובה.
- ד. אם נריץ את השיטה `what` פעם אחת על המערך $\{9, 8, 7, 6, 5, 4, 3, 2\}$ ופעם שניה על המערך $\{6, 18, 25, 9, 12, 10, 4, 20, 3, 15, 2\}$ נקבל אותה תשובה.
- ה. אם נריץ את השיטה `what` פעם אחת על המערך $\{7, 18, 5, 9, 8, 10, 4, 20, 2, 13, 25\}$ ופעם שניה על המערך $\{5, 10\}$ נקבל אותה תשובה.

ב ה צ ל ח ה