

## תרגיל בית מס' 0

### הנחיות לפתרון תרגילי הבית

על הקוד המוגש להיות מתועד היטב ועליו לכלול:

- מפרט, כפי שהודגם בהרצאה ובתרגול.
- תיעוד של כל מחלקה ומתודה ושל קטעי קוד רלוונטיים.
- במידת הצורך, יש להוסיף תיעוד חיצוני.

### הנחיות להגשת תרגילי בית

- תרגילי הבית 1-4 הם חובה.
- ההגשה בזוגות בלבד.
- עם סיום פתירת התרגיל, יש להגיש את:
  - כל קבצי הקוד והתיעוד.
  - פתרון לשאלות ה"יבשות" בקובץ pdf אחד. על הקובץ להכיל את מספרי תעודות הזהות של שני הסטודנטים המגישים.
- הקובץ הכולל את התיעוד יקרא `hwo_<id1>_<id2>.pdf` כאשר `<id1>` ו-`<id2>` הם מספרי הזהות של הסטודנטים המגישים (לדוגמא `hwo_12345678_9876541.pdf`, כמובן יש להשתמש במספרי הזהות שלכם).
- יש לארוז את קובץ ה-`pdf` ואת קבצי הקוד בתוך קובץ `zip` בשם `hwo_<id1>_<id2>.zip` כאשר `<id1>` ו-`<id2>` הם מספרי הזהות של הסטודנטים המגישים (לדוגמא `hwo_12345678_9876541.zip`, כמובן יש להשתמש במספרי הזהות שלכם).
- שני הסטודנטים המגישים צריכים להצטרף לקבוצה במשאב "בחירת קבוצות לתרגיל בית 0" במודל, ולאחר מכן על אחד מבני הזוג להגיש את קובץ ה-`zip` במשאב "תרגיל בית 0".

### **הקפידו על הוראות ההגשה על מנת למנוע אי נעימות מיותרת.**

- הגשת התרגיל היא אלקטרונית בלבד, דרך אתר הקורס ע"י אחד מבני הזוג בלבד.
- תרגיל שיוגש באיחור וללא אישור מתאים (כגון, אישור מילואים), יורד ממנו ציון באופן אוטומטי לפי חישוב של 5 נקודות לכל יום איחור ועד 2 ימי איחור שלאחריהם לא תתאפשר הגשה כלל.
- על הקוד המוגש לעבור הידור (קומפילציה). על קוד שלא עובר הידור יורדו 30 נקודות.
- על הקוד להיות מוגש על פי ההנחיות המפורטות כאן. תרגיל שלא יוגש בצורה הנדרשת, לא ייבדק.
- שאלות על תרגילי הבית – בפורום הרלוונטי באתר הקורס.

מועד ההגשה:  
יום ב', 5.5.2025

המטרות של תרגיל בית זה הן:

- ליצור בסיס משותף שממנו נתחיל את הקורס.
- להכיר את שפת Java ואת סביבת העבודה IntelliJ.
- להתנסות במימוש מפרט בסיסי.

## שאלה 1 (10 נקודות)

כתבו מסמך באנגלית המכיל רשימת כללים לאופן כתיבת קוד (coding style guidelines). עליכם להדגים את היישום של כללים אלה בשאלות הבאות ולהשתמש בהם באופן עקבי בכל תרגילי המחשב בהמשך הקורס. על רשימת הכללים לכלול:

- אופן נתינת שמות למחלקות, מתודות ומשתנים (naming conventions).
  - אורך שורה, צורת האינדנטציה והעימוד, רווחים ושורות ריקות.
  - צורת כתיבת הערות ותיעוד.
  - כל מידע רלוונטי אחר.
- במידת הצורך, יש לרשום במסמך קטעי קוד קצרים להדגמה.  
ניתן למצוא דוגמאות למסמכים המכילים רשימת כללים באתרים הבאים:

<https://geosoft.no/javastyle.html>  
<https://coderanch.com/wiki/718799/Style>  
<https://google.github.io/styleguide/javaguide.html>

להגשה "יבשה": מסמך הכללים הנ"ל.

## שאלה 2 (25 נקודות)

לצורך פתרון שאלה זו יש לקרוא את החומר לקריאה עצמית בנושא קלט/פלט.

עליכם לכתוב תוכנית שתקרא קובץ טקסט ותדפיס סטטיסטיקות על המספרים שמופיעים בו. התוכנית תקבל את שם הקובץ בשורת הפקודה (command line). אם לא סופק קובץ או שסופק קובץ שלא קיים, יש להדפיס הודעת שגיאה מתאימה. הקובץ יכיל מספרים שלמים (כולל שליליים), כאשר כל שורה יכולה להכיל מספר מספרים מופרדים ברווחים או באחד מהתווים הבאים: ' , . : ; { } [ ] = + - \_ ! @ # \$ % ^ & \* ( )

- מספר המספרים בקובץ.
- הממוצע של מספרי השורה (כל שורה בנפרד) עם שתי ספרות אחרי הנקודה.
- סטטיסטיקה כוללת של כמה אחוזים (עם שתי ספרות אחרי הנקודה) מהמספרים הם
  - חיוביים
  - שליליים
  - אפס

הנחיה 1: ניתן לטפל במקרי קצה שונים בכל דרך הגיונית.  
הנחיה 2: ניתן להיעזר במחלקות `java.io.BufferedReader`, `java.io.FileReader`, `java.util.StringTokenizer`, `Integer.parseInt` ובמתודה `String.format()` או `System.out.printf()`

להגשה ממוחשבת: קובץ המכיל את הפתרון (כולל תיעוד).  
להגשה "יבשה": פלט דוגמה של התוכנית על קבצי שני קבצי הטקס המצורפים לתרגיל.

#### שאלה 4 (45 נקודות)

נתונים הקבצים Book.java ו-Bookshelf.java המכילים מפרט ללא מימוש של מחלקות המייצגות ספר וספרייה של ספרים בהתאמה.

א.

ממשו את המחלקות כך שיעמדו במפרט הנתון וכתבו תכנית בדיקה קצרה שתדגים את פעולתן.

**הנחיה:** ניתן ומומלץ להיעזר במחלקה java.util.ArrayList. כדי להגדיר, למשל, רשימה של איברים מטיפוס String ניתן לרשום:

```
List<String> myList = new ArrayList<>();
```

ב.

לפי המפרט של המתודה getTotalPages() במחלקה Bookshelf, על מתודה זו להחזיר את סכום כל מספרי הדפים בספרייה. קיימות שתי דרכים למימוש מתודה זו – דרך א' היא לסכום את מספר הדפים בהכנסת ספר חדש לספרייה, ודרך ב' היא לבצע את כל הסכימה בעת הפעלת המתודה getTotalPages(). בסעיף א' בחרתם אחת משתי מדרכים אלה. בסעיף זה ממשו את המחלקה Bookshelf בדרך השנייה וקראו לקובץ Bookshelf2.java. הסבירו את היתרונות והחסרונות של כל אחד מהמימושים.

ג.

הוחלט להוסיף למפרט של המתודה addBook() במחלקה Bookshelf את הפסקה הבאה:

```
@requires ball != null
```

1. האם שינוי זה ידרוש שינויים נוספים במפרט של המתודה addBook()? הסבירו.

2. אילו שינויים ידרשו במימוש של מתודה זו כדי שתעמוד במפרט החדש? הסבירו.

3. האם המפרט החדש חזק יותר או חלש יותר כעת? הסבירו.

ד.

נתון כעת את הקובץ Library.java המכיל מפרט ללא מימוש של מחלקה המייצגת מדף. בדומה למדף הספרייה יכולה להכיל ספרים אך ההבדל הוא של - ספרייה יש כמות מוגבלת של דפים שהיא יכולה לאחסן, כלומר סכום דפי הספרים חסום מלמעלה. חסם זה מועבר לבנאי בעת יצירת האובייקט שהוא מופע של Library.

**הנחיה:** יש להיעזר במימוש שכתבתם ל-B ללא שימוש בהורשה.

להגשה ממוחשבת: הקבצים Book.java, Library.java, Bookshelf.java (בשתי גרסאות – סעיף א' וסעיף ב') ותוכנית הבדיקה שכתבתם.

להגשה "יבשה": א. פלט תכנית הבדיקה. ב. תשובות לסעיפים ב' וג'.

## שאלה 4 (20 נקודות)

נתונות שתי מתודות:

```
public static int findFirst(int[] arr, int val) {  
    for (int i = 0; i < arr.length; i++)  
        if (arr[i] == val)  
            return i;  
    return arr.length;  
}
```

```
public static int findLast(int[] arr, int val) {  
    for (int i = arr.length - 1; i >= 0; i--)  
        if (arr[i] == val)  
            return i;  
    return -1;  
}
```

- א. כתבו מפרט כללי מספיק כך שיתאים לשתי המתודות הנתונות (לפי הפורמט שנלמד בכיתה עם פסקאות @requires, @modifies ו-@effects).
- ב. כתבו מפרט חזק יותר מהמפרט שכתבתם בסעיף א', שיתאים רק למתודה findLast().

להגשה "יבשה": תשובות לסעיפים א' וב'.

עבודה נעימה!

