

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 ד', 3.7.2024

המטרות של תרגיל בית זה הן:

- ליצור בסיס משותף שממנו נתחיל את הקורס.
- להכיר את שפת Java ואת סביבת העבודה Eclipse.
- 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 (30 נקודות)

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

עליכם לכתוב תוכנית שתקרא קובץ טקסט ותדפיס סטטיסטיקות על הטקסט שנמצא בו. התוכנית תקבל את שם הקובץ בשורת הפקודה (command line). אם הפרמטר שסופק אינו שם קובץ קיים או אם מספר הארגומנטים שסופקו שגוי, תודפס הודעת שגיאה מתאימה.

מילה מוגדרת כרצף תווים המופרדים ברווח או באחד מהתווים הבאים: ' , . : ; { } [] _ ! @ # \$ % ^ & * ()

התוכנית תדפיס:

- מספר המילים בקובץ
- מספר המילים הממוצע בשורה (עם שתי ספרות אחרי הנקודה העשרונית)
- רשימה של כמה אחוז (עם שתי ספרות אחרי הנקודה העשרונית) מהמילים בקובץ הן בעלות אות אחת, בעלות שתי אותיות, בעלות שלוש אותיות וכו'. יש לתמוך במילים באורך של עד 15 אותיות.

הנחיה 1 : ניתן לטפל במקרי קצה שונים בכל דרך הגיונית.
הנחיה 2 : ניתן להיעזר במחלקות `java.io.BufferedReader`, `java.io.FileReader`, `java.util.StringTokenizer` 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 הטקסט המצורפים לתרגיל.

שאלה 3 (40 נקודות)

נתונים הקבצים Ball.java ו-BallContainer.java המכילים מפרט ללא מימוש של מחלקות המייצגות כדור ומכל של כדורים בהתאמה.

א.

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

הנחיה: ניתן 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 getVolume() במחלקה BallContainer, על מתודה זו להחזיר את סכום כל נפחי הכדורים במכל. קיימות שתי דרכים למימוש מתודה זו – דרך א' היא לסכום את הנפחים בזמן הוספת כדור חדש למכל, ודרך ב' היא לבצע את כל הסכימה בעת הפעלת המתודה getVolume(). בסעיף א' בחרתם אחת משתי מדרכים אלה. בסעיף זה ממשו את המחלקה BallContainer בדרך השנייה. הסבירו את היתרונות והחסרונות של כל אחד מהמימושים.

ג.

הוחלט להוסיף למפרט של המתודה add() במחלקה BallContainer את הפסקה הבאה:

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

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

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

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

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

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

שאלה 4 (10 נקודות)

כעת נרצה לממש אוסף של מטבעות. אוסף של מטבעות מכיל מטבע יחיד מכל סוג (10 אג', 50 אג' וכו').

השתמשו במחלקות Coin ו-Wallet על מנת לממש מחלקה בשם CoinCollection המייצגת אוסף מטבעות ללא שימוש בהורשה. נתונים הקבצים CoinCollection.java, Coin, Wallet המכילים את מפרטי המחלקות ללא מימוש. השלימו את המימוש כך שיעמוד במפרטים הנתונים.

להגשה ממוחשבת: הקבצים CoinCollection.java, Coin.java, Wallet.java וקובץ בשם CoinCollectionTest.java אשר בודק את המימוש של CoinCollection.

שאלה 5 (10 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!

