מכללת ספיר
המחלקה למדעי המחשב

מבוא לתכנות מונחה עצמים

עבודת הגשה 3

מועד הגשה: 16/05/2023 בשעה 23:50

הוראות הגשה:

1. אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.
2. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
3. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
4. הגשה דרך מערכת לי-מוד בלבד. שום עבודה לא מתקבלת במייל!
5. עליכם לכתוב כל מחלקה בקובץ java נפרד. כל חלק של העבודה תהיה בתיקיה נפרדת! לאחר מכן, תכווצו את הכל לקובץ אחד בפורמט RAR או ZIP המכיל את שתי התיקיות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.
6. יש להוסיף הערות באנגלית בלבד בעזרת javadoc.
7. עליכם להגיש קוד שמתקמפל. בדקו זאת לפני ההגשה. קוד שלא מתקמפל יקבל ציון 0.
8. אסור להוסיף שדות למחלקה, ניתן להוסיף פונקציות עזר אבל עליהן להיות פרטיות.
9. יש להקפיד על תכנות נכון:
   - שימוש בקבועים/ סטטי במידת הצורך.
   - הקפדה על הזחות, כיתוב נכון וקריא
   - מתן שמות משמעותיים למשתנים ופונקציות.
10. שימו לב: לא יינתו הארכות זמן מעבר לתאריך ההגשה המצויין. מי שמעוניין יכול להגיש עבודה באיחור של עד יומיים (כלומר עד 18/05/23) עם הורדת ציון. כל יום איחור מוריד 5 נקודות מציון הסופי של העבודה.
11. שום עבודה לא תתקבל דרך האימייל!
12. שאלות ובקשות בקשר לעבודה ניתן להפנות למרצה, יעל וקסלר דרך פורום שנפתח באתר לשאלות בנושא העבודה או באימייל: yaelva@mail.sapir.ac.il.

עבודת בית 3
ממשקים, קבצים, מחלקה גנרית וחריגות

משימה 1: ממשקים ומחלקה גנרית

מימוש תור (queue)

במדעי המחשב, תור (queue) הוא מבנה נתונים מופשט המוגבל בגודלו ומייצג אוסף נתונים המסודר על פי סדר מוגדר.
התור מוגדר על ידי הפעולות הבאות:

- הכנסה (enqueue) - הוספת אובייקט אחד חדש בסופו של התור.
- הוצאה (dequeue) - הוצאתו של האובייקט הנמצא בראש התור.
- בדיקת מהו הערך בראש התור (peek).
- בדיקת מהו מספר האובייקטים בתור (size).
- בדיקה האם התור ריק (isEmpty).
- בדיקה האם התור מלא (isFull).

כל הפעולות מתבצעות בסיבוכיות O(1) מלבד dequeuer שמתבצעת בסיבוכיות O(size). פעולות ההכנסה וההוצאה בתור מבוססות על העיקרון נכנס ראשון - יוצא ראשון FIFO (first in first out).

עליכם לבנות את **המחלקה הגנרית** ArrayQueue הממשת את **הממשק הגנרי** Queue.
(שימו לב לקבצים המצורפים לעבודה).

```
public interface Queue<T> {
      public void enqueue(T o);
      public T dequeue();
      public T peek();
      public int size();
      public boolean isEmpty();
      public boolean isFull();
}
```

```
public class ArrayQueue<T> implements Queue<T>
{
    private T[] queue;
    private int size;

    ...
}
```

המחלקה מכילה שני שדות בלבד **<u>אסור</u>** להוסיף או להוריד שדות מחלקה.
בנו בנאי עבור המחלקה, המקבל את הפרמטר capacity המייצג את קיבולת התור. הבנאי צריך לאתחל את השדות במחלקה כאשר:

- השדה queue מייצג מערך אובייקטים (מסוג גנרי T) שקיימים בתור בגודל capacity. ביצירה התור יהיה ריק מאובייקטים.
- השדה size מייצג את כמות האובייקטים שקיימים בתור. ביצירת התור, מספר האובייקטים בתור הוא אפס.

המחלקה תממש את הממשק queue<T> על פי ההנחיות הבאות:

❖ <u>הכנסה (enqueue):</u>
מוסיף אלמנט לסוף התור, מחזיר true אם האלמנט נוסף בהצלחה אחרת מחזיר false.
שימו לב לא ניתן להוסיף אובייקט לתור אם התור מלא.
כלומר, אם הקיבולת של התור הוא 20 ומספר האובייקטים בתוכו הוא 20, אז כאשר מנסים להכניס אובייקט נוסף לתור השיטה מחזירה false מפני שהתור מלא ולא ניתן להכניס אובייקטים נוספים לתור

❖ <u>הוצאה (dequeue) - הוצאתו של האובייקט הנמצא בראש התור:</u>
מסיר אלמנט מראש התור (במקום האפס של המערך) ומחזיר אותו, אם התור ריק יש להחזיר null אחרת יש לעדכן את השדות queue ו-size בהתאם כך שהאיבר הראשון במערך יהיה האיבר הראשון כעת בתור. כלומר, צריך לבצע הזזה של כל האובייקטים במערך שמאלה במערך queue .

❖ <u>בדיקת מהו הערך בראש התור (peek):</u>
מחזיר את האלמנט שנמצא בראש התור מבלי להסיר אותו, אם התור הוא ריק מחזירה null.
שימו לב התור לא משתנה.

❖ <u>בדיקת מהו מספר האובייקטים בתור (size):</u>
מחזיר את מספר האלמנטים שקיימים כעת בתור, אם התור ריק מחזיר 0.

❖ <u>בדיקה האם התור ריק (isEmpty):</u>
מחזיר true אם התור ריק, אחרת מחזיר false.

❖ <u>בדיקה האם התור מלא (isFull):</u>
מחזיר true אם התור מלא, אחרת מחזיר false. תור יחשב מלא אם מערך האובייקטים מלא. כלומר מספר האובייקטים במערך שווה לאורך המערך.

דוגמא:

המערך שמצוייר הוא השדה queue של המחלקה והתא השמאלי ביותר הוא אפס (האינדקס שלו 0)
התא מימינו הוא התא האחד וכך הלאה. התור בדוגמא בעל קיבולת 6, ביצירתו התור ריק

לאחר פעולת enqueue(7)

7					

לאחר פעולת enqueue(12)

7	12				

לאחר פעולת enqueue(8)

7	12	8			

לאחר פעולת enqueue(102)

7	12	8	102		

לאחר פעולת enqueue(12)

7	12	8	102	12	

לאחר פעולת dequeue(), מוחזר הערך 7

12	8	102	12		

פעולת peek(), תחזיר את הערך 12

12	8	102	12		

לאחר פעולת dequeue(), מוחזר הערך 12

8	102	12			

פעולת size(), תחזיר את הערך 3

8	102	12			

פעולת isFull(), תחזיר את הערך false

8	102	12			

פעולת isEmpty(), תחזיר את הערך false

8	102	12			

משימה 2 : קבצים וחריגות

סוכנות רכבים

בתרגיל זה אתם מתבקשים לבנות תוכנה המדמה מערכת המנהלת סוכנות רכבים.
כל הרכבים שנמכרים בסוכנות הם רכבים ששנת היצור שלהם היא לפחות 2017.
לכל רכב במגרש יש את הנתונים הבאים:

- מספר רכב (מספר בן 6 ספרות)
- שנת ייצור (מספר בין 2017-2023)
- שם היצרן
- קילומטראז'
- מחיר

עליכם לממש את המחלקות הבאות:

1. **מחלקת רכב (Car)** המוגדרת על פי המאפיינים שפורטו לעיל.

המחלקה תכיל את השיטות הבאות:

- בנאי המקבל את כל הפרמטרים –
בבנאי עליכם לבדוק תקינות הפרמטרים ובמידת הצורך תזרק חריגה עם הודעה מתאימה.
לדוגמה: המחיר חייב להיות חיובי, בכל מקרה שבו המחיר יהיה אי-חיובי תזרק חריגה שתציין שהמחיר חייב להיות חיובי.
- toString – מחרוזת עם פרטי הרכב כאשר בין כל נתון יש רווח אחד בלבד.
- שיטה המחשבת הורדת ערך לרכב – השיטה תקבל את אחוזי ההורדה ובהתאם לכך היא תבצע עדכון מחיר של הרכב.
במידה והפרמטר שהתקבל שלילי נזרוק חריגה. בנוסף, אם ההורדה במחיר אמורה להיות גבוהה מ5000 ש"ח לא נבצע את עדכון המחיר ונזרוק חריגה המציינת שהורדת הערך גבוה מדי.
- מכירת רכב – השיטה תקבל אובייקט מסוג PrintWriter הכותב לקובץ המתעד את כל הרכבים שנמכרו. השיטה תשמור את פרטי הרכב בקובץ על פי סדר הנתונים המתואר למעלה. כאשר בין הנתונים יש רווח אחד בלבד.

2. **מחלקת עובד (Employee)** המיישמת את הממשק Comparable.

עובד מוגדר על ידי המאפיינים הבאים:

- שם העובד
- ת.ז
- מספר מכירות

לכל עובד נספק את השיטות הבאות:

- בנאי המקבל את שם העובד ות"ז שלו. הבנאי יבדוק את תקינות הנתונים ובכל צורך תזרק חריגה עם הודעה מתאימה. שם העובד חייב להיות מורכב רק מאותיות ומספר ת.ז יהיה בעל 9 ספרות בדיוק.
- מכירת רכב – השיטה תקבל אובייקט מסוג רכב ואובייקט מסוג PrintWriter הכותב לקובץ המתעד את כל הרכבים שנמכרו. השיטה תעדכן את מספר המכירות של העובד ותמכור את הרכב ללקוח (שימו לב עליכם להפעיל את השיטה של מכירת רכב של מחלקת רכב).
- חישוב שכר – השיטה תחשב ותחזיר את השכר של העובד. שכר הבסיס של כל העובדים הוא 6000. ועבור כל מכירה שהם ביצעו הם מקבלים תוספת שכר של 100 ש"ח.
- toString – המחרוזת המשורשרת תכלול את כל פרטי העובד ובנוסף את השכר שלו.
- compareTo(Employee other) – השיטה תשווה בין שני עובדים ותחזיר 0 במידה ומספר המכירות שלהם זהה, 1- אם מספר המכירות של העובד האחר גדול יותר. אחרת תחזיר 1.

3. מחלקת סוכנות רכבים (CarDealership)

במחלקה הזו נממש שיטות גלובליות (public static)

- שיטה גנרית המקבלת מערך של אובייקטים (מסוג גנרי). השיטה תמיין את המערך מהגדול לקטן בעזרת פונקצית compareTo.
- תוכנית ראשית (main):

בתוכנית הראשית ניצור-

- אובייקט מסוג PrintWriter הכותב לקובץ בשם Sold.txt. קובץ זה מתעד את כל הרכבים שנמכרו.
- מערך של רכבים (Car) שנאתחל על פי הנתונים הכתובים בקובץ בשם CarDealership.txt. שימו לב: בתחילת הקובץ מופיע מספר המייצג את כמות הרכבים הכתובים בו.
- מערך של עובדים שנאתחל על פי הנתונים הכתובים בקובץ בשם Employee.txt. גם קובץ זה מתחיל ממספר העובדים ששמורים בקובץ.

לאחר מכן, נריץ תפריט:

1. הצגת רשימת העובדים בסוכנות.
2. הצגת הרכבים בסוכנות שטרם נמכרו.
3. מכירת רכב.
4. הוספת רכב לסוכנות.
5. סיום התוכנית.

אופציה 1 : הצגת רשימת העובדים בסוכנות –
התוכנית תציג את כל העובדים בסוכנות כך שסדר ההדפסות תהיה מהעובד עם מספר מכירות הגבוהה ביותר למספר העובד עם מספר המכירות הנמוך ביותר.

אופציה 2 : הצגת הרכבים בסוכנות שטרם נמכרו.
התוכנית תציג את פרטי כל הרכבים בסוכנות שטרם מכרו אותם.

אופציה 3: מכירת רכב –
תחילה תוצג בפני המשתמש רשימת העובדים (שם + ת.ז שלהם) לאחר מכן המשתמש יבחר עובד על ידי הקלדת הת.ז שלו.
במידה והת.ז שהמשתמש הקליד לא קיימת תזרק חריגה שהטיפול בה יהיה קליטה חוזרת.
שימו לב שכמות הפעמים שהמשתמש יכול להזין ת.ז תקין היא בלתי מוגבלת.
לאחר בחירת העובד התוכנית תציג למשתמש את רשימת הרכבים שטרם נמכרו. המשתמש יתבקש לבחור רכב על ידי הקלדת מספר הרכב. גם כן הגישה תהיה זהה לחוקיות הת.ז של העובד.
לאחר בחירת העובד והרכב נבצע מכירה של הרכב. מכירת הרכב תכלול עדכון של המספר המכירות של העובד הרלוונטי, שמירת פרטי הרכב בקובץ הרכבים הנמכרים ומחיקת הרכב ממערך הרכבים.

אופציה 4 : הוספת רכב לסוכנות-
המשתמש יכניס את פרטי הרכב (מספר רכב, שנת ייצור, שם היצרן, קילומטראז' ומחירו).
נוסיף את הרכב למערך הרכבים אם ניתן. כלומר אם הבנאי זרק חריגה מכוון שאחד מפרטי הרכב לא תקינים נודיע על כך למשתמש ונמשיך בתוכנית.

אופציה 5 : סיום התוכנית-
בסיום התוכנית, נשמור את פרטי הרכבים שטרם נמכרו בקובץ CarDealership.txt. ונסגור את האובייקטים הרלוונטיים.על מנת לשמור על אחידות, הקובץ CarDealership.txt יתחיל ממספר הרכבים השמורים בו ולאחר מכן יופיעו פרטי הרכבים.

נקודות חשובות:

- אסור להוסיף משתנים למחלקות. ניתן להוסיף פונקציות עזר ושיטות set/get הכרחיות. שיטות העזר יהיו בהרשאת private.
- הקבצים המכילים מידע על הרכבים בסוכנות ופרטי העובדים יכולים לכלול מידע לא תקין. כלומר, לדוגמה ת.ז של עובד יכול להיות מורכב מ8 ספרות ולא 9. במידה ולא ניתן ליצור אובייקט בגלל שהנתונים לא חוקיים רק את האובייקט הזה לא ניצור אבל כן נמשיך ביצירת האובייקטים האחרים שבאים לאחר מכן.
- בכל פעם שנמכר רכב נוסיף את פרטי הרכב לסוף הקובץ!
- בסיום התוכנית, נשמור את פרטי הרכבים שטרם נמכרו בקובץ המיועד לכך. שימו לב שפרטי הרכבים שמופיעים שם תחילה הם לא בהכרח רלוונטיים בסיום התוכנית ולכן צריך לדרוס את הנתונים המופיעים בו.
- **הקבצים המכילים את פרטי העובדים והרכבים בסוכנות מצורפים לעבודה.**