

פיתוח תוכנה מתקדם 2 – סמסטר קיץ מועד ב' תשפ"ד

תזכורת: כתובת מערכת הבדיקות: <https://cktest.cs.colman.ac.il/>. שם הקורס PTM2, מועד ב'. לאחר הורדת המבחן ממערכת הבדיקות. העתיקו את כל קובצי ה Java לתוך הפרויקט ב package בשם `test`. במבחן זה 3 שאלות, חובה לענות על כל 3 השאלות ולהגיש למערכת הבדיקות במוד הגשה סופית לפני סוף המבחן.

שאלה 1 – תכנות מקבילי באמצעות ת'רדים (30 נק')

בקובץ `MyCompletableFuture.java` עליכם לכתוב חיקוי קל של `CompletableFuture`. עליכם לממש את המחלקה `MyCompletableFuture<V>` עם המתודות הבאות:

- `set` תזין ערך מסוג `V` וכן תהווה טריגר לתגובת שרשרת (ראו בהמשך)
 - `thenApply` בהינתן `Function<V,R>` (פונקציה מטיפוס `V` לטיפוס `R`) היא תחזיר מיד אובייקט מסוג `MyCompletableFuture<R>` אשר עתיד להכיל אובייקט מסוג `R`.
 - כאשר יום יבוא ונזין ערך מסוג `V`, מיד תופעל עליו הפונקציה לעיל שמחזירה לנו `R` ואותו יש להזין לאותו `MyCompletableFuture<R>` שהחזרנו מבעוד מועד. דוגמה בהמשך.
 - `thenAccept` בהינתן `Consumer<V>` תגדיר את הפעולה אשר "תצרוך" את `V` לכשיוזן. דוגמה בהמשך.
 - `get` פשוט תחזיר את `V` ללא המתנה (כלומר יכול לחזור `null` אם בחרנו להחזיר לפני הזמן)
- כפי שלמדנו בכיתה, מימוש זה יאפשר לנו להגדיר מראש את שרשרת התגובה שתקרה לכשיוזן ערך מסוג `V`, מבלי שנצטרך להמתין לו עד שיחזור. לדוגמה:

```
MyCompletableFuture<String> fs=new MyCompletableFuture<>();
```

```
fs.thenApply(s->s.hashCode()).thenApply(i->i%2==0).thenAccept(System.out::println);
```

או בצורה מפורשת יותר:

```
MyCompletableFuture<String> fs=new MyCompletableFuture<>();
MyCompletableFuture<Integer> fi=fs.thenApply(s->s.hashCode());
MyCompletableFuture<Boolean> fb=fi.thenApply(i->i%2==0);
fb.thenAccept(b->System.out.println(b));
// after 2 years...
fs.set("hello world");
```

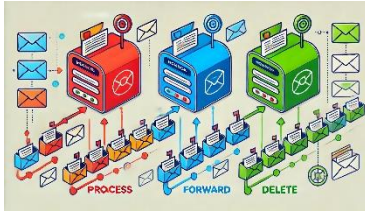
יצרנו את `fs` מסוג `MyCompletableFuture<String>`.

- הפעלנו מתוכו את `thenApply` אשר קבלה פונקציה שבהינתן מחרוזת היא מחזירה את ה `hashCode` שלה (מסוג `int`). לפיכך `thenApply` החזירה לנו מיד אובייקט (`fi`) שעתיד להחזיק `Integer`
- מתוך `fi` הפעלנו את `thenApply` אשר קבלה פונקציה שבהינתן `int` היא מחזירה האם הוא מתחלק ב 2 ללא שארית. לפיכך `thenApply` החזירה לנו מיד אובייקט (`fb`) שעתיד להחזיק `Boolean`.
- מתוך `fb` הפעלנו את `thenAccept` אשר קבלה פונקציה שבהינתן בוליאני היא מדפיסה את ערכו.

לאחר שנתיים ביצענו `set` ל `fs` והזנו את המחרוזת `hello world` ומיד נוצרה תגובה שרשרת שלבסוף הדפיסה `true`. מוד האימון זהה למוד ההגשה (הקלט אקראי). הקוד נבדק בצורה אוטומטית בלבד, עליכם להגיש קוד שמתקמפל ורץ ללא שגיאות.

שאלה 2 - תכנות מקבילי באמצעות ת'רדים (35 נק')

בקובץ MailBox.java נתונה לכם המחלקה האבסטרקטית MailBox. היא אינה לעריכה או להגשה. מחלקה זו מגדירה לכם:



- בנאי שמקבל ID מסוג int
- מתודה אבסטרקטית addMessage המקבלת מחרוזת
- מתודה אבסטרקטית close

עליכם לממש את המחלקה MyGenericMailBox ע"פ ההגדרות הבאות:

- היא סוג של MailBox
- בבנאי, מלבד ID, היא תקבל Consumer<String>
- היא תפעל כ Active Object
 - שמכיל ת'רד אחד בלבד!
 - שכל עוד לא הופעלה המתודה close()
 - הוא שולף מתוך תור שהוא thread safe הודעות מסוג String
 - אך אם התור ריק הוא "נח"
 - ומפעיל על הודעות אלו את ה Consumer. תהיה פעולתו אשר תהיה.
- במימוש ל addMessage פשוט נכניס את המחרוזת לתור
- במימוש למתודה close נגרום לכך שכל ההודעות שנתקבלו לפני הפעלת המתודה יעובדו כרגיל ע"פ הסדר, אך לאחר מכן פעולתו תיפסק והת'רד ייסגר.

מוד האימון זהה למוד הגשה. הקוד נבדק באופן אוטומטי בלבד. חובה שהקוד יעבור קומפילציה וירוץ ללא שגיאות.

שאלה 3 - אופטימיזציות קוד (35 נק')

בהינתן רשימה של נקודות (אובייקטי Point2D) אנו רוצים למצוא את המרחק המקסימלי הקיים בין 2 נקודות כלשהן בתוך הרשימה. בקובץ BadCode.java מצוי מימוש לא יעיל. עליכם לערוך או לממש מחדש את הקוד בקובץ GoodCode.java כך שהאימפלמנטציה תהיה לפחות פי 5 יותר מהירה.

כמובן, יש להחזיר את התוצאה הנכונה.

מוד האימון זהה למוד ההגשה והוא ניתן בקובץ MainTrain3.java.

תחילה מיצרים קלט של מערך אקראי.

לאחר מכן מתבצעת מדידת זמן של BadCode ושל המימוש שלכם ב GoodCode.

קוד שהוא לפחות פי 5 יותר מהיר יקבל את מלוא 35 הנק'

קוד שהוא פחות מפי 5 יותר מהיר יקבל את החלק היחסי.

שאלה זו תיבדק באופן אוטומטי בלבד. עליכם להגיש קוד שמתקמפל ורץ ללא שגיאות ריצה.

הגשה

עליכם להיכנס למערכת הבדיקות בכתובת: <https://cktest.cs.colman.ac.il/> ולהגיש ל PTM2 ומועד ב' את הקבצים GoodCode.java , MyGenericMailBox.java, MyCompletableFuture.java , בכל הגשה יש להגיש את כל הקבצים (ולהתייחס לפלט רק של השאלות שעניתם עליהן) ניתן להגיש במוד אימון ובמוד הגשה כמה פעמים שתמצאו עד לסוף המבחן. בסוף המבחן יש להגיש **במוד הגשה ואז במוד הגשה סופית**. אחריה תקבלו מס' אסמכתא בין 4 ספרות. לאחר הגשה במוד זה לא תוכלו להגיש יותר.

בהצלחה!