

פיתוח תוכנה מתקדם 2 – סמסטר א' מועד א' תשפ"ה

תזכורת: כתובת מערכת הבדיקות: <https://cktest.cs.colman.ac.il/>. שם הקורס PTM2, מועד א'. 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.

תזכורות נוספות ב Java:

- לכל ת'רד ניתן להגדיר עדיפות ע"י `setPriority` עם ערך 1-10 כאשר 10 זו העדיפות הגבוהה ביותר
- המתודה `add` של `ArrayBlockingQueue` מחזירה "אמת" אם הצלחנו להכניס איבר לתור, כלומר אם לא עברנו את הקיבולת (`capacity`) שהוגדרה לו בבנאי. אחרת יזרק `IllegalStateException` (ולא חוזר ערך `false`)

שאלה 1 – תכנות מקבילי באמצעות ת'רדים (40 נק')

בקובץ `Worker.java` עליכם להגדיר טיפוס שדומה בפעולתו ל `Active Object`:

- הטיפוס `Worker` צריך להיות סוג של ת'רד
- יקבל בבנאי ערכי קיבולת (`capacity`) ועדיפות (`priority`) שעליו להגדיר אותה לעצמו
- מתחזק `ArrayBlockingQueue` של `Runnable`-ים בקיבולת שניתנה בבנאי
- המתודה `add` תקבל משימה מסוג `Runnable` ותוסיף אותה לתור אם ורק אם לא נעבור את הקיבולת. הצלחה תחזיר "אמת" אחרת יוחזר "שקר".
- הפעולה של הת'רד, בדומה ל `Active Object`, היא להוציא מהתור `runnables` ולהריץ אותם אחד אחרי השני כאשר אם התור ריק יש להמתין.
- **טויסט קטן בעלילה:**
 - נניח מקרה בו הקיבולת של התור מלאה.
 - מייד לאחר הוצאה של משימת ה `Runnable` מהתור, מיד מתפנה בתור מקום חדש.
 - **אולם**, הדרישה הפעם היא שרק לאחר שהמשימה שנשלפה מהתור הסתיימה – רק אז מתפנה מקום בתור.
 - ייתכן שתצטרכו לשם כך לתחזק את גודל התור בנפרד באופן שהוא `thread safe`.
- המתודה `close` תפסיק מיידית את פעולת ה `worker` ותגרום לת'רד להיסגר.

בקובץ `PriorityPool.java` עליכם לממש `thread pool` עם עדיפויות באופן הבא:

- בבנאי תקבלו את מספר ה `worker threads` שיש לפתוח, ואת הקיבולת האחידה לכולם.
- עליכם לתחזק מערך של `Worker`-ים, כולם עם אותה הקיבולת
 - אך העדיפות של כל `Worker` גדלה בהתאמה למיקומו במערך
 - (בתא 0 יהיה `worker` עם עדיפות 1, בתא 1 `worker` עם עדיפות 2, וכך הלאה)
- המתודה `addTask`
 - תקבל משימה מסוג `Runnable` וערך עדיפות (1-10, ניתן להניח ערך תקין)
 - תנסה לעביר את המשימה ל `worker` עם עדיפות זו.
 - אם הקיבולת שלו מלאה, היא תנסה להעביר את המשימה ל `worker` עם עדיפות נמוכה יותר, וכך הלאה עד שבמקרה הגרוע לא ימצא `worker` שיכול לקבל את המשימה.
 - תחזיר את העדיפות של ה `worker` שקיבל את המשימה או 0 אם אף אחד לא קיבל.
- המתודה `stopAll` תסגור מיידית את כל ה `Workers`

המתודה testSingleWorker ב MainTrain1.java בודקת את ה Worker.
המתודה testPriorityPool1 בודקת PriorityPool עם Worker בודד וקיבולת של 5 ואילו המתודה testPriorityPool2 בודקת עם 3 Workers וקיבולת של 7.
מוד ההגשה דומה, כאשר הפרמטרים לעיל עשויים להשתנות. שאלה זו תיבדק באופן אוטומטי בלבד. עליכם להגיש קוד שמתקמפל ורץ ללא שגיאות ריצה.

שאלה 2 - תכנות מקבילי באמצעות ת'רדים (25 נק')

עליכם לממש את המחלקה Q2 בקובץ Q2.java באופן הבא:

- Q2 היא סוג של Recursive Task שמחזירה Integer
- בבנאי היא תקבל מערך של int-ים. ניתן להניח כי גודלו הוא חזקה של 2.
- הפעולה של ה recursive task היא לחשב את מספר האיברים הזוגיים במערך
- אולם עליה לעשות זאת ע"י הפרד ומשול, באמצעות Fork Join Pool.
 - בכל שלב של הרקורסיה יש לחשב את מספר האיברים הזוגיים כחיבור בין:
 - מספר האיברים הזוגיים בחצי המערך השמאלי
 - מספר האיברים הזוגיים בחצי חצי המערך הימני.
 - את החצי הימני יש לחשב כמשימה חדשה של ה Fork Join pool במקביל לחצי השמאלי.

הבדיקה בודקת שאכן הערך הרצוי מוחזר ושספר המשימות שנפתחו תואם את ההגדרה לעיל.

מוד האימון זהה למוד הבדיקה.

שאלה 3 - אופטימיזציות קוד (35 נק')

בהינתן רשימה של נקודות אקראיות (אובייקטי Point2D) מסתרת קבוצה **אחת בלבד** של נקודות בעלות אותם הערכים של x,y. בקובץ BadCode.java מצוי מימוש לא יעיל אשר מחזיר את מספר הנקודות בקבוצה. עליכם לערוך או לממש מחדש את הקוד בקובץ GoodCode.java כך שהאימפלמנטציה תהיה לפחות פי 5 יותר מהירה (ניתן להגיע לתוצאות גבוהות בהרבה).

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 שתרצו 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!