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

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.

שאלה 1 - תכנות מקבילי באמצעות ת'רדים (34 נק')

כפי שלמדנו, ה Future<V> הרגיל חושף מתודה get אשר גורמת לנו להמתין אם קראנו לה לפני שהערך V הוזן ל Future ע"י ה Thread Pool. כדי לעקוף את הבעיה, עליכם לממש את המחלקה ObservableFuture אשר תעטוף Future, תמתין ברקע לערך V, וכ Observable היא תודיע לכל Observers שלה כאשר ה V הגיע. בפרט במחלקה ObservableFuture:

- הבנאי יקבל אובייקט מסוג Future (הרגיל)
- המתודה get תחזיר את הערך V ללא כל המתנה (אם הוא אינו מוכן יחזור null)
  - ההנחה היא ש observers יקראו ל get רק לאחר שקבלו נוטיפיקציה על כך ש V הגיע.

שאלה זו תיבדק באופן אוטומטי בלבד. חובה שהקוד יעבור קומפילציה וירוץ ללא שגיאות ריצה כדי שהבדיקה תתאפשר. מוד ההגשה זהה למוד האימון.

שאלה 2 - fork join (31 נק')

נתונה לכם המחלקה BinTree עבור ייצוג של עץ בינארי.

- מחלקה זו אינה לעריכה ואינה להגשה.
- אובייקט של BinTree מיצג קודקוד בעץ.
  - תוכלו לבצע get לערך שהקודקוד מכיל (הערך מסוג int)
  - תוכלו לבצע get לבן השמאלי ולבן הימני של הקודקוד אם הם קיימים, אחרת יחזור null.

הבדיקה יוצרת עץ בינארי מלא (כלומר כל קודקוד מכיל בדיוק 0 או 2 בנים) עם ערכים אקראיים בקודקודים.

עליכם לחשב באופן רקורסיבי את סכום הקודקודים בעץ. אך כדי ליעל את החיפוש עליכם להשתמש ב fork join pool. בכל איטרציה החיפוש בתת העץ השמאלי יתבצע בת'רד אחר של ה fork join pool.

לשם כך עליכם לממש את המחלקה Par כסוג של RecursiveTask.

בבדיקה ב MainTrain2 אנו מייצרים עץ בינארי מלא שבקודקודיו ערכים אקראיים. לאחר מכן אנו מייצרים מופע של Par שמוזרק לתוך ה fork join pool. אנו בודקים ש:

- החישוב אכן מסתיים בתוך שנייה כפי שהוא אמור, אחרת הקוד נחשב כתקוע וכל ניקוד השאלה ירד
- הערך המוחזר אכן מהווה את סכום הקודקודים בעץ
- אכן ביצעתם שימוש ב fork join pool

שאלה זו אף תיבדק ידנית. עם זאת הקנס על קוד שאינו מתקפל או רץ ללא שגיאות ריצה = 10 נק'.

מוד ההגשה זהה למוד האימון.

שאלה 3 - אופטימיזציות קוד (35 נק')

נתון מערך דו-ממדי של בוליאניים המייצג את ההצבעה של נבחרי ציבור בפרלמנט כלשהו. ערך True משמעותו בעד ואילו False משמעותו נגד. בפרלמנט הזה לא ניתן להימנע מהצבעה. ברצוננו לבדוק האם היה רוב ל"בעד' או ל"נגד'.

בקובץ Q3bad.java מצויה המתודה vote (הצבעה) אשר בהינתן מערך דו-ממדי של בוליאניים היא מחזירה אמת או שקר בהתאמה לשאלה האם היה רוב ל"בעד'.

אולם, האימפלמנטציה בקובץ זה יכולה להיות מעט יעילה יותר. עליכם לערוך או לממש מחדש את הקוד בקובץ Q3good.java כך שהאימפלמנטציה תהיה לפחות פי 9 יותר מהירה.

כמובן, יש להחזיר את התוצאה הנכונה.

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

תחילה מיצרים קלט אקראי.

לאחר מכן מתבצעת מדידת זמן של Q3bad ושל המימוש שלכם ב Q3good.

קוד שהוא לפחות פי 9 יותר מהיר יקבל את מלוא 35 הנק' (כל עוד הוא אכן סופר את הקולות).

קוד שהוא פחות מפי 9 יותר מהיר יקבל את החלק היחסי.

שאלה זו אף תיבדק ידנית. עם זאת הקנס על קוד עם שגיאת ריצה או קומפילציה הוא 5 נק'.

הגשה

עליכם להיכנס למערכת הבדיקות בכתובת: https://cktest.cs.colman.ac.il/ ולהגיש ל PTM2 ומועד ב' את הקבצים Par.java, ObservableFuture.java , Q3good.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!