

פיתוח תוכנה מתקדם 2 – סמסטר ב' מועד א' 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')

תזכורות כלליות:

- במחלקה Thread קיים בנאי אשר מקבל Runnable ושם מסוג String – שמו של הת'רד.
 - גישה למחלקה שירשנו נעשית ע"י super.
- נתונה לכם המחלקה Tasker כסוג של מפה מ String לרשימה של Runnable. עליכם לממש את המתודות הבאות:
- המתודה start - לכל מפתח s (מסוג String) וערך runs (רשימה של runnables) המתודה תייצר ת'רד חדש ששמו s אשר יריץ בזה אחר זה את כל ה Runnables שב runs.
 - המתודה join – תמתין לסיום כל הת'רדים אשר נוצרו ב start.
 - בנוסף עליכם לדרוס את המתודה put אשר תאפשר הכנסה של מפתח וערך למפה אך ורק אם המתודה start עדין לא הופעלה. אם היא כבר הופעלה פשוט יש להחזיר null.
- בבדיקה של ה mainTrain1 מכניסים את הת'רדים בשמות t1 ו t2 עם מיפוי לשלושה Runnables שכל אחד מהם צריך להריץ. את t3 אין להכניס מכיוון שההכנסה מתבצעת לאחר הקריאה ל start.
- מוד ההגשה דומה.
- שאלה זו תיבדק באופן אוטומטי בלבד. חובה שהקוד יעבור קומפילציה וירוף ללא שגיאות ריצה כדי שהבדיקה תתאפשר.

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 לחפש באופן רקורסיבי את הערך המקסימאלי 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 ParMaxSearcher כסוג של RecursiveTask.

בבדיקה ב MainTrain2 אנו מייצרים עץ בינארי מלא שבקודקודיו ערכים אקראיים. לאחר מכן אנו מייצרים מופע של ParMaxSearcher שמזרק 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 המקסימלי אכן כזה
- 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')

בקובץ Q3bad.java מצויה המתודה fold אשר בהינתן ערך התחלתי ואופרטור בינארי המתודה תפעיל את האופרטור על כל איברי הרשימה ותחזיר את התוצאה. לדוגמה ב MainTrain3 מופעלת fold עם ערך התחלתי של 1 ופונקציית כפל, ולכן התוצאה שחוזרת היא המכפלה של כל האיברים.

האימפלמנטציה בקובץ זה היא לא יעילה. עליכם לערוך או לממש מחדש את הקוד בקובץ Q3good.java כך שהאימפלמנטציה תהיה לפחות פי 18 יותר מהירה.

כמובן, יש להחזיר את התוצאה הנכונה.

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

תחילה מייצרים קלט אקראי.

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

קוד שהוא לפחות פי 18 יותר מהיר יקבל את מלוא 35 הנק'

קוד שהוא פחות מפי 18 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 א' את הקבצים Q3good.java, Tasker.java, ParMaxSearcher.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!