

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

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 4 שאלות, חובה לענות על כל 4 השאלות ולהגיש למערכת הבדיקות במוד הגשה סופית לפני סוף המבחן. כל השאלות נבדקות אוטומטית בלבד.

שאלה 1 – Future אמיתי (25 נק')

בקובץ Q1.java עליכם לממש את האלמנטים הבאים

- מחלקה סטטית ופרטית בשם MyFuture המממשת את הממשק Future<V>
 - עליה לממש את כל המתודות של Future באופן הבא:
 - get - אם הערך V מוכן היא תחזיר אותו, אחרת היא תגרום לקורא להמתין.
 - get עם פרמטרים – תחזירו null. לא נשתמש בה בתרגיל זה.
 - isDone - תחזיר אמת או שקר בהתאמה להאם הערך V מוכן
 - cancel – אם הערך V מוכן היא תחזיר false. אחרת היא תבטל את ההמתנה ל V לכל מי שקרא ל get, ותחזיר true. אין צורך להתייחס לפרמטר של cancel.
 - isCancelled – תחזיר אמת או שקר בהתאמה להאם בוצע בפועל ביטול.
 - פרט למתודות לעיל, MyFuture תחיל גם מתודת set אשר תזין ערך מסוג V ל Future. כמובן, לאחר הזנה זו הערך V נחשב מוכן, וכל מי שהמתין לו בקריאה ל get צריך להתעורר.

- בנוסף למחלקה MyFuture לעיל, למחלקה Q1 יהיו המתודות (הרגילות) הבאות:
 - המתודה submit תקבל Callable<V> ותחזיר **מיד** Future<V>
 - האובייקט-בזמן-ריצה שחוזר הוא מסוג MyFuture אך הוא חוזר כ Future
 - על אובייקט זה להכיל את ערך ה V שה callable עתיד להחזיר
 - המתודה execute תגרום להרצה של ה callable שהתקבל במתודה submit.
- הבדיקה בודקת הזנה של Callable וקבלה של Future. נבדקת ההמתנה ב get עד לביטול ההמתנה או עד להפעלת ה execute. מוד ההגשה זהה למוד האימון.

* שימו לב לשימוש נכון ב synchronized

שאלה 2 – threads (25 נק')

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

- Q2 היא סוג של Thread
- המתודה setTask תזין Runnable כמשימה של ה Thread
 - אם קיימת כרגע משימה שרצה, המתודה פשוט תחזיר שקר,
 - אחרת, המתודה תזין את המשימה ותחזיר אמת.
- כאשר לא רצה משימה כלשהי,
 - ה Thread ימתין להזנה של משימה חדשה,
 - אחרת יריץ את המשימה שהוזנה.
- המתודה close תסגור את ה Thread.
 - אם רצה כרגע משימה היא תגיע לסיומה ואז ה thread ייסגר
 - אם מתבצעת המתנה למשימה חדשה, ההמתנה תסתיים וה thread ייסגר.

* שימו לב לשימוש נכון ב synchronized

שאלה 3 – אופטימיזציות קוד (25 נק')

בקובץ MainTrain3.java נתונה פונקציה לא יעילה (badSumCols) אשר בהינתן מטריצה, היא מחזירה ווקטור בו כל תא j הוא סכום העמודה ה j במטריצה. לסכום נלקחים רק הערכים האבסולוטיים (הערכים בערך מוחלט).

בקובץ Q3.java עליכם לממש את הפונקציה goodSumCols אשר גם היא מחשבת ווקטור בו כל תא j מכיל את סכום הערכים האבסולוטיים בעמודה ה j במטריצה. אולם על הפונקציה הזו לרוץ לפחות פי 15 יותר מהר מ badSumCols. עליכם להתייחס רק לקבועי הזמנים שבמערכת הבדיקות.

מוד האימון זהה למוד ההגשה. בבדיקה נבדוק שעבור מטריצה עם ערכים אקראיים הפונקציה הטובה מקבלת את אותן התוצאות אולם בזמן מהיר פי 15. הניקוד לבדיקה הוא ביחס ישיר למהירות הריצה של הקוד שלכם. ככל שהקוד שלכם קרוב יותר לפי 15 מהר יותר מהקוד הלא יעיל, כך ירדו לכם פחות נקודות.

שאלה 4 – אופטימיזציות קוד (25 נק')

בקובץ MainTrain4.java נתונה פונקציה לא יעילה (badCode) אשר בהינתן מטריצה היא מחשבת את סכום הערכים שבחלקה המשולשי העליון (כולל האלכסון).

בקובץ Q4.java עליכם לממש את הפונקציה goodCode אשר גם היא מחשבת ערך זה, אולם במהירות הגדולה פי 4. עליכם להתייחס רק לקבועי הזמנים שבמערכת הבדיקות.

מוד האימון זהה למוד ההגשה. בבדיקה נבדוק שעבור מערך עם ערכים אקראיים הפונקציה הטובה מקבלת את אותן התוצאות אולם בזמן מהיר פי 4. הניקוד לבדיקה הוא ביחס ישיר למהירות הריצה של הקוד שלכם. ככל שהקוד שלכם קרוב יותר לפי 4 מהר יותר מהקוד הלא יעיל, כך ירדו לכם פחות נקודות.

הגשה

עליכם להיכנס למערכת הבדיקות בכתובת: <https://cktest.cs.colman.ac.il/> ולהגיש ל PTM2 ומועד א' את הקבצים Q1.java, Q2.java, Q3.java, Q4.java,

בכל הגשה יש להגיש את כל הקבצים (ולהתייחס לפלט רק של השאלות שענייתם 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!