

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

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

בקובץ `MyFuture.java`, עליכם לממש מחלקה גנרית `MyFuture<V>`, שתשמש כהרחבה ל `Future` ותספק תוספות בהשראת `CompletableFuture`.

שימו לב: כפי שראינו בשיעורים, כל המתודות כרוכות אחת לשנייה בצורה כזו או אחרת, אז מומלץ לקרוא את כל הסעיפים בטרם המימוש.

המחלקה תתמוך בפונקציות הבאות:

- **`set(V v)`** מקצה ערך ל `Future`. כל קריאה ל- `get()` לאחר מכן צריכה להחזיר ערך זה.
 - כמובן, יש להעיר משנתם את כל מי שהיה `blocked` על `get` טרם הקריאה ל `set`.
 - ניסיון להזין ערך חדש אם ל `Future` כבר הוזן ערך, או שהיתה כבר פעולת ביטול, צריך לגרור זריקה מסוג `IllegalStateException`.
- **`get()`**
 - מחזירה את הערך `v` אם הוא כבר נקבע, או מחכה עד שהוא יהיה זמין.
 - אם בוצעה פעולת ביטול לפני הזנת הערך, יש לזרוק `CancellationException` (וכמובן לבטל את ההמתנה).
- **`get(long timeout, TimeUnit unit)`**
 - מקבלת ערך המתנה מסוג `long` (למשל 1) ויחידות זמן מסוג `TimeUnit` (למשל שניות)
 - מחזירה את הערך `v` אם הוא נקבע, או מחכה עד שיחלוף הזמן הנתון.
 - טיפ: ניתן להשתמש ב- `unit.timedWait(lock, timeout)`
 - אם הערך לא נקבע בזמן זה, יש לזרוק `TimeoutException`
 - אם בוצעה פעולת ביטול לפני הזנת הערך, יש לזרוק `CancellationException` (וכמובן לבטל את ההמתנה).
- **`cancel(boolean mayInterruptIfRunning)`**
 - גורם לביטול החישוב העתידי, אם עדיין לא הוזן ערך `V`.
 - מונע הגדרה עתידית של ערך.
 - מעיר משנתם את מי ש `blocked` על `get`
 - מחזיר `true/false` בהתאמה לאם אכן בוצע ביטול או לא.
 - הקלה - ניתן להתעלם מהפרמטר שקבלה.
- **`boolean isCancelled()`** מחזירה `true` אם הפעולה כבר בוטלה. אחרת `false`.
- **`boolean isDone()`** מחזירה `true` אם הערך נקבע או אם ה `Future` בוטל. אחרת `false`.

- **thenApply()** בהינתן 'פונקציה' מ V ל R היא
 - מחזירה מיידית $\text{MyFuture}<R>$ חדש.
 - כאשר ה $\text{MyFuture}<V>$ מקבל את ערכו מסוג V , מיד ימיר אותו ל R באמצעות הפונקציה, ויזין אותו ל $\text{MyFuture}<R>$ שהוחזר מבעוד מועד.
- **thenAccept()** בהינתן 'פונקציה' מ V ל void (צרכן) זו תופעל כאשר יוזן ערך מסוג V ל $\text{MyFuture}<V>$.

דגשים:

- המימוש צריך לאפשר הזנה תקינה כך שקריאות מקבילות לא יגרמו לבעיות.
- חשבו כיצד למנוע מצב שבו `get()` נתקע לנצח.
- שימו לב אלו מתודות צריכות להיות מסונכרנות עם מי וכיצד

מוד האימון זהה למוד הבדיקה. שאלה זו תיבדק באופן אוטומטי בלבד. עליכם להגיש קוד שמתקמפל ורץ ללא שגיאות ריצה.

בהצלחה!

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 האי-זוגיים במערך
- אולם עליה לעשות זאת ע"י הפרד ומשול, באמצעות Fork Join Pool.
- בכל שלב של הרקורסיה יש לחשב את מספר האיברים האי-זוגיים כחיבור בין:
 - מספר האיברים האי-זוגיים בחצי המערך השמאלי
 - מספר האיברים האי-זוגיים בחצי המערך הימני.
- את החצי הימני יש לחשב כמשימה חדשה של ה Fork Join pool במקביל לחצי השמאלי.

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

מוד האימון זהה למוד הבדיקה. שאלה זו תיבדק באופן אוטומטי בלבד. עליכם להגיש קוד שמתקמפל ורץ ללא שגיאות ריצה.

שאלה 3 - אופטימיזציות קוד (35 נק')

ברשימה של ערכים מספריים מסוג int מסתתרים רצפים שונים של מספרים עם אותם הערכים. בקובץ BadCode.java מצוי מימוש לא יעיל אשר מחזיר את אורך הרצף הגדול ביותר של מספרים עם אותו הערך.

לדוגמה ברשימה 2,5,1,5,5,5,7,2,9,6,6,6,6,1,5,8 ישנם שני רצפים – רצף של 5 באורך 3, ורצף של 6 באורך 4 ולכן יש להחזיר 4.

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 קובצי ה java לעיל השייכים לפתרון המבחן.

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!