פיתוח תוכנה מתקדם 2 – סמסטר א' 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.

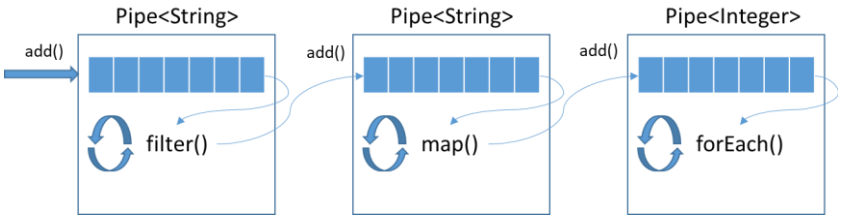
במבחן זה 3 שאלות, חובה לענות על כל 3 השאלות ולהגיש למערכת הבדיקות במוד הגשה סופית לפני סוף המבחן.

שאלה 1 - תכנות מקבילי באמצעות ת'רדים (35 נק')

ברצוננו לממש את המחלקה Pipe<E> (צינור). ראו את השורות הבאות ב MainTrain1.java:

```
Pipe<String> ps=new Pipe<String>();
int sum[]= {0};
// sum all lengths of strings with length under 4
ps.filter(s->s.length()<4).map(s->s.length()).forEach(x->sum[0]+=x);
```

לכל אובייקט Pipe<E> יש תור של E-ים שהוא thread safe ויכול להכיל עד 100 איברים. בדומה ל Active Object, לכל Pipe יש ת'רד אקטיבי ברקע. הת'רד פעיל רק כאשר יש נתונים בתור. הוא שולף E-ים מהתור בזה אחר זה ומבצע עליהם פעולה מוגדרת מראש. חלק מהפעולות מאפשרות העברת נתונים ל Pipe אחר. כך, ניתן להגדיר מראש "פס ייצור" של עיבוד שיחל לעבוד ברגע שהנתונים יתחילו לזרום. התרשים הבא ממחיש בצורה ויזואלית את פס הייצור שיוצר הקוד לעיל:



הפעולות הן:

- Filter – בהינתן תנאי, המתודה תעביר את כל ה E-ים שעליהם התנאי מחזיר אמת ל Pipe הבא
  - בדוגמה ps הוא Pipe<String> והתנאי הוא כל מחרוזת שאורכה קטן מ 4.
- Map – בהינתן פונקציה מ E ל R (טיפוסים פרמטריים), המתודה תמיר כל E בתור ל R ותעביר אותו ל Pipe הבא
  - בדוגמה, המתודה map הופעלה מאובייקט ה Pipe<String> שהחזירה filter. הפונקציה ממירה כל מחרוזת לאורך שלה.
- forEach – בהינתן "צרכן" של E, המתודה תצרוך כל E בתור. זו פעולה טרמינלית – אחריה לא ניתן לשרשר פעולות נוספות.
  - בדוגמה, המתודה forEach הופעלה מאובייקט ה Pipe<Integer> שהחזירה map. הצרכן צובר כל int בתור לתוך sum[0].
- Add – מכניסה אובייקט E לתור או ממתינה כל עוד התור מלא
- Stop – תעצור מידית את פעולת ה Pipe ואת ה Pipe שאחריו בהנחה והיה כזה.
  - כך, עצירה של ps תגרור עצירה של כל ה Pipe-ים.
  - מתודה זו הוגדרה בממשק Stoppable שאותה Pipe נדרשת לממש

הבדיקות ב MainTrain1:

א. לאחר הגדרת פס הייצור נוספו אך ורק 3 ת'רדים (אחד לכל Pipe שהוגדר) – 10 נק'
ב. לאחר הכנסה של כמה מחרוזות ל ps מתקבלת התוצאה הנכונה ל sum[0] – 10 נק'
ג. לאחר קריאה ל stop שכל הת'רדים שפתחנו נסגרו בהתאמה – 15 נק'

הבדיקות ב MainTest1 דומות. למען הסר ספק, הקוד צריך להיות גנרי ע"פ ההגדרות ולא רק מתאים לדוגמה לעיל. בפרט, ייתכן מספר Pipe-ים שונה, סדר הפעלה שונה, פרמטרים שונים לפונקציות, איברים מסוג שונה וכמות שונה של איברים. בנוסף הקלט אקראי.

שאלה זו תיבדק באופן אוטומטי בלבד. חובה שהקוד יעבור קומפילציה וירוץ ללא שגיאות ריצה כדי שהבדיקה תתאפשר.

שאלה 2 - סנכרון (30 נק')

בקובץ Count.java נמצאת מחלקה פשוטה שמחזיקה int, מגדילה אותו במתודה inc(), ומחזירה אותו ב get().

הביטו ב MainTrain2.java. אנו מפעילים שני ת'רדים שמריצים את אותו ה runnable. ה runnable קורא d פעמים (השווה בין מיליון ל 2 מיליון פעמים באופן אקראי) ל inc() של אותו מופע של Count. הציפייה היא שהערך של count יהיה שווה בדיוק ל 2d לאחר ששני הת'רדים סיימו את פעולתם. אולם, כפי שלמדנו, בכל ריצה נקבל ערך שונה בגלל שהפעולה inc() אינה פעולה אטומית.

עליכם לערוך את Count כך שאכן יתקבל הערך 2d ב MainTrain2.

אולם, חל איסור להשתמש ב synchronized.

שאלה זו אף תיבדק ידנית. עם זאת משקל קוד מתקפל ורץ ללא שגיאות ריצה = 10 נק'.

מוד ההגשה זהה למוד האימון.

שאלה 3 - אופטימיזציות קוד (35 נק')

בקובץ BadCode.java מצויה הפונקציה common (שכיח), אשר בהינתן מערך של ציונים (בין 0 ל 100 כולל) תפקידה להחזיר את הציון השכיח ביותר (הציון שהופיע הכי הרבה פעמים).

האימפלמנטציה בקובץ זה היא לא יעילה. עליכם לערוך או לממש מחדש את הקוד בקובץ GoodCode.java כך שהאימפלמנטציה תהיה לפחות פי 5 יותר מהירה.

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 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 Pipe.java, Count.java , GoodCode.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!