

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

**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 (30 נק')

בקובץ PooledThread.java ממשו את המחלקה PooledThread

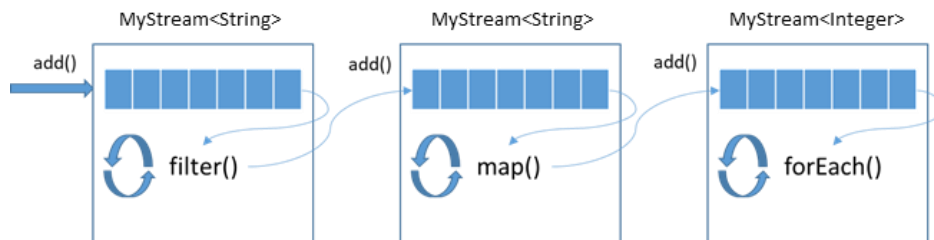
- כ ActiveObject
  - שמקבל משימות מסוג Runnable במתודה add.
  - אם במשך שנייה לא התקבלה משימה חדשה אז ה PooledThread נסגר אוטומטית.
- הבדיקה בודקת שאכן ה PooledThread פותח ת'רד פנימי אשר מבצע את כל המשימות שקיבל, ושלאחר שנייה ללא משימות הוא נסגר וסוגר עימו את הת'רד שפתח.
- שאלה זו נבדקת אוטומטית. הקפידו להגיש קוד שמתקפל ורץ ללא שגיאות ריצה. עם זאת, שגיאות קומפילציה או ריצה שמונעות בדיקה אוטומטית יגוררו קנס של 10 נק' ובדיקה ידנית.

### שאלה 2 - תכנות מקבילי באמצעות ת'רדים (35 נק')

ברצוננו לממש את המחלקה `MyStream<E>`. ראו את השורות הבאות ב `MainTrain2.java`:

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

לכל אובייקט `MyStream<E>` יש **תור** של E-ים שהוא thread safe ויכול להכיל עד 100 איברים. בדומה ל Active Object, לכל `MyStream` 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 למופע אחר של `MyStream`. 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 MyStream הבא
  - בדוגמה ms הוא `MyStream<String>` והתנאי הוא כל מחרוזת שאורכה קטן מ 4.
- Map – בהינתן פונקציה מ E ל R (טיפוסים פרמטריים), המתודה תמיר כל E בתור R ותעביר אותו ל MyStream הבא
  - בדוגמה, המתודה map הופעלה מאובייקט ה `MyStream<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 `MyStream<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 MyStream ואת ה MyStream שאחריו בהנחה והיה כזה.
  - כך, עצירה של ms תגרוור עצירה של כל ה MyStream-ים.
  - מתודה זו הוגדרה בממשק Stoppable שאותה MyStream נדרשת לממש

הבדיקות ב MainTrain1:

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

הבדיקות ב MainTest2 דומות. למען הסר ספק, הקוד צריך להיות גנרי ע"פ ההגדרות ולא רק מתאים לדוגמה לעיל. בפרט, ייתכן מספר MyStream-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 וירוף ללא שגיאות.

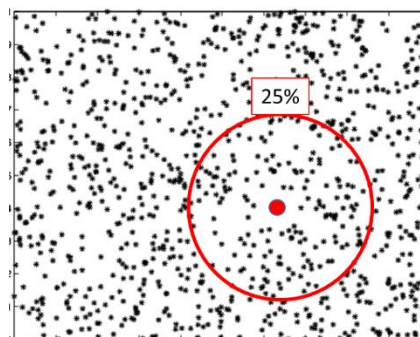
### שאלה 3 - אופטימיזציות קוד (35 נק')

בהינתן אוסף של נקודות דו-ממדיות, נקודת מרכז center, ופרמטר percent המייצג אחוז, ברצוננו למצוא את הרדיוס הקטן ביותר של מעגל שמרכזו center ומקיף לפחות percent אחוז מהנקודות.

לדוגמה:

המרכז הוא הנקודה אדומה. רוצים להקיף לפחות 25% ומצאנו את r כרדיוס (מס' שלם) שמשיג זאת. ייתכן שהקפנו קצת יותר מ 25% מהנקודות אך לו היינו משתמשים ב r-1 אז בפירוש היינו מקיפים פחות מ 25%.

בקובץ BadCode.java מצוי מימוש לא יעיל אשר מגדיל את הרדיוס עד לקבלת התוצאה הרצויה. עליכם לערוך או לממש מחדש את הקוד בקובץ GoodCode.java כך שהאימפלמנטציה תהיה לפחות פי 3 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 3 יותר מהיר יקבל את מלוא 35 הנק'

קוד שהוא פחות מפי 3 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 MyStream.java, PooledThread.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 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!