

## קורס תכנות מונחה עצמים מטלה 2 – חלק 1

### THREADS

#### יש שלושה חלקים

כללית: במטלה זו יוצרים מספר קיבצי טקסט ומחשבים את מס שורות הכולל בקבצים אלה. נשתמש בשלוש שיטות:

- שיטה רגילה ללא שימוש ב-threads,
- שימוש ב-threads,
- שימוש ב-ThreadPool

יש לכתוב מחלקה בשם Ex2 ובה לממש את הפונקציות הנדרשות.

#### חלק א

בחלק זה יש לכתוב מחלקה בשם Ex2\_1 ובתוכה לכתוב 4 פונקציות:

1. `public static String[] createTextFiles(int n, int seed, int bound){...}`

קלט:

**n** - מספר טבעי המייצג את מס קבצי טקסט.

שמות הקבצים הם, לדוגמה `file_1, file_2, ..., file_n`.

מספר שורות בכל קובץ הוא מס אקראי, המתקבל בעזרת מחלקת שימוש

במחלקה זו ובפרמטרים `seed` ו-`bound`

ניתן לראות בנספח שבסוף המסמך.

פלט: הפונקציה יוצרת **n** קבצי טקסט על הדיסק ומחזירה את מערך

של שמות הקבצים. בכל שורה ניתן לכתוב משפט כמו "Hello World"

המכיל לפחות 10 תווים.

2. יש לכתוב פונקציה

`public static int getNumOfLines(String[] fileNames){...}`

קלט: מערך שמכיל שמות הקבצים.

פלט: מספר שורות הכולל של הקבצים.

דוגמה: הפונקציה מקבלת שמות של שני קבצים, בקובץ ראשון יש 3250

שורות קובץ שני יש 1000 שורות הפונקציה מחזירה מספר 4250.

### 3. יש לכתוב פונקציה

```
public int getNumOfLinesThreads(String[] fileNames){...}
```

קלט: מערך שמכיל שמות הקבצים.

פלט: מספר שורות הכולל של הקבצים.

הדרכה: יש לכתוב מחלקה שיורשת ממחלקת Thread. הבנאי מקבל שם הקובץ ובפונקציה run יש לחשב את מס שורות של הקובץ הזה.

ומחשבת את מס' שורות באותו קובץ.

ניתן לכתוב מחלקות ומתודות עזר.

### 4. יש לכתוב פונקציה

```
public int getNumOfLinesThreadPool(String[] fileNames){...}
```

קלט: מערך שמכיל שמות הקבצים.

פלט: מספר שורות הכולל של הקבצים.

הדרכה:

יש לבנות מאגר של threads בגודל fileNames.length.

יש לכתוב מחלקת עזר המממשת את ממשק Callable, שבה מתודה

call מחשבת את מס' שורות של קובץ אחד.

5. **יש להשוות את זמני ריצה** של שיטות 2,3,4 ולהסביר במפורט (ב-README) למה זמני ריצה שונות (או שוות). לשם כך יש לקחת כמות גדולה של קבצי טקסט.

### 6. **יש לצרף דיאגרמת מחלקות עבור הפרויקט**

נספח שימוש במחלקת java.util.Random

```
int seed = 1, n = 10, bound = 1000;
Random rand = new Random(2);
for (int i = 1; i <= n; i++) {
    int x = rand.nextInt(bound);
    System.out.print(x+", ");
}
output: 108, 372, 40, 67, 389, 350, 606, 719, 847, 68,
```

כאשר

**Random(long seed)**

Creates a new random number generator using a single long seed.

**nextInt(int bound)**

Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.