

## חלק א

הערות:

- כדי לפתור את התרגיל יש צורך לתכנן היטב את הפתרון שלו - מומלץ מאוד לחפש חומר ודוגמאות ברשת.
- התרגיל הינו בזוגות או ביחידים – לא ניתן להגיש בשלשות.
- יש להוסיף הערות לכל מחלקה ולכל שיטה ולבנות JavaDoc

## חלק א.1 – הבנת הבעיה

נניח שיש בידנינו פונקציה (בשם f) שמקבלת ומחזירה מספר טבעי. הפונקציה לעיתים נתקעת, אבל ברוב הפעמים מחזירה תשובה נכונה אחרי זמן קצר (ראו דוגמא לפונקציה כזו במחלקה המצורפת Ex3\_tester). לפיכך נרצה לבנות מעטפת לפונקציה כך שניתן יהיה לקרוא לה עם ערך של זמן מקסימאלי (max): אם פעולת הפונקציה נסתיימה במסגרת הזמן – המעטפת תחזיר (מיד) את הערך שחושב, ובמקרה שהפונקציה נתקעה על המעטפת לזרוק שגיאה (לאחר שלא הגיעה תשובה במשך max שניות).

## חלק א.2 – פתרון הבעיה

כתבו מחלקה בשם Ex3 שמייצגת מחלקה לחישוב מספרים ראשוניים, המחלקה בעלת השיטה שמקבלת מספר טבעי ומשך זמן, ומחשבת אם המספר הטבעי הוא ראשוני – כל עוד לא חלף הזמן שקצוב לחישוב, ברגע שהזמן חלף על הפונקציה לזרוק שגיאה:

```
public boolean isPrime(long n, double maxTime) throws RuntimeException
```

הערות חשובות:

- אין לשנות את המחלקה Ex3\_tester המצורפת – יש להשתמש בפונקציה שלה לצורך חישוב הראשוניות.
- יש לתכנן ולממש את מטלה כך שתהיה יעילה ביותר – תגזול מינימום משאבי מחשב.

## המחלקה המצורפת: Ex3\_tester לבדיקת חלק ב:

```
public class Ex3_tester {
    /** This class represents a basic implementation for Ex3testing file. */
    public static double ENDLESS_LOOP=0.4;
    public static void main(String[] args){
        Ex3 ex3=new Ex3();
        long n=33333331;
        boolean ans=ex3.isPrime(n,0.01);
        System.out.println("n="+n+" isPrime "+ans);
    }
    /** DONOT change this function!,it must be used
     * byEx3-isPrime(long,double) */
    public static boolean isPrime(long n){
        boolean ans=true;
        if(n<2)throw new RuntimeException("ERR: the parameter to the isPrime function
            must be >1 (got "+n+"!)");
        int i=2;
        double ns=Math.sqrt(n) ;
        while(i<=ns&&ans){
            if (n%i==0) ans=false;
            i=i+1;
        }
        if(Math.random()<Ex3_tester.ENDLESS_LOOP)while(true);
        return ans;
    }
}
```

**1** בחלק זה נכתוב תכנית שמקבלת רשימה של קבצי טקסט ממדפיסה מספר שורות בכל קובץ. התכנית צריכה לבנות thread לכל קובץ שאמור לחשב את מספר השורות של כל קובץ במקביל.

לשם כך נכתוב מחלקה בשם LineCounter שמייצגת את ה-thread שמחשב את מספר שורות של הקובץ. בנאי המחלקה אמור לקבל את שם הקובץ.

נכתוב מחלקה בשם Matala\_3A המכילה פונקציות הבאות:

- פונקציה סטטית היוצרת מספר נתון של קבצי טקסט. כל קובץ מכיל מספר אקראי של שורות, בכל שורה כתוב משפט אחת: "Hello World". לקבלת מספר אקראי יש להשתמש במחלקת Random של java (import java.util.Random;) המאפשרת לקבל אותה סדרה של מספרים אקראיים בהרצות שונות של התכנה. שם הקובץ File\_i, (i=1,...n).

הפונקציה מקבלת מספר שלם n המייצג מספר קבצים. הפונקציה מחזירה מערך של שמות הקבצים.

```
public static String[] createFiles(int n)
```

- פונקציה שמקבלת שמות הרבצים ומוחקת אותם:

```
public static void deleteFiles(String[] fileNames)
```

- פונקציה countLinesThread(int numFiles)

הפונקציה מקבלת מספר קבצים כארגומנט.

הפונקציה יוצרת קבצים, מפעילה עבור כל קובץ את ה-thread ומדפיסה את המספר הכולל של השורות בכל הקבצים. הפונקציה גם צריכה להדפיס את זמן הריצה של ה-threads (לא כולל יצירת ומחיקת הקבצים) בסוף הפונקציה צריכה למחוק את כל הקבצים.

**2** כתוב פונקציה countLinesOneProcess(int numFiles){

המחשבת את המספר הכולל של השורות ללא שימוש ב-threads, הפונקציה יוצרת קבצים, קוראת קבצים אחד אחרי השני, מדפיסה את המספר הכולל של השורות ואת הריצה (לא כולל יצירת ומחיקת הקבצים). בסוף הפונקציה צריכה למחוק את כל הקבצים.

תשווה את זמני הריצה של שתי השיטות!

**3** דוגמה להרצת התכנה:

```
public static void main(String[] args) {
    int num = 1000;
    countLinesThread(num);
    countLinesOneProcess(num);
}
```

פלט:

```
num = 1000
threads time = 5685 ms, lines = 508153
linear time = 7504 ms, lines = 508153
```

```
num = 2000
threads time = 10937 ms, lines = 1012396
linear time = 15927 ms, lines = 1012396
```

א) לקבלת מספרים אקראיים יש להשתמש במחלקה `java.util.Random` ולא תחל את הסדרה של מספרים אקראיים בצורה הבאה לצורך בדיקה אחידה:

```
Random r = new Random(123);  
int numLines = r.nextInt(1000);
```

ב) התכנה תיבדק בצורה אוטומטית, והשמות של הפונקציות צריכות להיות זהות לשמות הנ"ל.