

## תרגול #8 – חריגות

שאלה 1:

נתון הקוד:

```
public class StackUnderflowException extends Exception
{
    StackUnderflowException () {
        super ();
    }
    StackUnderflowException (String message) {
        super(message);
    }
}

public class Stack<Person>
{
    private int index;
    ...
    public Object pop() throws StackUnderflowException {
        if (index < 0)
            throw new StackUnderflowException();
        ...
    }
}
```

עיינו בשני קטעי הקוד שלהלן ולגבי כל אחד מהם רשמו במשבצת שמתחת, מה לא הגיוני בדרך שבה משפט ה- try-catch כתוב (הניחו שאין משפטי תכנות היכן שאין שלוש נקודות).

<pre>Stack s = new Stack&lt;Person&gt;(); try {     Person p = s.pop();     ... } catch (Exception e) {     ... } catch (StackUnderflowException se) {     ... }</pre>	<pre>Stack s = new Stack&lt;Person&gt;(); try {     Person p = s.pop();     ... } catch (StackUnderflowException se) { } System.out.println(" Person is: " + p);</pre>
--	--

שאלה 2:

התרגול יעסוק בניהול חריגות.

**כתבו מחלקה: Exceptions** ובה שיטה מרכזית: `process( String fileName )` שמקבלת שם קובץ, קוראת את תכנו לזיכרון ובודקת את הדברים הבאים:

1. שהקובץ קיים

Exception program - file not found

2. שהקובץ איננו ריק

Exception program - file empty

Exception program - empty line at line number 0

3. שהקובץ מכיל מספרים בלבד

Exception program - not a number.

For input string: "abcdefg" after processing 10 lines.

4. שבכל שורה יש בדיוק מספר אחד

Exception program - multiple entries in a line.

2 entries in line 22

5. שאף אחד מהמספרים איננו גדול ממליון בערכו המוחלט

Exception program - entry too big

offending entry: -1905445.1886612335 after processing 47 lines.

6. שאין בקובץ למעלה מ-1000 שורות

Exception program - Too many entries

after processing 1001 lines.

1. שסכום כל הערכים שבו אינו גדול מ- $5 \cdot 10^8$

Exception program - sum too large.

is 5.00999499E8(exceeding 5.0E8) after processing 500 lines.

יש להקפיד על ניסוח הודעת השגיאה עבור כל אחד מהמצבים שתואר לעיל. והעיקר – השתמשו בחריגות בלבד! כמו כן, אל לתכנית לעצור אחרי שגיאה אחת: עליה לקרוא את הקלט עד תומו ולהודיע על כל השגיאות המצויות בו.

בכל הודעת שגיאה יש לציין גם את מספר השורה בקלט שגרמה לשגיאה.

בסיום טיפול בכל קובץ שי להדפיס לפלט הסטנדרטי הודעת סיכום.  
לדוגמא:

289 lines were processed.

Sum of all entries is 522028.4753156797

רמזים :

1. אפשר לפתוח קובץ קיים באמצעות המחלקה File. ל-Scanner יש בנאי המקבל עצם מטיפוס File וכך ניתן לקרוא קובץ כזה.
2. אפשר לקרוא שורה שלמה מהקלט באמצעות המתודה `getLine()` של `Scanner`.
3. המתודה `parseInt()` של `Integer` הופכת `String` למספר (`int`). אם הטקסט שבמחרוזת לא מייצג מספר, תיזרק חריגה מסוג `NumberFormatException`.

תזכורת: חריגה היא מחלקה שיורשת מ-`Exception`;  
כל מחלקה מוגדרת בקובץ נפרד ששמו כשם המחלקה.

כתבו את מחלקת: `Lab08Test` שמקבלת כפרמטר בשורת פקודה שמות של קבצי טקסט לבדיקה ומפעילה "עליהם" את `process` של `Exceptions`.

לדוגמא :

```
public static void main( String[] args ){
    Exceptions e = new Exceptions();

    for( String s : args)
    {
        e.process( s );

        /*
         * wait a second, to let any exception processing complete
         * before examining another input file.
         */
        try
        {
            Thread.sleep( 1000 );
        }
        catch( InterruptedException ie )
        {
            ie.printStackTrace();
        }
    }
}
```

שימו לב שאין טיפול בחריגות בשיטה `main` ולכן `process` צריכה לייצר ולטפל בכולן!

במעבדה זו בלבד, יתקבלו עבודות עם הערות קומפילציה ולא יורד ציון.

**בהצלחה!**