**מטלה 2**

Python VS java

**בקורס מבנה תוכנה**

**מוגש לידי:** **מר הראל ברגר**

**מגיש:**

1. **נוף חסון** ת"ז 308499557
2. דביר מרום ת"ז 302616248
3. **זמן פיתוח:**

את המטלה הראשונה לקח לי יותר זמן לממש וליצור כי זאת פעם ראשונה שנתקלתי בקבצי JSON וגם כי הייתי פחות מיומן בכתיבה נכונה לפי הכללים שנלמדו בקורס. לקח לי במטלה הראשונה 5 שעות ובמטלה של הפייתון לקח לי שעתיים סך הכל. הרבה יותר קצר ומובן.

1. **זמן ריצה:**

בשביל להשוות את זמן הריצה הפעלתי בשני הקודים בתחילת התוכנית טיימר START וטיימר END בסוף התוכנית וע"י חיסור הזמן סיום מזמן ההתחלה בדקתי כמה זמן לקח לקוד לרוץ,(ספריית Time בפייטון וספריית System בגאווה) שמרתי את התוצאה וביצעתי את הבדיקה בעזרת לולאה שרצה 10 פעמים כדי שיהיה מס' גדול של תוצאות (בקוד פייתון עשיתי שהוא יכתוב אוטומטית את הפלט של התוצאות אל קובץ טקסט בשם Python.txt ובקוד ג'אווה עשיתי שהוא בmain ישמור בתוך מערך את התוצאות של ההרצות וידפיס אותו, ואני בצורה ידנית הכנסתי את זה לקובץ טקסט בשם Java.txt.

קבצי התוצאות מצורפים למטלה אבל בכל מקרה אני ידביק כאן את התוכן שלהם:

קובץ JAVA: [0.013, 0.002, 0.004, 0.002, 0.005, 0.002, 0.004, 0.002, 0.004, 0.002]

קובץ Python: [0.0, 0.0, 0.0, 0.0166, 0.0015, 0.0014, 0.0019, 0.0005, 0.0, 0.0]

(\*בפלט של תוצאות הפייתון כאן מחקתי את הספרות אחרי הספרה הרביעית מטעמי נוחות)

1. **כמות שורות קוד:**

(בדקתי ע"י הסתכלות בide :)

קובץ JAVA - 70 שורות (כולל האיסוף תוצאות בשביל הגרף) נטו הפונקציה של הקוד 30 שורות.

קובץ Python – 25 שורות קוד(כולל איסוף תוצאות וכתיבה לtxt) נטו הפונקציה 17 שורות.

1. **שימוש בזיכרון:**

**PYTHON-PSUTIL**

בקוד פייתון השתמשתי בספריית psutil, להלן הסבר קצר על הספרייה ותפקידי הפונ' שהיא מאפשרת לנו להשתמש בהם:

**psutil (python system and process utilities)**

is a cross-platform library for retrieving information on running **processes** and **system utilization** (CPU, memory, disks, network, sensors) in **Python**. It is useful mainly for **system monitoring**, **profiling**, **limiting process resources** and the **management of running processes**. It implements many functionalities offered by command line tools such as: ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap. It currently

**הפקודה בה השתמשתי:**

**process=psutil.Process(os.getpid())**

**print("memory used by bytes: ", process.memory\_info().rss)**

**התשובה שימוש בזיכרון בפייטון: memory used by bytes: 15429632**

**JAVA-RunTime**

בשביל הקוד בג'אווה השתמשתי בפונקציות מהספריה RUNTIME להלן הסבר קצר על הספריה:

The total used / free memory of an program can be obtained in the program via java.lang.Runtime.getRuntime(); The runtime has several method which relates to the memory. The following code example demonstrate its usage.

בניגוד לפייתון, בג'אווה צריך להשתמש ביותר פונקציות כדי לחשב את הזכרון RAM בשימוש, אני יביא הסבר קצר על כל אחת מהפונק' שהשתמשתי וגם בסוף את הפקודה עצמה שהוספתי(בסוף הקוד)

public long freeMemory()

Returns the amount of free memory in the Java Virtual Machine

public long totalMemory()

Returns the total amount of memory in the Java virtual machine

public void gc()

Runs the garbage collector.

public static [Runtime](https://docs.oracle.com/javase/7/docs/api/java/lang/Runtime.html) getRuntime()

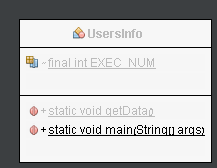
Returns the runtime object associated with the current Java application.

הפקודה שהשתמשתי מופיעה בסוף הקוד בשורות 68-72

התשובה בjava השימוש בזכרון: memory used by bytes is: 593544

* UML

את הדיאגרמת UML יצרתי בעזרת easyUML ומכיוון שאין הרבה מחלקות הוא יחסית קטן:



ואת הגרף בpython לקחתי מוכן OPEN SOURCE ושיניתי אותו כדי שיתאים לדרישות שלי, הוא משווה בין הקובץ תוצאות הרצה של גאווה לזה של פייתון ומייצר גרף בהתאם למילי שניות שנמדדו. כדי לקבל ערכים שונים בגרף צריך להריץ שוב את הקוד של EX2 (במקרה של פייטון כותב אוטומט לקובץ תוצאות הרצה) ובמקרה של ג'אווה להריץ את הinfoUser שזה בעצם מטלה 1 ולהכניס ידנית לקובץ Java.txt שזה הקובץ תוצאות הרצה בג'אווה את המערך שהוא מדפיס.

