מטלה מסכמת – אלגוריתמיקה ותכנות צד שרת-לקוח ב-Java

סטודנטים יקרים,

- באפשרותכם **לבחור ב-3 שאלות מתוך 5 שאלות בסה"כ**. אתם רשאים לבחור שאלה נוספת שתסמנו באדום ותהווה בונוס לעבודה.
- יש לבצע את הפרויקט **בקבוצות של 3-6 סטודנטים**
- **להלן קישור לתיקייה עם כל קוד המקור הרלוונטי לפתרון השאלות** והקלטות השיעורים https://drive.google.com/drive/folders/1-hElFvPVbYbhRflsaArfw6cKRWVfjb6h?usp=sharing
- שימו לב, השאלות קשורות באופן ישיר לקוד שכתוב בשאלה עצמה ולכן אני מציע שתנסו לענות על השאלות תוך שימוש בקוד זה ובקוד מהתיקייה, לפני צפייה בסרטונים (אם תצטרכו)
- **אנא מלאו את תשובותיכם על המסמך ה-Word המקורי ולאחר מכן יש ליצור ממנו קובץ PDF.**
- הנחיות להגשה - כל קבוצה תשלח למייל natandil@ariel.ac.il את הפתרון שלה בקובץ **בפורמט PDF** בלבד. שם הקובץ יכיל את תעודות הזהות של כל חברי הקבוצה, מופרדים בפסיקים.
- תאריך אחרון להגשה: **6.9**. היקף העבודה לא צריך לקחת לכם יותר מ2-1 ימים.

מלאו כאן את פרטי המגישים:

1. שם: , ת.ז:
2. שם: , ת.ז:
3. שם: , ת.ז:
4. שם: , ת.ז:
5. שם: , ת.ז:
6. שם: , ת.ז:

שאלה 1

*יש לענות על כל סעיף בלכל היותר 5-6 שורות, בעברית בלבד. ניתן לציין מונחים באנגלית.

להלן קוד ניתן להעתקה עבור השאלה (כל סעיף מסומן בהערה לנוחיותכם):

```
public class Q1 {
    public static void main(String[] args) {
        /*part 1*/
        String s1 = "John";
        String s2 = "John";
        System.out.println(s1==s2); // true

        /*part 2*/
        System.out.println("Id of s1 = " + s1.hashCode());
        s1 = s1 + "Doe";
        System.out.println("Id of s1 = " + s1.hashCode());
        // output is not equal!

        /*part 3*/
        String name1 = "Alex";
        String name2 = new String("Alex");
        System.out.println(name1==name2); // false
        System.out.println(name1==name2.intern()); // true

        /*part 4*/
        // define and create a ThreadPoolExecutor with only 1 thread
        ExecutorService threadPool = Executors.newSingleThreadExecutor();

       // define a task with no return value that may run in a separate thread
        Runnable task =
         ()-> System.out.println("Thread ID: " +
Thread.currentThread().getId() +
                ", Thread name: " + Thread.currentThread().getName() +
                ", Thread priority: " +
Thread.currentThread().getPriority());
```

```
        // submit the task to the thread pool
        threadPool.execute(task);

        // this time -  create a Thread object and pass the task.
        Thread t1 = new Thread(task);
        t1.start();
    }
}
```

1. s1 ו-s2 הם משתנים מסוג String. כיוון שמדובר ב-reference type , היינו מצפים ש-s1 ו-s2 יצביעו לשתי כתובות שונות של אובייקטים נפרדים ב-Heap memory space אך בעלי אותו ערך.
מדוע זהו לא המצב? איזה מנגנון מעורב כאן?
2. ניתן להשתמש במתודת ה-hashcode כדי להסיק על כתובת אובייקט. מדוע הערך של s1 לא נותר זהה?
3. מדוע בעקבות ביצוע הפקודה `name1==name2.intern()` הערך הוא true? מה יקרה אם נריץ שוב את הפקודה: `name1==name2.intern()`?
4. מה ההבדל בין השימוש באובייקט אחד מסוג Thread להרצת משימה שלא מחזירה ערך לבין Thread Pool עם Thread אחד בלבד להרצת אותה המשימה?

<u>שאלה 2</u>

להלן קוד השאלה:

```
public class SharedResources {
    private static  int result = 0;

    public static void function(int  numOfIterations){
            for (int i = 0; i <numOfIterations ; i++)
                result++;
        }

    public static void main(String[] args) throws InterruptedException {
        Scanner scanner = new Scanner(System.in);
        List<Thread> threads = new ArrayList<>();
        int numOfIterations = scanner.nextInt();

        for (int i = 0; i < numOfIterations; i++) {
            Thread aThread = new Thread(()->function(numOfIterations));
            threads.add(aThread);
            aThread.start();
        }

        for (int i = 0; i < numOfIterations; i++)
            threads.get(i).join();
        System.out.println(result);
    }
}
```

*אין מגבלת שורות. יש לכתוב בעברית בלבד. ניתן לציין מונחים באנגלית.
*ניתן להניח כי המשתמש אכן העביר מספר שלם אשר שווה או גדול ל-0.

1. מהו הערך המינימלי (באופן תיאורטי) של result? הסבירו
2. מהו הערך המקסימאלי (באופן תיאורטי) של result? הסבירו
3. כיצד נקראת הבעיה בתחום ה-concurrency בה עוסקת השאלה?

שאלה 3

עליכם לשנות את המתודה factorial **בלבד**, כך שהחישוב אשר היא מבצעת יתבצע באופן אסינכרוני (כלומר, לא במסגרת ה-main thread). **אסור להוסיף/לשנות** אף חלק אחר בקוד השאלה מלבד המתודה factorial.

```
public class Q2 {
    public static void factorial(int number) {
        // change only the factorial method in order to run it asynchronously
        System.out.println("Running Factorial method in " + MyLogger.threadInfo());
        if (number <20){
            long result = 1;
            for (int i = 2; i <= number; i++)
                result *= i;
            System.out.println("factorial of " + number + "= " + result);
        }
        else
            System.out.println("computing the factorial of "
             + number + " is not possible using a " + "long type.");
    }

    public static String threadInfo(){
        return "Thread ID: " +  Thread.currentThread().getId() +
                ", Thread name: " + Thread.currentThread().getName() +
                ", Thread priority: " + Thread.currentThread().getPriority();
    }

    public static void main(String[] args) {
        // compute factorial of 4
        factorial(4);
    }
}
```

שאלה 4

להלן קוד השאלה:

```
public class HeavyThreadPool<T> {
    private final int numOfProcessors;
    private final ThreadPoolExecutor threadPool;
    private static volatile HeavyThreadPool adaptedPool;

    private HeavyThreadPool(){
        numOfProcessors = Runtime.getRuntime().availableProcessors();
        threadPool = new ThreadPoolExecutor(1,numOfProcessors, 3000
,TimeUnit.MILLISECONDS,
                new LinkedBlockingQueue<>());
    }

    public static HeavyThreadPool getPool(){
        if (adaptedPool==null){
            adaptedPool = new HeavyThreadPool(); // critical section
        }
        return adaptedPool;
    }

    public void execute(Runnable command){
         adaptedPool.threadPool.execute(command);
    }

    public Future<T> submit(Callable<T> callableTask){
        return adaptedPool.threadPool.submit(callableTask);
    }

    public void shutdown(){
        adaptedPool.threadPool.shutdown();
```

```
    }

    public static void main(String[] args) {
        HeavyThreadPool commonThreadPool = HeavyThreadPool.getPool();
       commonThreadPool.execute(()-> System.out.println("Hi HeavyThreadPool"));
    }
}
```

*מפתח המחלקה לא לקח בחשבון שיכולים להיות כמה threadים מחוץ למחלקה אשר מעוניינים להשתמש ב-instance של `HeavyThreadPool`

1. איזה design pattern שלמדנו בקורס ממומשת במחלקה `HeavyThreadPool`?
2. מהי המוטיבציה לשימוש ב-design pattern זה באופן כללי?
3. עליכם לשנות את המתודה `getPool` כך שתתמוך באופן בטוח במערכת שיכולים להיות בה כמה threadים שינסו לקבל גישה ל-instance מסוג `HeavyThreadPool`. כאתם רשאים להוסיף שדות מידע/מתודות לנוחיותכם. כתבו כאן הקוד של המתודה:

<u>שאלה 5</u>

1. להלן קוד מחלקה ב-Kotlin:

```
class Human (val fullName:String){
    val descendants: MutableList<Human> = mutableListOf()

    constructor(fullName: String, ancestor:Human): this(fullName){
        ancestor.descendants.add(this)
    }

    override fun toString(): String {
        return "Human(fullName='$fullName', descendants=$descendants)"
    }
```

עליכם לתרגם את הקוד משפת Kotlin לשפת Java.

2. להלן קוד מתודת main:

```
fun main(){
    val names = listOf("Alex","Bob","Charlie")
    val numbers = listOf(100,200,300)

    numbers.filter { it in 200..250}.forEach{ println(it)}

    val mutableNumbers = mutableListOf(100,200,300)

    for (index in 0 until mutableNumbers.size){ // for i in
range(0,mutableNumbers.size)
        mutableNumbers[index] *= 10
    }
    println(mutableNumbers)

    for((index,name) in names.withIndex()){
        print("($index,$name)")
      }
}
```

עליכם לתרגם את הקוד משפת Kotlin לשפת Java.