

אוניברסיטת בר אילן – מדעי המחשב
מבחן בתכנות מתקדם (210-89) מועד א

הנחיות כלליות

- רשמו תשובות קצרות ומדוייקות.
- משך הבחינה שעתיים.
- יש לענות על כל השאלות.
- בהצלחה!

1. עבור אפלקצית גיליון אלקטרוני דמויית EXCEL מהו התיכון (High Level Design) המוצע בהתחשב בנתונים הבאים:

- ברשת התאים החישוב מתעדכן ברקע (ניתן להתעלם מתלויות).
- ממשק המשתמש זמין ופעיל תמיד (גם בזמן חישוב, הדפסה, שמירה).
- גרפים המתעדכנים בזמן אמת ע"פ ערכי התאים המהווים את מקור הנתונים.
- יכולת טעינת ושמירת הגיליון בפורמטים שונים, תוך יכולת הרחבה עתידית.

i. מהו סרטוט המערכת ופריסתה לתתי מערכות (ומה הם שיקולי ה- Design Pattern & Application Pattern).

- ii. תכנון מקביליות (Multi-tasking) - סרטוט אינטראקציה ושיקולי התכנון (עבור חישוב והדפסה).
- iii. כיצד (בדיוק) תמומש יכולת ההרחבה לשטעינה/שמירה בפורמטים שונים (וכיצד בפועל לאחר שהתוכנה הותקנה תתבצע ההרחבה ע"י צד שלישי).

2. הסבירו והדגימו בקצרה (פסקה):

- i. עקרון העדפת הרכבה על ירושה (Prefer composition over inheritance).
- ii. Flyweight design pattern
- iii. Thread affinity
- iv. Process entry point בראיית מערכת הפעלה לעומת המשתמש.

3. שאלת threads :

i. השלימו את הקוד הבא כך שבמסך יודפס הפלט (7 נקודות)

```
Home
Sweet
Home
Sweet
.
.
.
```



כאשר כל שורה מופיעה בדיוק חצי שנייה לאחר קודמתה

```
public class ThreadForTest {

    private class Home implements Callable<String>{

        public String call() throws Exception {
            Thread.sleep(250); // notice!!
            return "Home";
        }
    }

    private class Sweet implements Callable<String>{

        public String call() throws Exception {
            Thread.sleep(400); // notice!!
            return "Sweet";
        }
    }

    public void run() throws InterruptedException, ExecutionException{
        Home home=new Home();
        Sweet sweet=new Sweet();

        ExecutorService fh, fs;

        ExecutorService executor = Executors.newFixedThreadPool(2);

        while (true){
            fh=executor.submit(home);
            fs=executor.submit(sweet);
            Thread.sleep(500);
            System.out.println(fh.get());
            Thread.sleep(500);
            System.out.println(fs.get());
        }
    }

    public static void main(String[] args) throws InterruptedException, ExecutionException {
        new ThreadForTest().run();
    }
}
```

.ii ממשו את אותה התוכנית באמצעות Timer ו TimerTask (10 נקודות)

.iii פרט עבור כל טענה אם היא נכונה או לא ומדוע (8 נקודות)

טענה א: התוכנית בסעיף א' לעיל מסונכרנת ומתוזמנת היטב (לא יהיה למשל home לאחר home ותמיד תופיע הדפסה למסך בכל חצי שנייה בדיוק)
טענה ב: התוכנית שכתבת בסעיף ב' לעיל מסונכרנת ומתוזמנת היטב.
טענה ג: התוכנית הבאה מסונכרנת ומתוזמנת היטב:

```
public class ThreadTest {  
  
    public static void main(String[] args) throws InterruptedException{  
  
        new Thread(new Runnable() {  
            public void run() {  
                while(true) {  
                    System.out.println("Home");  
                    try {  
                        Thread.sleep(1000);  
                    } catch (InterruptedException e) {}  
                }  
            }  
        }).start();  
  
        Thread.sleep(500);  
  
        new Thread(new Runnable() {  
            public void run() {  
                while(true) {  
                    System.out.println("Sweet");  
                    try {  
                        Thread.sleep(1000);  
                    } catch (InterruptedException e) {}  
                }  
            }  
        }).start();  
  
    }  
}
```

4. נתון הקד הבא :

```
public class TempFileController {  
  
    BufferedWriter out;  
    static int count=0;  
  
    public TempFileController() throws IOException {  
        count++;  
        out=new BufferedWriter(new FileWriter(count+".tmp"));  
    }  
    public void writeData() throws IOException{  
        ...  
    }  
    public void readData() throws IOException{  
        ...  
    }  
    protected void finalize() throws Throwable{  
        out.close();  
        File f=new File(count+".tmp");  
        if(f.exists())  
            f.delete(); // deletes the file  
        else  
            throw new Exception("file not found!");  
    }  
}
```

<מחשבות המתכנת>

- בכל פעם שנוצר מופע של מחלקה זו, נוצר קובץ tmp (קובץ זמני) אליו יכולים להיכתב נתונים כלשהם.
- מכיוון שיכולים להיות בזמן הריצה מופעים רבים (מאד) של מחלקה זו, וכן כל מופע צריך לעבוד עם קובץ משלו, מומש count סטטי שגדל עם כל יצירת מופע ומרכיב את שם הקובץ.
- מחיקת הקובץ מתבצעת במתודה finalize כדי לוודא שהקובץ נמחק רק כאשר כבר לא יהיה שימוש במופע זה.

- i. זהו לפחות 2 טעויות מהותיות בתוכנית זו. (10 נקודות)
- ii. מהו הנזק שיגרם בעקבות כל טעות? (5 נקודות)
- iii. הציעו תיקון לקוד כך שתימנעו מהטעויות שזיהיתם. (10 נקודות)