



המסלול האקדמי
המכללה למינהל
להצלחה יש דרך

הנדסת תוכנה מוכוונת עצמים

concurrency design patterns – תרגיל 5

שאלה 1 – Active Object (25 נק')

בהינתן הממשק:

```
public interface I {  
    void run();  
    default int getPriority(){  
        return 0;  
    }  
}
```

- צרו decorator בשם D שהופך את ה I ל Active Object.
- תוסיפו ל D מתודה בשם stop() שתעצור אותו.
 - גם אם ישנה משימה שכרגע רצה נעצור הכל והת'רד של D יגיע לסיומו.
 - ישנו מקרה קצה קטן שעליכם להתמודד איתו.

שאלה 2 – Active Object (20)

- צרו Active Object (חדש, לא decorator) בשם MyActiveObject.
- המתודה execute() תקבל אובייקט מסוג I.
- המתודה start() תפעיל את ה ActiveObject (ולא הבנאי).
- המתודה stop() תעצור את ה Active Object בדומה לשאלה 1.
- ההוצאה מהתור תינתן ע"פ העדיפות של getPriority כך שערך נמוך יותר עדיף על פני ערכים גבוהים יותר.
- חישובו על thread safety, מה עלול להיפגע? פתרו את הבעיה ללא synchronized.

שאלה 3 – Thread Pool (40 נק')

צרו מחלקה בשם MyThreadPool המממשת את Executor באופן הבא:

- בבנאי היא תקבל את הכמות המקסימאלית של הת'רדים שיש ליצור.
- בכל פעם שנפעיל את execute():Runnable, וכל עוד לא הגענו למס' המקסימאלי של ת'רדים, ניצור מופע חדש של MyActiveObject (זה משאלה 2) עם הוראה להריץ את ה Runnable הזה.
- כשימשיכו להגיע Runnables נקצה אותם ל ActiveObject הראשון שהוא עם הכי פחות משימות.
- לרשימת ה Active Objects תקראו pool, תנו לה הרשאה דיפולטיבית.
- המתודה stop() תעצור את כל המשימות של כל ה ActiveObjects שב pool שלנו.
- המתודה getActiveThreadsCount() תחזיר ב int את מס' ה ActiveObjects הפעילים.
- במחלקה MyActiveObject תוסיפו את המתודה getTasksCount() שתחזיר ב int את מס' המשימות שממתינות בתור של ה active object.

שאלה 4 – Thread Pool + הרכבה מוכוונת עצמים (15)

צרו את המחלקה MyThreadPool2 המהווה סוג של MyThreadPool, שמוסיפה את המתודה:

```
MyFuture<V> submit(Callable<V> callable);
```

- MyFuture<V> היא מחלקה שמחזיקה ערך מסוג V.
- המתודה set(V v) תזין ל V ערך.
- המתודה get() תחזיר את ה V. אך אם הוא עדיין null, היא תגרום לקורא ל get() להמתין עד אשר יזון V כלשהו במתודת ה set.
- השתמשו ב guarded suspension כדי לממש זאת.
- מכיוון ש wait דורשת synchronized וזה דבר יקר, דאגו שהקריאה ל synchronized תהיה רק כאשר באמת צריך, ולא כל הזמן.
- עליכם לדאוג גם למקרה בו הרבה ת'רדים פנו ל get לפני ש V מוכן.

ה MyFuture יוחזר מיד, אך ה V שה callable מחזיר, יוכנס ל MyFuture רק לאחר שה call של ה callable סיימה את החישוב שלה.

הגשה

עליכם להגיש למערכת ההגשה תחת ex4 של software engineering את הקבצים הבאים ואותם בלבד:

- D.java
- MyActiveObject.java
- MyFuture.java
- MyThreadPool.java
- MyThreadPool2.java

על כל המחלקות להיות ב package בשם test.

****הערה:** חלק מהבדיקות יכולות להיכשל בגלל עומס רגעי בשרת. אם ל MainTrain יש שוני בין הריצה בבית לריצה במערכת אז פשוט תגישו שוב. כנ"ל לגבי מצב הגשה, תוכלו להגיש מס' פעמים כדי לוודא שלא נפלתם על עומס.

בהצלחה!