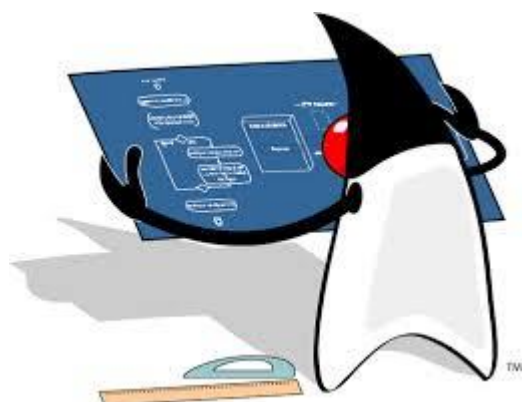




אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

מבחן לדוגמא

נושאים מתקדמים בתכנות



ד"ר אליהו חלסצ'י
khalastc@post.bgu.ac.il

הקדמה

במבחן זה תיבחנו על נושאים מתקדמים בתכנות. בפרט, המבחן מחולק לשאלות בלתי תלויות על קטעי קוד שיכולים להימצא בשכבות שונות של אפליקציית דסקטופ. עליכם לממש אלגוריתם כלשהו בצורה מכוונת עצמים, וליישם תבניות עיצוב היכן שצריך בקוד. הקוד נכתב בשפה Java, ואתם נדרשים לדעת כיצד להשתמש במבני נתונים שונים של java.util, לטפל ב IO (קבצים ולאוו תקשורת ב sockets), לכתוב קוד גנרי (<T>), ולנהל threads ע"פ concurrency patterns שונים.

עליכם לענות בטופס המבחן בלבד.

פרטים נוספים: חומר פתוח, שעתיים וחצי, 3/3 שאלות.

שאלה 1 - Controller (30 נק')

השלימו את הקוד הבא כך שניצור ActiveObject המריץ אובייקטי Command ע"פ סדר עדיפות.

- עדיפות תינתן לערך הקטן יותר
- כשתור העדיפויות ריק, יש לדאוג שלא מתבצע busy waiting
- המתודה stop() תגרום ליציאה מסודרת – המשימות שבתור ימשיכו לרוץ, אך לא יכנסו משימות נוספות מהרגע שהופעל stop. לאחר שהמשימות שבתור סיימו את ריצתן, הת'רד של ה active object יסתיים גם הוא, מיד.
- הכנסת אובייקטי Command לתור העדיפויות צריכה להיות thread safe, ואף לחסוך בנעילות כאשר אין בהן צורך.
- במידה ונעשה ניסיון להכניס Command לאחר שהופעלה המתודה stop() ייזרק אובייקט מסוג Exception המכיל את ההודעה:
"cannot insert new commands after the controller has stopped" ○

השלימו:

```
public class MyActiveController {  
  
    public interface Command{  
        public void execute();  
        default int getPriority(){return 0;}  
    }  
  
    _____ stop;    // 2 points  
  
    Thread activeThread;  
  
    PriorityQueue<_____> queue; // 2 points  
  
    public MyActiveController() {  
        stop=false;  
        queue=new PriorityQueue<>() // 5 points  
  
        _____  
        _____  
        _____  
        _____  
        _____  
  
    };
```

```

        activeThread=new Thread()->{ // 10 points
            _____
            _____
            _____
            _____
            _____
            _____
            _____
            _____
        });

        activeThread.start();
    }

    public void insertCommand(Command c) _____{ // 12 points:
        _____
        _____
        _____
        _____
        _____
    }

    public void stop(){ // 4 points
        _____
        _____
    }
}

```

שאלה 2 – Model (35 נק'):

סעיף א' (20 נק')

נתון לכם הפסאודו-קוד הבא, השלימו את הקוד הבא כדי לממש את האלגוריתם בצורה מכוונת עצמים. על המימוש להפריד בין האלגוריתם לבין הבעיה שהוא פותר.

Selection_sort(list of n elements)

1. For $i=0..(n-1)$
2. currentMin = list[i]
3. newMin = minimum value in the list from index $i+1$ to n
4. If currentMin \neq newMin then swap them

השלימו:

```
public interface Sorter { // 2 points

    <E> void sort(_____ sortable);

}
```

```
public interface Sortable<E> { // 3 points
```

```
void swap(int i,int j);
int size();
}
```

```
public class SelectionSorter implements _____{

    @Override
    public <E> void sort(_____ sortable) { // 15 points
```

[illegible]

שאלה 3 O/I ומבני נתונים (35 נק):

השלימו את הקוד הבא כך שבהינתן לקוח שמתחבר מעל TCP/IP, נקרא ממנו שורות מחרוזת עד להגעה של המחרוזת "done". כל מחרוזת שמתחילה ב "ERROR:" תיכתב לקובץ בשם errors.txt ואילו בכל מחרוזת אחרת, ננהג באופן הבא:

- נפריד את שורת המחרוזת למערך של מילים לפי ",
- נספור כמה פעמים הופיעה כל מילה בכל הקלט של הלקוח
- את 10 המילים הנפוצות ביותר נשלח כלקוח לשרת שנמצא ב 10.0.0.139 בפורט 6400.
 - השליחה תתבצע מהמילה הנפוצה ביותר לזו הפחות נפוצה

```
public class Handler {
```

```
wordsCount;
```

```
public Handler() {
    wordsCount=new _____;
}
```

```
public void start(int port) throws Exception{ // 15 points
```

[illegible]

```
// send the top 10 referenced words to the server
private void sendTop10Words() throws Exception{ // 10 points
```

```


```

```
// adds the count for each word given a line
private void countWords(String line) { // 10 points
```

```


```

```
}}                                     בהצלחה!
```