

המסלול האקדמי

המכללה למינהל

להצלחה יש דרך

מועד א' תשע"ז

פיתוח תוכנה מתקדם

ד"ר אליהו חלסצ'י
eliahukh@colman.ac.il

שאלה 1 – עדכונים, תקשורת (30 נק')

נתונה המחלקה Level. מחלקה זו מייצגת שלב במשחק כלשהו עם דמות. מבנה השלב מיוצג ע"י מערך דו ממדי של char, ומיקום הדמות מיוצג ע"י המשתנים cx,cy. מחלקה זו חושפת enum לכיוונים, ממשק עבור מדיניות התזוזה של הדמות ומתודה להזנת המדיניות (dependency injection). במתודה moveCharacter() נעשה שימוש במדיניות שהוזקה על מנת להזיז את הדמות אם זה צעד חוקי.

```
package test;
```

```
public class Level {
```

```
    public enum Direction{up,down,left,right}
```

```
    public interface MovePolicy{
        boolean isLegal(char[][] data,int cx,int cy,Direction d);
    }
```

```
    private char[][] data;
    private int cx,cy;
    MovePolicy mp;
```

```
    public void setMovePolicy(MovePolicy mp){
        this.mp=mp;
    }
```

```
    public void setData(char[][] data, int cx, int cy){
        this.data=data;
        this.cx=cx;
        this.cy=cy;
    }
```

```
    public void moveCharacter(Direction d){
        if(mp.isLegal(data, cx, cy, d)){
            switch(d){
                case up: cy--;break;
                case down: cy++;break;
                case left: cx--;break;
                case right: cx++;break;
            }
        }
    }
```

```
}
```

סעיף א' - 10 נקודות (שהן די נקודות מתנה, כדי להפחית blackout במבחן ולחמם מנועים... 🤪)

- העתיקו את קוד המחלקה לעיל למחלקה בשם LevelA ב package בשם test.
- שנו את הקוד של המחלקה הזו כך שבכל פעם שקרה איזשהו שינוי לאובייקט מסוג LevelA, כל מי שירצה יוכל לקבל על כך נטיפיקציה.
- עליכם להשתמש בפתרון סטנדרטי שמוכר למתכנתים אחרים
 - כלומר, יש להשתמש בקוד מתוך ספריות הסטנדרטיות של java שמתכנתים אחרים היו מצפים מכם להשתמש בו, ולא להמציא לשם כך מחלקות חדשות.

רמז: מדובר ב design pattern שלמדנו

סעיף ב' - 20 נקודות

העתיקו את המחלקה Level לעיל לפרויקט שלכם (הפעם מבלי לשנות את שמה). ממשו את קוד הלקוח הבא, כך שיתחבר לשרת ב ip וב port, ישלח לו אובייקט מסוג Level בפורמט XML ויתנתק. מותר לערוך שינויים במחלקה Level ע"פ הצורך.

```
public class MyClient {
    public static void sendLevel(String ip,int port,Level level) throws
    IOException{
        // write your answer here
    }
}
```

שאלה 2 חשיבה, Command, עיון קצר ב javadoc (35 נק')

נתונה המחלקה Controller החושפת ממשק פקודה שכולל אחזור של ID לצורך זיהוי הפקודה.

```
package test;

public class Controller {

    public interface Command{
        int getID();
        void execute();
    }

    PriorityQueue<_____> queue;

    public Controller() {}

    public void insertCommand(Command c){}

    public void executeOne(){
        if(!queue.isEmpty())
            queue.poll().execute();
    }

    public void executeAll(){
        while(!queue.isEmpty())
            queue.poll().execute();
    }
}
```

מטרת ה Controller היא למזער את הזמן שהפקודות ממתינות בתור. לפיכך, יש להריץ את הפקודות ע"פ הזמן שלוקח להן לרוץ – מהזמן הקצר ביותר ועד לזמן הארוך ביותר. אולם, לא ניתן להעריך את הזמן שלוקח לפקודה לרוץ ללא מדידה.

לכן, נגדיר את המנגנון הבא. נשתמש בתור עדיפויות כדי להעדיף פקודות עם זמן ריצה קצר יותר. עבור פקודות שלא הרצנו בעבר (ID חדש) נניח שזמן הריצה שלהן הוא 0. בכל פעם שנריץ פקודה כלשהי נמדוד את זמן הריצה שלה. נתון זה ישמש אותנו לפעם הבאה שנתבקש להריץ את אותה הפקודה – כך כל פקודה תתמקם בתור לפי זמן הריצה שלה בפעם הקודמת.

לדוגמא הכנסנו את הפקודות 1,2,3, וקראנו ל executeAll. הפקודות ירוצו (בסדר לא חשוב) וזמן הריצה של כל אחת יימדד. נניח ש 3 היא הקצרה ביותר, ו 1 היא הארוכה ביותר. לכן בפעם הבאה שנבקש להריץ את הפקודות סדר הריצה יהיה 3 ואז 2 ואז 1.

שימו לב שחל איסור לשנות את הממשק.

עליכם להשלים את הקוד לעיל כדי לממש מנגנון זה. בפרט, עליכם לממש את insertCommand עבור הכנסת פקודות ל Controller ע"פ המדיניות שהגדרנו.

טיפים:

- כדי למדוד זמן ריצה יש לדגום את שעון המחשב. בדקו מה עושה זאת במחלקה System.
- ע"פ הממשק, אובייקט מסוג Command לא נדרש למדוד לעצמו את זמן הריצה. ייתכן ותצטרכו ליצור מחלקת עזר פרטית (יתכנו מס' פתרונות שונים).
- אולי תצטרכו מבנה נתונים נוסף כדי לקשר בין פקודות לזמני ריצה (כנ"ל).

שאלה 3 Design Patterns (35 נק')

סעיף א' – 15 נק'

בפרויקט אחד נתונים האלמנטים הבאים:

```
public interface Moveable {
    public enum Direction{up,down,left,right}
    boolean moveMe(Direction d);
}

public class Box implements Moveable{
    @Override
    public boolean moveMe(Direction d) {
        // TODO Auto-generated method stub
        return false;
    }
}
```

ובפרויקט אחר יש צורך באובייקטים מסוג הממשק Pushable, המגדיר פעולת תזוזה בכיוון 0..359 מעלות.

```
public interface Pushable {
    void push(int heading); // 0..359 Degrees
}
```

כתבו מחלקה בשם PushableAdapter שתממש object adapter המאפשר להמיר אובייקט מסוג Box ל Pushable. הקפידו שהזנת "object" תהיה בבנאי של PhshableAdapter ושאינ בנאים אחרים. אופן ההמרה יהיה כך:

- מעלות 45 עד 134 יוגדר כתזוזה ימינה
- מעלות 135 עד 224 יוגדר כתזוזה למטה
- מעלות 225 עד 314 יוגדר כתזוזה שמאלה
- מעלות 315 (דרך 0) עד ל 44 יוגדר כתזוזה למעלה

יש להקפיד על עקרונות Solid.






סעיף ב' – 20 נק': נתונה המחלקה הבאה המהווה שאלה אמריקאית. המערך patterns מכיל שמות של design patterns שלמדנו במהלך הסמסטר. בכל מתודה מופיעה דוגמת קוד קטנה בהערה. עליכם להחזיר מחרוזת מתוך מערך patterns (לדוג' patterns[0]) בהתאמה לתבנית אליה משתייכת דוגמת הקוד.

```
public class AmericanQuestion {
    String[] patterns={"Singleton", "Strategy", "Class Adapter", "Object
Adapter", "Decorator", "Observer", "Facade", "Command", "Factory"};
    public String q1(){
        // which design pattern do we see here?
        // new Thread()->System.out.println("hello world!").start();
        return null;
    }
    public String q2(){
        // which design pattern does InputStreamReader applies?
        // new BufferedReader(new InputStreamReader(System.in));
        return null;
    }
    public String q3(){
        // which design pattern do we see here?
        // Model m=Model.getInstance();
        return null;
    }
    public String q4(){
        // which design pattern do we see here?
        // Model m=new MyModel();
        return null;
    }
}
```

הגשה

תחילה וודאו שכל המחלקות שבכוונתכם להגיש נמצאות ב package בשם test.

עליכם להגיש את הקבצים הבאים ואותם בלבד:

-  AmericanQuestion.java
-  Controller.java
-  LevelA.java
-  MyClient.java
-  PushableAdapter.java

בהצלחה!