



המכללה האקדמית להנדסה סמי שמעון

מדור בחינות ומערכת שעות

המחלקה להנדסת תוכנה

6/07/2022

13:30-16:30

תכנות מונחה עצמים מתקדם

מועד ב'

ד"ר מרינה ליטבק

תשפ"ב סמסטר ב'

חומר עזר : נא סמן במשבצת המתאימה את המתאים

___ * ניתן להשתמש בכל מחשבון

___ * לא ניתן להשתמש במחשבון Casio FX-991EX

___ * V_ לא ניתן להשתמש במחשבון

___ * V_ לא ניתן להשתמש בחומר עזר

___ * מותר שימוש בדף נוסחאות, כמפורט: _____

___ * הבחינה בחומר פתוח – מותר להשתמש בכל חומר עזר מודפס או כתוב

הערות

___ יש לענות על כל השאלות במקומות המיועדים ע"ג טופס השאלון בלבד

___ יש להחזיר את השאלון ביחד עם הכריכה/מחברת.

השאלון מכיל 16_ עמודים (כולל נספח, עמוד זה, וטיוטה) ומסתכם ל-109 נקודות.

יש לענות את התשובות בשאלון בלבד!

יש לענות תשובות מלאות בתמצות ובקיצור.

תשובות מסורבלות המכילות נתונים שלא קשורים לשאלה או פרטים שגויים יובילו לאובדן נקודות.

המקום שהושאר לכל תשובה הוא יותר מהדרוש.

בהצלחה !

=====

שאלה 1 (30 נק')

לגבי כל טענה – קבעו האם נכונה או לא ונמקו.

א. (3 נק') תכונת Reentrance מאפשרת להגביל מס' תהליכונים שיכולים לבצע אותו קטע קוד במקביל למס' יותר גדול מ-1.
נכון / לא נכון

נימוק: _____

ב. (3 נק') Vector מובנה הוא thread-safe ולכן מומלץ להשתמש בו תמיד, גם ללא ריבוי תהליכונים (multithreading).
נכון / לא נכון

נימוק: _____

ג. (3 נק') במחלקה שמממשת ממשק ב-Java ניתן לממש מתודות שלא קשורות לממשק.
נכון / לא נכון

נימוק: _____

ד. (3 נק') לא ניתן לנעול מנעול של מופע (this) בקוד של מתודה סטטית.
נכון / לא נכון

נימוק: _____

ה. (3 נק') הפעלת מתודה lock() על מופע של ReentrantLock ע"י תהליכון X גורמת לנעילת המופע ומונעת מכל תהליכון מתחרה אחר להפעיל כל קטע קוד שנועל את אותו המופע אך גישה לתהליכון X אינה מוגבלת.

נכון / לא נכון

נימוק: _____

ו. (3 נק') במידה ומתודה X מסונכרנת (synchronized) מכילה בתוכה הפעלה של מתודה מסונכרנת נוספת (שתיהן מתודות רגילות ולא סטטיות), שום תהליכון לא יוכל לסיים את הפעלתה של המתודה X כי הוא ייחסם על נעילת this שנייה.

נכון / לא נכון

נימוק:

ז. (3 נק') ב-JAVA, פעולת השמה רגילה על משתנה X מטיפוס int היא thread-safe כי מדובר על הטיפוס הפרימיטיבי.

נכון / לא נכון

נימוק:

ח. (3 נק') מחלקה אבסטרקטית (abstract class) לא יכולה להכיל מתודה רגילה שהיא מסונכרנת (synchronized) כי אין לה (למחלקה) this.

נכון / לא נכון

נימוק:

ט. (3 נק') שימוש בסמפור יכול לספק סדר יחסי בין תהליכונים שבו נקבע אילו תהליכונים יבצע את הפעולות שלהם לפני ואילו אחריו.

נכון / לא נכון

נימוק:

י. (3 נק') שימוש בסמפור יכול להגביל מס' תהליכונים המבצעים בו-זמנית קטע קוד מסויים למספר גדול מ-1.

נכון / לא נכון

נימוק:

שאלה 2 (49 נק')

בדף הבא נתון קוד של מחלקה המממשת כתיבה לקובץ לוגים.

א. (15 נק') יש לממש ניהול קובץ לוגים מותאם למדינה שבה קוד מופעל. כלומר, כתיבה תתבצע בשפה רשמית של המדינה. כמו כן, בפתיח של קובץ תהיה רשומה שמציינת מיקום של השרת וכל שורה תהיה ממוספרת החל מפתיחת קובץ (ראו דוגמת הרצה).

ממשו את הרעיון הזה בעזרת תבנית אדפטר (Adapter). יש להשלים קוד של 3 מחלקות למטה, אחת מהן מממשת את ההתאמה הכללית ו-2 מחלקות אחרות ממשות את ההתאמה הספציפית – לאה"ב וישראל. קחו לחשבון שמימוש שלכם אמור לאפשר הוספת אדפטרים חדשים (למדינות חדשות) ללא שינוי במחלקות הקיימות.

הערה חשובה: יש להימנע מקוד חוזר כמה שניתן!

דוגמת הרצה (מחייבת):

```
public class User {

    public static void main(String[] args) {
        LogAdapter adapter;
        try {
            adapter = new HeLogAdapter(new File("heLog.txt"));
            adapter.writeLog("שורה שלי");
            adapter.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

פלט (בקובץ heLog.txt):

Mon Jun 13 16:52:04 IDT 2022	שרת ישראלי
Mon Jun 13 16:52:04 IDT 2022	שורה מס' 0: שורה שלי

הערה: תאריך הוא רלוונטי לזמן הרצת דוגמה.


```

1
2
3
4 public class LogFile {
5
6     private Writer out;
7
8     public LogFile(File f) throws IOException {
9         FileWriter fw = new FileWriter(f);
10        this.out = new BufferedWriter(fw);
11    }
12
13    public void writeEntry(String message) throws IOException {
14        Date d = new Date();
15        out.write(d.toString());
16        out.write('\t');
17        out.write(message);
18        out.write("\r\n");
19    }
20
21    public void close() throws IOException {
22        out.flush();
23        out.close();
24    }
25
26    protected void finalize() {
27        try { this.close(); }
28        catch (IOException ex) { }
29    }
30 }

```




המכללה האקדמית להנדסה סמי שמעון

```
_____class LogAdapter {  
  
_____  
_____  
  
public LogAdapter(_____) throws IOException {  
  
_____  
_____  
  
}  
  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
  
public void close() throws IOException {  
  
_____  
  
}  
  
}
```




המכללה האקדמית להנדסה סמי שמעון

```
public class EnLogAdapter _____{
```

```
}
```

```
public class HeLogAdapter _____{
```

```
}
```


ב. (6 נק') יש כמה תהליכונים שכותבים לאותו קובץ לוגים דרך אותו מופע של LogFile. יש להתאים את המימוש שלו כך שיהיה thread-safe, כלומר שכמה תהליכונים יכלו לכתוב לאותו קובץ במקביל.
הערה: יש לכתוב רק שורות קוד חדשים ועדכונים הרלוונטים לדרישה ולציין את מיקומן.

ג. (12 נק') יש להוסיף מתודות חדשות למחלקות רלוונטיות עבור הדפסה של כל רשומה שנכתבת לקובץ (לאחר התאמה למדינה). הדפסה גם צריכה להיות thread-safe.
הערה: חובה להקפיד על הפתרון היעיל!

ד. (10 נק') יש להוסיף תמיכה ביצירה אוטומטית של אדפטר מטיפוס המתאים לפי שם המדינה. תשתמשו בתבנית עיצוב Factory Method

דוגמת הרצה (מחייבת):

```
public static void main(String[] args) {
    LogAdapter adapter;
    try {
        adapter = Factory.getAdapter("Israel", new File("heLog.txt"));
        adapter.writeLog("שלי שורה");
        adapter.close();
    } catch (IOException e) { e.printStackTrace();}
}
```

ה. (6 נק') תוסיפו טיפול במקרה של בעיית כתיבה לקובץ (IOException). יש להדפיס הודעת שגיאה מתאימה למשתמש.

שאלה 3 (30 נק')

שאלה זאת מתייחסת לעבודות בית בנושא של אקווריום שמימשתם השנה.
נניח שהחלטתם להכליל את המימוש בכמה כיוונים:

1. להוסיף אפשרות לעקוב אחרי שינויים באקווריום ע"י ניהול קובץ לוגים. כל דג או מדוזה תרשום הודעה (לוג) על שינוי שזה עובר. למשל, כש מצב של דג הופך מ-"רעב" ל-"שבע", תירשם רשומה בקובץ לוגים עם שעה, תיאור השינוי ומס' הדג. יהיה קובץ אחד לכולם.
 2. להכליל את 1 כך שבוודאות יהיה מופע יחיד של LogFile באקווריום (טיפוס שמממש את הכתיבה לקובץ לוגים) במצב של ריבוי תהליכים.
 3. במקום שהתהליכים המייצגים דגים ומדוזות יתעסקו בניהול קובץ לוגים, כל הרישומים לקובץ תתבצע ע"י התהליך האחראי לזה (מנהל קובץ לוגים). הוא יממש אופן רישום (תוכן ופורמט) לכל מצב אפשרי, ותהליכים של דגים יפעילו את זה בזמן אמת.
- זהו 3 תבניות עיצוב שמתחייבות מתיאור זה. לכל תבנית פרטו בדיוק מהי והסבירו (במשפט אחד או שניים בלבד) מדוע היא מחויבת. תנו פירוט של מחלקות/ממשקים ויחסים ביניהם שתבנית מחייבת לממש (כולל הקשר, כלומר מחלקות הקיימות במימוש שלכם שהתבנית הנתונה מתקשרת איתן).

רמז: כל 3 תבניות הן מקטגוריה Concurrency

א. (10 נק')

a. (2 נק') שם התבנית הראשונה (שעונה לדרישות של סעיף 1):

b. (3 נק') הסבר מדוע היא מתאימה :

c. (5 נק') מחלקות/ממשקים וקשרים ביניהם (דיאגרמת UML אמורה להכיל אלמנטים כללים של התבנית והמחלקות הרלוונטיות מהמימוש הנוכחי)

a. (2 נק') שם התבנית השנייה (שעונה לדרישות של סעיף 2):

b. (3 נק') הסבר מדוע היא מתאימה :

c. (5 נק') מחלקות/ממשקים וקשרים בניהם (דיאגרמת UML אמורה להכיל אלמנטים כללים של התבנית והמחלקות הרלוונטיות מהמימוש הנוכחי)

a. (2 נק') שם התבנית השלישית (שעונה לדרישות של סעיף 3):

b. (3 נק') הסבר מדוע היא מתאימה :

c. (5 נק') מחלקות/ממשקים וקשרים בניהם (דיאגרמת UML אמורה להכיל אלמנטים כללים של התבנית והמחלקות הרלוונטיות מהמימוש הנוכחי)

Concurrency:

- Double-checked lock
- Read write lock
- Thread pool
- Observer/Listener

Creational:

- Singleton
- Factory method
- Abstract Factory
- Builder
- Prototype

Structural:

- Adapter
- Delegator
- Bridge
- Decorator
- Proxy

Behavioral:

- Iterator
- Chain of Responsibility
- Memento
- State
- Visitor
- Mediator



המכללה האקדמית להנדסה סמי שמעון
טיוטה



המכללה האקדמית להנדסה סמי שמעון
טיוטה



המכללה האקדמית להנדסה סמי שמעון
טיוטה

