

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

ת.ז. הסטודנט:	ברקוד נבחן
מספר חדר:	
מספר נבחן:	
מספר אסמכתא:	

מבחן בקורס: תכנות פיתוח תוכנה מתקדם 2

תאריך הבחינה: 26.07.18

שנת הלימודים: תשע"ח, סמסטר: ב', מועד: א'.

משך הבחינה: 4 שעות

שם המרצה/ים:	
ד"ר אליהו חלסצ'י	
שם המתרגל/ים:	

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות בבחינה: 10 שאלות אמריקאיות ושאלה תכנותית אחת

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- מותר השימוש בכל חומר עזר לרבות חומר מודפס, עמדת המחשב במעבדה, דיסק און קי.
 - אסור השימוש בכל אמצעי תקשורת לרבות מחשב אישי, מחשב לוח או טלפון.
- יש לענות במחשב
- לא נדרש להחזיר את השאלון
- לא מצורף נספח לבחינה
- מחברת טיוטה: אין
- מחברת נפרדת לכל שאלה: לא
- למבחן בדיקה אוטומטית ולכן יש להגיש ע"פ הוראות ההגשה במדויק!!!
- שאלה שלא תתקמפל או שתהיה לה שגיאת ריצה ציונה אוטומטית 0. אין בדיקה ידנית למבחן. יש מספיק זמן למבחן כדי לנפות את כל השגיאות.
- אזהרה: למבחן מתקיימת בדיקת העתקות אוטומטית ובעבר סטודנטים שהעתיקו הועלו לוועדת משמעת והורחקו מהמכללה לשנה. אל תעתיקו במבחן.

בהצלחה!!

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

חלק אמריקאי (60 נק')

בסמסטר זה עבדתם על אבני דרך 4-6.

- באבן דרך 4 שכללתם את צד השרת כך שיעבד מספר לקוחות במקביל ע"י thread pool.
- באבן דרך 5 כתבתם את אפליקציית הלקוח עם GUI ב javaFX, יצרתם רכיב גרפי משלכם, ומימשתם ארכיטקטורה של MVC + OBSERVER או MVVM.
- באבן דרך 6 כתבתם אפליקציית רשת המאפשרת לעבוד מעל שכבת ORM ולנהל נתונים כמו אלו שחקנים פתרו אלו שלבים במסד נתונים רלציוני.

לפניכם 10 שאלות אמריקאיות אודות חומר הפרויקט לפי הנושאים לעיל. במחלקה American קיימות המתודות q1 עד q10 בהתאמה המחזירות char. עליכם להחזיר "d"-"a" בהתאמה לאופציה שבחרתם בתשובות. לכל שאלה יש רק תשובה אחת. הניקוד לכל שאלה מופיע לצידה.

1. מה מהמשפטים הבאים מתאר נכון synchronized? (6 נק')
 - a. ת'רד שהפעיל מתודה מסונכרנת באובייקט נועל את האובייקט ואף ת'רד אחר לא יכול להפעיל אף מתודה אחרת כל עוד האובייקט נעול
 - b. ת'רד שהפעיל מתודה מסונכרנת באובייקט נועל את האובייקט ואף ת'רד אחר לא יכול להפעיל אף מתודה מסונכרנת אחרת באף אובייקט מאותו הסוג כל עוד האובייקט נעול
 - c. ת'רד שהפעיל מתודה מסונכרנת באובייקט נועל את האובייקט ואף ת'רד אחר לא יכול להפעיל אף מתודה אחרת באותו האובייקט כל עוד האובייקט נעול
 - d. ת'רד שהפעיל מתודה מסונכרנת באובייקט נועל את האובייקט ואף ת'רד אחר לא יכול להפעיל אף מתודה מסונכרנת באותו האובייקט כל עוד האובייקט נעול
2. מה מהבאים נכון עבור משתנה מסוג volatile? (6 נק')
 - a. הוא משתנה אטומי, וככזה הוא יכול להחליף את השימוש ב synchronized
 - b. הוא מתאים למקרים בהם יש צורך לכמה תהליכים במקביל לראות את אותו הערך של המשתנה כאשר סדר הקריאות אינו משנה
 - c. הוא מתאים למקרים בהם יש צורך לכמה תהליכים במקביל לראות את אותו הערך של המשתנה כאשר סדר הקריאות כן משנה
 - d. הוא נמצא תמיד ואך ורק ב RAM, וזו התכונה היחידה שמייחדת אותו
3. מה מהבאים נכון עבור המשתנים Callable c, Future f? (6 נק')
 - a. f מכיל את c
 - b. f יכול בעתיד את c כאשר c יסיים להפעיל את המתודה call המחזירה ערך מסוג פרמטרי V.
 - c. f יכול בעתיד איזשהו ערך מסוג פרמטרי V כאשר c יסיים להפעיל את המתודה call שמחזירה V.
 - d. המתודה call של c מחזירה את f מיד, כך שיכיל ערך מסוג פרמטרי V בבוא העת.

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

4. איזה סוג אובייקט חזר בכל אחד מהמקומות המסומנים בהערה בקוד הבא: (2 נק')

```
public String deepThought() {
    // takes a really really long time...
    return "42";
}
CompletableFuture.supplyAsync( ()->{return deepThought();}) // A: _____
    .thenApply(answer->Integer.parseInt(answer)) // B: _____
    .thenApply(x->x*2) // C: _____
    .thenAccept(answer->System.out.println("answer: "+answer));
// D: _____
```

- a. A: CompletableFuture<String>
B: CompletableFuture<Integer>
C: CompletableFuture<Integer>
D: Integer
- b. A: CompletableFuture<String>
B: CompletableFuture<Integer>
C: CompletableFuture<Integer>
D: void
- c. A: Future<String>
B: Future<Integer>
C: Future<Integer>
D: void
- d. A: CompletableFuture<String>
B: CompletableFuture<Integer>
C: CompletableFuture<Integer>
D: CompletableFuture<void>

5. כשיצרתם רכיב גרפי משלכם לפרויקט, היה עליכם להפריד בין הפונקציונאליות של הרכיב, לבין התצוגה שלו, ובין דברים שעשויים להשתנות לאלו שלא. חלק מאלה נממש בתוך המחלקה המייצגת את הרכיב, ואילו את השאר נעדיף לקבל כפרמטר "מבחוץ". עבור לוח משחק הצינורות, מה מבין הדברים הבאים נעדיף לא לקבל כפרמטר מבחוץ? (6 נק')

- a. שמות קובצי התמונות שמייצגות את חלקי הצינור
- b. Event handler ל double click \ click על חלקי צינור
- c. Event handler למצב פתור
- d. Event handler לשינוי בצעדים שביצע המשתמש

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

6. בצד הלקוח נתון ה controller הבא. לחיצה על כפתור solve כרוכה לפונקציה ב controller שבתורה מבקשת מהמודל לפתור את השלב הנוכחי. כשמגיע פתרון למודל הוא שולח נטיפיקציה ל controller ששולף מהמודל את הפתרון ומתרגם אותו לפורמט שה view מבין, ושומר את הנתון בתכונה אצלו שה view כרוך אליה ולכן ה view גם יציג אותה. מה מבין המשפטים **אינו** הבאים נכון? (10 נק')

- a. ה controller הוא ViewModel
- b. ה controller לא מהווה באמת controller מפני שהוא לא מחליט על מדיניות הפעלה כלשהי
- c. ה controller מהווה באמת controller מפני שהוא כן מחליט על מדיניות הפעלה כלשהי
- d. ה controller הפריד בין ה view למודל

7. באפליקציית הלקוח, על איזו שכבה נטיל את הפונקציונאליות הבאה? בהינתן שלב של משחק צינורות, "ננרמל" את הצינורות לכיוון מסוים ע"פ כללים קבועים כך ששליבים המהווים תמורה (פרמוטציה) אחד של השני ייחשבו מעשית כאותו השלב.

- a. מודל
- b. View
- c. Controller
- d. ViewModel

8. מה מהבאים **לא בהכרח נכון** לגבי שכבת ORM?

- a. מספקת קישוריות למסד נתונים רלציוני
- b. מאפשרת שמירה וטעינה של אובייקטים מתוך מסד הנתונים
- c. יוצרת שפת שאליות מוכוונת עצמים לשליפת אובייקטים
- d. ניתן ליצור שכבה כזו גם לבד ללא עזרה של ספריית קוד חיצונית

9. מה מהבאים הכי נכון עבור Restful API? (6 נק') כשנרצה להוסיף, למחוק, לעדכן או לקבל את פרטיו של משתמש כלשהו...

- a. ניגש לאותה כתובת אינטרנט, עם פקודת ה HTTP הבאות בהתאמה:
get, put, delete, post
לדוגמא http://localhost:8080/users/12345789
- b. ניגש לכתובות אינטרנט שונות, כאשר הפעולה הרצויה תיבחר ע"י query parameter (נתונים המופיעים לאחר "?")
לדוגמא http://localhost:8080/users/12345789?action=delete
- c. ניגש לכתובות אינטרנט שונות, כאשר הפעולה הרצויה תיבחר ע"י path parameters (מילים שבין "/"-ים)
לדוגמא http://localhost:8080/users/12345789/delete
- d. נשלח מחרוזת מעל TCP/IP בפורמט קבוע ומוסכם מראש

10. את הפרויקט ככל הנראה ממשתם על מחשבים אישיים. אולם, במוצר אמתי צריך לדאוג לסקלבליות הפתרון. מה מהבאים יכול לקבוע את הפרמטר ההולך וגובר של הבעיה? (6 נק')

- a. מספר המשתמשים
- b. מספר הבקשות לפתרון שלבים בזמן נתון
- c. זמן הריצה של האלגוריתמים השונים לפתרון השלבים
- d. כמות השרתים שבהם נשתמש

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

שאלה תכנותית (40 נק')

ברצוננו ליצור חיקוי Java-אי ומשודרג ל delegates & events הקיים ב C#. עליכם לממש את המחלקה Event בקובץ Event.java כך שה main הבא יתקמפל וירוך ללא שגיאות, ויצור את הפלט הרצוי המוגדר בהערות.

```
public class MainTrain2 {

    public static void func(Event invoker, Object about){
        System.out.println("func was invoked by "+invoker);
        if(about!=null){
            System.out.println("this is about "+about);
        }
    }

    public static void main(String[] args) {
        Event e1=new Event("E1");
        Event e2=new Event("E2");

        e1.addDelegate((i,a)->func(i,a));
        e1.apply(); // output: func was invoked by E1
        System.out.println();
        e1.apply("hello world!"); // output:
                                //func was invoked by E1
                                //this is about hello world!

        System.out.println();
        e2.addDelegate((i,a)->func(i,a));
        e2.apply(); // output: func was invoked by E2

        System.out.println();
        e1.addDelegate((i,a)->System.out.println("just another
event handler")));
        e1.bindTo(e2);
        e2.apply(); // output:
                                //func was invoked by E2
                                //func was invoked by E1
                                //just another event handler

    }
}
```

הסבר:

1. הגדרנו פונקציה סטטית בשם func שבהינתן משתנה מסוג event ואובייקט נוסף (אודות=about) היא מדפיסה את שניהם.
2. ב main יצרנו שני משתנים מסוג Event עם השמות E1, E2.
3. ל e1 הוספנו delegate (פונקציה כביטוי למדה) שמפעילה את func
4. הפעלת המתודה apply ללא פרמטרים תגרום להפעלת כל הפונקציות שנוספו כאשר about==null – כרגע זו רק ההפעלה של func, ולכן יופיע הפלט שבהערה
5. הפעלה של apply עם פרמטר תגרום להפעלת כל הפונקציות שנוספו כאשר הפרמטר about == – כרגע זה עדיין רק ההפעלה של func ולכן יופיעו שתי השורות שבהערה כפלט

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

6. לאחר מכן הוספנו את func כ delegate של e2. לכן בהפעלה של apply ניתן לראות שהפעם ה invoker הוא E2.

7. הוספנו ל e1 עוד delegate שפשוט מדפיסה טקסט נוסף

8. הפעולה e1.bindTo(e2) כרכה את e1 ל e2. כלומר, כאשר נפעיל את e2 נגרום גם להפעלה של כל מי שכרוך אליו – שזה e1 במקרה זה.

9. לכן הפעלה של e2 תגרום לפלט שמופיע בהערות:

func was invoked by E2 – E2 זו הפונקציה הראשונה שנרשמה ל E2

ובגלל הכריכה כעת יופעל גם e1 ולכן נקבל את שאר הפלטים של הפונקציות שנרשמו אליו:

func was invoked by E1

just another event handler

שימו לב!

- עליכם להשלים את הקוד החסר במחלקה Event בלבד
- בפרט, שימו לב לממשק הפנימי והפומבי Delegate שעליכם להגדיר
- למחלקה Event אסור לרשת מחלקה כלשהי או לממש ממשק כלשהו. בפרט, אסור לה לממש את Observable / Observer. זה חלק מהאתגר. כתבו קוד קצר ואלגנטי.

אופי הבדיקה דומה ל main לעיל, וכולל בדיקה נוספת ש Even לא ירשה \ מימשה שום דבר.

עליכם להגיש את הקבצים American.java ו Event.java בלבד למערכת הבדיקות ע"פ הוראות ההגשה שניתנו לכם במסמך בתוך קובץ ה ZIP שהורדתם ממערכת הבדיקות.

בהצלחה!