



התוכנית להנדסת תוכנה

תאריך הבחינה: 22.01.2023
כ"ט טבת, תשפ"ג
שם המרצה: ד"ר אחיה אליסף
שם הקורס: הנדסת איכות תוכנה
מספר הקורס: 372-1-3501
סמסטר: א'
מועד: א'
משך הבחינה: שעתיים
חומר עזר: אסור

מספר נבחן: _____

הנחיות:

- יש לענות על השאלות רק בתוך שאלון הבחינה ורק בתוך המקומות המיועדים. תשובות מחוץ לשאלון או למקומות המיועדים — לא יבדקו.
- אלא אם צוין אחרת במפורש, בכל שאלה יש לבחור תשובה אחת בלבד.
- אם לפי דעתכם יש מספר תשובות נכונות, בחרו בתשובה הנכונה ביותר.

בהצלחה!

שאלה 1 (10 נקודות) — function point

BGUAPP היא מערכת לניהול מידע הסטודנטים של אוניברסיטת בן גוריון. בין היתר המערכת מאפשרת הצגה נוחה לצורך מעקב על מידע על למבחנים בקורסים הנלמדים, ובנוסף ניתן להרחיב את המידע המוצג על כל אחד מהמבחנים בעזרת לחיצה על הכפתור שמשמאלו (ראו תמונה לדוגמה).

זהו את התהליכים הקיימים בפיצ'ר זה של המערכת, וחשבו עבורו ועבור מסד הנתונים הנתון נקודות פונקציונליות ופרטו את חישובכם.

(שימו לב – יש להתייחס למסך המוצג בלבד)

הנחות:

- יש להראות את הדרך לחישוב.
- ניתן להעזר בטבלאות המופיעות בנספח 1.
- יש לחשב על פי מסך זה בלבד, אין להשתמש בידע שנובע מההיכרות שלכם עם המערכת ועם המסכים האחרים שיש.
- ניתן להניח שהטבלאות הבאות קיימות:

Users: id, first name, last name, email

Courses: id, name

StudentsRegistration: courseId, studentId

Exam: courseId, date, term, room, building

18:45 58%

מבחנים >

שם

מבוא לבינה מלאכותית
תאריך: 20/01/2023 | 09:00
מועד: בוחן-2
חדר: טרם נקבע מקום
בניין: טרם נקבע מקום

מקורות מיתולוגיים לתולדות האמנות
תאריך: 23/01/2023 | 13:30
מועד: א'
חדר: טרם נקבע מקום
בניין: טרם נקבע מקום

דטא סיינס בתעשייה
תאריך: 24/01/2023 | 13:30
מועד: א'
חדר: טרם נקבע מקום
בניין: טרם נקבע מקום

מבוא לאימות תוכנה בשיטות פורמליות
תאריך: 01/02/2023 | 09:00
מועד: א'
חדר: טרם נקבע מקום
בניין: טרם נקבע מקום

- זהו תהליך EO (ניתן להיות לפי ה-| בתאריך ולפי ה"טרם נקבע מקום" בבניין ובחדר):
- FTR 3 – כל הטבלאות חוץ מ Users (את userId יש לנו כי הוא עשה login. לכן לא חייבים את טבלת users. כן חייבים את טבלת registration בשביל לחשב ולהציג את הנתונים)
 - DET 5 – שם קורס, תאריך, מועד, חדר, בניין
 - Function points = 4

בסיס הנתונים פנימי ILF:

- יש 4 טבלאות RET
- DET 8 – לפי מה שביקשנו צריך לחשב לפי מה שרואים ואז לא מחשיבים את המזהה של הקורס וגם לא את הפרטים של הסטודנט. כן מחשבים שדות זרים ולכן: course.name, courseId, studentId, courseId, date, term, room, building
- function points = 7

בנוסף, יש כפתור שמוביל אותנו למידע נוסף ולכן זהו כפתור input EI. אומנם אנחנו לא צריכים להתייחס למה שאנחנו לא רואים ולכן לא צריך לחשב את המסך הבא, אבל כן כתוב במפורש שלחיצה על הכפתור מובילה אותנו למידע מורחב, כלומר זה מסך שמצריך שליפה נוספת וצריך להתייחס לכפתור:

- FTR 0
- DET 1 (סופרים את החץ פעם אחת ולא 4 כי זו פונקציונליות אחת)
- Function point = 3

נשים לב שלחיצה לא יכולה להספר כשדה DET. לא מתחשבים בכפתור חזרה למסך קודם.

לכן בסה"כ – $total\ function\ points = 7 + 4 + 3 = 16$

EIF		DET		
RET	I-19	20-50	51+	
I	Low (5)	Low (5)	Average(7)	
2-5	Low (5)	Average(7)	High (10)	
6+	Average(7)	High (10)	High (10)	

ILF		DET		
RET	I-19	20-50	51+	
I	Low (7)	Low (7)	Average(10)	
2-5	Low (7)	Average(10)	High (15)	
6+	Average(10)	High (15)	High (15)	

EQ		DET		
FTR	I-5	6-19	20+	
0-1	Low (3)	Low (3)	Average(4)	
2-3	Low (3)	Average(4)	High (6)	
4+	Average(4)	High (6)	High (6)	

EO		DET		
FTR	I-5	6-19	20+	
0-1	Low (4)	Low (4)	Average(5)	
2-3	Low (4)	Average(5)	High (7)	
4+	Average(5)	High (7)	High (7)	

EI		DET		
FTR	I-4	5-15	16+	
0-1	Low (3)	Low (3)	Average(4)	
2	Low (3)	Average(4)	High (6)	
3+	Average(4)	High (6)	High (6)	

שאלה 2 (5 נקודות) — CMMI

מה המטרה של (CMMI) Capability Maturity Model Integration

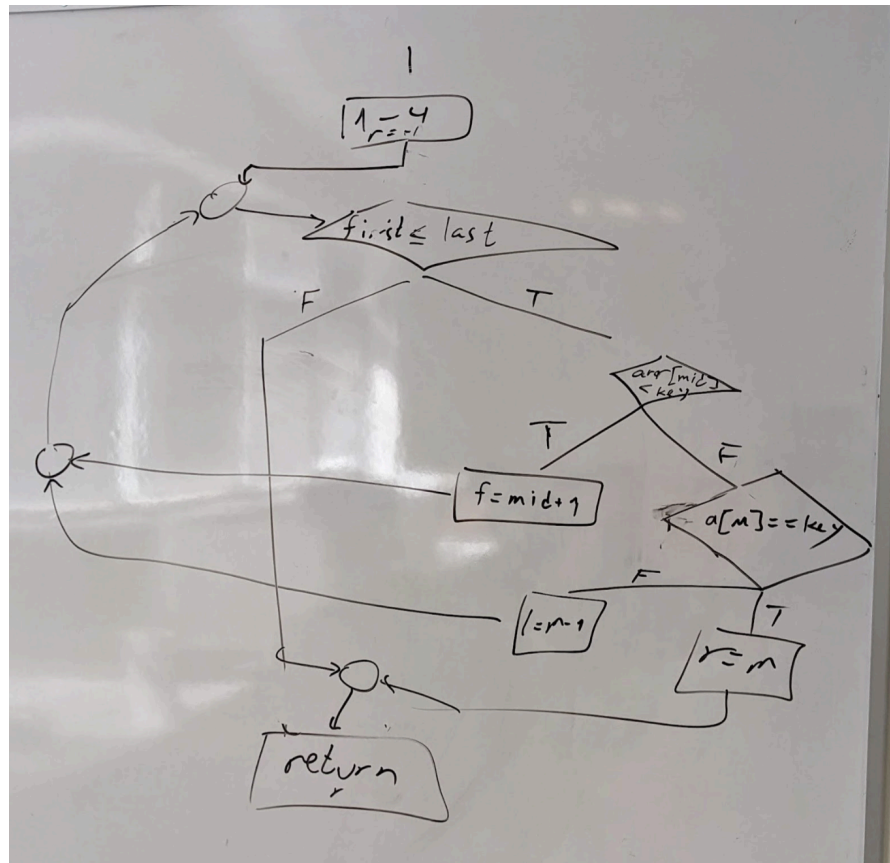
CMMI זה מדד לבשלות חברה מבחינת תהליכי העבודה שלה. המדד כולל מספר רמות של בשלות כגון רמת הארגון של תהליכים, איך הם מוגדרים ומנוהלים, האם יש תהליך שיפור פנימי וכדו'. CMMI יכול לשמש חברות באופן פנימי בשביל להבין איך לשפר את הארגון. בנוסף, ניתן לקבל גם דרגה מגוף רשמי חיצוני.

שאלה 3 (15 נקודות) — Control-flow testing

לפניכם מתודה המבצעת חיפוש בינארי במערך (ניתן להניח כי המערך arr ממורכז):

```
1. public int binarySearch(int[] arr, int key) {
2.     int first = 0;
3.     int last = arr.length - 1;
4.     int mid = (first + last) / 2;
5.     while (first <= last) {
6.         if (arr[mid] < key) {
7.             first = mid + 1;
8.         } else if (arr[mid] == key) {
9.             return mid;
10.        } else {
11.            last = mid - 1;
12.        }
13.        mid = (first + last) / 2;
14.    }
15.    return -1;
16. }
```

סעיף א (7 נקודות) — עליכם לצייר את ה control-flow graph עבור מתודה זו.



סעיף ב (8 נקודות)

לכל אחד מסוגי הכיסויים הבאים, תנו דוגמה לסט קלטים מינימלי שנותן כיסוי מלא:
(התקבלו תשובות של קלטים ושל קודקודים)

Branch coverage

--

שאלה 4 (20 נקודות) — Symbolic execution

להלן קוד:

```

1. public static boolean dividesInN(int number, int n) {
2.     return number % n == 0;
3. }
4.
5. public static int stupidFunction(int x, int y, int z) {
6.     int res = 0;
7.     if (dividesInN(x, 7)) {
8.         if (dividesInN(y, 3)) {
9.             res = x * y;
10.        } else {
11.            res = x + y;
12.        }
13.    }
14.    if (res == z) {
15.        // ERROR
16.    }
17.    return res;
18. }
19.
20. public static void main(String[] args) {
21.     int x = read("Enter num1: ");
22.     int y = read("Enter num2: ");
23.     int z = read("Enter num3: ");
24.     stupidFunction(x, y, z);
25. }

```

בסעיפים הבאים עליכם למצוא את התנאי על הקלט (PCT) שגורם לתוכנית להגיע לשגיאה באמצעות הרצה סימבולית (symbolic execution).

סעיף א (15 נקודות) — יש להראות את הדרך:

At the end of row	Symbolic Store σ_s	Path constraint (PCT)
23	$x \rightarrow x_0 ; y \rightarrow y_0 ; z \rightarrow z_0$	true
6	אפשר לכתוב את כל הערכים או	אפשר לכתוב true או [23], שזה אומר אותם תנאים של

	סוף שורה 23.	$res \rightarrow 0$; שזה אומר אותו σ_s כמו בסוף 23 + השמה ל res
7	[6]	$x_0 \% 7 = 0$
8	[6]	$[7] \wedge y_0 \% 3 = 0$
9	$[23]; res \rightarrow x_0 y_0$	[8]
11	$[23]; res \rightarrow x_0 + y_0$	$[7] \wedge y_0 \% 3 \neq 0$
13a	[6]	$x_0 \% 7 \neq 0$
13b	[9]	[8]
13c	[11]	[11]
14a	[6]	$[13a] \wedge z_0 = 0$
14b	[9]	$[8] \wedge z_0 = x_0 y_0$
14c	[11]	$[11] \wedge z_0 = x_0 + y_0$

הערות לפתרון: ניתן להשמיט את סוף שורה 8 כי מה שמעניין זה סוף שורה 9 (כפי שנעשה ב else עם שורות 10-11). כמו כן - כל מה שקורה אחרי ה ERROR כבר לא מעניין מבחינת הניתוח. הניתוח צריך לכלול את כל האפשרויות להגיע ל ERROR.

סעיף ב (5 נקודות) — יש לכתוב את התנאי (PCT) הסופי ולתת דוגמה לקלט שמוביל לשגיאה:

Final PCT that leads to an error (i.e., to the end of line 14):

$[14a] \vee [14b] \vee [14c]$, where:

$$[14a] = (x_0 \% 7 \neq 0 \wedge z_0 = 0)$$

$$[14b] = (x_0 \% 7 = 0 \wedge y_0 \% 3 = 0 \wedge z_0 = x_0 y_0)$$

$$[14c] = (x_0 \% 7 = 0 \wedge y_0 \% 3 \neq 0 \wedge z_0 = x_0 + y_0)$$

Example for input that leads to this error:

We will give an example for an input where $[14a] = \text{true}$:

$$[14a] = \text{true} \Rightarrow x_0 \% 7 \neq 0 \wedge z_0 = 0$$

The symbolic store of 14a is: $x \rightarrow x_0$; $y \rightarrow y_0$; $z \rightarrow z_0$; $res \rightarrow 0$

Therefore a possible answer is: $x = 6$, $y = 1$, $z = 0$

(Note that the answer is on x,y,x and not on the symbolic variables)

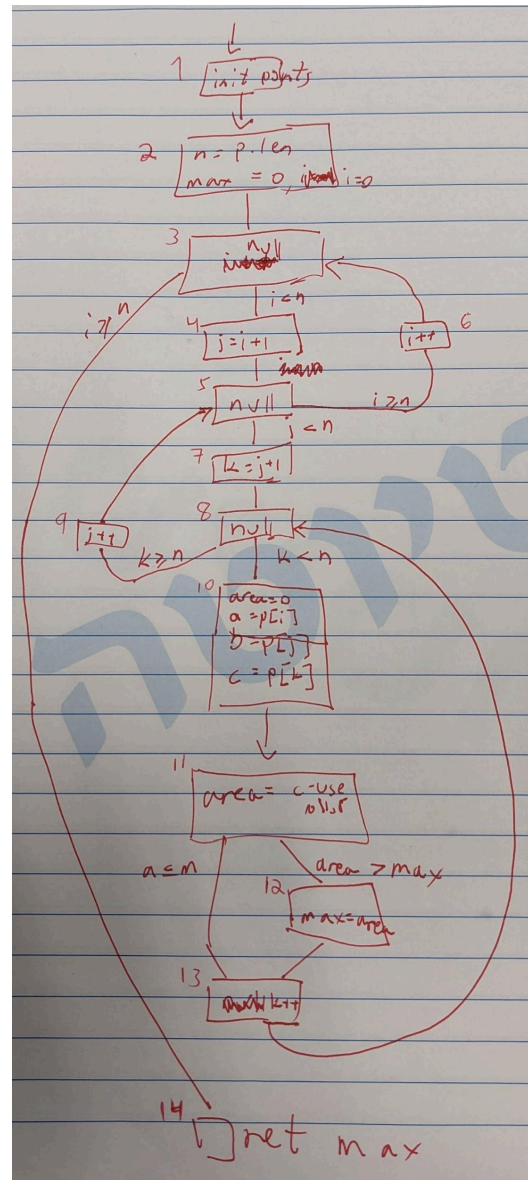
שימו לב שנדרשתם לכתוב גם את התנאי וגם קלט אפשרי

שאלה 5 (15 נקודות) — Data flow testing

נתונה הפונקציה largestTriangleArea המקבלת מערך של נקודות (נקודה מיוצגת ע"י מערך של שתי נקודות x,y). הפונקציה מחזירה את שטח המשולש הגדול ביותר שניתן ליצור בעזרת 3 נקודות כלשהן מהמערך.

```
1. public double largestTriangleArea(int[][] points) {
2.     int n = points.length;
3.     double max = 0;
4.
5.     for (int i = 0; i < n; i++) {
6.         for (int j = i + 1; j < n; j++) {
7.             for (int k = j + 1; k < n; k++) {
8.                 double area = 0;
9.                 int[] a = points[i];
10.                int[] b = points[j];
11.                int[] c = points[k];
12.                area = Math.abs(area(a, b) + area(b, c) + area(c, a));
13.                if (area > max)
14.                    max = area;
15.            }
16.        }
17.    }
18.    return max;
19. }
```

סעיף א (10 נקודות) — עליכם לצייר את ה data-flow-graph עבור מתודה זו.



סעיף ב (5 נקודות)

באילו קודקודים המשתנה `area` הוא global definition? התשובה היא 11.
 נשים לב שהמשתנה מוגדר בקודקודים 10 ו-11. אבל להגדרה בקודקוד 10 אין `use` ולכן היא לא `g-def`

כתבו סט מינימלי של מסלולים שלמים עבור המשתנה `area` והקריטריון `All c-use/Some p-use`.

יש לסמן בתוך המסלולים את ה `definition-clear path` שחייבים לעבור בהם.

להלן קוד:

```
public class WeatherService {
    private final WebService webService;

    public WeatherService (WebService service) {
        this.webService = service;
    }

    public double getTemperature(boolean celsius) {
        String temperatureString = webService.getTemperature();
        double temperature = Double.parseDouble(temperatureString);
        if(!celsius) {
            temperature = celsiusToFahrenheit(temperature);
        }
        return temperature;
    }

    private double celsiusToFahrenheit(double temperature) {
        return ((9.0/5) * temperature + 32);
    }
}
```

כתבו למחלקה WeatherService בדיקות יחידה. אין צורך לזכור תחביר מדויק:

```
public class WeatherServiceTest {
    private WeatherService weatherService;
    private WebService webServiceMock;

    @BeforeEach
    public void setUp() {
        webServiceMock = Mockito.mock(WebService.class);
        this.weatherService = new WeatherService(webServiceMock);
    }

    @ParameterizedTest
    @CsvSource({"25.0", "0.0", "-10.0", "10.0", "5.3", "150.0"})
    public void getTemperature_celsius_success(double temp) {
        when(webServiceMock.getTemperature()).thenReturn(temp + "");
        double weather = weatherService.getTemperature(true);
        assertEquals(temp, weather, "Temperature should be " + temp);
    }

    @ParameterizedTest
    @CsvSource({"25.0, 77", "0.0, 32.0", "-10.0, 14.0", "10.0, 50.0",
        "5.3, 41.54", "150.0, 302.0"})
    public void getTemperature_fahrenheit_success(double celsius,
        double fahrenheit) {
        when(webServiceMock.getTemperature()).thenReturn(celsius + "");
        double weather = weatherService.getTemperature(false);
        assertEquals(fahrenheit, weather,
            "Temperature should be " + fahrenheit);
    }
}
```

הערות:

1. כל קלט הוא חוקי ולכן אין בדיקות rainy-day. אין סיבה לבדוק מה קורה אם webService מחזיר מחרוזת שהיא לא מספר כי בבדיקות יחידה אנחנו מניחים נכונות של מחלקות אחרות.
2. אין צורך לבדוק את הפונ' private משתי סיבות: (א) הפונ' הזו היא בסה"כ הגדרה; (ב) היא פרט מימוש ובאותה מידה בגרסה אחרת של המערכת אפשר יהיה להכניס אותה לתוך הפונ' getTemperature.
3. יש לכתוב מספר בדיקות של תקינות ולא אחד (בפתרון זה מופיע כמספר ערכים csv). הסיבה היא שאולי הקוד מתנהג בצורה לא תקינה (כרגע/בגרסה עתידית). גם אם טמפרטורה אחת מכסה את כל מסלולים בקוד — מי אמר שהקוד נכון? שהוא עושה את מה שהוא אמור לעשות?
4. אמנם לא צריך לזכור תחביר מדויק אבל יש ציפייה שיופיעו הרעיונות של: beforeAll, שימוש נכון ב when then (להבין למי עושים את זה), של parameterized test. כמו-כן יש להשתמש בקונבציה של שמות הטסטים (כפי שמופיע בפתרון).
5. טעות נפוצה — עטיפה של weatherService עם spy. זו טעות כיוון שאין בזה צורך. הוא לא מאפשר

לנו בדיקה נוספת מעבר למה שצריך.

6. טעות נפוצה - בדיקה של `parseDouble` או `mock של Double`. אין שום סיבה לבדוק את זה - אנחנו מניחים שמחלקות אחרות ממומשות נכון.

שאלה 7 (10 נקודות) — integration testing

ציינו שתי שיטות לביצוע בדיקות אינטגרציה ומה היתרונות של כל שיטה?

יש 4 שיטות לאינטגרציה: Bottom-up, Top-down, Sandwich, Big-bang
היתרונות והחסרונות של השיטות מופיעים במצגת מספר 7 של בדיקות אינטגרציה.

שאלה 8 (10 נקודות) — model-based testing

מה מתאר המודל ב model-based testing, ואיך הוא משמש את הבודק?

המודל מתאר מה אפשר ואי אפשר לעשות במערכת. בנוסף המודל מתאר מה נרצה לבדוק בכל מצב במערכת (במילים אחרות — במידה ואי אפשר לעשות משהו במערכת — המודל יכיל וידוא שאכן אי אפשר לעשות את זה).

שימושים של המודל:

1. יצירת test suite
2. הערכה של כיסוי מסלולים