

האוניברסיטה הפתוחה

20554

**תכנות מתקדם**

**בשפת Java**

חוברת הקורס – סתיו 2023א

כתבה: תמר בניה

אוקטובר 2022 – סמסטר סתיו – תשפ"ג

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנט
ב	1. לוח זמנים ופעילויות
ד	2. תיאור המטלות
ד	2.1 מידע כללי
ד	2.2 מבנה המטלות ואופן הגשתן
ה	2.3 משקל המטלות ואפיוניהן
ן	3. התנאים לקבלת נקודות זכות
1	ממ"ן 11
5	ממ"ן 12
9	ממ"ן 13
11	ממ"ן 14
15	ממ"ן 15
19	ממ"ן 16



## אל הסטודנט,

אנו מקדמים את פניכם בברכה עם הצטרפותכם אל הלומדים בקורס "תכנות מתקדם בשפת Java".

בחוברת זו תמצאו את התנאים לקבלת נקודות זכות בקורס, לוח זמנים ופעילויות ומטלות.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:

<http://telem.openu.ac.il>

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט [www.openu.ac.il/Library](http://www.openu.ac.il/Library).

אפשר לפנות אלי בדואר אלקטרוני [tamar@openu.ac.il](mailto:tamar@openu.ac.il). במידת הצורך אפשר לתאם פגישה. לצורך בירורים בנושאים אדמיניסטרטיביים יש לפנות בכתב או טלפונית למחלקת האוניברסיטה הפתוחה.

**- שאילתא -** לפניות בנושאים אקדמיים שונים כגון מועדי בחינה מעבר לטווח זכאות ועוד, אנא עשו שימוש מסודר במערכת הפניות דרך שאילתא. לחצו על הכפתור פניה חדשה ואחר כך לימודים אקדמיים > משימות אקדמיות, ובשדה פניות סטודנטים: השלמת בחינות בקורס. המערכת תומכת גם בבקשות מנהלה שונות ומגוונות.

**לתשומת לב הסטודנטים הלומדים בחו"ל:**

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.

הפרטים החיוניים על הקורס נכללים בחוברת הקורס וכן באתר הקורס. מומלץ מאד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

בברכת לימוד מהנה,

תמר בניה  
מרכזת ההוראה בקורס



1. לוח זמנים ופעילויות (20554/א/2023)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	28.10.2022-23.10.2022	פרק 1-3	מפגש 1	
2	04.11.2022-30.10.2022	פרק 4-6		
3	11.11.2022-6.11.2022	פרק 7-8	מפגש 2	ממ"ן 11 11.11.2022
4	18.11.2022-13.11.2022	פרק 9-10		
5	25.11.2022-20.11.2022	פרק 10-11	מפגש 3	ממ"ן 12 25.11.2022
6	02.12.2022-27.11.2022	פרק 12		
7	09.12.2022-04.12.2022	פרק 12-13	מפגש 4	
8	16.12.2022-11.12.2022	פרק 14-15		ממ"ן 13 16.12.2022

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	23.12.2022-18.12.2022 (ב-ו חנוכה)	פרק 16	מפגש 5	
10	30.12.2022-25.12.2022 (א-ב חנוכה)	פרק 20-21	מפגש 6	ממ"ן 14 30.12.2022
11	06.01.2023-01.01.2023	פרק 22-23		
12	13.01.2023-08.01.2023	פרק 23	מפגש 7	ממ"ן 15 13.1.2023
13	20.01.2023-15.01.2023	פרק 23,28		
14	27.01.2023-22.01.2023	פרק 28	מפגש 8	ממ"ן 16 27.1.2023

מועדי בחינות הגמר יפורסמו בנפרד

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".



## 2. תיאור המטלות

קראו היטב עמודים אלו לפני שתתחילו לענות על השאלות

### 2.1 מידע כללי

על מנת לתרגל את החומר הנלמד ולבדוק את מידת הבנתכם, עליכם לפתור את המטלות המצורפות. פתרון המטלות הוא חלק בלתי נפרד מלימוד הקורס - הבנה מעמיקה של חומר הלימוד דורשת תרגול רב. המטלות ייבדקו על-ידי המנחה ויוחזרו אליכם בצירוף הערות המתייחסות לתשובות.

#### לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי.

זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.

**זכרו!** ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

### 2.2 מבנה המטלות ואופן הגשתן

כל מטלה מורכבת מכמה שאלות. בראש כל שאלה מצוין משקלה היחסי בקביעת ציון המטלה. את הפתרונות למטלה עליכם להריץ במחשב. יש להגיש את המטלות דרך מערכת המטלות האלקטרונית באופן הבא:

מטלה תהייה מורכבת מקובץ אחד המכוון בפורמט ZIP. הקובץ יכיל ספריות כך שכל פתרון לשאלה יהיה בספרייה נפרדת. מלבד הספריות האלה אין ליצור ספריות נוספות. כל ספרייה תכיל: קבצי מקור (עם סיומת java), קבצי הרצה (עם סיומת class), קובץ הרצה (run.bat) כפי שיתואר בהמשך.

לחיצה כפולה על קובץ ההרצה תפעיל את התוכנית, יש לוודא את תקינות קובץ ההרצה לפני ההגשה. אין לצרף קבצים מיותרים (קבצים המתווספים על ידי סביבות עבודה).

קובץ ההרצה (run.bat) יכיל את השורות :

```
java ClassFileName  
pause
```

אין צורך לכלול בקובץ ההרצה את פקודת ה**הידור** `javac SourceFileName`

אם השאלה בממ"ן אינה ברורה לכם, אל תהססו להתקשר אל אחד המנחים (בשעות הייעוץ הטלפוני) לצורך קבלת הסבר.

### 2.3 משקל המטלות ואפיוניהן

מספר מטלה	הפרק שאליו מתייחסת המטלה	משקל המטלה
11	עד פרק 7 כולל	4
12	עד פרק 11 כולל	4
13	עד פרק 15 כולל	4
14	עד פרק 22 כולל	4
15	עד פרק 23 כולל	4
16	עד פרק 28 כולל	4

### **לתשומת לבכם:**

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס. סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע בטלפון **09-7782222** או **יעדכנו בעצמם** באתר שאילתא <http://www.openu.ac.il/sheilta>

**קורסים ← ציוני מטלות ובחינות ← הזנת ציון 0 למטלות רשות שלא הוגשו.** יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהממוצע המשוקלל של המטלות והבחינה יהיה נמוך מ- 60).

**כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.**

### **3. התנאים לקבלת נקודות זכות**

כדי לקבל נקודות זכות בקורס זה עליכם לעמוד בדרישות הבאות:

- א. הגשת 4 מטלות לפחות.
- ב. ציון של לפחות 60 נקודות בבחינת הגמר.
- ג. ציון סופי בקורס של 60 נקודות לפחות.

### **שימו לב:**

פרקים נבחרים מספר הקורס משמשים גם כחומר לימוד בסדנה בתכנות מתקדם בשפת Java 20503. אם הנכם מתכוונים בעתיד ללמוד את הסדנה **אנא שימרו על הספר כך שיוכל לשמש אתכם גם בסדנה.**

# מטלת מנחה (ממ"ן) 11

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 7 (כולל)

משקל המטלה: 4 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 11.11.2022

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 (50 נקודות)

כתבו יישום המממש את המשחק "בול פגיעה" באופן הבא: התכנית תבחר מספר בן ארבע ספרות (בלי חזרות) והמשתמש ינסה לנחש את המספר הנבחר. עבור כל ניחוש התכנית תציג את המספר יחד עם המידע הבא: מספר הפגיעות המדויקות (בול) ומספר הספרות שמופיעות במספר המקורי אך במיקום אחר (פגיעה).

דוגמה: בהינתן המספר 1307, והניחוש 3201 התוצאה תהייה: בול אחד ושתי פגיעות.

הסבר: הבול הוא עבור הספרה 0 שנמצאת בדיוק במקום המתאים, והפגיעות הן עבור הספרות 3 ו-1 שמופיעות במספר אך לא באותו המקום.

המשתמש ימשיך לנחש עד שיצליח לגלות את המספר. לבסוף יש להציג את מספר הניחושים להן הזדקק המשתמש עד שהצליח לנחש. הקלט והפלט יתבצע באמצעות תיבת דו-שיח, בכל פעם יש להציג את כל הניחושים הקודמים כולל התוצאות שהתקבלו עבור כל ניחוש.

לאחר סיום המשחק, יש לשאול את המשתמש (באמצעות תיבת דו-שיח) אם הוא מעוניין לשחק משחק חדש.

## הדרכה:

- כתבו מחלקה המממשת את לוגיקת המשחק, המחלקה תכיל משתנה מופע המייצג את המספר שצריך לנחש ומתודה המקבלת מחרוזת המייצגת את הניחוש ומחזירה תשובה למשתמש.

- התכנית הראשית תנהל את המשחק, כלומר תקלוט ניחושים מהמשתמש, תשתמש במחלקת המשחק ותציג הודעות למשתמש.

- במקרה של קלט שאינו מורכב מספרות בלבד או קלט באורך שגוי או קלט הכולל ספרות החוזרות על עצמן יש להציג הודעה מתאימה.

- עליכם לחפש במחלקה String מתודות המתאימות לפירוק מחרוזת הקלט.

**שאלה 2 (50 נקודות)**

משחק החיים הינו משחק סימולציה שפותח על ידי המתמטיקאי John Conway כמודל חקר מחזור החיים של האורגניזם החי. המשחק משוחק על מטרצת ענק שאיבריה מהווים אתרי מחייה (קיום) אפשריים : בכל אתר יכול להיות אחד משני המצבים הבאים :

א. "יש חיים" - אתר מחייה מלא - נסמן כריבוע מלא

ב."אין חיים" - אתר מחייה ריק - נסמן כריבוע ריק

לדוגמא, בהינתן המטריצה הבאה:

אפשר לראות שבאתר  $(1,1)$  אין חיים, ושלאחר זה

יש 4 שכנים חיים והם:  $(0,0)$ ,  $(0,1)$ ,  $(1,2)$  ו-  $(2,1)$ .

לעומת זאת באתר  $(0,3)$  יש חיים, ולאתר זה יש שכן

חי אחד שהוא  $(1,2)$ .

## חוקי הגנטיקה הבסיסיים של CONWAY:

- **לידה** - בכל אתר בו "אין חיים" שלו בדיוק 3 שכנים חיים, תהיה לידה בדור הבא. אחרת האתר נשאר "ללא חיים" - ריק.
- **מוות** - בכל אתר בו "יש חיים" שלו 0 או 1 שכנים חיים יתרחש מוות בדור הבא כתוצאה מבדידות. בכל אתר בו "יש חיים" ולו 4 שכנים חיים ומעלה, יתרחש מוות בדור הבא כתוצאה מ"פיצוץ אוכלוסין".
- **קיום** - כל אתר בו "יש חיים" והינו בעל 2 או 3 שכנים חיים, ימשיך להתקיים גם בדור הבא. תהליכי הלידה, המוות והקיום מתרחשים בו זמנית בכל האתרים ויוצרים מצב חיים חדש הנקרא דור חדש.

כתבו יישום **JavaFX** המבצע סימולציה למשחק החיים על מטריצה בגודל  $10 \times 10$  באופן הבא:

- תחילה התכנית תיצור מטריצה עם ערכים אקראיים המייצגים את מצבי החיים ההתחלתיים.
- התכנית תציג את מטריצת החיים בחלון שעליו יש לצייר את האתרים כריבועים ריקים ומלאים לפי מצבי החיים. בצעו את הציור על רכיב מסוג Canvas הממוקם בתוך חלון מסוג Application.
- הציגו כפתור בפינה השמאלית העליונה של משטח הציור בדומה לדוגמאות שמופיעות בסעיפי ממשק המשתמש בסוף פרקים 4-7 בספר הלימוד. בכל לחיצה על הכפתור, התכנית תחשב את מצבי החיים של הדור הבא ותעדכן את התצוגה.
- שימו לב שבכל פעם שמעדכנים את מטריצת החיים, יש להציג מחדש את המטריצה כולה.

## הדרכה:

- הגדירו מחלקה המייצגת את מטריצת החיים. המחלקה תכלול בנאי המקבל את מימדי המטריצה ופעולות מתאימות לעדכון ואחזור המידע.
- אין לשנות את מצבו של תא מסוים לפני שהוא נבדק לצורך קביעת מצב שכניו.
- קווי המטריצה צריכים להתפרס על פני כל משטח הציור בהתאם לגודלו.

- את הציור יש לבצע מתוך מחלקת Controller הכוללת מתודה המגיבה ללחיצת הכפתור.
- מילוי ריבועים מתבצע באמצעות המתודה fillRect של GraphicsContext.
- הריצו את התכנית ולחצו על הכפתור כמה פעמים, עליכם לוודא שהמטריצה מתעדכנת בהתאם לכללים.



# מטלת מנחה (ממ"ן) 12

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 11 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 25.11.2022

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 (50 נקודות)

ברצוננו לייצג ביטויים חשבוניים ב-Java. הביטויים החשבוניים יכילו רק את הפעולות חיבור וחיסור. להלן דוגמאות לביטויים חשבוניים כאלה –

2

$5 + 6$

$4 - 3 + 2$

ביטוי חשבוני יכול להיות אחד משני הדברים הבאים:

ביטוי אטומי – מספר ממשי (למשל, 2, 6, 10)

ביטוי מורכב – ביטוי שמורכב משני אופרנדים שהם ביטויים חשבוניים, ואופרטור חיבור או חיסור. למשל  $3+5$  או  $5-7+12$ .

א. כתבו מחלקה מופשטת בשם Expression. מחלקה זו תייצג ביטוי חשבוני. הוסיפו למחלקה מתודה מופשטת בשם:

```
public double calculate();
```

מתודה זו תחזיר את תוצאת הביטוי החשבוני. שימו לב, שתוצאת ביטוי אטומי היא פשוט ערך המספר.

דרסו (override) במחלקה את המתודה equals שנורשת מ-Object. נאמר ששני ביטויים חשבוניים הם שווים אם תוצאת החישוב שלהם היא שווה. למשל הביטויים 8 ו-4-12 הם שווים.

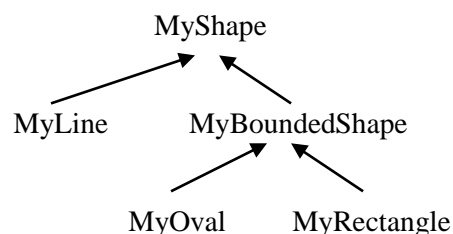
ב. כתבו מחלקה בשם AtomicExpression היורשת מ-Expression. מחלקה זו תייצג ביטוי חשבוני אטומי. כתבו במחלקה בנאי המקבל מספר ממשי ומאתחל באמצעותו את הביטוי. הוסיפו למחלקה את המתודה toString המחזירה את תוצאת הביטוי בצורת מחרוזת.



- ג. כתבו מחלקה מופשטת בשם `CompoundExpression` היורשת מ-`Expression`. מחלקה זו מייצגת ביטוי חשבוני מורכב. לביטוי מורכב ישנם שני אופרנדים שהם ביטויים חשבוניים. כתבו במחלקה בנאי המקבל שני פרמטרים מסוג `Expression` ומאתחל את התכונות.
- ד. כתבו שתי מחלקות היורשות מ-`CompoundExpression`: `AdditionExpression` ו-`SubtractionExpression`. המחלקות ייצגו ביטוי חיבור וביטוי חיסור בהתאמה. כתבו במחלקות בנאים המקבלים שני פרמטרים מסוג `Expression` ומאתחלים איתם את התכונות.
- הוסיפו במחלקות את המתודה `toString` המחזירה את הביטוי בצורת מחרוזת. פורמט הביטוי יהיה הדפסת הביטוי השמאלי, אחריו רווח, אחריו סימן האופרטור (חיבור או חיסור), אחריו רווח ואחריו הביטוי הימני.
- ה. כתבו תכנית ראשית (`main`) המדגימה את השימוש במחלקות – צרו בתוכנית מערך מסוג `ArrayList` המכיל ביטויים מסוגים שונים. מלאו אותו בצורה רנדומלית בביטויים שונים, (אפשר להגביל את טווח האופרנדים כרצונכם) עברו על המערך והציגו את הביטויים השונים יחד עם התוצאות המתקבלות מחישובם.
- עבור כל ביטוי במערך, בדקו האם יש ביטויים נוספים השווים (`equals`) לו.

## שאלה 2 (50 נקודות)

- א. הגדירו היררכיה של צורות גאומטריות כפי שמתואר בתרשים 10.17 בספר הלימוד:



- בראש ההיררכיה נמצאת מחלקה מופשטת בשם `MyShape` המגדירה את המשותף לכל הצורות. המחלקה המופשטת `MyBoundedShape` מגדירה את המשותף לצורות בעלות שטח, והמחלקות `MyLine`, `MyOval` ו-`MyRectangle` מייצגות את הצורות: קו, אליפסה ומלבן.
- כל הצורות בהיררכיה מיוצגות על ידי 4 מספרים  $(x1, y1)$ ,  $(x2, y2)$  וצבע (מסוג `Color`) והן יודעות לצייר עצמן באמצעות מתודה המקבלת פרמטר מסוג `GraphicsContext`. במקרה של קו הערכים מייצגים את נקודת ההתחלה והסיום של הקו. במקרה של מלבן (או אליפסה), אפשר לשם הפשטות להתייחס ל- $(x1, y1)$  כמייצגים את הפינה השמאלית העליונה של המלבן (או המלבן החוסם), ול- $(x2, y2)$  כמייצגים את הרוחב והגובה של המלבן (או המלבן החוסם).

- הצורות אליפסה ומלבן מכילות תכונה נוספת המציינת אם הצורה מלאה או ריקה - לדוגמה, מלבן מלא:  ומלבן ריק:
  - כל מחלקה תכלול בנאי המקבל את נתוני הצורה ומאתחל אותה בהתאם.
  - הקפידו להגדיר את כל המשתנים כפרטיים, ספקו מתודות set ו-get לעדכון והחזרת הערכים.
  - ממשו את המתודה equals שנורשת מ-Object. במקרה של קווים, המתודה תחזיר true עבור קווים שהם באותו אורך, ללא קשר למיקומם. במקרה של אליפסה ומלבן יוחזר true כאשר הרוחב והגובה שלהם שווים, ללא קשר למיקומם.
  - הוסיפו למחלקה MyShape אפשרות לשכפול צורות באמצעות מימוש הממשק Cloneable והגדרה מחדש של המתודה clone של Object. בדקו ב-API כיצד מוגדרת המתודה clone.
- ב. כתבו תכנית ציור המייצרת ומציגה צורות באופן הבא:
- צרו 2 צורות מכל סוג באופן הבא: קבעו את ערכי  $(x1, y1)$  ו-  $(x2, y2)$  באופן רנדומלי (ערכים בין 0 ל-200). אתחלו את כל הצורות כצורות מלאות בצבע אדום. שמרו את הצורות במבנה נתונים **משותף** מסוג ArrayList.
  - שכפלו את כל הצורות באמצעות מתודת השכפול והכניסו אותן ל-ArrayList נוסף.
  - שנו את הצורות **המשוכפלות** באופן הבא: הזיזו את נקודת ההתחלה של כל אחת מהן 10 פיקסלים ימינה ולמטה. כלומר הגדילו את ערכו של  $x1$  ב-10 ואת ערכו של  $y1$  ב-10. שנו את צבע הצורות לירוק ושנו את תכונת ה"מילוי" שלהן מצורה מלאה לריקה.
  - הציגו את כל הצורות (המקוריות והמשוכפלות) על משטח ציור מסוג Canvas הממוקם בתוך חלון מסוג Application (בגודל  $400 \times 400$ ).



# מטלת מנחה (ממ"ן) 13

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 15 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 16.12.2022

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 (50 נקודות)

כתבו יישום המממש את המשחק "איש תלוי" (Hang-man) הפועל באופן הבא:

התכנית בוחרת מלה כלשהי מתוך מאגר מילים, והמשתמש מנסה לנחש את המלה. בכל תור מנחשים אות אחת, אם האות מופיעה במלה, יש להציבה במקומות המתאימים בתוך המלה ולהציגה. למשל המלה "polymorphism" לאחר שהמשתמש ניחש את האותיות "m", "ו", "o", תוצג באופן הבא:

\_ o \_ \_ m o \_ \_ \_ \_ m

אם הניחוש שגוי התכנית מוסיפה איבר אחד ל"איש התלוי". המשחק נמשך עד שהמשתמש מצליח לנחש את המלה או עד שהציור של ה"איש התלוי" מושלם. בנוסף יש להציג את אותיות הא"ב שנבחרו עד כה. המשחק מסתיים כאשר משלימים את האיש התלוי, או כאשר מצליחים לנחש את המלה. הממשק הגרפי של המשחק יהיה מעוצב לפי שיקול דעתכם, למשל אפשר להשתמש ברכיבים הבאים:

- תוויות - להצגת המילה והאותיות שכבר נבחרו.
  - משטח ציור מסוג Canvas - להצגת תרשים האיש התלוי. את האיש התלוי ניתן לצייר באמצעות גרפיקה פשוטה.
  - סדרה של כפתורים או כפתור בחירה (ComboBox) לקבלת האות מהמשתמש.
- התכנית תקרא את מאגר המילים מקובץ. עליכם לאפשר למשתמש להתחיל משחק חדש (מלה חדשה מתוך המאגר). עליכם לתכנן פתרון לפי עקרונות תכנות מונחה עצמים, כלומר עליכם להגדיר את האובייקטים העיקריים ולהרכיב מהם את התכנית.

## שאלה 2 (50 נקודות)

כתבו תכנית המממשת **משחק טריוויה** המאפשר למשתמש לקבל שאלות בנושא כלשהו לפי בחירתכם. לכל שאלה יש ארבע תשובות אפשריות כאשר רק תשובה אחת מתוכן נכונה. התכנית תציג למשתמש את השאלה, וארבעת התשובות האפשריות. המשתמש בוחר בתשובה מתוך האפשרויות המוצגות. אם הוא בחר בתשובה הנכונה הוא מקבל 10 נקודות, אחרת יורדות לו 5 נקודות. המשחק מסתיים כאשר השחקן מבקש לסיים או כאשר השאלות במאגר הסתיימו.

- התכנית תקרא את מאגר השאלות והתשובות מתוך קובץ. עליכם להכין קובץ טקסט כאשר כל שאלה תהיה מורכבת מחמש שורות: השורה הראשונה מייצגת את השאלה, השורה אחריה היא התשובה הנכונה ושלושת השורות הבאות הן תשובות לא נכונות. קריאה מתוך קובץ מתבצעת באמצעות המחלקות `java.util.Scanner` ו-`java.io.File` באופן הבא:

פתיחת הקובץ `trivia.txt` לקריאה:

```
Scanner input = new Scanner(new File("trivia.txt"));
```

קריאת מחרוזות:

```
while (input.hasNext()){  
    String st = input.next();  
}
```

סגירת הקובץ:

```
input.close();
```

שימו לב, כי פתיחת הקובץ לקריאה עלולה לעורר מצב חריג הדורש טיפול.

- התכנית תציג שאלה אקראית ותמתין לקבלת תשובה. המשתמש יקבל חיווי לגבי נכונות התשובה. לאחר מכן התכנית תמשיך עם הצגת השאלה הבאה.
- התכנית תקבע את הניקוד המצטבר ותציג אותו בסיום המשחק.
- יש להקפיד לא להציג את אותה השאלה יותר מפעם אחת במהלך משחק.
- תכננו את הממשק הגרפי בתוך חלון שבו תוצג בכל פעם שאלה אחת עם ארבעת התשובות האפשריות בסדר אקראי. בחרו ברכיב גרפי שיאפשר למשתמש לבחור תשובה אחת בלבד. הוסיפו כפתורים שיאפשרו למשתמש להתחיל משחק חדש ולסיים משחק.
- בסיום המשחק יוצג למשתמש הניקוד שצבר ויתאפשר לו לשחק משחק חדש.

הדרכה:

הגדירו מחלקות מתאימות לשאלה, למאגר שאלות, ולמשחק.

# מטלת מנחה (ממ"ן) 14

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 22 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 30.12.2022

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 (50 נקודות)

- א. כתבו מחלקה גנרית המממשת תור עם  $n$  עדיפויות כאשר  $n$  הוא מספר בתחום  $1..10$ . התור יכיל איברים מטיפוס  $T$  כלשהו כאשר כל איבר יהיה בעל עדיפות  $(1..n)$ . עדיפות 1 היא הגבוהה ביותר ועדיפות  $n$  היא הנמוכה ביותר. האיבר שנמצא בראש התור הוא האיבר בעל העדיפות הגבוהה ביותר שנמצא בתור. כלומר אם הוכנס לתור איבר בעל עדיפות 3 ואחריו הוכנסו לתור שני איברים בעלי עדיפות 1, אז האיבר בראש התור יהיה האיבר הראשון שהוכנס מבין שני האיברים בעלי העדיפות 1.
- שימו לב** שלכל איבר מוגדרת רמת עדיפות אך העדיפות אינה חלק מהאיבר עצמו אלא מידע נלווה.
- המחלקה תכלול את האלמנטים הבאים:
- **בנאי** המקבל את  $n$  שהוא מספר העדיפויות המקסימלי בתור (מספר בין  $1..10$ ). הבנאי יוצר תור ריק.
  - **add** - פעולה המקבלת איבר ועדיפות ומוסיפה אותו לתור. אם התקבלה עדיפות שאינה בתחום  $1..n$ , עדיפות האיבר תהיה  $n$ .
  - **poll** - פעולה המסירה ומחזירה את האיבר שנמצא בראש התור (האיבר בעל העדיפות הגבוהה ביותר). במקרה שהתור ריק (כלומר אין אף איבר בשום עדיפות) יש להחזיר `null`.
  - **contains** – פעולה בוליאנית המקבלת איבר ובודקת אם קיים בתור איבר בעדיפות כלשהי השווה לאיבר שהתקבל.
  - **remove** – פעולה בוליאנית המקבלת איבר ומסירה מהתור את האיבר הראשון (בעל העדיפות הגבוהה ביותר) השווה לאיבר המצויין. הפעולה מחזירה `true` אם איבר הוסר בהצלחה אחרת יוחזר `false`.

- size - פעולה המחזירה את סה"כ מספר האיברים בתור (בכל העדיפויות).
- iterator – פעולה המחזירה Iterator המאפשר מעבר על כל איברי התור בכל העדיפויות. האיברים יהיו מסודרים לפי סדר העדיפות.
- ממשו את התור בעצמכם, למשל באמצעות מערך באורך מספר העדיפויות, כאשר כל איבר במערך יהיה ArrayList המכיל את האיברים בעדיפות מסויימת.
- ב. בדקו את המחלקה הגנרית שכתבתם עבור איברים מסוג String. כתבו במחלקה נפרדת תכנית ראשית הבודקת את הפעולות השונות - הכניסו והוציאו כמה איברים בעדיפויות שונות והשתמשו באיטרטור כדי להציג את האיברים. לוו את הבדיקה בהדפסות מתאימות.
- ג. בדקו את המחלקה הגנרית עבור אובייקטים מסוג "פניות של לקוחות". הגדירו מחלקה המייצגת פנייה של לקוח. המחלקה תכלול שם ומספר ת.ז. של הלקוח ומחרוזת הכוללת את פרטי הפנייה. שתי פניות תחשבנה לשוות כאשר הן בעלות אותו מספר ת.ז. ובעלות תוכן פנייה זהה. כתבו במחלקה נפרדת תכנית ראשית המוסיפה כמה פניות של לקוחות בעדיפויות שונות, ובדקו את הפעולות poll, remove ו-contains ולוו את הבדיקה בהדפסות מתאימות.

## שאלה 2 (50 נקודות)

ברצוננו לכתוב אפליקציה של לוח שנה אלקטרוני המציג את חודשי השנה בפורמט סטנדרטי של טבלה המכילה משבצת עבור כל יום בחודש. כותרות הטבלה תכלול את השנה ואת החודש. הטבלה תכלול 7 עמודות המיועדות לימי השבוע ראשון עד שבת. אפשר לקבוע את מספר השורות בטבלה ל-6 כך שיתאים לכל האפשרויות של מספר הימים בחודש ושל היום בשבוע בו הוא מתחיל. למשל במקרה של חודש שיש בו 30 יום ושמתחיל ביום שבת, נזדקק ל-6 שורות. לוח השנה יאפשר למשתמש לבחור את השנה והחודש המבוקש לתצוגה. משתמש יוכל לבחור יום בחודש ולהזין פגישות המיועדות ליום הנבחר. הפגישות יהיו מורכבות מטקסט חופשי. התכנית אינה צריכה להתייחס לשעות הפגישות אך המשתמש יכול כמובן לרשום את השעה כחלק מהטקסט החופשי. לחיצה על יום מסויים בטבלה תציג בתיבת דו-שיח את רשימת הפגישות של אותו היום. המשתמש יוכל לערוך את הטקסט ולשמור אותו.

### הדרכה:

ממשק המשתמש יכלול אזור בקרה המאפשר לבחור את השנה והחודש המבוקש לתצוגה, טבלה של כפתורים עבור ימי החודש.

עליכם לשמור את הפגישות במבנה נתונים מסוג HashMap או TreeMap כפי שמוצג בסעיף 16.10 בספר הלימוד, כאשר מפתח הטבלה הוא התאריך והתוכן הוא מחרוזת (או מערך של מחרוזות) המייצגת את הפגישות באותו היום.

העזרו במחלקה java.util.Calendar לצורך עבודה עם תאריכים. שימו לב למתודות הבאות:

- getInstance - מתודה סטטית ליצירת אובייקט מסוג Calendar המייצג את התאריך הנוכחי. לדוגמה:

Calendar c = Calendar.getInstance();

- set – מתודה המקבלת שנה חודש ויום ומעדכנת את התאריך לפי ערך הפרמטרים, למשל  
עבור האובייקט c שהוגדר לעיל אפשר לקבוע את התאריך 1 בדצמבר 2011 באופן הבא:  
c.set(2011,Calendar.DECEMBER,1)

- מתודה get המקבלת קבועים שונים מהמחלקה Calendar ומחזירה את ערכם. למשל,  
עבור האובייקט c שהוגדר לעיל יוחזרו הערכים הבאים:

c.get(Calendar.YEAR) = 2011

c.get(Calendar.MONTH) = 11

c.get(Calendar.DAY\_OF\_MONTH) = 1

c.get(Calendar.DAY\_OF\_WEEK) = 5

שימו לב שמספרו של החודש דצמבר הוא 11 ולא 12 (המספור מתחיל מ-0).

הקבוע DAY\_OF\_MONTH מחזיר את מספר היום בחודש.

הקבוע DAY\_OF\_WEEK מחזיר את מספר היום בשבוע, במקרה זה ה-1 בדצמבר 2011 חל ביום  
חמישי ולכן מוחזר הערך 5.





# מטלת מנחה (ממ"ן) 15

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 23 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 13.1.2023

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

### שאלה 1 (50 נקודות)

ברצוננו לכתוב תכנית המדמה פקידי בנק המעדכנים חשבונות בנק. המערכת כוללת חשבונות בנק, פקידים המעדכנים חשבונות אלה, מאגר תנועות שיש לבצע ומחלקת הבנק המאתחלת את החשבונות, הפקידים והמאגר.

א. **הגדירו את מחלקת חשבון בנק.** המחלקה מכילה מספר חשבון ויתרה והיא כוללת את האלמנטים הבאים:

- בנאי המאתחל את מספר החשבון ואת היתרה.
- מתודה transaction המאפשרת לבצע הפקדות ומשיכות. המתודה מקבלת פרמטר המייצג את סכום הפעולה כאשר ערך חיובי בפרמטר מציין את סכום ההפקדה וערך שלילי מציין את סכום המשיכה.
- מתודה לאחזור היתרה
- כתבו את המחלקה כך שתתפקד בצורה תקינה כאשר מספר פקידים בנק ניגשים אליה במקביל. עליכם לספק מימוש המשיכה פקידים המנסים לבצע משיכה שתגרום להיווצרות יתרה שלילית, כלומר עליכם לדאוג לכך שהחשבון לא ייכנס לחובה. הפקידים המושהים ימתינו עד אשר יתווסף מספיק כסף בחשבון. דאגו לכך שההמתנה לא תהיה המתנה פעילה, (עליכם להימנע מ-busy wait).

ב. **הגדירו מחלקה המייצגת את מאגר תנועות.** המחלקה תכלול מערך של תנועות כאשר כל תנועה כוללת מספר חשבון וסכום (חיובי או שלילי) המציין את ההפקדה או המשיכה שיש לבצע.

- המאגר יקבל בבנאי את מערך התנועות שיש לבצע.
- המאגר יכיל מתודה המאפשרת לפקידים לשלוח תנועה מהמאגר. בכל פנייה של פקיד תוחזר תנועה אחת והיא תוסר מהמאגר. פקידים שיפנו למאגר לאחר שכל התנועות נשלפו יקבלו ערך null המציין שאין עוד תנועות לביצוע.

- ג. **הגדירו תהליך (Thread) המייצג פקיד בנק.** הפקיד יקבל בבנאי מצביע למאגר התנועות ומצביע למבנה נתונים המכיל את חשבונות הבנק.
- הפקיד יפנה למאגר וישלוף רשומה המייצגת תנועה.
  - הפקיד יפנה לחשבון המבוקש ויבצע עליו את התנועה המבוקשת.
  - לאחר הצלחת ביצוע התנועה, הפקיד ינוח זמן רנדומלי (עד 100 מילישניות) ואז יפנה שוב למאגר לקבלת התנועה הבאה.
  - הפקיד יסיים את עבודתו לאחר שלא יישארו עוד תנועות במאגר (כלומר לאחר קבלת null מהמאגר).
- ד. **הגדירו את מחלקת הבנק** המייצרת את האלמנטים שהוגדרו לעיל:
- 5 חשבונות בנק עם מספרי חשבון 4..0. אתחלו את החשבונות עם יתרה 0.
  - מאגר עם תנועות עבור 5 החשבונות עם הפקדות ומשיכות רנדומליות בטווח של 1000- עד 1000.
  - 10 פקידים המקבלים בבנאי הפנייה למאגר התנועות והפנייה למבנה נתונים המכיל את החשבונות.
  - הריצו את הפקידים עם הדפסות מלוות כדי שתוכלו לעקוב אחר פעילות המערכת.

## שאלה 2 (50 נקודות)

נתונה המחלקה Data המכילה שני שלמים. המחלקה מכילה מתודה המאפשרת לעדכן את הערכים ומתודה המחזירה את ההפרש בין הערכים.

```
public class Data{
    private int x = 0;
    private int y = 0;
    public Data (int x, int y){
        this.x = x;
        this.y = y;
    }
    public int getDiff(){
        return (Math.abs(x,-y));
    }
    public void update(int dx, int dy){
        x = x + dx;
        y = y + dy;
    }
}
```

- א. הגדירו שני תהליכים המקבלים reference לאובייקט מסוג Data (אותו אובייקט).
- התהליך הראשון מייצר 10 זוגות של מספרים אקראיים ומעדכן באמצעותם בזה אחר זה את האובייקט מסוג Data (באמצעות המתודה update). עליכם לבצע השהייה בין עדכון לעדכון ולהציג בפלט הסטנדרטי את ערכי המספרים.

- התהליך השני פונה 10 פעמים לאובייקט בכדי לקבל את ההפרש (באמצעות המתודה `getDiff`). עליכם לבצע השהייה בין פנייה לפנייה ולהציג בפלט הסטנדרטי את התוצאות שהתקבלו.
- הפעילו את התהליכים במקביל ובדקו האם התוצאות שהתקבלו ע"י התהליך השני משקפות את כל העדכונים שהתבצעו ע"י התהליך הראשון.
- ב. הוסיפו למחלקה `Data` מנגנון המסנכרן בין שני התהליכים כך שיפעלו לסירוגין. הפעילו את התהליכים במקביל ובדקו את התוצאות.
- ג. שנו את `Data` כך שתאפשר "**ריבוי קוראים וכותב יחיד**" באופן הבא :
  - מספר תהליכים יוכלו לבצע חישוב (`getDiff`) בו זמנית, ובזמן זה לא יתבצע אף עדכון (`update`).
  - יש לוודא שתהליך המבצע עדכון (`update`) יהיה היחיד שניגש באותו זמן לנתונים המשותפים, כלומר חישובים ועדכונים נוספים לא יתבצעו בזמן שהנתונים מתעדכנים.
- עליכם לבצע את השינוי ב- `Data` ולא בתהליכים. הפעילו את התכנית עם 4 תהליכים מכל סוג ועקבו אחר פעילותם באמצעות הדפסות מתאימות.



# מטלת מנחה (ממ"ן) 16

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 28 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 27.1.2023

סמסטר: 2023א

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

## שאלה 1 (60 נקודות)

כתבו תכנית שרת לקוח המממשת את "משחק הזיכרון" בין זוגות של שחקנים הנמצאים במחשבים שונים ברשת.

המשחק מציג מטריצה בגודל  $n \times n$  של קלפים כאשר על כל קלף נמצאת תמונה כלשהי. כל תמונה מופיעה על שני קלפים בדיוק. הקלפים מעורבבים (כך שסדר התמונות לא ידוע) והם מוצגים במטריצה כשהם הפוכים, כך שלא ניתן לראות את התמונות שמוצגות עליהם.

המשחק מתנהל בתורות באופן הבא:

כל שחקן בתורו הופך שני קלפים ע"י לחיצות עכבר. אם הקלפים בעלי תמונה זהה, הם מהווים זוג ויש להשאיר אותם גלויים והשחקן שחשף את הזוג צובר נקודה. אם הקלפים לא מהווים זוג יש להפוך אותם בחזרה לאחר 3 שניות.

שני השחקנים רואים את אותו הלוח עם אותם הקלפים.

המשחק מסתיים כאשר כל הזוגות נחשפו. בסיום המשחק יש להציג לשני השחקנים תיבת דו-שיח עם הודעת סיום הכוללת הודעת ניצחון לשחקן שצבר את מירב הנקודות.

## הדרכה:

**תכנית השרת** פועלת באופן הבא: השרת ממתין לשחקנים, עבור כל זוג שחקנים שמתחברים, השרת פותח עבורם משחק. השרת מנהל את המשחק בכך שהוא מגריל את לוח המשחק ושולח אותו לכל שחקן. אין צורך להעביר את התמונות עצמן, אפשר להסתפק בהעברת מספרן הסידורי. השרת מודיע לשחקן הראשון שהתחבר שהמשחק מתחיל והוא ממתין לבחירתו של השחקן הראשון. במהלך המשחק השרת מקבל מכל שחקן את פרטי הקלפים שנבחרו ומעביר את המידע לשחקן השני. השרת צובר את הניקוד עבור השחקנים ושולח הודעת ניצחון בסוף המשחק. השרת אינו מכיל ממשק גרפי והוא מסוגל לנהל כמה משחקים במקביל.

**תכנית הלקוח** תכלול את הממשק הגרפי של המשחק. עם הפעלת התכנית, תתבצע פנייה לשרת המהווה בקשה להצטרפות למשחק.

לאחר התחברות יש להמתין להתחברות של שחקן נוסף ולקבלת פרטי הלוח מהשרת. בשלב זה יוצג הלוח והשרת יודיע לשחקן אם הוא ראשון או שני. תכנית הלקוח תאפשר למשתמש בתורו לבחור שני קלפים ותשלח את המידע לשרת. לאחר מכן התכנית תמתין לקבל את פרטי תורו של השחקן השני ותעדכן את הלוח בהתאם. התכנית תדאג לכך שהמשתמש יוכל לשחק בתורו בלבד. בסוף המשחק, התכנית תמתין לקבל מהשרת את הודעת הניצחון ואת הניקוד של שני השחקנים. תכנית הלקוח תציג את המידע למשתמש. בסיום המשחק, יש לאפשר לשחקן לבקש לשחק משחק חדש. בקשה זו תטופל כפנייה חדשה לשרת.

#### **הערות :**

- המערכת תמומש באמצעות תקשורת TCP, יש להגדיר פרוטוקול מדויק בין השרת ללקוח. את התקשורת יש לממש באמצעות העברת אובייקטים.
- עליכם לאפשר למשתמש לספק לתכנית הלקוח פרמטרים הכוללים את שם המחשב ומספר ה-port עליו מאזין השרת.
- עליכם לאפשר למשתמש לספק לתכנית השרת פרמטר המציין את מימדי לוח המשחק. יש להקפיד על כך ש-n יהיה מספר זוגי (ניתן להגביל כרצונכם).
- ניתן להריץ את השרת והלקוח על אותו המחשב.

#### **שאלה 2 (40 נקודות)**

כתבו שרת חדשות המספק מבזקי חדשות ללקוחות הפונים לשרת.

#### **תכנית השרת**

- תכנית השרת תאזין בפורט 7777 ותקבל פניות של לקוחות המעוניינים להרשם לקבלת מבזקי חדשות ופניות של לקוחות המעוניינים להסיר עצמם מהרשימה.
- במקביל, התכנית תקבל מבזקי חדשות בקלט (מאזור טקסט) ותפיץ את המידע לכל הלקוחות הרשומים אצלה.
- תכנית השרת לא תשמור אצלה את המבזקים הישנים, כל מבזק יופץ ללקוחות שבאותו רגע רשומים.
- הגבלת זמן ההמתנה להודעה מתבצעת באמצעות הפעלה חד פעמית של המתודה  
: DatagramSocket של setSoTimeout

```
public void setSoTimeout(int timeout)
    throws SocketException
```

### תכנית הלקוח

- לקוח יוכל לפנות לשרת כדי להרשם אצלו לקבלת מבזקים או כדי להסיר עצמו מהרשימה.
  - המבזקים שיגיעו ללקוח יוצגו באזור טקסט יחד עם התאריך והשעה בו הם הגיעו.
  - המשתמש יוכל לנקות את אזור הטקסט באמצעות כפתור clear.
- המערכת תמומש באמצעות תקשורת UDP.
- עליכם לאפשר למשתמש לספק לתכנית הלקוח מידע הכולל את שם המחשב עליו רץ השרת.
- ניתן להריץ את השרת והלקוח על אותו המחשב.