

תרגיל בית 2 – תהליכונים

אופן ההגשה:

- במידה ובתרגיל מופיעה דוגמא של פלט מסוים, הקפידו שהפלט שלכם יהיה זהה
- אם בתרגיל הוזכרו במפורש שמות מחלקות/משתנים וכד', או קבצים בהם יש להשתמש – יש לדייק בשמות, ולהשתמש בקבצים שהוגדו. יש לממש את השאלות ב java ולהגיש רק קבצי java ללא קבצים אחרים של הפרוייקט.
- יש לרשום את שם פרטי, שם משפחה ות"ז המגישים בראש כל קובץ המהווה חלק מפיתרון התרגיל. בנוסף יש להגיש קובץ word המכיל את השמות שלכם בעברית ותשובות על השאלות בכתב.
- במידה והמטלה מכילה יותר מקובץ אחד, יש לכווץ את כלל הקבצים לקובץ zip (ולא פורמטים אחרים כמו rar), כאשר שם הקובץ המכווץ כולל את ת"ז המגישים. למשל, אם הגישו את העבודה בעלי ת"ז 11111 ו-22222, שם הקובץ יהיה 11111_22222.zip. לאחר הכיווץ, העלו את קובץ ה zip בלבד ללימוד.
- בנוסף, בקובץ word ובקבצים עם ה-main של כל שאלה יש לרשום קישור ל-GitHub עם הקוד שלכם שפיתחתם אותו ב-Git.
- ההגשה בזוגות בלבד (אלא אם כן אושר אחרת ע"י המרצה)
- יש להגיש את הפתרון עד למועד שנקבע בלימוד
- חלק מהנבדק בתרגיל הוא עמידה בנקודות הנ"ל. הקפידו לעמוד בהן.
- אישורי הארכה יינתנו ע"י המרצה בלבד.
- את כל השאלות יש לשאול בפורום לעבודה 2.

בהצלחה!

שאלה 1

צרו חבילה (package) בשם **stocks** וממשו את התוכנית הבאה:

בבורסא שבארץ "לעולם לא", סוחרים יכולים לצפות (צפייה בלבד) בשערי מניות דרך גישה לאתר stocker.com.

נהג האוטובוס תמי טאן, שחקן הכדורגל טים סרולי והמורה לספורט סימה דידס, מנסים לצפות בנתוני המניות שקנו. שלושתם צופים בנתונים בו - זמנית.

כל אחד מהם צופה בנתונים של מניה כלשהי, הולך לישון למשך זמן ראנדומאלי שבין שנייה אחת ל 3, ואז צופה בנתונים שוב – ושוב חוזר על פעולות אלו - 10 פעמים.

1. כתבו תוכנית המדמה את התהליך המוזכר מעלה, ומספקת שירות לשלושת האנשים במקביל. בכל קריאה של מידע, הקורא מדפיס את הנתונים שקרא למסך. למשל: אם הנהג תמי טאן קרא שהמנייה של מיקרוסופט שווה 215 דולר, הוא ידפיס על המסך

"Name: Tami Tan, Microsoft Stock: 215 USD"

לשם מימוש התוכנית, פתחו package חדש בשם **stocks** וחלצו לתוכו את הקובץ **StockServer.java** המצורף לעבודה.

כדי לקרוא ערך של מניה מסויימת מהשרת, השתמש בפונקציה **GetStock** של המחלקה **StockServer**. למשל: כדי לקרוא את ערך המנייה של מיקרוסופט:

```
int sockValue = stock.GetStock(Stock.MICROSOFT);
```

המניות הזמינות בשרת הן:

MICROSOFT, APPLE, GOOGLE

2. במקרה המתואר בשאלה, מהי הדרך הטובה ביותר להפוך את המחלקה **StockServer** לבטוחה להרצה מקבילית? בחר את אחת מהאופציות הבאות, ותאר מדוע היא טובה יותר מהאופציה האחרת

a. אופציה א': להפוך את המחלקה ל **immutable**

b. אופציה ב': להשתמש בסנכרון על קטע קוד קריטי ובכך להשיג מניעה הדדית

3. מפתחי האתר stocker.com גילו להפתעתם כי הערכים של המניות באתר לא מתעדכנים עקב קוד שגוי במחלקה **StockServer**. ערכי המניות במחלקה קבועים, ואין פונקציה המאפשרת לשנותם.

הוסיפו למחלקה פונקציה בשם **UpdateStock(Stock stock, int value)** המקבלת את סוג המנייה, ומעדכנת את השדה המתאים של ערך המנייה בערך ראנדומאלי בין 100-500, כך שעבור קריאות שונות של הפונקציה **GetStock** יתקבלו ערכים שונים.

- בנוסף, כתבו תהליכון בשם `StockUpdateThread` שתפקידו לעדכן את המחלקה `StockServer` באמצעות הפונקצייה `UpdateStock`. התהליכון יעדכן את כל סוגי המניות, ילך לישון למשך שנייה אחת, ויחזור על פעולות אלו 10 פעמים.
4. לאחר הוספת התהליכון `StockUpdateThread`, האם תשובתכם לסעיף 2 תשתנה? הסבירו מדוע, וערוך את המחלקה `StockUpdateThread` כך שתתאים לתשובתכם ותהיה בטוחה להרצה מקבילית.
5. כתבו פונקציית `main`, וצרו בתוכה:
- a. אובייקט `StockServer` המייצג את השרת של המניות
- b. 3 תהליכונים שקוראים מניות (אחד לכל אחד מהאנשים)
- תהליכון אחד שמעדכן את שערי המניות בשרת

שאלה 2

בשאלה זו נדרש מכם לממש תוכנית המדמה תחרות ריצה, כאשר כל משתתף במירוץ הוא בעצם תהליכון.

צרו חבילה חדשה (package) בשם **Race** וממשו את המחלקות הבאות:

המחלקה Track – מייצגת את מסלול המירוץ

שדות:

- `finishedRacers` – שדה השומר את מספר הרצים שסיימו את המירוץ

המחלקה Racer – מייצגת משתתף במירוץ. מממשת את Runnable

שדות:

- `globalId` - שדה סטאטי המאותחל ל-1, וישמש ליצירת מס' מזהה ייחודי לכל משתתף
- `id` – מאחסן את מס' הזיהוי הייחודי של המשתתף, הנוצר באמצעות המשתנה הקודם
- `speed` – מהירות הריצה של המשתתף, חייבת להיות בין 1 – 10 (עליכם לוודא זאת בקבלת הקלט, ולהדפיס שגיאה במידה והקלט שגוי)
- `track` – שדה המאחסן את מסלול הריצה

בנאים:

- בנאי המקבל מהירות המשתתף ומסלול ריצה, ומאתחל את שדות המחלקה בהתאם (שימו לב – בצעו שימוש ב globalld ע"מ לאתחל מס' זיהוי המשתתף כך שיהיה ייחודי)

מתודות:

- go – מתודה הקובעת את עדיפות התהליכון שקרא לה להיות שווה למהירות המשתתף (speed) ואז מריצה לולאה מ – 1 עד 100, ומדמה את מספר המטרים שהמשתתף עבר. לאחר כל מטר שהמשתתף עובר, יודפס מספר המשתתף, ומספר המטרים שעבר. לדוגמא:

Runner 1 ran 19 meters

במידה והמשתתף השלים 100 מטר, תודפס הודעה כמו בשורה הקודמת, המודיעה על הגעתו ל 100 מ', ומייד לאחר מכן תודפס הודעה על כך שהמשתתף סיים את המירוץ, והמקום בו סיים. למשל:

Runner 1 finished 1st

שימו לב לחלק המודגש בפלט: עבור המקום הראשון יודפס 1st, המקום השני 2nd, השלישי 3rd, ומהרביעי ואילך: 4th, 5th וכו'.

- Run – המתודה המגיעה מהמחלקה Runnable. מתודה זו תריץ את המתודה go.

מחלקה main – מחלקת התוכנית הראשית

בתוך קובץ ה zip של המטלה תמצאו קובץ JAVA בשם Main. הוסיפו אותו ל package של שאלה זו כמחלקה ראשית אשר ממנה תרוץ התוכנית.

אין לשנות את הקוד במחלקה זו, על הקוד שלכם לרוץ עם הקוד המקורי שבקובץ זה.
במידה ומתקבלות שגיאות, בצעו התאמות בקוד שלכם כך שיעבוד עבור התוכנית הראשית, אבל זכרו לעשות זאת תוך הקפדה על הכללים והפרטים שהוזכרו.
מעבר לרשום מעלה, הנכם רשאים להוסיף בכל המחלקות מתודות ומשתנים כפי שידרש לכם.

ענו בכתב:

1. באחת מהרצות התוכנית התקבל הפלט הבא:

```
Runner 4 ran 100 meters  
Runner 3 ran 74 meters  
Runner 4 finished 3rd
```

ניתן לראות כי למרות שמשתתף מס' 4 סיים את המירוץ, והלוגיקה של התוכנית מכתובה לו להדפיס שהגיע ל 100 וגם סיים את המירוץ, הוא מדפיס שהגיע ל – 100, ורק לאחר הדפסה של משתתף אחר, הוא מדפיס שסיים.

- a. הסבירו את התופעה, מהו שמה, ומהו הגורם לה
- b. האם יתכן שהתופעה "תחמיר" באחת ההרצות, כך שלמרות שמשתתף 4 סיים את הריצה והגיע למטר ה – 100, משתתף שלוש ידווח שסיים את המירוץ לפניו? הסבירו.

2. האם תמיד מהירות גבוהה יותר של משתמש (עדיפות גבוהה יותר של תהליכון)

מבטיחה סיום המירוץ מהר יותר? מדוע?

3. בכיתה למדנו 2 שיטות ליצירת תהליכון.

a. פרטו מהן, ומה היתרון והחסרון של כל אחת מהשיטות?

b. הביטו בקטע הקוד הבא:

```
public static void main(String[] args) {  
    MyThread t1 = new MyThread();  
    Thread th1 = new Thread(t1, "Thread A");  
    th1.start();  
}
```

נתון שהמחלקה MyThread מממשת את Runnable (שיטה 2 ליצירת תהליכון).

מדוע בשיטה זו יש ליצור מופע מהמחלקה MyThread וגם מופע של המחלקה Thread כדי להריץ את התהליכון (ולא מספיק ליצור רק מופע של MyThread כמו שראינו בשיטה 1)?

שאלה 3

בשאלה זו עליכם ליצור משחק איקס עיגול אשר בו שני שחקנים. במשחק הנ"ל הלוח יהיה בגודל 5x5 וינצח שחקן שמצליח לעשות 4 סימנים (O או X) צמודים זהים בשורה, בעמודה או בצורה אלכסונית.

למשל בדוגמה הבאה מנצח O:

X	O	X	O	
X		O		X
O	X		O	
	X	X		O
			O	

פתחו חבילה בשם XO. כל שחקן יהיה תהליכון נפרד.

יש לממש 2 גרסאות של המשחק:

1. בגרסה ראשונה כל שחקן הוא תהליכון עצמאי.
2. בגרסה שנייה, שחקן אחד יהיה תהליכון עצמאי ובתור שחקן השני ישחק משתמש. כלומר המשתמש יבחר את התא עבור המהלך שלו.

כתבו תוכנית מקבילית המדמה משחק איקס עיגול באופן הבא:

- ממשו את המחלקה אבסטרקטית Game.java אשר מכילה שדות ומתודות הבאות:
 - שדה gameBoard – מערך דו ממדי בדולל 5x5 המשמש כלוח של המשחק.
 - printBoard – מתודה המדפיסה את הלוח הנוכחי עם X ו-O ים.
 - getTurn – מתודה המחזירה את סוג השחקן שתורו לשחק. הוסיפו מחלקה או enum המייצג את סוג השחקן (X או O).
 - getFreeCells – מתודה המחזירה את כל התאים הפנויים של הלוח. אפשר לממש תא בעזרת מחלקה עם 2 קואורדינטות.

- getters ו-setters מתאימים.
- הוסיפו שדות ומתודות לפי הצורך.

- ממשו מחלקות קונקרטיות SelfGame.java ו-UserGame.java היורשות ממחלקה Game ומנהלות את הגרסה של המשחק המתאימה.
- צרו מחלקה אבסטרקטית בשם Player המקבלת בבנאי את סוג השחקן (האם מייצג שחקן מסוג איקס או עיגול) וצרו 2 מחלקות קונקרטיות (תהליכונים עם מתודות run) SelfPlayer ו-UserPlayer עבור שחקן עצמאי ועבור שחקן שמנוהל ע"י המשתמש.
- במידה והשחקן הוא עצמאי, התהליכון ירוץ בלולאה, ובכל איטרציה הוא ילך לישון 500 מילי שניות, ולאחר מכן יבדוק האם תורו (למשל: במידה והוא שחקן מסוג איקס, עליו לוודא שכעת התור של איקס לשחק). במידה ואכן תורו הגיע, והלוח אינו מלא (עדיין קיימים תאים פנויים), הוא יבחר רנדומאלית תא בלוח מבין כל התאים הפנויים, ויסמן שם איקס או עיגול (לפי הסוג שהוא מייצג). לאחר מכן ידפיס את הלוח.
- במידה ושחקן מנוהל ע"י משתמש, התהליכון ירוץ בלולאה וכל פעם שתורו מגיע הוא יבקש מהמשתמש להכניס קואורדינטות של התא, יסמן שם איקס או עיגול (כמובן אם תא הזה ריק) ולאחר מכן ידפיס את הלוח.
- אם תהליכון זיהה שהלוח מלא, עליו להדפיס "Board is full" ולא לנסות לשחק את הצעד שלו (מה שיגרום למשחק להעצר).
- אם אחד השחקנים ניצח את המשחק, יש לעצור את המשחק ולהדפיס מי ניצח.
- ממשו מחלקה Main אשר, בהתאם לבחירה של המשתמש, יוצרת משחק ו-2 שחקנים מתאימים, מריצה את המשחק ומדפיסה מי ניצח.