תרגיל בית 3

אופן ההגשה:

- במידה ובתרגיל מופיעה דוגמא של פלט מסוים, הקפידו שהפלט שלכם יהיה זהה.
- תשובות בכתב הקפידו להזין לתוך קובץ pdf, תוך ציון מספר השאלה ומספר הסעיף
- 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
- **יש לרשום את שם ות"ז המגישים בראש כל קובץ המהווה חלק מפיתרון התרגיל. (לדוגמא: אם בניתם 3 מחלקות, וקובץ pdf לתשובות, אז השם והת"ז יופיע בראש שלוש המחלקות וגם בראש קובץ pdf). בקובץ pdf יש לרשום את השם בעברית**
- 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.
- 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.
- את כל השאלות יש לשאול בפורום של תרגיל 3.

בהצלחה!

שאלה 1

יש לנו 3 תהליכונים t1, t2 ו-t3. כל תהליכון מבצע בלולאה נצחית קטע קוד. כלומר קוד של 3 תהליכונים נראה כך:

```
t1:
void run() {
   while (true) {
          A
   }
}

t2:
void run() {
   while (true) {
          B
   }
}

t3:
void run() {
   while (true) {
          C
   }
}
public static void main(String[] args) {
    Thread t1 = new Thread();
    Thread t2 = new Thread();
    Thread t3 = new Thread();
    t1.start();
    t2.start();
    t3.start();
}
```

כאשר A, B ו-C זה בלוק של שורות קוד שהם מבצעים.

עליכם לסנכרן את התהליכונים (בעזרת אחת השיטות שלמדתם) כך שיתבצעו באופן הבא: קודם תהליכון t1 יבצע את קטע קוד A מתחילתו ועד סופו, לאחר מכן תהליכון t2 יבצע את קטע קוד B מספר כלשהו של פעמים, מתחילתו ועד סופו, ורק לאחר מכן t3 יבצע את הקטע קוד C מתחילתו ועד סופו.

אחרי הסיבוב הזה הכל מתחיל מהתחלה: A, ואז B מספר כלשהו של פעמים, ואז C וכך הלאה עד שעוצרים את ריצת התוכנית.

הערה: מספר פעמים שקטע קוד B מתבצע ברצף אינו נתון מראש, הדבר תלוי בתזמון התהליכונים בסוף ביצוע קטע קוד B:

- אם t3 מתוזמן בדיוק כאשר t2 סיים קטע קוד B, אז t3 רשאי וחייב להתחיל לבצע את קטע קוד C.
- אם t2 מתוזמן אחרי שסיים לבצע קטע קוד B ולפני ש-t3 התחיל לבצע את קטע קוד שלו C, אז t2 חוזר ומבצע שוב את קטע קוד שלו B.

שאלה 2

פתחו חבילה בשם assig3_2 וממשו תוכנית המדמה משחק עץ או פלי, עם החוקים הבאים:

1. במשחק ישנם שני משתתפים, ומטבע אחד. לכל משתתף במשחק ישנו מונה, הסופר את כמות הפעמים שהתקבל עץ (מתקבל הערך true מהפונקציה flipCoin()).
2. המשחק נגמר, לאחר ששני המשתתפים הטילו את המטבע 10 פעמים
3. המנצח הוא השחקן עם מספר ההטלות הרב ביותר שבהן התקבל עץ

המחלקה GamePlay מייצגת את המשחק עצמו, את המטבע ואת מספר ההטלות

ממשו את המחלקה GamePlay באופן הבא:

1. שדה coin_available_ המתאר האם המטבע זמין
2. שדה rounds_counter_ המתאר את כמות ההטלות שבוצעו במשחק
3. פונקציה makeCoinAvail(boolean val):
   a. הופכת את המטבע לזמין או לא זמין לפי הערך val
   b. במידה והמטבע הופך זמין, מודיעה על כך לשאר התהליכונים שממתינים למטבע
   c. מעדכנת את השדה coin_available_ בהתאם לערך שקיבלה
4. פונקציה flipCoin():
   a. אם המטבע אינו זמין, התהליכון:
      i. ימתין עד שהמטבע יהפוך זמין (זכרו להשתמש ב guarded suspension)
      ii. ידפיס את שמו (באמצעות getName() של המחלקה Thread), ואת העובדה שהוא נכנס להמתנה. למשל:
      Thread-0 is waiting for coin
   b. אם המטבע זמין, התהליכון:
      i. ידפיס שהוא מטיל מטבע. למשל:
      Thread-1 is flipping coin
      ii. יהפוך את המטבע ללא זמין במהלך ההטלה
      iii. יקדם את מספר ההטלות ב-1
      iv. יגריל ראנדומאלי 0 או 1
      v. יהפוך את המטבע שוב לזמין, ויודיע על כך לתהליכונים אחרים
   c. הפונקציה תחזיר את תוצאת ההטלה (0 = שקר/כשלון, 1 = אמת/הצלחה)

5. פונקציה getNumOfRounds() :
   a. מחזירה את מספר ההטלות במשחק

המחלקה Gamer מייצגת תהליכון שחקן אשר מטיל את המטבע שוב ושוב

ממשו את המחלקה Gamer באופן הבא:

1. שדה goodFlipsCounter הסופר את כמות ההטלות שהשחקן ביצע והצליחו
2. שדה מסוג GamePlay המאותחל בבנאי (האובייקט מתקבל מבחוץ)
3. פונקציית play() המתבצעת בלולאה, ובכל איטרציה:
   a. כל עוד שלא ביצעו לתהליכון INTERRUPT וגם מספר ההטלות <u>במשחק כולו</u> קטן או שווה ל 10:
      i. נסה להטיל מטבע, ואם הצלחת קדם את goodFlipsCounter ב-1
      ii. לך לישון למשך שנייה אחת
4. GETTER בשם getScore המחזיר את מספר ההטלות שהצליחו

המחלקה Judge מייצגת שופט במשחק, ההופך את המאשר לשחקנים מתי הם רשאים להטיל את המטבע ומתי לא, ע״י הפיכת המטבע לזמין / לא זמין

ממשו את המחלקה Judge באופן הבא:

1. לולאה, בה כל עוד לא ביצעו לתהליכון interrupt:
   a. השופט הופך את המטבע ללא זמין למשך שנייה
   b. השופט הופך את המטבע לזמין למשך חצי שנייה

במחלקה הראשית של התוכנית:

1. צרו משחק חדש
2. צרו שני שחקנים
3. לאחר שהשחקנים <u>מסיימים</u> לשחק, מודפסת הודעה על השחקן שניצח (השחקן בעל הניקוד הגבוה ביותר). למשל:

   player 1 wins

   במידה ולשחקנים ניקוד זהה, יודפס תיקו כך:

   tie

שאלה 3

פתח חבילה בשם assig3_3 , והכנס לתוכה את קבצי הקוד שתחת התיקייה assig3_3 שצורפה למטלה.

בשאלה זו עליכם לממש תוכנה עבור מכונה אוטומטית להכנת סלט. המכונה מורכבת מתא בעל ראש קוצץ, שאליו מוזנים באופן אוטומטי מלפפונים ועגבניות. ברגע שלתא הוזנו 3 מלפפונים ו-2 עגבניות, הלהבים מכינים מהם סלט אחד.

בתחילת התוכנית המשתמש בוחר כמות סלטים להכנה, והמכונה תכין את הסלטים אוטומטית תחת האילוצים הבאים:

- תהליך הזנת מלפפונים לתא הקוצץ יכול לקרות **רק אם יש בתא פחות מ 3 מלפפונים**. בכל מקרה אחר הזנת המלפפונים צריכה להיעצר עד שיתפנה מקום בתא הקוצץ.
- תהליך הזנת עגבניות לתא הקוצץ יכול לקרות **רק אם יש בתא פחות מ 2 עגבניות**. בכל מקרה אחר הזנת העגבניות צריכה להיעצר עד שיתפנה מקום בתא הקוצץ.
- הקוצץ יכין סלט **רק כאשר בתא יש 3 מלפפונים ו-2 עגבניות**. בכל מקרה אחר הקוצץ ייעצר.
- כאשר הקוצץ הכין N סלטים (לפי בקשת המשתמש), **הוא יעצר, ואחריו יעצרו גם התהליכונים** שמזינים את הקוצץ בירקות, והתוכנית תסיים כאשר כל התהליכונים **נסגרו**.

לשימושכם בקוד שקיבלתם מחלקות שאותן יש לשנות ולהשלים:

- מחלקה בשם CucumberThread האחראית על הזנת מלפפונים לקוצץ.
- מחלקה בשם TomatoesThread האחראית על הזנת עגבניות לקוצץ.
- מחלקה בשם SlicerThread האחראית על קיצוץ הירקות.
- מחלקה בשם SlicerMachine המייצגת את הפונקציות של המכונה:
  - addOneCucumber – הזנת מלפפון אחד לתא הקוצץ
  - addOneTomato – הזנת עגבניה אחת לתא הקוצץ
  - sliceVegetables – קיצוץ הירקות
  - getNumOfPreparedSalads – מחזירה את מספר הסלטים שהמכונה הכינה

עליכם לממש את המחלקות של התהליכונים כך שישתמשו במחלקה SlicerMachine להשגת המטרה שהוגדרה.

המחלקה SlicerMachine מכילה קוד ראשוני בלבד שיעבוד בהרצה **סדרתית**, שנו אותה כך שתתאים להרצה מקבילית שתקיים את הלוגיקה שהוגדרה תוך **שמירה על בטיחות להרצה מקבילית**.

דוגמא לפלט תקין אפשרי של התוכנית (שימו לב שהפלט מעיד שהוכנו 3 סלטים כמו שהמשתמש בחר, ובנוסף כל הכנת סלט קרתה רק אחרי שהיו מספיק ירקות בקוצץ) :

Please Type How Many Salads To Prepare:

3

Preparing 3 Salads...

adding one tomato to the machine

adding one cucumber to the machine

adding one cucumber to the machine

adding one tomato to the machine

adding one cucumber to the machine

== preparing one more salad ==

adding one tomato to the machine

adding one cucumber to the machine

adding one tomato to the machine

adding one cucumber to the machine

adding one cucumber to the machine

== preparing one more salad ==

adding one tomato to the machine

adding one cucumber to the machine

adding one tomato to the machine

adding one cucumber to the machine

adding one cucumber to the machine

== preparing one more salad ==

adding one tomato to the machine

adding one cucumber to the machine

adding one tomato to the machine

Done