

תרגיל 2

בתרגיל זה אנו נממש משחק איקס עיגול.
בגרסה שלנו הלוח יהיה בגודל $n \times n$ (למשתנה n נקרא בשם *SIZE* והוא יהיה בגודל 4),
וכדי לנצח נצטרך רצף של סימוני איקס או עיגול באורך k (למשתנה k נקרא בשם *WIN_STREAK* והוא יהיה בגודל 3).
המשחק מיועד בשלב הזה עבור שני שחקנים אנושיים ויתנהל בתורות.
בכל תור השחקן יספק קורדינטה של תא ריק בלוח והסימון שלו יופיע שם.
המשחק נגמר כשאחד השחקנים יוצר רצף מנצח (מאונך, מאוזן או אלכסוני), או
כאשר לא נותרו מקומות פנויים בלוח.

כדי לממש את המשחק נעזר בארבע מחלקות -

1. *Renderer* המסופקת לכם באתר

2. *Game*

3. *Board*

4. *Player*

כמו כן צרו קובץ נפרד והגדירו בו $\text{enum Mark} \{BLANK, X, O\}$
תפקידו של ה *enum* הוא לייצג את הסימונים בקורדינטות על הלוח.

בקובץ זה אנו נעבור על ה *API* בלבד (מתודות עם גישה פומבית).

טיפים למימוש תוכלו למצוא באתר

תפקידי המחלקה board -

1. השחקנים צריכים שהלוח יציג את הסימונים שלהם.
2. הלוח צריך לדעת כשהמשחק נגמר ואיך.
3. הלוח צריך לספק לנו מידע לגבי הסימונים בקואורדינטות השונות.

מתודה ראשונה שנרצה שתממשו היא putMark
המתודה תקבל משחקן Mark (מה הסימן שלו איקס או עיגול) וקורדינטות
(שתי מספרים שלמים המייצגים קורדינטות על הלוח) ותחזיר ערך אמת
שיהווה אינדיקציה לכך שהתוצאה הצליחה או לא.

כמו כן, נרצה מתודה שנקראת getMark שתקבל קורדינטות
ותחזיר את הסימן שיש בהן.
במקרה והקורדינטות לא חוקיות על המתודה להחזיר Mark.BLANK

נשים לב שעל השחקנים לדעת את גודל הלוח ומה הרצף הנדרש לניצחון.
כיוון שאלו בבירור תכונות של הלוח נשים את המשתנים הללו במחלקה board
אבל נרשה גישה פומבית אליהם, כלומר חתימתם תהיה

```
public static final int SIZE = 4
```

```
public static final int WIN_STREAK = 3
```

בנוסף למחלקה יהיה בנאי שאינו מקבל קלט ומגדיר לוח ריק חדש, ושתי getters

אחת עם חתימה `public boolean gameEnded`

המדווחת האם המשחק נגמר

והשנייה עם חתימה `public Mark getWinner`

המדווחת מי המנצח

תפקידי המחלקה `player` -

השחקן מייצג לוגיקה או אסטרטגיה מסויימת לביצוע תור במשחק בהינתן לוח מסויים.

כלומר, מופע אחד של שחקן יכול לבצע תור בכמה משחקים שונים.

בפועל נצטרך שהשחקן יבצע תור יחיד.

נעשה זאת באמצעות מתודת `playTurn` המקבלת לוח וסימן כקלט.

המתודה מבקשת קורדינטות מהמשתמש(שימו לב שמשתמשים

מתחילים לספור מ 1 ולא מ 0)

וממקמת את הסימן במידה והקורדינטות שניתנו תקינות.

במידה והקורדינטות לא תקינות המתודה תבקש מהמשתמש קורדינטות אחרות.

ערך ההחזרה של המתודה הוא

תפקידי המחלקה `Game`-

מופע של המחלקה מייצג משחק יחיד.

מה יהיו התכונות שמזהות משחק במצבו הראשוני, שיפרידו בינו לבין משחקים אחרים?

במילים אחרות, מה הבנאי יצטרך לדרוש כפרמטרים?

ראשית המחלקה Game צריכה אובייקט `Renderer` כדי להציג את הלוח.

כמו כן, מסיבות שיובהרו בתרגיל 3 נבקש גם מהבנאי לקבל גם את שני השחקנים.

תפקידה של מחלקת Game בבנייה שלנו היא להיות המנוע של המשחק,

ולכן טבעי שתהיה לה מתודת `run` מתודה זו למעשה תהיה אחראית

לריצתו של משחק יחיד ותחזיר את ערך המנצח(על ידי החזרת `Mark`).

מתודה זו אינה מקבלת דבר.

בתרגיל זה נבקש שמתודת ה `main` של התכנית כולה תשב במחלקה Game.

בתרגיל 3 נזיז אותה משם.

הנחיות :

אנו מצפים ששמות של משתנים, מתודות, קבועים והודעות יהיו אינפורמטיביים, מתן שמות זו אומנות עדינה(ראו כל ספר פנטזיה אפשרי) ונדרש מכם להקדיש לכך תשומת לב. נקודות יורדו על שמות לא אינפורמטיביים.

עליכם להגיש קובץ jar בשם ex2.jar המכיל את הקבצים הבאים:

1. Game.java לפי ההנחיות שניתנו

2. Board.java לפי ההנחיות שניתנו

3. Player.java לפי ההנחיות שניתנו

3. מחלקה המכילה את MARK

5. קובץ README (ללא סיומת) –

בשורה הראשונה בקובץ זה יופיע שם המשתמש שלכם, בשורה השנייה יופיע מספר תעודת הזהות שלכם, השורה השלישית תהיה ריקה, אין צורך להוסיף יותר בתרגיל הזה.