

# תכנות מונחה עצמים תרגיל 4

## תוכן עניינים

|   |                |
|---|----------------|
| 1 | התרגיל         |
| 2 | חומרים מסופקים |
| 4 | מבנה הספרייה   |
| 5 | מחלקות למימוש  |
| 5 | פרטי מימוש     |
| 6 | אופן ההגשה     |

## התרגיל

בתרגיל זה שני חלקים:

### חלק ראשון

בחלק זה תרחיבו את משחק הבריקר שבניתם בתרגיל הקודם.

מתוך התנהגויות הלבנים המתוארות באתר, אנחנו מבקשים שתממשו את ההתנהגויות הבאות:

1. כדורים נוספים – כאשר הלבנה נשברת מופיעים במקומה 3 כדורים.
2. דיסקית נוספת – כאשר הלבנה נשברת מופיעה דיסקית נוספת.
3. שינוי מצלמה – כאשר לבנה עם התנהגות זו נשברת, המצלמה עוברת לעקוב אחר הכדור עד שהתנגש ב-4 דברים.
4. התנהגות נוספת לבחירתכם מתוך אינדקס ההתנהגויות.
5. התנהגות כפולה –
  - a. ללבנה עם התנהגות זו יהיו שתי התנהגויות, שיכולות להיות או התנהגויות הקודמות, או התנהגות כפולה.
  - b. נרצה שיהיה ניתן לקבל לבנה עם 3 התנהגויות שונות.
  - c. עם זאת, נרצה שללבנה נתונה לא יהיו בסך הכל יותר מ-3 התנהגויות.
  - d. התנהגות נתונה יכולה לחזור על עצמה בלבנה.
- 6.

### חלק שני

בחלק זה תתרגלו מימוש אלגוריתמים בג'אווה.

אנו מבקשים שתפתרו את הבעיות האלגוריתמיות הבאות:

1. לסטודנט יש  $n$  מטלות ו- $m$  חלונות זמן. למטלה ה- $i$  נדרשות  $\text{task}(i)$  שעות להשלמה. בחלון הזמן ה- $i$  יש  $\text{timeSlot}(i)$  שעות. הסטודנט יכול לעשות את המטלה האיי אם יקצה לה חלון זמן ג'י כלשהו. הוא יכול להקצות לה את חלון הזמן הזה רק אם אורכו גדול או שווה למשך הדרוש להשלימה. כתבו אלגוריתם שמוצא בזמן הקצר ביותר שאפשר מה המספר המקסימלי של מטלות שהסטודנט יכול לעשות. ממשו כמתודה `alotStudyTime` כמפורט ב-API של החבילה `algo_questions`. הנחות מקילות ופרטים נוספים נמצאים ב-API.
2. צפרדע יושב על העלה ה-1 בסדרה של  $n$  עלים. העלים שונים במידת יציבותם על המים כך שהמרחקים שניתן לכסות בקפיצה מן העלים שונים. לכל עלה יש מספר טבעי  $\text{leapNum}[i]$ , המציין כמה עלים קדימה אפשר לקפוץ ממנו. למשל, אם  $\text{leapNum}[3]=4$ , אפשר לקפוץ מעלה 3 לעלים 4, 5, 6. כתבו אלגוריתם שמחזיר את המספר המינימלי של קפיצות שבהן יכול הצפרדע להגיע מהעלה הראשון לאחרון. כתבו אלגוריתם יעיל ככל האפשר. ממשו כמתודה `minLeap` כמפורט ב-API של החבילה `algo_questions`. הנחות מקילות ופרטים נוספים נמצאים ב-API.
3. ראו תיאור הבעיה ב-API. ממשו ע"י הפונקציה `bucketWalk`.
4. ראו תיאור הבעיה ב-API. ממשו ע"י הפונקציה `numTrees`.

## חומרים מסופקים

מסופקים לכם לצורך המימוש החומרים הבאים:

1. אוסף של קבצי תמונה לצורך רינדור אובייקטי המשחק – כדור, לבנים, מטקה, וכו'
  1. `assets_files.txt`
  2. `ball.png`
  3. `botBad.png`
  4. `botGood.png`
  5. `brick.png`
  6. `buffNarrow.png`
  7. `buffWiden.png`
  8. `gravity.png`
  9. `heart.png`
  10. `mockBall.png`
  11. `paddle.png`
  12. `quicken.png`
  13. `slow.png`
2. מספר קבצי סאונד לאפקטים של התנגשות וכו'
  1. `Bubble5_4.wav`
  2. `blorp_cut_silenced.wav`
3. קובץ לרינדור רקע
  1. `DARK_BG2_small.jpeg`
4. ספריית מנוע המשחק `DanoGameLab`. ניתן למצוא אותה יחד עם הוראות להתקנה (סרטון ומדריך כתוב) באתר הקורס המקוון.

פרטים נוספים והערות:

1. כל החומרים הנ"ל נמצאים במודל תחת החוצץ של תרגיל 3.
2. החומרים נלקחו מאתר קמפוס IL נכון לתאריך 11-11-21.
3. לצורך מניעת שוני גרסאות, **נא להשתמש בקובץ בחומרים הנמצאים במודל לצורך התרגיל.**
4. אתם תגישו את הקוד שתכתבו בלי קבצי התמונה והסאונד, לכן הקפידו להשתמש בקוד רק בשמות הקבצים הנ"ל, אחרת הקוד עלול לא להתקמפל או לא לרוץ.

## מבנה הספרייה

1. ספריית הפרוייקט שלכם תכיל את הספריות:

src .a

assets .b

2. מבנה הספרייה של הספריות הנ"ל יראה כך, ויכלול את כל הקבצים המתוארים להלן לפי מבנה ספריות

זה בדיוק, אך בניגוד לתרגיל הקודם, הוא יכול לכלול גם קבצי .java. נוספים.

לדוגמא, מלבד הקבצים המופיעים ברשימה, בספרייה src/brick\_strategies/ יכול להימצא גם הקובץ

. WidePaddleStrategy.java

```
./src/
./src/BrickerGameManager.java
./src/brick_strategies
./src/brick_strategies/AddPaddleStrategy.java
./src/brick_strategies/BrickStrategyFactory.java
./src/brick_strategies/ChangeCameraStrategy.java
./src/brick_strategies/CollisionStrategy.java
./src/brick_strategies/PuckStrategy.java
./src/brick_strategies/RemoveBrickStrategy.java
./src/brick_strategies/RemoveBrickStrategyDecorator.java
./src/gameobjects
./src/gameobjects/Ball.java
./src/gameobjects/BallCollisionCountdownAgent.java
./src/gameobjects/Brick.java
./src/gameobjects/GraphicLifeCounter.java
./src/gameobjects/MockPaddle.java
./src/gameobjects/NumericLifeCounter.java
./src/gameobjects/Paddle.java
./src/gameobjects/Puck.java
./algo_questions/
./algo_questions/Solutions.java
./assets
./assets/assets_files.txt
./assets/Attribution.txt
./assets/ball.png
./assets/blow_cut_silenced.wav
./assets/botBad.png
./assets/botGood.png
./assets/brick.png
./assets/Bubble5_4.wav
./assets/buffNarrow.png
./assets/buffwiden.png
./assets/DARK_BG2_small.jpeg
./assets/gravity.png
./assets/heart.png
./assets/mockBall.png
./assets/paddle.png
./assets/quicken.png
./assets/slow.png
```

3. כל קובץ java. מהרשימה הזו יכיל מימוש של בדיוק מחלקה אחת, ששמה כשם הקובץ.
4. הקפידו לשייך כל מחלקה לחבילה.
5. הקפידו שמבנה החבילות יתאם למבנה הספריות.

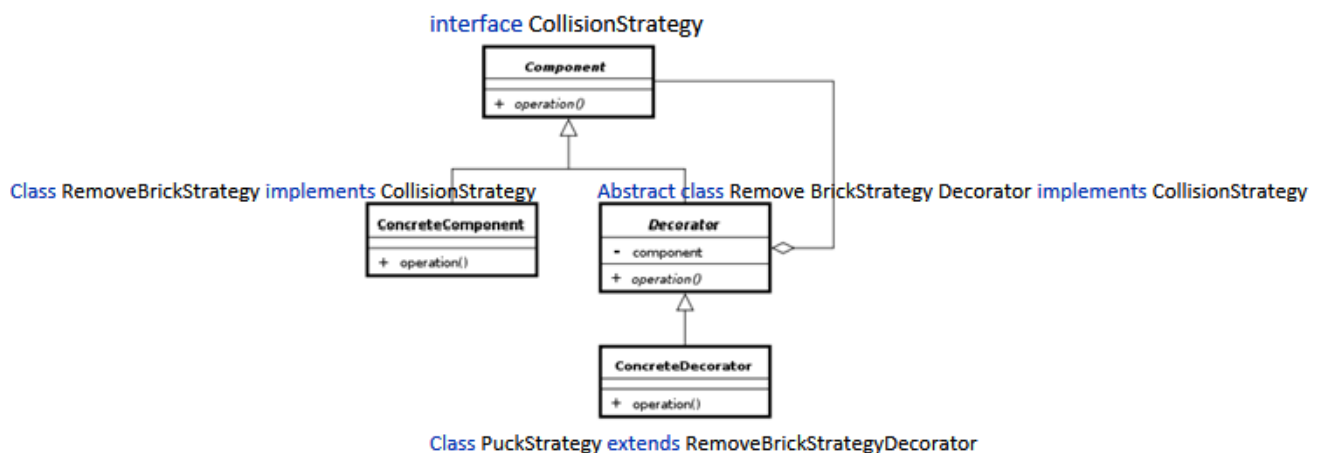
## מחלקות למימוש

ממשו את ה-API הנתון לכם כאן:

[https://yoavwoller.github.io/HUJI-OOP-22-ex3\\_2-javadoc/](https://yoavwoller.github.io/HUJI-OOP-22-ex3_2-javadoc/)

## פרטי מימוש

לצורך כתיבת ההתנהגויות השונות של הלבנים דן משתמש באתר בתבנית strategy. בתרגיל אנחנו נוסיף למימוש גם שימוש בתבנית decorator עליה למדנו בתרגולים. הסכמה הכללית של המימוש שאנו מבקשים משתקף ב-API של מחלקות החבילה brick\_strategy, וגם בשמות המחלקות.



From: Wikipedia, Decorator Pattern

שאלה: אם מימשנו בעזרת decorator pattern, מהי הדרך הטבעית לתת ללבנה שתי התנהגויות?

## אופן ההגשה

1. הגישו את התרגיל כקובץ jar. בשם ex3\_2.jar.
2. בתוך קובץ זה, ימצאו הקבצים הבאים:
  1. הספריות src ו-algo\_questions, בעלות המבנה והקבצים המתוארים כאן:

```
./src/  
./src/BrickerGameManager.java  
./src/brick_strategies  
./src/brick_strategies/AddPaddleStrategy.java  
./src/brick_strategies/BrickStrategyFactory.java  
./src/brick_strategies/ChangeCameraStrategy.java  
./src/brick_strategies/CollisionStrategy.java  
./src/brick_strategies/PuckStrategy.java  
./src/brick_strategies/RemoveBrickStrategy.java  
./src/brick_strategies/RemoveBrickStrategyDecorator.java  
./src/gameobjects  
./src/gameobjects/Ball.java  
./src/gameobjects/BallCollisionCountdownAgent.java  
./src/gameobjects/Brick.java  
./src/gameobjects/GraphicLifeCounter.java  
./src/gameobjects/MockPaddle.java  
./src/gameobjects/NumericLifeCounter.java  
./src/gameobjects/Paddle.java  
./src/gameobjects/Puck.java  
./algo_questions/  
./algo_questions/Solutions.java
```

2. קבצי מימוש (קבצי java). נוספים לצרכי המימוש שלכם. כל קבצי המימוש של משחק הבריקר צריכים להימצא בתוך הספרייה src שאתה מגשים, ולשמור על מבנה חבילות/ספריות תקין.
3. קובץ בשם README. ראו פירוט לגביו למטה.

5. מבנה קובץ ה-README :

1. שורה ראשונה: שם מלא.
2. שורה שניה: מספר ת"ז.
3. שורה שלישית: ריקה
4. בשורה רביעית והלאה - פירוט לגבי המימוש שלכם:
  - א. הסבירו כיצד תכנתם כל התנהגות שממשתם – רשימות מחלקות, תבניות עיצוב שהמחלקות מממשות, בין אם בעקבות החלטכם ובין אם בעקבות התכנון של DanoGameLab.
  - ב. אם מדובר בתבנית עיצוב שאתם בחרתם – מדוע בחרתם בתבניות עיצוב אלו? מדוע היה עדיף להשתמש בהן מאשר לתכנן את הקוד אחרת.
  - ג. הסבר קצר על האלגוריתמים שהגשמתם, מחולק לסעיפים הבאים (כל סעיף בנפרד)
    - i. הסבר נכונות תמציתי וקצר

ii. ניתוח זמן ריצה תמציתי וקצר  
ד. כל מידע נוסף שהייתם רוצים להוסיף ולפרט.

בהצלחה!