

**תרגיל 7 – תכנות פונקציונאלי ומבני נתונים**

תאריך פרסום: 24.12.2020

תאריך הגשה: 03.01.2021

מתרגל אחראי: ישעיה צברי

הנחיות כלליות:

- העבודה תבוצע ביחידים.
- קראו את ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- מומלץ לקרוא את כל העבודה לפני תחילת הפתרון.
- כתבו תיעוד (הערות) שמסביר את הקוד שלכם. אין לכתוב הערות בעברית.
- עליכם להוריד את הקובץ "hw7.py" מתיקיית "תרגיל בית 7" מהמודל, ולהכניס את הקוד שלכם בשורות המתאימות.
- אין להשתמש בחבילות או במודולים חיצוניים, אלא אם נאמר במפורש.
- **ניתן להניח שהקלט תקין**, אלא אם נכתב אחרת בשאלה.
- העבודה תיבדק באופן אוטומטי ולכן על הפלטים להיות בדיוק כפי שמוגדר בתרגיל (ללא רווחים מיותרים בהדפסות).
- העתקת קוד (משנים קודמות, מחברים או מהאינטרנט) עלולה להוביל כישלון בקורס. **אל תעתיקו!**
- שאלות בנוגע לעבודה ישאלו ב-"פורום שאלות לתרגיל בית 7" במודל.
- את העבודה יש להגיש דרך מערכת ההגשה בכתובת :  
<https://subsys.ise.bgu.ac.il/submission/login.aspx>
- טרם ההגשה אנא וודאו:
  - כל אחד מהקבצים מתקמפל ורץ כנדרש.
  - המשתנים שכתבתם עם שמות משמעותיים (ללא שמות כמו a).
  - אין הדפסות מיותרות (למשל הרצה של טסטים).
- מטרות העבודה: תרגול פונקציות מסדר גבוה ומחסנית.
- מדריך קצר על ביטויי lambda –  
<https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>
- מדריך קצר על High Order Functions (HOF) -  
<https://www.codingame.com/playgrounds/1434/introduction-to-functional-programming-with-python/built-in-higher-order-functions-filter-map-and-reduce>

**בהצלחה!**

בעבודה זו נמשיך לעזור לעדי וישי מעבודה מס' 4 בתחזוקת החצר. לאחר שכלל הצמחים נבחרו ונשתלו, עדי וישי החליטו שע"מ לשמור על החצר במצב מיטבי, עליהם לפתח מערכת חכמה להשקיית הצמחים.

קראו בעיון את תיאור את שתי המחלקות בטבלה, מימוש שתי המחלקות מצורף לעבודה זו בקובץ hw7.py. אין להוסיף שיטות או שדות מלבד המצוינים בטבלה. המחלקות ישמשו אתכם לבדיקה של הסעיפים שבהמשך השאלה.

שם מחלקה – Plant המייצגת צמח					
שיטות			שדות		
שם	סוג	ערך ברירת מחדל	חתימה	קלט	פלט (ערך החזרה)
name	str	-	__init__(self, name, aesthetics, water_consumption_month, average_month_yield)	בנאי המחלקה, שדות המחלקה כמתוארים בטבלה.	-
aesthetics – the aesthetic value the plant provides to the owner.	int	1	get_maintenance_cost(self, func1)	משתנה func1 מסוג פונקציה	הפלט של הפונקציה func1 הוא הפלט של func1 בהינתן הקלט self.
water_consumption_month – the amount of water (litres) the plant consumes each month.	int	1	purchase_decision(self, func1, func2)	משתנה func1 מסוג פונקציה.	הפלט של func1 הוא
average_month_yield – the average amount of flowers the plant blooms monthly.	Int	1		משתנה func2 מסוג פונקציה.	הפלט של func2 בהינתן self כקלט.
Seasonal – whether the plant blossoms (and consumes water) for 6 or 12 months in a entire year.	boolean	False			

שם מחלקה – GardenManager – מייצגת מנהל גינה		
שיטות	שדות	
- בנאי המחלקה, <code>__init__(self, plants_in_garden, n)</code> שדות המחלקה כמתוארים בטבלה.	<code>plants_in_garden</code> <code>list</code>	-
הפלט של הפונקציה <code>func1</code> הוא הפלט של <code>func1</code> בהינתן הקלט <code>self</code> .	<code>action(self, func1)</code>	משתנה <code>func1</code> מסוג פונקציה

### חלק א' – ביטויי `lambda`

בכל סעיף בחלק זה תתבקשו לממש ביטוי `lambda` קצר, אותו עליכם לאחסן במשתנה בעל השם המתאים, ראו את קובץ `hw7.py` לדוגמה. על הביטוי להיות מאוחסן בפריים הגלובלי של הסקריפט פתרון שאתם מגישים (ולא בתוך מחלקה או פונקציה).

1. אחסנו במשתנה בשם `get_cost_lmbd` ביטוי `lambda` המקבל כקלט צמח ומחזיר את צריכת המים החודשית שלו.

2. אחסנו במשתנה בשם `get_yearly_cost_lmbd` ביטוי `lambda` המקבל כקלט צמח ומחזיר את צריכת המים השנתית של הצמח.

3. אחסנו במשתנה בשם `worth_investing_lmbd` ביטוי `lambda` המקבל כקלט צמח ומחזיר משתנה מסוג `bool` שערכו `True` אם כמות הפרחים שפורחים כל חודש גדולה **ממש** מצריכת המים החודשית שלו (השוואה היא כמות ליטרים מול כמות פרחים), ו-`False` אחרת.

4. הערך האסתטי משמעותי לעדי וישי יותר מכמות הפרחים של הצמח רק במעט, אחסנו במשתנה בשם `declare_purchase_lmbd` ביטוי `lambda` המקבל כקלט צמח ומשתנה בוליאני המציין האם הפרח "שווה" רכישה (על פי החלטה שרירותית של עדי/ישי שהתרחשה במשתלה ללא בדיקות מעמיקות) ומחזיר מחרוזת בפורמט הבא: "A:B", כאשר A הוא שם הפרח, ו B הוא "yes" אם הפרח "שווה" בהחלטה המקדימה (`worth=True`) או אם הפרח לא "שווה" בהחלטה המקדימה, אך הערך האסתטי של הצמח גדול או שווה לערך צריכת המים החודשית שלו. אחרת B הוא "no".

הדרכה: [בניית מחרוזות עם מתודת `format`](#)

5. אחסנו במשתנה בשם `get_plants_names_lmdb` ביטוי `lambda` המקבל מנהל גינה כקלט. הביטוי מחזיר רשימה ממויינת (לפי ערך `ascii`) של שמות הצמחים במנהל הגינה.

### חלק ב' – High Order Functions

שימו לב. בחלק זה אין להשתמש בלולאות ובפונקציות מובנות של פייתון למעט `filter`, `map`, ו-`reduce` (יבאו באמצעות `functools`), אין להשתמש ב-`list/dictionary/etc. comprehensions` גם כן.

1. ממשו פונקציה `retrospect` המקבלת מנהל גינה ומחזירה רשימה של שמות הצמחים עם כמות פרחים גדולה מעלות צריכת המים בליטרים.

2. ממשו פונקציה `get_total_yearly_cost` המקבלת מנהל גינה ומחזירה את ערך סכום צריכת המים השנתית המצטברת של הצמחים בגינה.

לדוגמה, בהינתן מופע של `GardenManager`:

```
GardenManager([Plant("Rose", 0, 1, 2, True), Plant("Hibiscus", 0, 5, 1)])
```

הפונקציה תחזיר את המספר 66.

3. ממשו פונקציה `get_aesthetics` המקבלת כקלט מנהל גינה ומחזירה רשימה של ערכם האסתטי של כל הפרחים בגינה.

לדוגמה, בהינתן מופע של `GardenManager`:

```
GardenManager([Plant("Tulip", 0, 1, 2), Plant("Sunflower", 8, 5, 1)])
```

הפונקציה תחזיר את הרשימה:

```
[0, 8]
```

בחלק ב' כולו ניתן להניח קלט תקין:

- קלט מספרי יתקבל לפי הדרישה המפורטת (ארגומנט שלם לא שלילי אכן יקבל ערך שלם ולא שלילי)
- רשימות המכילות אובייקטים מסוגים מסוימים יכילו את סוגי האובייקטים האלה בלבד.

## חלק ג' – תור

בכניסה לאוניברסיטה הקולנירית "הטחיניון" שער כניסה אוטומטי יחיד המאופיין בעומס גבוה בשעות הבוקר. על מנת לגזור קופון, ההנהלה פתחה מכירת כרטיסי סטודנט מיוחדים, המעניקים לכל סטודנט שרכש אותם קדימות בתור על פני כל סטודנט שלא רכש את הכרטיס.

עליכם לממש מחלקה בשם GateLine שתנהל את התור בהתאם לדרישות הבאות:

- a. מימוש המחלקה למעט השיטות המפורטות בסעיפים הבאים הוא כפי שאתם רואים לנכון, אתם יכולים להוסיף שדות ושיטות נוספות כפי שתמצאו ע"מ לעמוד בדרישות המפורטות בהמשך, עם זאת, עליכם לקבל ארגומנט בשם max\_capacity בבנאי המחלקה ולהכניס את ערכו לשדה של המופע תחת אותו השם (ראו דוגמה 1 בנספח).
- b. ממשו במחלקה שיטה בשם new\_in\_line המוסיפה סטודנט לתור (הפונקציה אינה מחזירה ערך). השיטה מקבלת שני פרמטרים, מחרוזת student\_id המייצגת את תעודת הזהות של הסטודנט (חד ערכית לכל סטודנט) ומשתנה בוליאני priority\_id\_holder המציין האם הסטודנט רכש תעודת קדימות. מכיוון שגודלו המקסימלי האפשרי של התור מוגבל (ע"י ערך השדה max\_capacity), המחלקה תוסיף את הסטודנט לתור באופן הבא:
  - i. אם התור לא מלא, הסטודנט יצטרף לתור.
  - ii. אם התור מלא, הסטודנט יוכל להצטרף לתור רק אם ברשותו כרטיס קדימות: ע"מ לפנות לו מקום בתור, המערכת מגרשת סטודנטים ללא כרטיס קדימות (לפי סדר הגעתם, האחרון שהגיע הוא הראשון להיות מגורש) עד שיתפנה מקום לסטודנט בעל כרטיס הקדימות החדש. אם אין מקום בתור לסטודנט החדש גם לאחר הסרת כל הסטודנטים שלא רכשו כרטיס קדימות, הסטודנט החדש מוכנס למרות זאת לתור (מגבלת המקום מופרת ללא כל משמעויות נוספות).
 ראו דוגמה 2 בנספח.
- c. ניתן להניח קלט תקין ושאינן שני סטודנטים עם אותה תעודת זהות.
  - i. ממשו במחלקה שיטה בשם open\_gate המחזירה את תעודת הזהות של הסטודנט הבא בתור להיכנס, ומסירה אותו מהתור. כניסת הסטודנטים בתור היא עפ"י מדיניות ההעדפה הבאה:
    - i. הסטודנט עם כרטיס הקדימות הוותיק ביותר בתור.
    - ii. הסטודנט הוותיק ביותר בתור.

בהינתן תור ריק – הפונקציה תחזיר את הערך None.

- d. ממשו במחלקה שיטה בשם is\_empty המחזירה ערך בוליאני המציין האם התור ריק. ראו דוגמה 3 בנספח.
- e. ממשו במחלקה שיטה בשם show\_who\_is\_in\_line המחזירה רשימה של תעודות הזהות של הסטודנטים בתור, לפי סדר היציאה העתידי (הראשון לצאת באינדקס 0, השני באינדקס 1 וכך הלאה) ראו דוגמה 4 בנספח. הניחו קלט תקין.

נספח א' - דוגמאות הרצה וקידוד:

– דוגמה 1

```
class GateLine:

    def __init__(self, max_capacity=2):

        self.max_capacity=max_capacity
```

– דוגמה 2

```
gl = GateLine(2)
gl.new_in_line(student_id='012345678', priority_id_holder=False)
print(gl.open_gate())
```

– דוגמה 3

```
gl = GateLine(2)
gl.new_in_line(student_id='012345678', priority_id_holder=False)
print(gl.open_gate())
print(gl.is_empty())
```

– דוגמה 4

```
gl = GateLine(1)
gl.new_in_line(student_id='012345678', priority_id_holder=False)
gl.new_in_line(student_id='123456789', priority_id_holder=True)
print(gl.show_who_is_in_line())
```

['123456789']

