

תרגיל בית 7

הנחיות כלליות:

- קראו בעיון את השאלות והקפידו שהתוכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ `ex7_123456789.py` המצורף לתרגיל, לאחר החלפת הספרות 123456789 במספר ת.ז. שלכם, כל 9 הספרות כולל, ספרת ביקורת.
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון).
- חלק מהשאלות נבדקות באופן אוטומטי. לכן, עליכם לרשום את הקוד שלכם אך ורק במקומות המתאימים לכך בקובץ השלד.
- ניתן להניח כי הקלט שמקבלות הפונקציות תקין (אלא אם נכתב אחרת).
- אין לשנות שמות אובייקטים (עצמים) או משתנים שקיימים בקובץ השלד של התרגיל.
- אין למחוק את ההערות שמופיעות בשלד.
- אין להשתמש בקריאה לספריות חיצוניות. אסור לעשות `import`.
- מועד אחרון להגשה: כמפורסם באתר.

שאלה 1

בשאלה זו נממש את המחלקה Beverage שתייצג משקה שניתן להזמין במסעדה.

כל אובייקט מסוג Beverage יכיל את השדות הבאים:

1. שדה name- שדה מסוג string המייצג את שם המשקה.
2. שדה price- שדה מסוג float המייצג את המחיר של המשקה.
3. שדה is_diet- שדה מסוג boolean המכיל True אם המשקה הוא דיאטטי ו- False אם לא.

א. ממשו את בנאי המחלקה שחתימתו היא:

```
__init__(self, name, price, is_diet):
```

הבנאי מקבל את הנתונים הדרושים ושומר אותם בשדות הרלוונטיים של האובייקט החדש שהוא מייצר.

- ניתן להניח כי טיפוס משתני הקלט הוא תקין והמחרוזות אינן ריקות.
- יש לוודא שהמשתנה price גדול מ-0, אחרת יש להעלות חריגה מסוג ValueError עם הודעה לבחירתכם.

ב. למחלקת Beverage, ממשו את המתודה:

```
get_final_price(self, size):
```

המתודה (Method) תחזיר את המחיר של המשקה כתלות בגודל המבוקש. המשתנה size הינו מסוג string ויכול לקבל את אחת משלושת המחרוזות הנ"ל: 'Normal', 'Large', 'XL'.

- במידה וגודל המשקה מצויין כ 'Large', יש להחזיר את המחיר הרגיל (שהגדרנו בסעיף א').
- במידה וגודל המשקה מצויין כ 'XL', יש להחזיר 125% מהמחיר הרגיל.
- במידה וגודל המשקה מצויין כ 'Normal', יש להחזיר 75% מהמחיר הרגיל.
- במידה והקלט שונה מהמחרוזות הנ"ל, יש להעלות חריגת ValueError עם הודעה לבחירתכם.
- נרצה לתמוך בקריאה של המתודה בלי לציין את גודל המשקה. כלומר, עם ערך ברירת מחדל. קריאה לפונקציה בלי פרמטר של גודל, תתנהג כאילו הוכנס 'Large'. מותר לצורך זה לשנות את הגדרת הפונקציה, אך לא את שמה.

פייתון למהנדסים 0509-1820 , סמסטר א' תש"ף 2019

דוגמאות הרצה:

הרצה תקינה: שימו לב שאכן חוזר 125% של 15

```
In[2]: diet_coke = Beverage("Coca Cola", 15, True)
In[3]: price = diet_coke.get_final_price('XL')
In[4]: price
Out[4]: 18.75
```

הרצה בלי משתנה גודל:

```
In[5]: diet_coke = Beverage("Coca Cola", 15, True)
In[6]: price = diet_coke.get_final_price()
In[7]: price
Out[7]: 15
```

הרצה עם מחיר שלילי:

```
In[9]: pepsi = Beverage("Pepsi", -10, True)
```

```
ValueError: Price must be greater than 0
```

הרצה עם size שגוי:

```
In[10]: pepsi = Beverage ("Pepsi", 10, False)
In[11]: pepsi.get_final_price('Small')
```

```
ValueError: invalid size
```

שאלה 2

בשאלה זו נממש את המחלקה Pizza שתייצג פיצה שניתן להזמין במסעדה.

כל אובייקט מסוג Pizza יכיל את השדות הבאים:

1. שדה name - שדה מסוג string המייצג את שם הפיצה.
2. שדה price - שדה מסוג float המייצג את המחיר של הפיצה.
3. שדה calories - שדה מסוג int המייצג את מספר הקלוריות בפיצה.
4. שדה toppings - שדה מסוג list באורך כלשהו המכיל את שמות התוספות על הפיצה כאלמנטים רשימה (כל אלמנט הינו מסוג string).

א. ממשו את בנאי המחלקה שחתימתו היא:

`__init__(self, name, price, calories, toppings):`

הבנאי מקבל את הנתונים הדרושים ושומר אותם בשדות הרלוונטיים של האובייקט החדש שהוא מייצר.

- ניתן להניח כי טיפוסים משתנים תקינים, הרשימה איננה ריקה והמחרוזות אינן ריקות.
- יש לוודא שהמשתנים price ו-calories גדולים מ-0, אחרת יש להעלות חריגה מסוג ValueError עם הודעה לבחירתכם.

ב. למחלקת Pizza, ממשו את המתודה:

`get_final_price(self, size):`

המתודה (Method) תחזיר את המחיר של הפיצה כתלות בגודל המגש המבוקש. המשתנה size הינו מסוג string ויכול לקבל את אחת משלושת המחרוזות הנ"ל: 'XL', 'Family', 'Personal'.

- אם גודל מגש הפיצה מצויין כ 'Family', יש להחזיר את המחיר הרגיל (שהגדרנו בסעיף א').
- אם גודל מגש הפיצה מצויין כ 'XL', יש להחזיר 115% מהמחיר הרגיל.
- אם גודל המשקה מצויין כ 'Personal', יש להחזיר 60% מהמחיר הרגיל.
- במידה והקלט שונה מהמחרוזות הנ"ל, יש להעלות חריגת ValueError עם הודעה לבחירתכם.
- נרצה לתמוך בקריאה של המתודה בלי לציין את גודל הפיצה. כלומר, עם ערך ברירת מחדל. קריאה לפונקציה בלי פרמטר של גודל, תתנהג כאילו הוכנס 'Family'. מותר לצורך זה לשנות את הגדרת הפונקציה, אך לא את שמה.

ג. ללקוחות בררניים, נרצה להוסיף אפשרות לערוך את התוספות שעל הפיצה.

1. ממשו את המתודה:

`add_topping(self, topping, calories, price):`

המקבלת את שם התוספת (`topping`) כ- `string`, מספר קלוריות של התוספת (`calories`) כ- `int`, ואת המחיר של התוספת (`price`) כ- `float`.
המתודה תצרף את התוספת החדשה לרשימת התוספות הקיימות על הפיצה, תוסיף את המחיר של התוספת החדשה למחיר של הפיצה, ותוסיף את מספר הקלוריות של התוספת החדשה למספר הקלוריות הכולל של הפיצה.

- אם התוספת כבר מופיעה ברשימת התוספות שעל הפיצה, יש להעלות הודעת חריגה `ValueError` עם הודעה לבחירתכם.
- ניתן להניח שהמחרוזת `topping` איננה ריקה, ושהערכים של הקלוריות והמחירים של התוספות החדשות הם מספרים הגדולים מ-0.

2. ממשו את המתודה:

`remove_topping(self, topping, calories, price):`

המקבלת את שם התוספת (`topping`) כ- `string`, מספר קלוריות של התוספת (`calories`) כ- `int`, ואת המחיר של התוספת (`price`) כ- `float`.
המתודה תוריד את התוספת המבוקשת מרשימת התוספות הקיימות על הפיצה, תוריד את המחיר של התוספת המבוקשת מהמחיר של הפיצה, ותוריד את מספר הקלוריות של התוספת המבוקשת ממספר הקלוריות הכולל של הפיצה.

- אם התוספת איננה מופיעה ברשימת התוספות שעל הפיצה, יש להעלות הודעת חריגה `ValueError` עם הודעה לבחירתכם.
- ניתן להניח שהמחרוזת `topping` איננה ריקה, ושהערכים של הקלוריות והמחירים של התוספות החדשות הם מספרים הגדולים מ-0.
- לבסוף, יש להעלות הודעת חריגה `ValueError` עם הודעה לבחירתכם, במידה וכמות הקלוריות החדשה של הפיצה או המחיר החדש של הפיצה, קטן או שווה ל-0.
- שימו לב: יש חשיבות לאותיות גדולות\קטנות. כלומר, לצורכי השאלה "Tomato" ו-"tomato" הם מרכיבים שונים.

פייתון למהנדסים 0509-1820 , סמסטר א' תש"ף 2019

דוגמאות הרצה:

הרצה תקינה:

```
In[12]: personal_pizza = Pizza('Four Cheese', 50, 1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
In[13]: personal_pizza.get_final_price('Personal')
Out[13]: 30.0
In[14]: personal_pizza.remove_topping('Gouda', 100, 10)
In[15]: personal_pizza.get_final_price('Personal')
Out[15]: 24.0
In[16]: personal_pizza.add_topping('Feta', 120, 9)
In[17]: personal_pizza.get_final_price('Personal')
Out[17]: 29.4
In[18]: personal_pizza.toppings
Out[18]: ['Mozzarella', 'Roquefort', 'Goat Cheese', 'Feta']
In[19]: personal_pizza.calories
Out[19]: 1220
```

הרצה עם מחיר שלילי:

```
In[25]: personal_pizza = Pizza('Four Cheese', -50, 1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
```

```
ValueError: Price must be greater than 0
```

הרצה עם קלוריות שליליות:

```
In[26]: personal_pizza = Pizza('Four Cheese', 50, -1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
```

```
ValueError: Calories must be greater than 0
```

ניסיון להוסיף תוספת קיימת:

```
In[2]: personal_pizza = Pizza('Four Cheese', 50, 1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
In[3]: personal_pizza.add_topping('Gouda', 120, 9)
```

```
ValueError: Four Cheese already contains Gouda
```

ניסיון להוריד תוספת בעלת יותר קלוריות מסך הקלוריות של הפיצה:

```
In[4]: personal_pizza = Pizza('Four Cheese', 50, 1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
In[5]: personal_pizza.remove_topping('Gouda', 10000, 10)
```

```
ValueError: remaining calories must be greater than 0
```

פייתון למהנדסים 0509-1820 , סמסטר א' תש"ף 2019

ניסיון להוריד תוספת אשר אינה קיימת בתוספות שעל הפיצה:

```
In[2]: personal_pizza = Pizza('Four Cheese', 50, 1200, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
In[3]: personal_pizza.remove_topping('Goudax', 100, 10)
ValueError: Four Cheese does not contain Goudax
```

שאלה 3

בשאלה זו נממש את המחלקה Meal שתייצג ארוחה שניתן להזמין במסעדה.

כל אובייקט מסוג Meal יכיל את השדות הבאים:

1. שדה beverage - שדה מסוג Beverage המייצג את המשקה בארוחה.

2. שדה pizza שדה מסוג Pizza המייצג את הפיצה בארוחה.

א. ממשו את בנאי המחלקה שחתימתו היא:

```
__init__(self, beverage, pizza):
```

הבנאי מקבל את הנתונים הדרושים ושומר אותם בשדות הרלוונטיים של האובייקט החדש שהוא מייצר.

- ניתן להניח כי משתני הקלט הם תקינים.

ב. ממשו את המתודה:

```
get_final_price(self, beverage_size, pizza_size):
```

המקבלת beverage_size כגודל המשקה כ- string, ואת pizza_size כגודל הפיצה כ- string. המתודה מחזירה את הסכום של מחיר המשקה ומחיר הפיצה.

- שימו לב: יש להשתמש במתודות get_final_price שהגדרנו למשקה ולפיצה, בשאלות 1 ו- 2 בהתאמה, לקבלת הסכום.
- על מנת למנוע בלבול של הלקוחות, במתודה הזאת לא נגדיר ערכי ברירת מחדל. בניגוד למה שעשינו במתודות get_final_price הקודמות שהגדרנו בשאלות 1 ו- 2.

ג. ממשו את המתודה:

```
is_healthy(self):
```

אשר מחזירה משתנה boolean במידה והארוחה בריאה. נגדיר ארוחה בריאה כארוחה המכילה משקה דיאטטי ובעלת פחות מ 1000 קלוריות בסך הכל.

פייתון למהנדסים 0509-1820 , סמסטר א' תש"ף 2019

דוגמאות הרצה:

הרצה תקינה:

```
In[20]: personal_pizza = Pizza('Four Cheese', 50, 900, ['Mozzarella', 'Gouda', 'Roquefort', 'Goat Cheese'])
In[21]: diet_coke = Beverage("Coca Cola", 15, True)
In[22]: meal = Meal(diet_coke, personal_pizza)
In[23]: meal.is_healthy()
Out[23]: True
In[24]: meal.get_final_price('Large', 'Family')
Out[24]: 65
In[25]: meal.get_final_price('XL', 'Personal')
Out[25]: 48.75
```

הארוחה מוגדרת כבריאה במקרה זה, כי יש בה פחות מ- 1000 קלוריות (900) והמשקה הוא דיאטטי.

ניסיון הרצה עם גודל לא נכון:

```
In[26]: meal.get_final_price('XXXXL', 'Personal')
```

```
ValueError: invalid size
```

אנא שימו לב שהחריגה כאן נובעת מהעובדה שהגדרנו אותה בשאלה 1, ב- Beverage.get_final_price.