**10806 – תכנות ותיכון מונחה עצמים פיתון – הנדסת תעו"נ – תרגיל בית מספר 2**

1.

יש לכתוב תוכנית בפיתון המכילה מחלקה בשם ShoppingCart (עגלת קניות). תכונות המחלקה הן: items – list המכיל את המוצרים בעגלת הקניות, כאשר כל מוצר הוא tuple המכיל רביעייה: שם המוצר, כמות מאותו מוצר, מחיר ליחידה, וקוד זיהוי של המוצר, total – סכום הקניה של המוצרים בעגלת הקניות, shopper – שם הקונה, address – כתובת למשלוח המוצרים שנקנו.

כול תכונות המחלקה צריכות להיות מסוג private.

מתודות המחלקה:

\_\_init\_\_ : מתן ערך לאובייקט מסוג ShppingCart. מקבלת כפרמטר רק את שם הקונה.  
  
add\_item : מוסיפה מצרך לעגלת הקניות. מקבלת כפרמטר tuple שמכיל את רביעיית המאפיינים של המוצר (שם, כמות, מחיר ליחידה, קוד זיהוי). המתודה מוסיפה את המוצר לעגלת הקניות ומעדכנת את סכום הקנייה של המוצרים בעגלת הקניות. יכולות להיות רביעיות זהות.  
  
delete\_item : מורידה מצרך מעגלת הקניות. מקבלת כפרמטר tuple שמכיל את רביעיית המאפיינים של המוצר (שם, כמות, מחיר ליחידה, קוד זיהוי). אם המצרך אינו בעגלה תצא הודעה מתאימה. אם המצרך בעגלה הוא יוסר ממנה ויתעדכן סכום הקנייה של המוצרים בעגלת הקניות.  
  
print\_cart: מדפיסה את עגלת הקניות (יש לראות דוגמת פלט בהמשך).

set\_address: מקבלת כתובת למשלוח המוצרים ומעדכנת בהתאם את האובייקט.

get\_address: מחזירה את הכתובת למשלוח המוצרים.

\_\_str\_\_: מחזירה מחרוזת שהיא ההדפסה של print\_cart.  
  
הוסף אחרי המחלקה את השורות הבאות:

**def** demo():  
 *"""Demonstrate use of class"""* r = ShoppingCart(**'Rich'**)  
 r.add\_item((**'aa'**, 3, 23, 123))  
 r.add\_item((**'bb'**, 2, 13, 246))  
 r.add\_item((**'cc'**, 2, 13, 246))  
 r.delete\_item((**'cc'**, 1, 1, 1))  
 r.delete\_item((**'cc'**, 2, 13, 246))  
 r.set\_address(**'123 Main, Somwhere, AZ'**)  
 print(**"from print\_cart:"**)  
 r.print\_cart()  
 print(**"from \_\_str\_\_(self)"**)  
 print(r)

כתוצאה מהן התוכנית תדפיס את הפלט הבא:

Delete Error: item ('cc', 1, 1, 1) is not in cart.

from print\_cart:

Shipping: 123 Main, Somwhere, AZ

Cart:

item name: aa quantity: 3 unit price: 23 item ID: 123

item name: bb quantity: 2 unit price: 13 item ID: 246

total payment: 95

from \_\_str\_\_(self)

Shipping: 123 Main, Somwhere, AZ

Cart:

item name: aa quantity: 3 unit price: 23 item ID: 123

item name: bb quantity: 2 unit price: 13 item ID: 246

total payment: 95

Process finished with exit code 0

2.

Create a class named player that initiates a player with default name as player, default level as 0, and default character type as "Default". Make two instances with random levels 1-n. n is an input: integer from 1 to 99. You have to force the user to enter correct n, by try-expect (ValueError). Create a function (name of function: battle) that compares two players levels and prints the players name, level, and character type that has the higher level. If both characters are the same level print "tie". All class attributes must be private.

These are the last rows of the main function:

player1 = player("Player1", random.randint(1, n), "Mage")

player2 = player("Player2", random.randint(1, n))

print()

print(player1, "\n\n", player2, "\n", sep="")

battle(player1, player2)

This is the sample output after running:

Enter an integer from 1 to 99:

r

number must be integer and from 1 to 99. the wrong values is: invalid literal for int() with base 10: 'r'

Enter an integer from 1 to 99:

5.5

number must be integer and from 1 to 99. the wrong values is: invalid literal for int() with base 10: '5.5'

Enter an integer from 1 to 99:

0

number must be integer and from 1 to 99. the wrong values is: 0

Enter an integer from 1 to 99:

101

number must be integer and from 1 to 99. the wrong values is: 101

Enter an integer from 1 to 99:

44

Name: Player1

Level: 24

Type: Mage

Name: Player2

Level: 7

Type: Default

Player1 Wins.

Process finished with exit code 0

3.

תוכנית שמשחקת את המשחק אבן, נייר ומספריים – אדם מול מחשב.

choices = ["rock", "paper", "scissors"]

יש להזין שם של אדם, ואחריו יש להזין שם של מחשב. האדם בוחר כתשובה לשאלה אפשרות מתוך choices, והמחשב מגריל אפשרות אחת מתוך choices. התוכנית תשאל את האדם לגבי בחירתו כול עוד לא יזין בחירה נכונה.

המשחק משוחק 3 פעמים שלאחריהם תצא הודעה מי ניצח האדם או המחשב - זה שזכה יותר פעמים. תצא גם הודעה אחרי כול משחק מתוך ה-3 מי זכה בו, האדם או המחשב.

פונקציית main מבקשת מהמשתמש את שם האדם ואת שם המחשב המשחקים, יוצרת אובייקטים מתאימים, ובלולאה בת 3 איטרציות מבקשת מהמשתמש את בחירת האדם המשחק – הפעלת מתודה במחלקת Human, מגרילה את בחירת המחשב – מתודה במחלקת Computer, מבררת מי ניצח באותו משחק, ומעדכנת את הניקוד באובייקט המתאים (Human או Computer). ביציאה מהלולאה מוכרז המנצח.

מחלקות:

מחלקת התחרות בשם Contestant. במחלקה זו יש שתי תכונות: שם (שם האדם או שם המחשב שמשחקים), וניקוד – כל פעם שהאדם או המחשב מנצחים, ערך תכונה זו באובייקט עולה ב-1. מהמחלקה הזו לא יוצרים אובייקט אלא רק מבניה, היא קיימת בשל ההורשה – כדי לא להכפיל תכונות ומתודות.

מחלקת האדם המשחק בשם Human שיורשת מהמחלקה Contestant. במחלקה הזו מבקשים מהאדם את הבחירה שלו ומחזירים אותה אחרי שמוודאים שזו בחירה תקנית.

מחלקת המחשב בשם Computer בה מגרילים עבור המחשב המשחק בחירה.

דוגמא לריצת התוכנית:

Enter name of human: avi levi

Enter name of computer: moon 3

avi levi, enter your choice: paper

moon 3 chooses rock

avi levi: 1 moon 3: 0

avi levi, enter your choice: rock

moon 3 chooses rock

avi levi: 1 moon 3: 0

avi levi, enter your choice: paper

moon 3 chooses rock

avi levi: 2 moon 3: 0

avi levi WINS

Process finished with exit code 0