תרגיל בית 8

OOP 2

הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ ex8_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת.
- מועד אחרון להגשה: כמפורסם באתר.
- 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 וכי התוכנית אינה קורסת).
- היות ובדיקת התרגילים עשויה להיות אוטומטית, יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- אין למחוק את ההערות שמופיעות בשלד.

בכל השאלות ניתן להניח את תקינות הקלט על פי המפורט בשאלה

שאלה 1

בשאלה זו נכתוב תוכנה לניהול נתוני סטודנטים באוניברסיטה יוקרתית.

סעיף א'

ממשו את המחלקה **Student** אשר מייצגת סטודנט באוניברסיטה. המחלקה כוללת את המתודות הבאות:

1. **__init__(self, name, id, courses)**

מתודת בנאי זו תאתחל אובייקט חדש הכולל את השדות הבאים:

- **name** – מחרוזת (str) המייצגת את שם הסטודנט.
- **id** – מחרוזת (str) המייצגת את מספר תעודת הזהות של הסטודנט (ניתן להניח שהמחרוזת מכילה רק ספרות)
- **courses** - מילון (dict) הממפה כל שם קורס שהסטודנט לקח (str) ל-tuple המכיל שני נתונים: הראשון (אינדקס 0) הוא מספר נקודות הזכות של הקורס (int גדול מ-0) והשני (אינדקס 1) הוא הציון של הסטודנט בקורס (int). יש לבדוק את תקינות ערכי הציונים: במקרה ואחד הציונים קטן מ-0 או גדול מ-100 יש לזרוק שגיאה מסוג ValueError עם הודעה לבחירתכם.

✓ ניתן להניח שהמילון מכיל לפחות רשומה אחת.

2. **__repr__(self)**

מתודה זו תחזיר מחרוזת המייצגת את האובייקט על פי דוגמת הפלט המוצגת בהמשך. המחרוזת תכיל בשורות נפרדות את שם הסטודנט, תעודת הזהות שלו, ורשימת הקורסים שהסטודנט לקח כאשר הקורסים ממוינים ע"פ שמם באופן אלפביתי (לקסיקוגרפי), בסדר עולה.

✓ שימו לב שבמחרוזת שהמתודה מחזירה ישנו תו רווח בודד לאחר כל סימן נקודתיים, וכן שברשימת הקורסים ישנו תו רווח בודד לאחר כל שם קורס, מספר נקודות זכות וציון.

3. **get_average(self)**

מתודה זו תחזיר את הממוצע המשוקלל של ציוני הסטודנט, שהינו סכום של מכפלות הציונים במספר נקודות הזכות של כל קורס, חלקי סכום מספר נקודות הזכות של כל הקורסים. ראו דוגמת חישוב בהמשך.

דוגמא לשימוש במחלקה:

```
>>> A = Student('Or','123456789', {'calculus':(7,80),
'algebra':(7,90), 'programming':(4,100)})
>>> A
Name: Or
Id: 123456789
Courses list: algebra 7 90 calculus 7 80 programming 4 100
>>> A.get_average()
88.33333333333333
```

הערך המוחזר 88.33 הינו תוצאת חישוב הממוצע המשוקלל של ציוני הסטודנט:

`(7*90+7*80+4*100)` לחלק לסכום נקודות הזכות של הקורסים `(7+7+4)`

סעיף ב'

ממשו את המחלקה **GradStudent** המייצגת סטודנט לתואר מתקדם. המחלקה תכיל את כל שדות המחלקה **Student** בנוסף לשדה מסוג מחרוזת בשם degree המציין את התואר שאליו לומד הסטודנט. על המימוש להיעשות בהינתן שקיים בידכם הקוד של סעיף א'.
ניקוד מלא יינתן לפתרונות שיעשו שימוש נכון בתכנות מונחה עצמים לצורך צמצום שכפול קוד.

על המחלקה לכלול את המתודות הבאות:

1. **__init__(self, name, id, courses, degree)**

מתודת בנאי זו תאתחל אובייקט חדש הכולל את השדות name, id, courses (באופן דומה לסעיף א', כולל ווידוא תקינות הציונים), בנוסף לשדה בשם degree שהינו מחרוזת המציינת את התואר שאליו לומד הסטודנט (ניתן להניח שערך הפרמטר שנשלח לבנאי עבור שדה זה יהיה תמיד 'msc' או 'phd').

2. **__repr__(self)**

מתודה זו תחזיר מחרוזת המייצגת את האובייקט בדומה לייצוג של אובייקט Student, כאשר בסופה שורה נוספת בה יצוין התואר שאליו לומד הסטודנט. ראו דוגמא לפלט בהמשך.

3. **get_average(self)**

מתודה זו תחזיר את הממוצע המשוקלל של הסטודנט. הממוצע המשוקלל של סטודנט מתקדם יחושב בדומה למתודה בסעיף א', אך כולל גם בונוס התלוי בתואר אליו לומד הסטודנט:

- עבור סטודנט הלומד לתואר שני ('msc' בשדה degree), המתודה תחזיר את ממוצע ציוניו המשוקלל, כפול 1.05.
- עבור סטודנט הלומד לתואר שלישי ('phd' בשדה degree), המתודה תחזיר את ממוצע ציוניו המשוקלל, כפול 1.15.
- <u>הממוצע המשוקלל לא יכול להיות גדול מ-100.0. עבור סטודנט שממוצע ציוניו המשוקלל לאחר הבונוס גדול מ-100.0, יוחזר הערך 100.0.</u>

<u>דוגמא לשימוש במחלקה</u>:

```
>>> B = GradStudent('Guy','987654321', {'calculus':(7,84),
'quantummechanics':(4,95), 'imageprocessing':(4,90) }, 'msc')
>>> B
Name: Guy
Id: 987654321
Courses list: calculus 7 84 imageprocessing 4 90
quantummechanics 4 95
Degree: msc
>>> B.get_average()
92.96000000000001
```

שימו לב שערך זה הוא הציון המשוקלל של הסטודנט (המחושב באופן דומה לחישוב המוצג בדוגמא של סעיף א'), כפול 1.05.

```
>>> C = GradStudent('Dan','192837465', {'phdseminar':(4,95),
'programming':(4,90)}, 'phd')
```

```
>>> C
Name: Dan
Id: 192837465
Courses list: phdseminar 4 95 programming 4 90
Degree: phd
>>> C.get_average()
100.0
```

הציון המשוקלל של הסטודנט יחד עם הבונוס (הציון המשוקלל כפול 1.15) הינו גדול מ-100.0, ולכן המתודה מחזירה 100.0.

סעיף ג'

ממשו את המחלקה **InternationalStudent** המייצגת סטודנט בינלאומי שלומד בפקולטה. המחלקה תכיל את כל השדות והמתודות של המחלקה **Student** אך מילון הקורסים שמקבל הבנאי יכיל ציונים אמריקאיים ולא ציונים מספריים.

על המימוש להיעשות בהינתן שקיים בידכם הקוד של סעיף א'. ניקוד מלא יינתן לפתרונות שיעשו שימוש נכון בתכנות מונחה עצמים לצורך צמצום שכפול קוד.

על המחלקה לכלול את המתודות הבאות:

4. **__init__(self, name, id, courses)**

מתודת בנאי זו תקבל 3 ארגומנטים באופן בדומה למחלקה Student, אך הציונים במילון courses (הערך השני בטאפל של ערכי המילון) יהיו מחרוזת המייצגת ציון אמריקאי (כאשר 'A' מקביל ל-100, 'B' מקביל ל-90, 'C' מקביל ל-80, 'D' מקביל ל-70 , ו-'F' מקביל ל-60).

במידה ואחד מהציונים במילון הקורסים איננו אחת מהאותיות הנ"ל יש להעלות שגיאת ValueError הכוללת את הודעת השגיאה 'Invalid American grade' .

<u>שימו לב</u>: חישבו כיצד לשמור את ערכי הארגומנט courses באובייקט המאותחל כך שהמרת הציונים מציון אמריקאי למספרי תתבצע רק פעם אחת לכל אובייקט.

5. **__repr__(self)**

מתודה זו תחזיר מחרוזת המייצגת את האובייקט בדומה לייצוג של אובייקט Student, כאשר הציונים הם אמריקאיים.

6. **get_average(self)**

מתודה זו תחזיר את הממוצע המשוקלל של הסטודנט (בדומה לנעשה במחלקה Student) לאחר שכל ציון אמריקאי הומר לציון מספרי לפי הפירוט שהובא בהגדרת הבנאי.

<u>דוגמא לשימוש במחלקה</u>:

```
>>> D = InternationalStudent('Nadav', '220376541',
{'calculus': (7, 'A'), 'babies 101': (2, 'C')})
>>> D
Name: Nadav
Id: 220376541
```

```
Courses list: babies 101 2 C calculus 7 A
>>> D.get_average()
95.55555555555556
```

שימו לב שערך זה הוא הציון המשוקלל של הסטודנט כפי שחושב במחלקה Student לאחר ש-A הומר ל-100, ו-C הומר ל-80.

סעיף ד'

לסיום, נגדיר מחלקה בשם **Faculty** אשר תייצג פקולטה באוניברסיטה. המחלקה תשמור רשימה של סטודנטים וכן את שמה של הפקולטה. על המחלקה לכלול את המתודות הבאות:

1. **__init__(self, name, students)**

מתודת בנאי זו תאתחל אובייקט חדש הכולל שדה בשם name (str) שישמור את שם הפקולטה, ושדה בשם students (list) שישמור רשימה של סטודנטים (אובייקטים מסוג Student או GradStudent). ניתן להניח שרשימת הסטודנטים אינה ריקה.

2. **__repr__(self)**

מתודה זו תחזיר מחרוזת המייצגת את האובייקט על פי דוגמת הפלט המוצגת בהמשך. ראשית יופיע שם הפקולטה, ולאחר מכן יופיעו כלל הקורסים שלקחו הסטודנטים בפקולטה. כל קורס יופיע רק פעם אחת, ולצידו יופיע מספר הסטודנטים מהפקולטה שלקחו קורס זה.

סדר הופעות הקורסים יהיה לפי מספר הסטודנטים שהשתתפו בכל קורס בסדר יורד. הקורס שיופיע ראשון הינו הקורס אותו לקחו הכי הרבה סטודנטים. עבור קורסים עם אותו מספר של סטודנטים שהשתתפו בהם, לא חשוב איזה קורס יופיע קודם.

- ✓ שימו לב שישנו רווח אחד לפני ואחד אחרי המקף שבין שם הקורס ומספר הסטודנטים שלקחו את אותו קורס, וירידת שורה לאחר ציון שם הפקולטה, ולאחר כל אחד מהקורסים.
- ✓ בשאלה זו ניתן להניח שהאותיות המופיעות במחרוזות הינן ב-lower case (אותיות קטנות).
- ✓

<u>דוגמא לשימוש במחלקה. (A, B, C,D) הם שלושת הסטודנטים שהוגדרו בסעיפים א',ב' ו-ג':</u>

```
>>> E = Faculty('engineering', [A,B,C,D])
>>> E
Faculty of engineering
calculus - 3 students
programming - 2 students
algebra - 1 students
quantummechanics - 1 students
imageprocessing - 1 students
phdseminar - 1 students
babies 101 - 1 students
```

שאלה 2

בשאלה זו נממש חלק ממערכת לניהול הזמנות עבור בית דפוס.
ניקוד מלא יינתן לפתרונות שיכללו תכנון נכון של המחלקות הנדרשות ויעשו שימוש נכון בתכנות מונחה עצמים לצורך צמצום שכפול קוד.

סעיף א'

בית הדפוס מאפשר לבצע הזמנה של שני סוגי הדפסות: פוסטר (Poster) ומכתב (Letter).

ממשו את המחלקות PosterOrder ו-LetterOrder על פי הפירוט הבא, **ותוך שימוש במחלקת בסיס משותפת (שיהיה עליכם להגדיר בעצמיכם)** שתאפשר מניעת שכפול קוד ככל שניתן.

PosterOrder

המחלקה **PosterOrder** תייצג הזמנה של הדפסת **פוסטר**. המחלקה תכלול את המתודות הבאות:

1. בנאי המחלקה __init__(self, client_name, copies, size) יקבל את הפרמטרים הבאים וישמור כל אחד מהם לשדה:
   1.1. client_name – מחרוזת המייצגת את שם הלקוח.
   1.2. copies – מספר שלם המציין את מספר העותקים.
   1.3. size – זוג מספרים שלמים (tuple) המציינים את אורך ורוחב הפוסטר במטרים.

   טיפול בשגיאות (מלבד בדיקות אלו ניתן להניח שהקלט תקין):
   - יש להעלות שגיאה מסוג ValueError (עם ההודעה "Invalid input") אם המחרוזת המציינת את שם הלקוח ריקה או אם מספר העותקים קטן מ-1.
   - יש להעלות שגיאה מסוג ValueError (עם ההודעה "Invalid poster size") אם הפרמטר size אינו באורך 2 או אם אחד מאיבריו קטן מ-1.
2. המתודה __repr__(self) תחזיר מחרוזת המייצגת את פרטי האובייקט על פי דוגמת הפלט שבהמשך.
3. המתודה calc_cost(self) תחזיר את עלות עבודת ההדפסה בשקלים על פי הנוסחה הבאה: מספר העותקים * שטח הפוסטר (אורך* רוחב) * 30 ש"ח.

LetterOrder

המחלקה **LetterOrder** תייצג הזמנה של הדפסת **מכתב**. המחלקה תכלול את המתודות הבאות:

1. בנאי המחלקה __init__(self, client_name, copies, paper_type, paper_prices) יקבל את הפרמטרים הבאים וישמור כל אחד מהם לשדה:
   1.1. client_name – מחרוזת המייצגת את שם הלקוח.
   1.2. copies – מספר שלם המציין את מספר העותקים.
   1.3. paper_type – מחרוזת המייצגת את סוג הדף.
   1.4. paper_prices – מילון הממפה לכל סוג דף (מחרוזת) את מחיר הדף (מספר שלם).

טיפול בשגיאות (מלבד בדיקות אלו ניתן להניח שהקלט תקין):

- יש להעלות שגיאה מסוג ValueError (עם ההודעה "Invalid input") אם המחרוזת המציינת את שם הלקוח ריקה או אם מספר העותקים קטן מ-1.
- יש להעלות שגיאה מסוג ValueError (עם ההודעה "Invalid paper type") אם המחרוזת paper_type אינה כלולה במפתחות המילון paper_prices.

2. המתודה __repr__(self) תחזיר מחרוזת המייצגת את פרטי האובייקט לפי דוגמת הפלט שבהמשך.
3. המתודה calc_cost(self) תחזיר את עלות עבודה ההדפסה בשקלים על פי הנוסחה הבאה: מספר העותקים * המחיר לדף על פי סוג הדף (כפי שמוגדר במילון paper_prices).

דוגמאות הרצה:

```
poster1 = PosterOrder ("Dvir", 1, (1, 1))
print (poster1)
print (poster1.calc_cost())
```

פלט:

```
Poster order: Client name: Dvir, Copies: 1, Size: (1, 1)
30
```

```
paper_prices = {"A4 80gr": 3, "A5 100gr": 5}
letter1 = LetterOrder ("Danielle", 1, "A4 80gr", paper_prices)
print (letter1)
print (letter1.calc_cost())
letter2 = LetterOrder ("Roy", 1, "A5 100gr", paper_prices)
print (letter2)
print (letter2.calc_cost())
letter3 = LetterOrder ("Roy", 5, "A5 100gr", paper_prices)
print (letter3)
print (letter3.calc_cost())
```

פלט:

```
Letter order: Client name: Danielle, Copies: 1, Paper type: A4 80gr
3
Letter order: Client name: Roy, Copies: 1, Paper type: A5 100gr
5
Letter order: Client name: Roy, Copies: 5, Paper type: A5 100gr
25
```

<u>שימו לב:</u> עליכם לכתוב מימוש למחלקות **PosterOrder** ו-**LetterOrder** כך שיפעלו בדיוק על פי ההוראות ודוגמאות ההרצה שהובאו לעיל.

התחילו במימוש על פי שיקול דעתכם של מחלקת בסיס בשם **PrintOrder** אשר לא תשמש אותנו ליצירה של אובייקטים בצורה ישירה (ולכן אינה חייבת לממש את כל המתודות שהוזכרו), אבל תאפשר מימוש של המחלקות הנ"ל תוך שימוש בירושה לצורך מניעת שכפול קוד <u>ככל שניתן</u>.

סעיף ב'

ממשו את המחלקה **PrintShop** שתייצג בית דפוס על פי הפירוט הבא:

1. בנאי המחלקה (self)__init__לא יקבל פרמטרים כלשהם, אך יאתחל את השדות הבאים:
   1.1. orders – רשימה ריקה שתחזיק אובייקטים מסוג PosterOrder או LetterOrder ותייצג את תור עבודות ההדפסה הממתינות לביצוע.
   1.2. revenues – שדה מסוג מספר שלם המאותחל ל-0 ומייצג את הכנסות בית הדפוס החל מרגע אתחול האובייקט.
2. המתודה add_order(self,order) תקבל אובייקט בשם order המייצג עבודת הדפסה ותוסיף אותו לסוף רשימת ההזמנות הממתינות לביצוע בשדה orders.
3. המתודה print_next_order(self) מבצעת את עבודת ההדפסה הבאה בתור (זו שהוכנסה הכי מוקדם לרשימת ההזמנות) על ידי הוספת עלות ההזמנה לשדה revenues והסרת ההזמנה מרשימת ההזמנות שמאוחסנת בשדה orders.
4. המתודה (self)__repr__ תחזיר מחרוזת המייצגת את פרטי אובייקט בית הדפוס כפי שמוצג בדוגמת ההרצה שבהמשך.

שימו לב: המחרוזת המוחזרת תפרט כמה הזמנות סה"כ ממתינות לביצוע ברשימה שבשדה orders <u>עבור כל לקוח</u> (אין חשיבות לסדר בו יופיעו שמות הלקוחות). היעזרו במילון!

דוגמת הרצה תוך שימוש באובייקטים שנוצרו בדוגמא של סעיף א':

```
print_shop = PrintShop()
print_shop.add_order(poster1)
print_shop.add_order(letter1)
print_shop.add_order(letter2)
print_shop.add_order(letter3)

print (print_shop)

print_shop.print_next_order() #prints poster1 whose cost is 30
print (print_shop)
```

הפלט:

```
Print shop orders:
Dvir : 1 orders
Danielle : 1 orders
Roy : 2 orders
Revenues: 0

Print shop orders:
Danielle : 1 orders
Roy : 2 orders
Revenues: 30
```

בהצלחה!