

עבודת בית מס' 4:

הנחיות כלליות:

- קראו את כל ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- קראו את כל העבודה לפני שתתחילו לפתור אותה.
- תאריך פרסום: 1.06.22
- תאריך הגשת התרגיל: 10.06.22 בשעה 23:59.
- יתאפשר איחורי הגשות בסך כולל של 5 ימים עבור כל התרגילים בסמסטר, לתרגיל שלא יוגש במסגרת זמן זה, יינתן ציון 0.
- כתבו תיעוד (הערות) שמסביר את הקוד שלכם. **אסור לכתוב הערות בעברית!**
- שאלות בנוגע לעבודה יישאלו בפורום המתאים במודל או בשעות הקבלה.
- את העבודה יש לכתוב בעורך הקוד המקובל בקורס (PyCharm) הגשת העבודה תתבצע באתר המודל בתיקייה הייעודית.
- **קבצי ההגשה יהיה קובץ מסוג .py. במידה ויש יותר מקובץ אחד, על הקבצים להיות מרוכזים בתוך תיקייה .zip. כששם התיקייה הוא תעודת הזהות של הסטודנט.**
- השימוש בחבילות מוכנות של פיתון **אסור בהחלט** ויגרור ציון 0.
- **על הפלטים להיות בדיק כפי שמוגדרים בשאלות (ללא רווחים מיותרים).**
- בעבודה זו ניתן להניח כי כל הקלטים תקינים כפי שהוגדרו בסעיפים הרלוונטיים, לכן אין צורך לבדוק תקינות קלט לשיטות ולבנאים אלא אם כן צוין אחרת במפורש.

בדיקות ידניות:

לתרגיל זה מצורפת תכנית ובה כלל קבצי המחלקות בנוסף למחלקת ה-main, במחלקת ה-main נכללו מספר בדיקות בסיסיות עבור הבנאים והשיטות שתממשו בעבודה – שימו לב כי הבדיקות יעבדו רק לאחר מימוש כלל המחלקות בעבודה.

אין לשנות את חתימת המחלקות, התכונות והפונקציות המצוינות בתרגיל.

הגשת התרגיל:

תיקיית ההגשה ובה כל הקבצים הרלוונטיים של המחלקות תוגש במודל ותיהיה קובץ מסוג .zip. כאשר שם הקובץ הינו תעודת הזהות של הסטודנט. לדוגמא: 123456789.zip

שאלה 1:

בשאלה זו תממשו בשפת python ייצוג תכנותי של בית ספר לקוסמים.

בית ספר לקוסמים וקוסמות (School) מחולק לבתים (House), כך שכל בית מייצג קבוצה נפרדת בתוך בית הספר בדומה לכיתות בבית ספר רגיל, כל בית מכיל בתוכו רשימת מכשפים (Wizard) המשויכים לאותו הבית.

לכל אחת מן המחלקות שתממשו (בית ספר, בית, מכשף ומכשף בוגר) יהיו תכונות, בנאים ופעולות משלהן אותן תצטרכו לממש בסעיפים הבאים.

סעיף א – מחלקת מכשף (Wizard):

מחלקת Wizard מייצגת מכשף והיא מוגדרת ע"י השדות הבאים:

- name – שם המכשף (מחרוזת) לדוגמא 'Harry Potter'
- house – הבית אליו שייך המכשף (מסוג House)
- in_dorm – האם המכשף נמצא במגורים כרגע (בוליאני), ברירת מחדל False.
- potions – מייצג את הציון של המכשף בקורס "שיקויים" (מספר שלם 0-100)
- charms – מייצג את הציון של המכשף בקורס "כישופים" (מספר שלם 0-100)

ממשו את הפעולות הבאות:

- `def __init__(self, name, house, potions, charms, in_dorm=False):`
 - בנאי המאתחל את כל השדות באובייקט מסוג מכשף.
- `def get_avg(self):`
 - שיטה המחזירה את ממוצע הציונים של המכשף כמספר עשרוני (float), לדוגמא: מכשף שציונו בקורס שיקויים הוא 75 ובקורס כישופים הוא 90, יוחזר הממוצע 82.5.
- `def enter_dorm(self, password):`
 - שיטה המקבלת כקלט סיסמא בצורת מחרוזת, בודקת האם הסיסמא (באובייקט בית) נכונה, במידה וכן, תשנה את הערך של `is_dorm` ל-True, אחרת תדפיס "wrong password, try again".
- `def exit_dorm(self):`
 - השיטה תוציא את המכשף מביתו, ע"י שינוי הערך של `is_dorm` ל-False.
- `def set_grade(self, course, grade):`
 - שיטה המקבלת כקלט מספר שלם המייצג את הקורס, כאשר 1 מייצג את הקורס שיקויים ו-2 מייצג את הקורס כישופים, ומספר שלם המייצג את הציון החדש באותו הקורס ומשנה את הציון של הקורס לאותו המכשף, לדוגמא: עבור הקריאה לפונקציה באופן הבא – `set_grade(wiz_01, 1, 85)` יעודכן הציון של המכשף "wiz_01" בקורס שיקויים ל-85.
- `def is_wizard_in_dorm(self):`
 - השיטה תקבל מכשף ותחזיר האם הוא נמצא במגורים כרגע או לא כבוליאני (True/False).
- `def __repr__(self):`
 - השיטה תחזיר מחרוזת המייצגת את המכשף בפורמט הבא, כאשר הטקסט באדום והערכים בשחור:

name: {name}, **average:** {(points + charms)/2}, **house:** {house}, **in_dorm:** {in_dorm}

כך לדוגמא הפעולה `print(w_harry)` תדפיס:

```
name: Harry Potter, average: 80.0, house: Gryffindor, in_dorm: False
```

- `def __eq__(self, other):`
 - השיטה תחזיר ערך בוליאני (True/False) המציין אם האובייקט הנוכחי (self) והאובייקט other שווים. שני אובייקטים מסוג מכשף שווים אם ממוצע הציונים שלהם שווה, לדוגמא:

```
>>>w_harry == w_malfoy
False
```

- יש לממש את כלל שיטות ההשוואה במחלקה (`__lt__`, `__ne__`, `__ge__`, `__le__`, `__gt__`)

סעיף ב – מחלקת בית (House):

מחלקת House מייצגת בית (קבוצה בתוך בית הספר) והיא מוגדרת ע"י השדות הבאים:

- name – שם הבית (מחרוזת) לדוגמא: 'Slytherin'
- school – בית הספר אליו שייך הבית (מסוג School)
- head – שם ראש הבית (מחרוזת) לדוגמא: 'Severus Snape'
- password – סיסמת הכניסה למגורים (מחרוזת)
- score – לכל בית יש ניקוד שיכול להשתנות במהלך הזמן (מספר שלם)
- wizard_list – רשימה (list) של אובייקטים מסוג מכשף המכילה את המכשפים השייכים לבית, יש להשתמש בהעתקה רדודה

ממשו את הפעולות הבאות:

- `def __init__(self, name, school, head, password, score = 0, wizard_list=[]):`
 - בנאי המאתחל את כל השדות באובייקט מסוג בית.
- `def get_wizards(self):`
 - השיטה תחזיר את רשימת המכשפים השייכים לאותו הבית.
- `def set_head(self, head):`
 - השיטה תקבל כקלט את שמו של ראש הבית החדש כמחרוזת ותשנה את הערך של head לשם שהתקבל.
- `def add_score(self, points):`
 - השיטה תקבל מספר שלם ותוסיף אותו לניקוד של הבית.
- `def change_password(self, new_password):`
 - השיטה תקבל סיסמא חדשה כמחרוזת ותשנה את סיסמת הכניסה של הבית לערך שהתקבל.
- `def add_wizard(self, wiz):`
 - השיטה תקבל רשימת מכשפים (רשימה של אובייקטים מסוג מכשף) ותוסיף את כל המכשפים ברשימה שאינם נמצאים כבר בבית לרשימת המכשפים של הבית.
- `def is_wizard_in_house(self, wiz):`
 - שיטה רקורסיבית, תקבל מכשף ותבדוק האם הוא נמצא ברשימת המכשפים של הבית, השיטה תחזיר True אם כן ואחרת False (בוליאני), אין להשתמש בשיטה `contains` שתממשו מאוחר יותר בסעיף זה.
- `def rank_wizards(self, rank):`
 - השיטה תקבל מספר שלם ותחזיר את המכשף מאותו הבית הנמצא במיקום הנ"ל בדירוג לפי ממוצע ציוניו, לדוגמא: עבור הקלט 3, השיטה תחזיר את המכשף בעל הממוצע

השלישי הגבוה ביותר, במידה והמספר שהתקבל גדול מאורך הרשימה, יוחזר המכשף בעל הממוצע הנמוך ביותר. אחרת (קלט שאינו מספר חיובי ושלם (0, -1)) יוחזר None – יש לממש את השיטה באחד מאלגוריתמי המיון שלמדנו, הרשימה המקורית באובייקט House לא תשתנה.

- `def __repr__(self):`
 - השיטה תחזיר (לא תדפיס) ייצוג מחרוזת של אובייקט מסוג בית, כך שהדפסת הערך המוחזר מהפונקציה ידפיס באופן הבא כאשר הטקסט באדום והערכים בשחור:

`House of {name} has {score} points. The proud wizards on {name} are: \n{wizard_list}`

לדוגמא – הפעולה `print(h_gryffindor)` תדפיס:

```
House of Gryffindor has 50 points. The proud wizards of Gryffindor are:
Harry Potter | Ron Wisely
```

- `def __gt__(self, other):`
 - השיטה תחזיר ערך בוליאני (True/False) המייצג האם ניקוד הבית הנוכחי (self) גדול יותר מניקוד הבית other.

לדוגמא:

```
>>> gryffindor > slytherin
```

True

- יש לממש את כלל שיטות ההשוואה (`__lt__`, `__gt__`, `__eq__`, `__ne__`, `__ge__`, `__le__`)
- `def __contains__(self, wizard_name):`
 - השיטה תחזיר ערך בוליאני (True/False) אם קיים מכשף בעל אותו השם בבית, לדוגמא:

```
>>> wiz_name = 'Harry Potter'
```

```
>>> wiz_name in gryffindor
```

True

סעיף ג' – מחלקת בית ספר (School):

מחלקת School מייצגת את בית הספר והיא מוגדרת ע"י השדות הבאים :

- name – שם בית הספר (מחרוזת)
- headmaster – שם מנהל בית הספר (מחרוזת) לדוגמא: 'Albus Dumbledore'
- houses – רשימת הבתים בבית הספר, יש להשתמש בהעתקה רדודה

ממשו את הפעולות הבאות :

- `def __init__(self, name, headmaster, houses=[]):`
 - בנאי המאתחל את כל השדות באובייקט מסוג בית.
- `def set_name(self, new_name):`
 - השיטה תקבל מחרוזת ותשנה את שם בית הספר.
- `def add_house(self, wiz):`
 - השיטה תקבל רשימת בתים (רשימה של אובייקטים מסוג בית) ותוסיף את כל הבתים ברשימה שאינם נמצאים כבר בבית הספר לרשימת הבתים של בית הספר.
- `def set_headmaster(self, new_headmaster):`
 - השיטה תקבל מחרוזת ותשנה את שם מנהל בית הספר.
- `def best_house_avg (self):`
 - השיטה תחזיר את הבית אשר למכשפים השייכים אליו ממוצע הציונים הגבוה ביותר, את השיטה יש לממש ע"י ביצוע מיון לפי ממוצע על פי אחד מאלגוריתמי המיון שלמדנו.
- `def best_house_score (self):`
 - השיטה תחזיר את הבית בעל הניקוד הגבוה ביותר ע"י מיון בסיבוכיות של $O(n \log n)$ או בסיבוכיות של $\Omega(n)$ כאשר n מספר הבתים בבית הספר.
- `def __repr__(self):`
 - השיטה תחזיר (לא תדפיס) ייצוג מחרוזת של אובייקט מסוג בית ספר באופן הבא כאשר הטקסט באדום והערכים בשחור:

{name} **school for witchcraft and wizardry, under head master** {headmaster}

לדוגמא – `print(s_hogwarts)` ידפיס :

```
Hogwarts school of witchcraft and wizardry, under head master Albus Dumbledore
```

סעיף ז – מחלקת מכשף בוגר (Guider):

מחלקת Guider מייצגת מכשף בוגר, המחלקה יורשת ממחלקת Wizard והיא מוגדרת ע"י השדות הנוספים הבאים:

- play_quidditch – האם המכשף הבוגר משחק את משחק הספורט קווידיץ' (בוליאני)
ממשו את הפעולות הבאות:
- def __init__(self, name, house, potions, charms, play_quidditch, in_dorm=False):
 - בנאי המאתחל את כל השדות באובייקט מסוג מכשף בוגר ע"י שימוש בתורשה.
- def __repr__(self):
 - השיטה תחזיר ייצור כמחרוזת של האובייקט באופן הבא:
Guider (name, house, average) לדוגמא:

```
Guider (Fred Weasley, Gryffindor, 60.0)
```