

234124 - מבוא לתכנות מערכות

תרגיל בית 5

סמסטר אביב 2022

תאריך פרסום: 23/06/2022
תאריך הגשה: 01/07/2022
מתרגלים אחראים: מוחמד גראמנה

1 הערות כלליות

- תרגיל זה מהווה 4% מהציון הסופי
- התרגיל להגשה בזוגות בלבד.
- מענה לשאלות בנוגע לתרגיל יינתן אך ורק בפורום התרגיל בפיאצה או בסדנות. לפני פרסום שאלה בפורום אנא בדקו אם כבר נענתה – מומלץ להיעזר בכלי החיפוש שהוצגו במצגת האדמיניסטרציה בתרגול הראשון.
- שימו לב: לא תינתנה דחיות במועד הגשת התרגיל פרט למקרים חריגים. תכננו את הזמן בהתאם.
- ראו את התרגיל עד סופו לפני שאתם מתחילים לממש. חובה להתעדכן בעמוד ה-F.A.Q של התרגיל, הכתוב שם מחייב.
- העתקות קוד בין סטודנטים ובפרט גם העתקות מסמסטרים קודמים תטופלנה. עם זאת – מומלץ ומבורך להתייעץ עם חברים על ארכיטקטורת המימוש.
- קבצי התרגיל נמצאים ב-GitHub repository הבא: <https://github.com/CS234124/ex5>
- המסמך נכתב בלשון זכר מטעמי נוחות בלבד ומיועד לשני המינים.
- מטרת תרגיל זה היא היכרות עם תכנות ב-Python.

2 מערכת רישום קורסים

הטכניון מקים בימים אלה מערכת חדשה לניהול קורסים, המבוססת על קבצי JSON. בעקבות זאת, הוא מבקש מכם להכין 3 פונקציות שיעזרו לו במימוש המערכת. בכל אחד מהסעיפים הבאים תממשו אחת מהפונקציות האלה.

2.1 מציאת הסטודנטים הרשומים לקורס מסוים

בסעיף זה נחזיר את שמות הסטודנטים הרשומים לקורס ספציפי. המידע מאוחסן בקובץ JSON שבו הסטודנטים מופיעים לפי מספרי תעודות הזהות שלהם, ושמואים בו גם שמם וההקורסים אליהם הם רשומים. ממשו פונקציה בשם `names_of_registered_students` המקבלת 2 ארגומנטים, וחתימתה:

```
def names_of_registered_students(input_json_path, course_name)
```

הסבר על הארגומנטים:

1. ארגומנט ראשון - מסלול לקובץ JSON המכיל את מאגר הסטודנטים. המידע שמור בפורמט הבא:
<student_id>: { "student_name": <name>, "registered_courses": course_list }

כאשר:

`student_id` הוא מספר תעודת הזהות (במחרוזת)

`student_name` הוא שם הסטודנט

`course_list` הוא רשימה של שמות הקורסים שהסטודנט רשום אליהם (רשימה של מחרוזות)

לדוגמה:

```
{
  "318467833" : { "student_name" : "Mohamed Gramne", "registered_courses" : ["Introduction to Systems Programming",
    "Data Structures 1","Introduction to Algorithms"]},
  "318422301" : { "student_name" : "Mhameed", "registered_courses" : ["Introduction to Systems Programming",
    "Data Structures 1"]},
  "319856388" : { "student_name" : "Efrat Levkovizh", "registered_courses" : ["Introduction to Systems Programming"]},
  "318464432" : { "student_name" : "Yazeed Falah", "registered_courses" : ["Introduction to Systems Programming",
    "Data Structures 1","Introduction to Algorithms"]},
  "318423455" : { "student_name" : "Zoabi", "registered_courses" : ["Introduction to Algorithms"]}
}
```

2. ארגומנט שני – שם הקורס המבוקש.

למשל, עבור קובץ ה-JSON בדוגמה למעלה ושם הקורס - "Introduction to Systems Programming"

על הפונקציה להחזיר (אין דרישה לסדר מסוים):

['Mohamed Gramne', 'Mhameed', 'Efrat Levkovizh', 'Yazeed Falah']

2.2 מציאת מספר הרשומים לכל קורס

בסעיף זה נחשב את מספר הסטודנטים הרשומים לכל קורס במערכת, ונכתוב את המידע לקובץ. בכל שורה בקובץ

הפלט יצויין שם קורס מסוים ומספר הרשומים אליו.

ממשו פונקציה בשם enrollment_numbers שחתימתה:

```
def enrollment_numbers(input_json_path, output_file_path)
```

הפונקציה תקבל 2 ארגומנטים:

1. ארגומנט ראשון - מסלול לקובץ JSON עם נתוני הסטודנטים בטכניון כפי בתואר בסעיף 2.1.

2. ארגומנט שני – מסלול (path) לקובץ בו ייכתב הפלט – output_file_path. הפונקציה תכתוב לקובץ את

שמות כל הקורסים בטכניון, ואת מספר הרשומים לכל אחד. הקורסים ימויינו על פי שם הקורס בסדר

לקסיקוגרפי עולה (ניתן להניח שאין 2 קורסים עם אותו שם). בכל שורה בקובץ הפלט יופיע שם של קורס

אחד בתוך מרכאות, לאחריו רווח אחד, ולאחריו מספר הרשומים לקורס זה, באופן הבא:

"<course name> <num_enrolled>"

לדוגמה, עבור קובץ ה-JSON בדוגמה בסעיף 2.1, תוכן קובץ הפלט יהיה:

"Data Structures 1" 3

"Introduction to Algorithms" 3

"Introduction to Systems Programming" 4

2.3 מציאת קורסים בכל הסמסטרים והמרצים המעבירים אותם

בסעיף זה נניח שיש מספר קבצי JSON, אחד לכל סמסטר, המאוחסנים בתיקייה שתקבל מהמשתמש. כל קובץ

JSON יכיל מידע לגבי הקורסים המועברים באותו הסמסטר, ושמות המרצים המעבירים אותם. הפונקציה תאתר את

כל הקורסים-המרצים שהעבירו קורסים המועברים בכל הסמסטרים, ולכל המרצים-קורס את רשימת כל המרצים

הקורסים שהעבירו אותם. הפונקציה תקבל כפרמטר את הספרייה שבה מאוחסנים

קבצי ה-JSON. אנו נניח שכל קובץ JSON בתיקייה זו מכיל מידע עבור סמסטר כלשהו.

ממשו פונקציה בשם courses_for_lecturers המקבלת 2 ארגומנטים וחתימתה:

```
def courses_for_lecturers(json_directory_path, output_json_path)
```

הסבר על הארגומנטים:

1. ארגומנט ראשון - מסלול לתיקייה המכילה קבצים מסוג JSON, כל אחד מכיל מידע עבור סמסטר אחד.

פורמט קבצי ה-JSON יהיה:

```
<course_id>: {"course_name": <name>}, "lecturers": <lecturer_list>
```

כאשר:

course_id הוא מספר הקורס (במחרוזת)

course_name הוא שם הקורס

lecturer_list הוא רשימת שמות המרצים שהעבירו את הקורס בסמסטר המתאים (רשימה של מחרוזות)

לדוגמה, נניח שקיימים שני קבצי JSON עבור 2 סמסטרים, בשמות 2022a.json, 2022b.json, המאוחסנים בתיקייה בשם "semesters_databases". מבנה התיקייה יהיה כדלקמן:

```
semesters_databases
├── 2022a.json
└── 2022b.json
```

תוכן הקובץ 2022a.json:

```
{
  "234124" : { "course_name" : "Introduction to Systems Programming", "lecturers" : ["Liav Adi","Gal Lalouche"]},
  "234247" : { "course_name" : "Introduction to Algorithms", "lecturers" : ["Hadas Shachnai","Roy Schwartz"]},
  "234292" : { "course_name" : "Logic for Computer Science", "lecturers" : ["Keren Censor-Hillel","Liav Adi"]}
}
```

תוכן הקובץ 2022b.json:

```
{
  "234124" : { "course_name" : "Introduction to Systems Programming", "lecturers" : ["Liav Adi","Gal Lalouche"]},
  "234218" : { "course_name" : "Data Structures 1", "lecturers" : ["Erez Petrank","Rami Cohen","Hadas Shachnai"]}
}
```

שימו לב שהתיקייה עשויה להכיל **קבצים נוספים** שאינם קבצי JSON. יש להתייחס לקבצים בתיקייה עם סיומת json. בלבד.

2. ארגומנט שני – מסלול (path) לקובץ בו ייכתב הפלט, בפורמט json. הקובץ ייכתב בפורמט הבא:

```
<lecturer_name>: <list_of_courses>
```

המפתחות יהיו שמות המרצים, והערכים יהיו רשימות של שמות הקורסים שהועברו על ידי אותו מרצה בכל

הסמסטרים. לדוגמה, עבור הדוגמה לעיל עם 2 קבצי JSON, יתקבל קובץ פלט עם התוכן הבא:

```
{
  "Liav Adi": ["Introduction to Systems Programming", "Logic for Computer Science"],
  "Gal Lalouche": ["Introduction to Systems Programming"],
  "Keren Censor-Hillel": ["Logic for Computer Science"],
  "Hadas Shachnai": ["Introduction to Algorithms", "Data Structures 1"],
  "Roy Schwartz": ["Introduction to Algorithms"],
  "Erez Petrank": ["Data Structures 1"],
  "Rami Cohen": ["Data Structures 1"]
}
```

3 הערות

- עם התרגיל אתם מקבלים את הקבצים הבאים:
 - ex5.py - עליכם להשלים את מימוש הפונקציות בקובץ זה ולהגישו.
 - test.py - בקובץ זה מסופקים לכם שלושה טסטים לשלושת הפונקציות שעליכם לממש. על מנת להריץ את הטסט יש למקם אותו באותה תיקייה כמו ex5.py עם המימוש שלכם ולהריץ את הפקודה הבאה בשרת – “python3 test.py”.
- הטסט ממומש בעזרת הספרייה – unittest המספקת כלים לבנייה והרצת טסטים בפייתון.
- students_database.json – דוגמה לקובץ נתוני הסטודנטים.
- semesters_databases – תיקייה לדוגמה המכילה 2 קבצי json – 2022a.json ו-2022b.json
- expected_output_test2_enrollment_numbers.txt – קובץ פלט לדוגמה של הפונקציה enrollment_numbers.
- expected_output_test3_courses_for_lecturer.json – קובץ פלט לדוגמה של הפונקציה courses_for_lecturers.
- וודאו שאתם מריצים פייתון גרסה 3.6. שימו לב כי זאת לא גרסה ברירת מחדל על השרת. כדי להריץ פייתון 3.6 השתמשו בפקודה `python3`.
- פתרון התרגיל צריך לעבוד גם במערכת ההפעלה לינוקס וגם במערכת ההפעלה ווינדוס.
- ניתן להניח שמתקבלים ארגומנטים תקינים.

4 הגשה

את ההגשה יש לבצע דרך אתר הקורס, תחת Electronic Submit -> HW5 -> Assignments .

הקפידו על הדברים הבאים:

- יש להגיש את קבצי הקוד מבווצים לקובץ zip (לא פורמט אחר).
- אין להגיש קבצים נוספים מלבד קובץ אחד בשם **ex5.py** בתוך הzip
- ניתן להגיש את התרגיל מספר פעמים, רק ההגשה האחרונה נחשבת.
- על מנת לבטח את עצמכם נגד תקלות בהגשה האוטומטית, שימרו את קוד האישור עבור ההגשה. עדיף לשלוח גם לשותף. כמו כן שימרו עותק של התרגיל על חשבון ה-CSL3/Github/Google Drive שלכם לפני ההגשה האלקטרונית ואל תשנו אותו לאחריה (שינוי הקובץ יגרור שינוי חתימת העדכון האחרון).
- כל אמצעי אחר לא יחשב הוכחה לקיום הקוד לפני ההגשה.

בהצלחה !