

## עקרונות שפות תוכנה

תש"פ, סמסטר א'

### עבודת הגשה מס' 4

#### הנחיות:

- יש להגיש את העבודה עד **26.01.2020**
- מותר להכין את העבודה בזוגות או בודדים בלבד. חל איסור להעתיק או "לחלוק" חלקים מהפתרונות.
- מגישים קובץ zip אחד שבתוכו קובץ py. אחד ו-pdf
- חשוב מאוד לציין את שמות שני המגישים בתוך הקובץ + תז.
- חובה להשתמש בשמות הפונקציות המוגדרות.
- על הקובץ להכיל פקודות הרצה של הפונקציות (אפשר מיד אחרי הפונקציה גם לקרוא לה עם ערכים מתאימים לצורך הדגמה)
- ההגשה היא באתר moodle – על שני הסטודנטים להגיש את אותו הקובץ RAR/ZIP עם שם זהה שיהיה מורכב מהמילה "HW4" ושני שת"ז עם קווים תחתונים ביניהם. לדוגמה:  
HW4\_123456789\_123456789.zip
- כל שאלה ופניה בנוגע לתרגיל יש להפנות אך ורק לאחראית על התרגיל – תמר שרוט בדוא"ל: [tammarm@gmail.com](mailto:tammarm@gmail.com) פניות בכל בדרך אחרת – לא יענו!

\* \* \*

**בהצלחה!**

\* \* \*

---

## תרגיל 1

בשאלה זאת אנו מתמקדים ב-Shmyton (התעלמות ממערכת האובייקטים הקיימת בשפת python וסימלוצה על ידי פונקציות ו dispatch function או dispatch dictionary).

(א) בכיתה בנינו את הכלים המאפשרים לנו לבנות מחלקות כולל הורשה יחידה. בתרגיל זה אתם מתבקשים לשנות את הקוד הנ"ל על מנת לתמוך **הורשה מרובה**. בתרגיל אתם מתבקשים לתמוך בהורשה של יותר ממחלקה אחת. שימו לב שכמו בפייתון גם פה הבעיה של הורשה מרובה תיפתר על ידי נתינת עדיפות לאבא הקודם יותר.  
רמז: ניתן להיעזר ב-build-in type – list.  
על מנת לפשט את המימוש, בשונה מפייתון, במימוש שלכם לאבות של האבא הראשון יש קדימות על פני האבא השני.

(ב) היעזרו בכלים המשודרגים על מנת לבנות את המחלקות הבאות:

### מחלקת MyDate

מחלקה שמתארת תאריך. השדות: יום, חודש ושנה.

ובעלת המתודות:

- בנאי שמקבל את הערכים ערך דיפולט של 2020 (לשנה)
- repr
- str – תדפיס תאריך כפי שמקובל בארץ (dd.mm.yyyy).
- getDay, getMonth, getYear – מחזירות את ערכי השדות בהתאמה.
- setDay, setMonth, setYear – מקבלות פרמטר ומשנות את הערך של השדה הרלוונטי, במידה ומדובר בערך חוקי (יום בין 1 ל-30, חודש בין 1 ל-12 ושנה בין 1900 ל-2100).

### מחלקת Person

מחלקה שמתארת בן אדם. השדות: שם פרטי, שם משפחה, תאריך לידה ות"ז (int)

ובעלת המתודות:

- בנאי שמקבל ערכים (2 מחרוזות, אובייקט MyDate, ו-int).
- repr
- str – תדפיס את פרטי האדם בצורה הבאה, לדוגמא:  
Name: Moshe Choen  
DoB: 01.01.2020  
ID: 1234
- מתודות get מתאימות לכל אחד מהשדות
- מתודות set מתאימות לכל אחד מהשדות. ערכים תקינים לת"ז הם ערכים חיוביים.

### מחלקת Student – יורשת ממחלקת Person

מחלקה המתארת סטודנט. השדות הנוספים במחלקה: שם המחלקה בה הוא לומד, ממוצע ציונים, וותק בלימודים

ובעלת המתודות:

- בנאי
- repr
- str – תשתמש בstr של אדם ותדפיס בנוסף עוד שלוש שורות, לדוגמא:  
Learning: Software Engineering  
Avg: 98.0  
Seniority: 3
- get ו-set לשדות הנוספים

### מחלקת Faculty – יורשת ממחלקת Person

מחלקה המתארת חבר צוות בפקולטה. השדות הנוספים במחלקה: שם המחלקה בה הוא מלמד, משכורת וותק בהוראה

ובעלת המתודות:

- בנאי
- repr
- str – תשתמש בstr של אדם ותדפיס בנוסף עוד שלוש שורות, לדוגמא:  
Teaching: Software Engineering  
Salary: 1000.0  
Seniority: 3
- get ו-set לשדות הנוספים

### מחלקת TA – יורשת ממחלקת Student וממחלקת Faculty

מחלקה המתארת עוזר הוראה. אין לו שדות נוספים

ובעלת המתודות:

- בנאי
- repr
- str – שתדפיס את כל המידע אודותיו – בלי חזרות! (במקרה של חפיפה הולכים לפי האבא בעל העדיפות הגדולה יותר).

שימו לב כי מותר לכם להוסיף עוד מתודות למימוש, אך חישבו היטב איך למזער את ההוספה על מנת למקסם את היעילות בקוד למנוע את הרחבת הAPI.  
כמו כן ממשו את הקוד כמה שיותר איכותי, תוך reuse של קוד עד כמה שניתן.

## תרגיל 2 – פונקציות כלליות על ידי DDP

בתרגיל זה נשתמש במחלקות `Rlist`, `Rational` שהגדרנו בהרצאה וכן בפרמיטיבי `int`.

יש לבנות פונקציה ג'נרית `apply` שמבצעת פעולות שונות על זוג ארגומנטים מאחד משלושת הטיפוסים שהוגדרו לעיל.

עליכם לממש, כרגע, תמיכה בשתי פעולות חיבור (`add`) וכפל (`mul`).

יש לתמוך בכל השילובים הבאים:

- שילובים לפעולת `add`:
  - `Rlist` ו-`Rational` – הפעולה תחזיר רשימה (`Rlist`) חדשה שהיא הרשימה המקורית עם הוספת הערך של `Rational` כאיבר אחרון ברשימה. **חובה** לממש תוך שימוש ב`cons`.
  - `Rlist` ו-`Rlist` – הפעולה תחזיר רשימה חדשה היא שירשור שתי הרשימות.
  - `Rational` ו-`Rational` – הפעולה תחזיר את סכום שני הארגומנטים
  - `int` ו-`int` – הפעולה תחזיר את סכום שני הארגומנטים
  - `int` ו-`Rational` – הפעולה תחזיר את סכום שני הארגומנטים כאובייקט `Rational`
- שילובים לפעולת `mul`:
  - `Rlist` ו-`int` – פעולה שתחזיר רשימה שהיא שירשור הרשימה המקורית עם עצמה כמספר הפעמים של הארגומט השלם.
  - `Rational` ו-`Rational` – הפעולה תחזיר את מכפלת שני הארגומנטים
  - `int` ו-`int` – הפעולה תחזיר את מכפלת שני הארגומנטים
  - `int` ו-`Rational` – הפעולה תחזיר את מכפלת שני הארגומנטים כאובייקט `Rational`

יש צורך לתמוך באופרטור גם כשם (`add/mul`) וגם כאופרטור מובנה (`+/-`).

## תרגיל 3 – שאלות תאורטיות

- עבור כל אחת מהרמות של עיקרון שפות התכנות – הפשטה, כיתבו מהי, כיתבו לפחות עיקרון תיכנותי אחד שנובע ממנה והסבירו במשפט מדוע.
- הסבירו ב-2 שורות (בלבד) מדוע עיקרון הרקורסיה חשוב לבניית שפות?
- מה בעצם בונים כאשר בונים שפה? הסבירו בפירוט!
- שימוש ב-`memoized` משפר דרסטית את זמן ריצה. מה הוא עושה לסיבוכיות מקום? האם הוא עדיף על הגישה האיטרטיבית ומדוע?