המכללה האקדמית להנדסה סמי שמעון

**עבודת הגשה מס' 4**

תאריך הגשה הוא **15/01/2021**

- ✓ ניתן להכין את המטלה **בזוגות**
- ✓ רק **חבר אחד** בצמד **יגיש** בפועל את העבודה
- ✓ יש להגיש את הפתרון **תחת שם המכיל את מספרי ת"ז של כל המגישים**.
- ✓ יש להגיש קובץ בפורמט PY.
- ✓ **חובה** להשתמש בשמות הפונקציות המוגדרות.
- ✓ שימו לב, הפלט של דוגמאות ההרצה הוא בהתאם לסביבת הפיתוח Python IDLE.
- ✓ חובה לכל פונקציה להוסיף doc strings .
- ✓ הגשה דרך **מודל** בלבד!
- ✓ כל שאלה בנוגע לתרגיל יש להפנות אך ורק לאחראי על התרגיל - מיכאל באימייל: finkm@ac.sce.ac.il . פניות בכל בדרך אחרת – לא יענו! בפנייה, יש לציין את : שם הקורס (**PPL**), מסי עבודה (**HW4**) ופרטים מזהים.
- ✓ **אישורי ההארכה** יינתנו ע"י מרצה בלבד !

* הערה חשובה: קיים הבדל עקרוני בין הדפסה לבין החזרה של ערך מפונקציה! ברירת המחדל בהיעדר הוראת הדפסה מפורשת היא **החזרה בלבד**.

המכללה האקדמית להנדסה סמי שמעון

## חלק 1 (Python) OOP ומימוש מערכת אובייקטים (Shmython)

### שאלה 1

ממש מחלקות הבאות ב-**Python** :

**א)** **Date** - תאריך כולל פירוט של שנה, חודש ויום בחודש. יש לממש בנאי שאם לא מקבל פרמטרים מאתחל אובייקט בתאריך נוכחי ופונקציה גנרית- **str**.

**ב)** **Temperature**– מדידת טמפרטורה שכוללת טמפרטורה ותאריך מדידה. יש לממש בנאי שאם לא מקבל פרמטרים לתאריך מאתחל אובייקט בתאריך נוכחי, פונקציה גנרית – **str** ומתודה **compareTemp** שמקבלת כפרמטר אובייקט מדידת טמפרטורה אחר ומחזירה אובייקט עם טמפרטורה הגבוה ביותר.

**ג)** **Location** - מדידות טמפרטורה במקום מסוים כולל שם מקום ורשימת מדידות טמפרטורה. יש לממש בנאי שמקבל מקום ומתודות: **addTemp** שמוסיפה מדידות טמפרטורה לרשימה, **printLocation** שמדפיסה מקום ורשימת מדידות, **getAverage** שמחזירה ממוצע מדידות טמפרטורה, **getMaxTemp** שמחזירה מדידת טמפרטורה מקסימלית ומתודה **compareLocation** שמקבלת אובייקט נוסף ומחזירה אובייקט עם ממוצע מדידות טמפרטורה יותר גבוה.

**<u>דוגמה להרצה(מחייבת):</u>**

```
>>> d1=Date(1,2,2020)
>>> d2=Date()
>>> print(d1,'-',d2)
01/02/2020 - 20/12/2020
>>> t1=Temperature(-12,1,2,2020)
>>> t2=Temperature(0)
>>> t3=Temperature(32,20,8,2020)
>>> print(t1,',',t2,',',t3)
-12.0°C:01/02/2020 , 0.0°C:20/12/2020 , +32.0°C:20/08/2020
>>> print(t1.compareTemp(t3))
+32.0°C:20/08/2020
>>> loc1=Location('London')
>>> loc1.printLocation()
London
no temperature measurements available
>>>
loc1.addTemp(Temperature(9),Temperature(7,1,12,2020),Temperature(23,21,8,2020),Temperature(16,
4,5,2020))
>>> loc1.printLocation()
London
+9.0°C:20/12/2020 +7.0°C:01/12/2020 +23.0°C:21/08/2020 +16.0°C:04/05/2020
>>> loc1.getAverage()
13.75
>>> print(loc1.getMaxTemp())
+23.0°C:21/08/2020
>>> loc2=Location('Berlin')
>>>loc2.addTemp(Temperature(6),Temperature(28,12,8,2020),Temperature(3,1,12,2020),Temperature(
-3,2,1,2020))
>>> loc2.printLocation()
Berlin
+6.0°C:20/12/2020 +28.0°C:12/08/2020 +3.0°C:01/12/2020 -3.0°C:02/01/2020
>>> loc2.getAverage()
8.5
>>> loc2.compareLocation(loc1).printLocation()
London
+9.0°C:20/12/2020 +7.0°C:01/12/2020 +23.0°C:21/08/2020 +16.0°C:04/05/2020
```

המכללה האקדמית להנדסה סמי שמעון

## שאלה 2

ממש אותן מחלקות ב -**Shmython**:

**א) Date** - יש לממש פונקציה make_date_class()

**ב) Temperature** - יש לממש פונקציה make_temperature_class()

**ג) Location** - יש לממש פונקציה make_location_class()

הערה: יש לצרף קוד של מערכת אובייקטים שנבנה בכיתה להרצה.

**דוגמה להרצה(מחייבת):**

```
>>> d1=Date['new'](1,2,2020)
>>> d2=Date['new']()
>>> print(d1['get']('str')(),'-',d2['get']('str')())
01/02/2020 - 20/12/2020
>>> t1=Temperature['new'](-12,1,2,2020)
>>> t2=Temperature['new'](0)
>>> t3=Temperature['new'](32,20,8,2020)
>>> print(t1['get']('str')(),',',t2['get']('str')(),',',t3['get']('str')())
-12.0°C:01/02/2020 , 0.0°C:20/12/2020 , +32.0°C:20/08/2020
>>> print(t1['get']('compareTemp')(t3)['get']('str')())
+32.0°C:20/08/2020
>>> loc1=Location['new']('London')
>>> loc1['get']('printLocation')()
London
no temperature measurements available
>>>loc1['get']('addTemp')(Temperature['new'](9),Temperature['new'](7,1,12,2020),Temperature['new'](23
,21,8,2020),Temperature['new'](16,4,5,2020))
>>> loc1['get']('printLocation')()
London
+9.0°C:20/12/2020 +7.0°C:01/12/2020 +23.0°C:21/08/2020 +16.0°C:04/05/2020
>>> loc1['get']('getAverage')()
13.75
>>> print(loc1['get']('getMaxTemp')()['get']('str')())
+23.0°C:21/08/2020
>>> loc2=Location['new']('Berlin')
>>>loc2['get']('addTemp')(Temperature['new'](6),Temperature['new'](28,12,8,2020),Temperature['new'](3
,1,12,2020),Temperature['new'](-3,2,1,2020))
>>> loc2['get']('printLocation')()
Berlin
+6.0°C:20/12/2020 +28.0°C:12/08/2020 +3.0°C:01/12/2020 -3.0°C:02/01/2020
>>> loc2['get']('getAverage')()
8.5
>>> loc2['get']('compareLocation')(loc1)['get']('printLocation')()
London
+9.0°C:20/12/2020 +7.0°C:01/12/2020 +23.0°C:21/08/2020 +16.0°C:04/05/2020
```

המכללה האקדמית להנדסה סמי שמעון

## שאלה 3

**א)** שנו את המימוש הקיים של הפונקציה **make_class**, כך שלכל מחלקה יהיה מאפיין של שם המחלקה (**name**) (ראו דוגמה להרצה).

```
def make_account_class():
    return make_class('Account', {'interest' : 0.05})

def make_save_account_class():
    def init(self, owner):
        self['set']('owner',owner)
        self['set']('balance',0)
    return make_class('SaveAccount', {'__init__' : init, 'interest' : 0.03}, Account)

Account = make_account_class()
SaveAccount = make_save_account_class()
```

**דוגמה להרצה:**

```
>>> Account['get']('name')
'Account'
>>> SaveAccount['get']('name')
'SaveAccount'
```

**ב)** שנו את המימוש הקיים של פונקציה **make_class**, כך שעבור כל מחלקה תהיה אפשרות לקבל מסלול הורשה של כל המחלקות מהבסיס עם הסימן **'::'** המפריד בין שמות המחלקות.

```
>>> Account['class_path']()
'Account'
>>> SaveAccount['class_path']()
'Account::SaveAccount'
```

## חלק 2: פונקציות גנריות ( generic functions)

### שאלה 4

יש להוסיף תמיכה במספר אקספוננציאלי (exponential number) לחבילה שמימשנו בכיתה לפעולות אריתמטיות על מספרים מרוכבים ומספרים רציונליים. יש לממש פונקציה גנרית **apply** שבהינתן שם של פעולה ושמות טיפוסים של ארגומנטים מחשבת ומחזירה את התוצאה של הפעולה על הארגומנטים. יש לממש את המחלקה **Exponential** שמגדירה את מספר אקספוננציאלי. בנאי של מס' אקספוננציאלי יקבל שני מספרים: בסיס וחיזקה של 10 (מספר שלם). למשל, מספר 0.00002 ניתן לייצג כ-$2*10^{-5}$ (2 זה בסיס ו-(-5) זאת חיזקה של 10). יש לתמוך ב-2 פעולות:

**א)** חיבור (add )

**ב)** כפל (mul)

**<u>דוגמה להרצה(מחייבת):</u>**

```
>>> apply('add', Exponential(2,-4), Rational(3,4))
Exponential(7502, -4)    # OR Rational(3751, 5000)OR ComplexRI(0.7502, 0)
>>> apply('add', Exponential(2,-4), ComplexRI(3,4))
ComplexRI(3.0002, 4)
>>> apply('add', Exponential(2,-4), Exponential(3,-5))
Exponential(23,-5)        # OR Rational(23,100000) OR ComplexRI(0.00023, 0)
>>> apply('mul', Exponential(2,-4), Exponential(3,-5))
Exponential(6,-9)
>>> apply('mul', Exponential(2,-4), ComplexMA(10,1))
ComplexMA(0.002, 1)
>>> apply('mul', Exponential(2,-4), Rational(1,3))
Rational(1,15000)                # OR Exponential(0.6666666666666666,-4) OR ComplexMA(…)
```

<u>הערה</u>: ניתן להחליט לבד על הטיפוס של התוצאה (אקספוננציאלי, רציונלי או מרוכב) במקרה של "ערבוב" טיפוסים. יש לספק פונקציות לפעולות אריתמטיות עבור כל קומבינציית טיפוסים אפשרית. אין להמיר את כל הטיפוסים לטיפוס אחד!

### שאלה 5

יש לממש פונקציה גנרית **coerce_apply** שבהינתן שם של פעולה ושמות טיפוסים של ארגומנטים מחשבת ומחזירה את התוצאה של הפעולה על הארגומנטים ע"י <u>המרה</u> של כל טיפוס נומרי (רציונלי ואקספוננציאלי) לטיפוס של מספר מרוכב והפעלת פעולה על שני מספרים מרוכבים.

<u>הערה</u>: ניתן להמיר אקספוננציאלי לרציונלי ורציונלי למרוכב או להמיר כל טיפוס למרוכב בנפרד (אקספוננציאלי למרוכב ורציונלי למרוכב). <u>תוצאה תמיד תהיה מטיפוס מספר מרוכב</u>!

**<u>דוגמה להרצה(מחייבת):</u>**

```
>>> coerce_apply('add', Exponential(2,-4), Rational(3,4))
ComplexRI(0.7502, 0)
>>> coerce_apply('add', Exponential(2,-4), ComplexRI(3,4))
ComplexRI(3.0002, 4)
>>> coerce_apply('add', Exponential(2,-4), Exponential(3,-5))
ComplexRI(0.00023, 0)
>>> coerce_apply('mul', Exponential(2,-4), Exponential(3,-5))
ComplexMA(6.000000000000001e-09, 0.0)
>>> coerce_apply('mul', Exponential(2,-4), ComplexMA(10,1))
ComplexMA(0.002, 1)
>>> coerce_apply('mul', Exponential(2,-4), ComplexRI(3,4)))
ComplexMA(0.001, 0.9272952180016122)
>>> coerce_apply('mul', Exponential(2,-4), Rational(1,3))
ComplexMA(6.666666666666667e-05, 0.0)
```

המכללה האקדמית להנדסה סמי שמעון

## חלק 3: חריגות (Exceptions)

## שאלה 6

שאלה זאת מתייחסת לשאלה **5** בעבודת בית מס' **3** (מימוש של **sequence**).

יש לשדרג את המימוש שלכם ולהוסיף טיפול בחריגות (**ValueError**, **TypeError**, **IndexError**) של Python במקרים:

**א)** בהפעלת **make_sequence** יש לבדוק פונקציה מקבלת פרמטר והוא רצף.

**ב)** בהפעלת **'filter'** פונקציה מקבלת פרמטר.

**ג)** בהפעלת **'filter_iterator'** פונקציה מקבלת פרמטר.

**ד)** בהדפסה לא מעגלית (**'next'**,**'reverse'** ) יש לטפל שמקום קיים ברצף.

**דוגמה להרצה(מחייבת):**

```
>>> s1=make_sequence()
<class 'TypeError'> : no sequence argument
>>> s1=make_sequence(200)
<class 'TypeError'> : no sequence argument
>>> s1=make_sequence((1,2,3,4,5))
>>> s1['filter']()
<class 'TypeError'> : No filter function
(1, 2, 3, 4, 5)
>>> p1=s1['filter_iterator']()
<class 'TypeError'> : No filter function
>>> p1=s1['filter_iterator'](lambda x: x<4)
>>> for _ in range(6):
p1['next']()
1
2
3
<class 'IndexError'> : tuple index out of range
<class 'IndexError'> : tuple index out of range
<class 'IndexError'> : tuple index out of range
>>> p1=s1['filter_iterator'](lambda x: x>1)
>>> p1['next']()
2
>>> p1['next']()
3
>>> p1['next']()
4
>>> for _ in range(6):
p1['reverse']()
5
4
3
2
<class 'IndexError'> : Index error
<class 'IndexError'> : Index error
```

המכללה האקדמית להנדסה סמי שמעון

## חלק 4:מבני נתונים רקורסיביים ( Recursive Data Structures)

### שאלה 7

נתון מטה מימוש של מחלקה בשם **Tree** המייצגת עץ כללי. לכל צומת בעץ יש ערך פנימי (**value**) ורשימת בנים (**nodes**). העלים הם מופעים של **Tree** ללא רשימת ילדים.

```
class Tree():
    def __init__(self, value, nodes=None):
        self.value = value
        self.nodes = nodes
    def __repr__(self):
        if self.nodes:
            return 'Tree({0},{1})'.format(self.value,repr(self.nodes))
        return 'Tree({0})'.format(self.value)
```

**א)** השלימו פונקציה **BuildTree**, שבהנתן עץ (**tree**) מיוצג כ-**tuple** מחזירה עץ חדש (מופע של **Tree**) כך שהעלים הם בעלי ערכים וערך לכל צומת פנימית גובה תת-עץ. הפונקציה חייבת להיות **רקורסיבית!**

```
def BuildTree (tree):
    if        <1>:
        return        <2>
    <3>
    return    <4>
```

**ב)** (8 נק') כתוב פונקציה **is_AVL_tree**, שבהנתן עץ (**tree**) הפונקציה המחזירה את ה-**True** אם עץ הוא עץ מאוזן(**AVL**), אחרת **False**. עץ מאוזן כאשר הפער בין גובהם של תתי-העצים לכל צמתים, הוא לכל היותר **1**.

```
def is_AVL_tree (tree):
```

**דוגמה להרצה:**

```
>>> t1=BuildTree((((1,2), 3), (4, (5, 6))))
>>> t1
Tree(3,[Tree(2,[Tree(1,[Tree(1), Tree(2)]), Tree(3)]), Tree(2,[Tree(4), Tree(1,[Tree(5), Tree(6)])])])
>>> is_AVL_tree(t1)
True
>>> t2=BuildTree(((2, 3), (4, (5, 6, (8, 2)))))
>>> t2
Tree(4,[Tree(1,[Tree(2), Tree(3)]), Tree(3,[Tree(4), Tree(2,[Tree(5), Tree(6), Tree(1,[Tree(8),
Tree(2)])])])])
>>> is_AVL_tree(t2)
False
>>> t3=BuildTree((((19,1,6), (1,(2,3))), (((1,2),6), (5, 6, (8, 2)))))
>>> t3
Tree(4,[Tree(3,[Tree(1,[Tree(19), Tree(1), Tree(6)]), Tree(2,[Tree(1), Tree(1,[Tree(2), Tree(3)])])]),
Tree(3,[Tree(2,[Tree(1,[Tree(1), Tree(2)]), Tree(6)]), Tree(2,[Tree(5), Tree(6), Tree(1,[Tree(8),
Tree(2)])])])])
>>> is_AVL_tree(t3)
True
```

המכללה האקדמית להנדסה סמי שמעון

## חלק 5:מפרש ( Interpreter)

### שאלה 8

אתם מתבקשים להרחיב/לעדכן את המפרש באופן הבא:

**א)** שהמחשבון יתמוך בנוסף למספרים שלמים (עשרוניים) גם במספרים מבסיסים: 2(**b**), 8(**q**), 16(**h**). כל מספר שלם יכול להיות רשום לפי בסיס. לדוגמה: **1afh**, **101101b**(תו בסוף המספר מציין בסיס). על המחשבון לזהות אם מספר נתון באחד משלושת הבסיסים, להפוך ולאחסן כמספר שלם עשרוני.

**דוגמת להרצה:**

```
calc> add(1001b, ah)
19
calc> add(1001b, mul(1ah, 21q))
451
```

**רמז:** פונקציה **int**, בנוסף להפיכת מחרוזת למספר שלם גם יכולה לבצע את הפעולה לפי בסיסים.
לדוגמה:

```
>>> int('10010',2)
18
```

**ב)** שלמחשבון תהיה אפשרות לבצע פעולת משלים ל-(**n-1**),במקרה שלנו **9**,ע"י אופרטור **compl** או סימן **!** אשר מקבל מספר שלם ומחזיר משלים ל-**9**(לכל ספרה) עבורו.
לדוגמה: **1564** -> **8435**, **12083** -> **87916**
את החישוב עצמו יש לבצע ע"י השלמת **pipeline**

**int( ''.join( ( <1>( lambda x:<2>, list(<3>)) ) ) )**

**דוגמת להרצה:**

```
calc> compl(12083)
87916
calc> !(12083,12,24)
TypeError: compl requires exactly 1 argument
calc> compl(12083.12)
TypeError: 12083.12 is not <class int>
calc>
```

**הערה לסעיפים א' ו-ב':**
יש לטפל בחריגות הרלוונטיות לפעולות אלו.

## חלק 6: שאלות תיאורטיות

**א)** במערכת אובייקטים שבנינו בכיתה (של **Shmython**) פונקציה **get** של אובייקט תמיד מחזירה מתודה כשמקבלת שם המאפיין שהוא פונקציה.

**ב)** במימוש מחשבון יש מקרים (בריצות מסוימות), כאשר פונקציה רקורסיבית **calc_eval** לא מבצעת חישוב רקורסיבי.

**ג)** בשימוש בפונקציה גנרית **coerce_apply** ניתן להמיר טיפוס אחד לטיפוסים שונים לפי הצורך.

**ד)** פונקציה שמפילה פונקציה אחרת שעלולה לעלות חריגה חייבת לטפל באותה חריגה ע"י תפיסתה או העברתה לפונקציה המפעילה.

**ה)** בשימוש OOP (Python) לכל האובייקטים של אותה מחלקה יש בדיוק אותם שדות (תכונות) ולא יתכן מצב שלאובייקט אחד יהיה שדה שלא מופיע אצל אובייקטים אחרים.

*בהצלחה !*