



משימות למעבדה מס' 10

(dispatch dictionary, מחלקות, אובייקטים, אופרטורים ב-Python)

1. השלימו את הקוד כך שיעבוד עם dispatch dictionary מבוסס הודעות :

```
def shmeasy_park(fee):          # fee – לשעה –
    <1>
    def charge(amount):        #amount – (בשקלים)
        <2>
    def park(time):             # time – (בשעות)
        <3>
    def dispatch( <4> ):
        <5>
    return <6>
```

דוגמת הרצה(מחייבת):

```
>>> k = shmeasy_park(5)
>>> k['charge'](100)
>>> k['park']( 10)
balance left: 50.0
```

2. ממשו את המחלקה **Time** המייצגת שעות דיגיטלי (שעות, דקות ושניות). עליכם לממש את הפונקציות הבאות:

- (א) פונקציה לאתחול נתונים `__init__` (יש לבצע בדיקה לפרמטרים המעוברים ולהכניס 0 אם אינם מתאימים).
- (ב) פונקציית הדפסת נתוני השעון בפורמט הבא **03:09:45** (וירטואלית)
- (ג) פונקציה **TimeToInt** הממירה את השעה ביום למספר השניות שחלפו מאז חצות (למשל, השעה **01:10:15** תומר למספר **4215**).
- (ד) פונקציה **IntToTime** הממירה מספר שניות לשעה המתאימה ומעדכנת את האובייקט הנוכחי.
- (ה) פונקציה בוליאנית **Later** המקבלת אובייקט-שעון נוסף ובודקת האם השעון הנוכחי מכיל שעה מאוחרת יותר מהשעון הנוסף. למשל, עבור השעונים **09:45:00** (נוכחי) ושעון נוסף **01:35:00** יוחזר ערך **True**.
- (ו) פונקציה **addSecond** המקדמת את נתוני השעון הנוכחי בשנייה אחת קדימה (עבור **23:59:59** - **00:00:00**).
- (ז) **אופרטור +** מקבל שעון נוכחי ושעון נוסף, מחבר את נתוני השעונים ומכניס את התוצאה לשעון הנוכחי. למשל עבור **09:45:00** (נוכחי) + **01:35:00** (נוסף) השעון הנוכחי יעודכן ל-**11:20:00**.
- (ח) **אופרטור -** מקבל שעון נוכחי ומספר שניות, מוריד מנתוני השעון הנוכחי את מספר השניות ומעדכן את השעון הנוכחי. למשל עבור **01:10:16** (נוכחי) - **5** (מספר השניות) יתעדכן השעון הנוכחי ל-**01:10:11**.



(ט) פונקציה הממירה את אובייקט השעון למחרוזת אשר מכילה את כל פרטי האובייקט בפורמט **hh:mm:ss**. למשל, **01:10:11** (מאובייקט המכיל שעות -1, דקות -10, שניות -11).

כתבו דרייבר המפעיל את כל פונקציות

דוגמת הרצה (#הדפסה צפויה):

```
start = Time( 9, 45, 0 )
end = Time( 1, 35, 0 )
test = Time( 1, 10, 15 )
start.printTime() #09:45:00
print('-----')
Time.printTime( start )
print('-----')
print(start.Later( end )) #True
print(test. TimeToInt ()) #4215
help=test. IntToTime (4215)
help.printTime() #01:10:15
help.addSecond()
help.printTime() #01:10:16
(start+end).printTime() #09:45:00 + 01:35:00 = 11:20:00
(help-5).printTime() #01:10:11
print (help.__str__()) #01:10:11
```

3. ממשו את המחלקה **ZoneTime** המרחיבה את המחלקה **Time** כך שתכיל גם את שם אזור הזמן (**Zone**).

בנוסף, עליכם להגדיר את הפונקציות הבאות:

- (א) פונקציית אתחול (יש להפעיל את הפונקציה המתאימה במחלקת האב).
 - (ב) פונקציה המדפיסה את כל הפרטים כולל שם האזור.
 - (ג) פונקציה המדפיסה את כל פרטי השעון, כולל שם האזור (דריסה של הפונקציה הווירטואלית!).
- כתבו דרייבר המפעיל את כל פונקציות.
- דוגמת הרצה (#הדפסה צפויה):

```
child=ZoneTime(10, 5, 34, 'Montreal')
zt=ZoneTime(10, 55, 34) #10:55:34 \n Jerusalem
print('-----')
zt.printTime()
print('-----')
child.printTime() #10:05:34 \n Montreal
```

בהצלחה !