מטלת מנחה (ממ"ן) 14

הקורס: תכנות וניתוח נתונים בשפת פייתון (20606)

חומר הלימוד למטלה: יחידה 12 | **נושאי המטלה:** תכנות מונחה עצמים

מספר השאלות: 4 | **משקל המטלה:** 3 נקודות

סמסטר: 2025א | **מועד אחרון להגשה:** 25.1.2025

(ת)

שימו לב:

- **יש להקפיד על שמות המחלקות בדיוק כמו שנכתבו.**
- **יש לתעד את התכניות בתיעוד פנימי באנגלית בלבד (בתחילת התכנית התיעוד מסביר מה מבצעת התכנית באופן כללי ובמהלך התכניות התיעוד מסביר את הקוד) על פי תקן PEP 8 ועל פי הדוגמאות בפרק 1.9 באתר הקורס.**
- **אין להוסיף פונקציות מעבר לאלה הנדרשות במטלה במפורש. אלא אם נכתב במפורש שניתן. יחד עם זאת, ניתן להגדיר פונקציות עזר פרטיות (ולא ציבוריות!) במחלקות כרצונכם.**
- **אין להשתמש בחומר מתקדם או שלא נלמד בקורס.**
- **יש להשתמש בקבועים היכן שאפשר.**
- **הקפידו להגדיר את משתני המחלקה כמשתנים פנימיים, אלא אם צוין במפורש אחרת.**
- **יש להקפיד על הזחה (אינדנטציה - עימוד) נכונה, ועל שמות משתנים בעלי משמעות (באנגלית) ולפי המוסכמות בקורס.**
- **יש להקפיד על פורמט הפלט בדיוק כפי שמצוין בשאלה: איות נכון, אותיות גדולות וקטנות, רווחים, וכו.**
- **הגשת המטלה נעשית אך ורק בעזרת מערכת המטלות המקוונת שבאתר הקורס.**
- **אל תשכחו לשמור את מספר האסמכתא שתקבלו מהמערכת לאחר ההגשה.**

מעוניינים לפתח מערכת מידע עבור מרכז מעקב גדילה לתינוקות בקופות חולים.

לצורך כך נגדיר ארבע מחלקות:

- המחלקה Date שמייצגת תאריך. **מחלקה זו נתונה לכם וניתן להורידה באתר הקורס תחת "מטלה 14"**. תיאור המחלקה והפונקציות המוגדרות בה מפורטות בסיום המטלה.
- המחלקה Weight שמייצגת משקל
- המחלקה Baby שמייצגת תינוק
- המחלקה HealthCare מייצגת סניף טיפת חלב בקופת חולים

תזכורת – בכל המטלה עליכם להשתמש בקבועים ולא במספרים, כשצריך.

שאלה 1 - 25 נקודות

המחלקה Weight מייצגת משקל.

למחלקה Weight יש את משתני המופע פנימיים (private members) הבאים:

- תכונה בשם __kilos – שמייצגת את מספר הקילו (שלם חיובי). ערך ברירת המחדל של תכונה זו הוא 1.
- תכונה בשם __grams – שמייצגת את מספר הגרם (שלם בטווח 0-999). ערך ברירת המחדל של תכונה זו הוא 0.

הקפידו להשתמש פעמיים בסימן underscore על מנת לשמור על פרטיות המשתנים.

למחלקה Weight הוגדר בנאי (constructor) המקבל שני פרמטרים (מספר הקילו, מספר גרם) של המשקל.

```
def __init___(self, kilos, grams)
```

אם אחד הפרמטרים (או יותר) אינו חוקי, יש לאתחל את המשקל כולו על פי ערכי ברירת המחדל, כלומר יתקבל המשקל: 1.0

בנוסף הוגדרו במחלקה השיטות הבאות:

- **שיטות האחזור:**

```
get_kilos(self), get_grams(self).
```

שיטות אלו מחזירות את התכונות __kilos וכן __grams בהתאמה.

- השיטה __eq__ המקבלת כפרמטר משקל נוסף ובודקת אם הוא שווה למשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה. במידה והאובייקט המתקבל אינו מסוג Weight יש להחזיר NotImplemented.

חתימת השיטה:

```
def __eq__(self, other)
```

- השיטה __lt__ המקבלת כפרמטר משקל נוסף ובודקת האם המשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה, **קל יותר** מהמשקל שהתקבל כפרמטר. במידה והאובייקט המתקבל אינו מסוג Weight יש להחזיר NotImplemented.

חתימת השיטה:

```
def __lt__(self, other)
```

- השיטה __gt__ המקבלת כפרמטר משקל נוסף ובודקת האם המשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה, **כבד יותר** מהמשקל שהתקבל כפרמטר. **השיטה gt חייבת להשתמש אך ורק בשיטה lt** (היא לא יכולה לחשב את התשובה לפי ערכי התכונות, ולא יכולה להשתמש בשיטה eq או בשיטות אחרות).

חתימת השיטה:

```
def __gt__(self, other)
```

- השיטה __str__ מחזירה מחרוזת תווים המייצגת את המשקל כך: kg.grams **שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים. לדוגמא:**
  - המשקל 3 קילו ו- 55 גרם יוחזר כך: 3.055 .
  - המשקל 4 קילו ו- 5 גרם יוחזר כך: 4.005 .
  - שימו לב, אם יש 0 בסוף ערך הגרמים, הוא לא יופיע במחרוזת. למשל:
  - המשקל 4 קילו ו- 70 גרם יוחזר כך: 4.07 ולא 4.070.
  - המשקל 3 קילו ו- 200 גרם יוחזר כך: 3.2 ולא 3.200
  - המשקל 4 קילו יוחזר כך: 4.0 ולא 4.000

חתימת השיטה:

```
def __str__(self)
```

- השיטה add המקבלת תוספת של מספר גרמים grams ומוסיפה אותם למשקל הנכחי. שימו לב שמספר הגרמים grams יכול להיות גם שלילי. אם לאחר התוספת (השלילית) המשקל יהיה מתחת לערך ברירת המחדל של משקל המוגדר כ- 1.0 ק״ג, השיטה לא תוסיף את הערך grams שהתקבל כפרמטר והמשקל יישאר כי שהיה. במידה והפרמטר grams אינו מטיפוס מספר שלם, יש להחזיר NotImplemented.

לדוגמה,

  - אם המשקל עליו מופעלת השיטה הוא 3.15 ו- grams=900 אזי השיטה add תעדכן את המשקל להיות 4.05
  - אם המשקל עליו מופעלת השיטה הוא 3.15 ו- grams=-250 אזי השיטה add תעדכן את המשקל להיות 2.9

- אם המשקל עליו מופעלת השיטה הוא 3.15 ו- grams=-3250 אזי השיטה add לא תבצע שינוי במשקל כלל.

חתימת השיטה:

```
def add(self, grams)
```

עליכם לכתוב את המחלקה Weight לפי ההגדרות לעיל.

שאלה 2 - 40 נקודות

המחלקה Baby מייצגת תינוק.

למחלקה Baby יש את משתני המופע פנימיים (private members) הבאים:

- תכונה בשם __first_name מטיפוס str – שמייצגת את השם הפרטי של התינוק
- תכונה בשם __last_name מטיפוס str – שמייצגת את שם המשפחה של התינוק
- תכונה בשם __id_num מטיפוס str – שמייצגת את מספר זהות התינוק
- תכונה בשם __date_of_birth מטיפוס Date – שמייצגת את תאריך לידת התינוק
- תכונה בשם __birth_weight מטיפוס Weight – שמייצגת את משקל התינוק בזמן הלידה
- תכונה בשם __current_weight מטיפוס Weight – שמייצגת את משקלו העכשווי של התינוק

הקפידו להשתמש פעמיים בסימן underscore על מנת לשמור על פרטיות המשתנים.

- **למחלקה Baby הוגדר הבנאי (constructor) המקבל את הפרמטרים הבאים:**
  - שמו הפרטי של התינוק. ניתן להניח כי הערך המתקבל מאותחל והמחרוזת אינה ריקה.
  - שם המשפחה של התינוק. ניתן להניח כי הערך המתקבל מאותחל והמחרוזת אינה ריקה.
  - מספר זהות התינוק. **אם כמות התווים אינה חוקית, כלומר מספר הזהות אינו מורכב מ- 9 תווים או התווים במחרוזת אינם מייצגים ספרות, יש לאתחל למחרוזת "000000000"**.
  - יום, חודש ושנה של מועד הולדת התינוק. טיפול תקינות ערכי התאריך יטופלו במסגרת הבנאי במחלקה Date.
  - משקל התינוק בזמן הלידה בגרמים. טיפול תקינות ערך המשקל יטופל במסגרת הבנאי במחלקה Weight.

שימו לב שהבנאי לא מקבל את משקל התינוק העכשווי כפרמטר. כשנוצר האובייקט, המשקל העכשווי יהיה בדיוק כמו המשקל בזמן הלידה בגרמים. **יש להקפיד להימנע מ- aliasing.**

חתימת הבנאי היא:

```
def __init__(self, f_name, l_name, id_num, day, month, year, birth_weight_in_grams)
```

בנוסף הוגדרו במחלקה השיטות הבאות:

הוגדרו שיטות האחזור (get) והשיטות הקובעות (set) לפי החתימות הבאות:

- **שיטות האחזור:**

```
def get_first_name(self)
def get_last_name(self)
def get_id (self)
def get_date_of_birth(self)
def get_birth_weight(self)
def get_current_weight(self)
```

- **השיטה הקובעת:**

```
def set_current_weight(self, weight_to_set)
```

במידה והאובייקט המתקבל אינו מסוג Weight יש לעורר חריגה TypeError. יש להקפיד להימנע מ- aliasing.

שימו לב שיש במחלקה Baby רק שיטה קובעת אחת.

- השיטה __str__ מחזירה מחרוזת תווים המייצגת את התינוק כך:

Name: Ariel Israeli

Id: 123456789

Date of Birth: 03/08/2024

Birth Weight: 3.005

Current Weight: 3.425

שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים. שימו לב לרווח אחרי התו : (נקודותיים) ולירידות השורה. **לאחר השורה האחרונה של המשקל העכשווי יש להוסיף ירידת שורה.**

חתימת השיטה:

```
def __str__(self)
```

- השיטה הבוליאנית __eq__ המקבלת כפרמטר תינוק נוסף ובודקת אם הוא זהה לתינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה. השיטה תחזיר True אם ערכי התינוקות זהים בשמם, תאריך הולדתם ומספרי הזהות שלהם. אחרת, תחזיר False. במידה והאובייקט המתקבל אינו מסוג Baby יש להחזיר NotImplemented.

חתימת השיטה:

```
def __eq__(self, other)
```

- השיטה הבוליאנית are_twins המקבלת כפרמטר תינוק נוסף ובודקת אם הוא תאום של התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה. התינוקות הם תאומים אם שם המשפחה שלהם זהה, השם הפרטי שלהם שונה, מספרי ת"ז של התינוקות שונים זה מזה ותאריך הלידה שלהם זהה או שונה ביום אחד בדיוק (כלומר, נולדו באותו יום או בימים עוקבים). אם כל התנאים מתקיימים, השיטה תחזיר True, ואחרת, תחזיר False. במידה והאובייקט המתקבל אינו מסוג Baby יש להחזיר NotImplemented.

חתימת השיטה:

```
def are_twins(self, other)
```

- השיטה הבוליאנית __gt__ המקבלת כפרמטר תינוק נוסף. השיטה תחזיר True אם התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה כבד יותר לפי המשקל העכשווי. אחרת, תחזיר False. במידה והאובייקט המתקבל אינו מסוג Baby יש להחזיר NotImplemented.

חתימת השיטה:

```
def __gt__(self, other)
```

- השיטה update_current_weight המקבלת כפרמטר grams מספר גרמים לעדכון המשקל העכשווי. השיטה תעדכן את משקל התינוק בתוספת grams גרמים. הערך grams יכול להיות שלילי.

אם לאחר התוספת (השלילית) המשקל יהיה מתחת לערך ברירת המחדל של משקל המוגדר כ- 1.0 ק"ג, השיטה לא תעשה כלום, ולא תוסיף את הערך grams.

שימו לב להבדל בין השיטה הזו לבין השיטה set_current_weight.

חתימת השיטה:

```
def update_current_weight(self, grams)
```

- השיטה הבוליאנית older המקבלת כפרמטר תינוק נוסף. השיטה תחזיר True אם תאריך הולדת התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה **קודם** לתאריך הולדת התינוק שהתקבל כפרמטר. אחרת, יוחזר False. במידה והאובייקט המתקבל אינו מסוג Baby יש להחזיר NotImplemented.

חתימת השיטה:

```
def older (self, other)
```

עליכם לכתוב את המחלקה Baby לפי ההגדרות לעיל.

שאלה 3 - 25 נקודות

המחלקה HealthCare מייצגת סניף טיפת חלב בקופת חולים ומכילה את התינוקות שבמעקב גדילה.

למחלקה HealthCare יש את משתני המופע פנימיים (private members) הבאים :

- תכונה בשם __name מטיפוס str המייצגת את שם הסניף.
- תכונה בשם __babies מטיפוס list המייצגת רשימה של תינוקות המטופלים בסניף.

התינוקות (כלומר האובייקטים מהמחלקה Baby) נמצאים **מתחילת הרשימה ממוינים בסדר עולה על פי תאריך לידתם.**

יש לממש את המחלקה HealthCare לפי הסעיפים להלן:

1. הגדרת התכונות של המחלקה. **הקפידו להשתמש פעמיים בסימן underscore על מנת לשמור על פרטיות המשתנים.**
2. בנאי שמקבל את שם הסניף ומאתחל את שמו לפרמטר שהתקבל וכן בונה רשימת תינוקות ריקה.

חתימת הבנאי היא:

```
def __init__(self, branch_name)
```

3. כתבו את השיטה get_name המחזירה את שם סניף.

חתימת השיטה:

```
def get_name(self)
```

4. כתבו את השיטה num_of_babies המחזירה את מספר התינוקות המטופלים בסניף.

חתימת השיטה:

```
def num_of_babies(self)
```

5. כתבו את השיטה add_baby שמוסיפה תינוק לרשימה. היא מקבלת כפרמטר תינוק baby. יש להוסיף את התינוק תוך שמירה על מיון התינוקות בסדר עולה על פי תאריך לידתו לאחר הוספת התינוק החדש. אפשר להניח שהתינוק החדש לא נמצא כבר ברשימה. במידה וקיימים תינוקות עם תאריך לידה זהה, יש להוסיף את התינוק החדש לפני כל התינוקות עם תאריך הלידה הזהה. במידה והאובייקט המתקבל אינו מסוג Baby יש לעורר חריגה TypeError.

חתימת השיטה:

```
def add_baby(self, baby)
```

6. כתבו את השיטה how_many_above_weight המקבלת משקל ~~בגרמים~~ ומחזירה כמה תינוקות הם מעל משקל זה.

חתימת השיטה:

```
def how_many_above_weight(self, weight)
```

7. כתבו את השיטה average_weight המחזירה את המשקל הממוצע בקרב כל התינוקות במעקב בסניף. האובייקט שיוחזר יכיל מספר קילו ומספר גרם שלמים, כלומר הקפידו לשמור את החלק השלם בלבד (אין צורך בעיגול). במידה ואין תינוקות ברשימה, יש להחזיר None.

חתימת השיטה:

```
def average_weight(self)
```

8. כתבו את השיטה most_heaviest_baby המחזירה את התינוק בעל המשקל המקסימלי. במידה ויש שני תינוקות או יותר בעל משקל מקסימלי, יש להחזיר את התינוק הראשון מביניהם. במידה ואין תינוקות ברשימה, יש להחזיר None. יש להקפיד להימנע מ- aliasing.

חתימת השיטה:

```
def most_heaviest_baby(self)
```

9. כתבו את השיטה babies_above_weight המקבלת משקל w ומחזירה רשימת תינוקות שמשקלן הוא מעל משקל זה. במידה ואין תינוקות מתאימים, יש להחזיר None. במידה והאובייקט המתקבל אינו מסוג Weight יש לעורר חריגה TypeError.

חתימת השיטה:

```
def babies_above_weight(self, weight)
```

10. כתבו את השיטה __str__ המחזירה מחרוזת המכילה את המידע על כל התינוקות המטופלים בסניף. שימו לב להשתמש בשיטות שכבר כתבתם.

חתימת השיטה:

```
def __str__(self)
```

לדוגמה, עבור סניף Raanana ובו התינוקות הבאים:

```
Branch Raanana has 3 babies:
Name: Daniel Cohen
Id: 333333333
Date Of Birth: 25/11/2023
```

```
Birth Weight: 3.05
Current Weight: 3.425

Name: Lior Levi
Id: 222222222
Date Of Birth: 01/07/2024
Birth Weight: 3.15
Current Weight: 3.64

Name: Ariel Goren
Id: 111111111
Date Of Birth: 20/08/2024
Birth Weight: 2.99
Current Weight: 3.303
```

אם אין תינוקות רשומים בסניף תוחזר המחרוזת:

```
"There are no babies in this Health Care branch."
```

שימו לב להקפיד שהמחרוזת שתוחזר תהייה בדיוק בפורמט שלעייל: ללא רווחים מיותרים, ללא תוספת תווים, ללא שגיאות כתיב או החלפה בין אותיות קטנות לגדולות.

שאלה 4 - 10 נקודות

הגדירו מחלקה כרצונכם היורשת מאחת מהמחלקות Weight, Baby או HealthCare במטלה זו.

המחלקה היורשת תכלול משתנה פנימי אחד, לפחות (מעבר לתכונות הנורשות).

בנוסף, יש להגדיר פונקציית בנאי (__init__), פונקציה __str__ ופונקציית השוואת אובייקטים (__eq__). יש להקפיד על תיעוד.

יש להקפיד על מימוש מושכל והימנעות מ- aliasing בהתבסס על תכנות מונחה עצמים ובפרט בפונקציות קיימות במחלקת העל (מחלקת הבסיס).

שימו לב, בכל שאלות המטלה :

- **אסור להוסיף משתנה מופע או מחלקה למחלקות.**
- **מותר להוסיף שיטות פרטיות אבל לא ציבוריות.**
- **אין להשתמש במספרים בקוד. יש להוסיף קבועים עבור כל מספר קבוע ולהשתמש בקבוע בקוד מלבד מקרים טריוויאליים (למשל, הערך 0).**

- **שימו לב לא לבצע aliasing במקומות המועדים.**
- **הקפידו לעשות שימוש בשיטות הקיימות במחלקות שכתבתן ובמחלקה Date הנתונה לכם.**
- **אתם צריכים לכתוב בעצמכם docstring למחלקה, לבנאי ולשיטות לפי הנהוג בכתיבת docstring. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט. דוגמאות לתיעוד ניתן למצוא בפרק 1.9 באתר הקורס.**

שימו לב ששמנו טסטרים לשלושת המחלקות באתר הקורס. חובה שטסטרים אלו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרים לא ירוצו בגלל שגיאות קומפילציה הציון במטלה יהיה אפס.

הגשה

1. הגשת הממ״ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לשאלה 1 יכלול את הקובץ Weight.py, הפתרון לשאלה 2 יכלול את הקובץ Baby.py, הפתרון לשאלה 3 יכלול את הקובץ HealthCare.py. הפתרון לשאלה 4 יכלול את הקובץ ClassName.py, בהתאם לשם המחלקה שהגדרתם.
3. ארזו את קובץ הפתרון בקובץ zip **(ולא rar)** יחיד ושלחו אותו בלבד.
4. **אל תשכחו לשמור את מספר האסמכתא שקיבלתם מהמערכת לאחר ההגשה. אם לא קיבלתם מספר אסמכתא, סימן שההגשה לא התקבלה.**
5. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורסת את ההגשה הקודמת. **אבל עשו זאת אך ורק עד לתאריך ההגשה**. אחרי התאריך, ייחשב לכם כאילו הגשתם באיחור, גם אם ההגשה הראשונה היתה בזמן! כמו כן, אם המנחה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

בהצלחה

תיאור המחלקה Date (מייצגת תאריך)

למחלקה Date יש את משתני המופע פנימיים (private members) הבאים:

- תכונה בשם __day – שמייצגת את היום (שלמים בין 1 ל- 31)
- תכונה בשם __month – שמייצגת את החודש (שלמים בין 1 ל- 12)
- תכונה בשם __year – שמייצגת את השנה (שלמים בין 1000 ל- 9999)

למחלקה Date הוגדר הבנאי (constructor) הבא:

בנאי המקבל שלושה פרמטרים (יום, חודש ושנה) של התאריך.

```
def Date(self, day=1, month=1, year=2024)
```

אפשר להניח שהפרמטרים הם מספרים שלמים אבל אי אפשר להניח שהתאריך שמתקבל הוא חוקי. שימו לב, אם אחד הפרמטרים (או יותר) אינו חוקי (למשל, הוא מספר שלילי), או שהתאריך אינו חוקי (למשל 30.2.2013), האובייקט שצריך להיווצר הוא של ה- 1 בינואר בשנת 2024. יש להתייחס לשנים מעוברות (בלוח הגרגוריאני) בהן בחודש פברואר יש 29 ימים. ראו: כאן.

אם לא מתקבלים פרמטרים, נוצר התאריך ה-1 בינואר 2024.

בנוסף הוגדרו במחלקה השיטות הבאות:

- **שיטות האחזור:**

```
get_day(self), get_month(self), get_year(self).
```

- **השיטות הקובעות:**

```
set_day(self, day_to_set), set_month(self, month_to_set),
set_year(self, year_to_set).
```

בשיטות הקובעות, אם אחד הפרמטרים אינו חוקי או שלאחר ההשמה ייווצר תאריך שאינו חוקי, התאריך שבאובייקט **לא ישתנה** בכלל, ויישאר כמו שהיה.

- השיטה __eq__ המקבלת כפרמטר תאריך מסוים ובודקת אם הוא זהה לתאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה. במידה והאובייקט המתקבל אינו מסוג Date יש להחזיר NotImplemented.

חתימת השיטה:

```
def __eq__(self, other)
```

- השיטה __lt__ המקבלת כפרמטר תאריך מסוים ובודקת האם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **קודם** לתאריך שהתקבל כפרמטר. במידה והאובייקט המתקבל אינו מסוג Date יש להחזיר NotImplemented.

חתימת השיטה:

```
def __lt__(self, other)
```

- השיטה __gt__ המקבלת כפרמטר תאריך מסוים ובודקת האם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **מאוחר** מהתאריך שהתקבל כפרמטר. **השיטה gt חייבת להשתמש אך ורק בשיטה lt** (היא לא יכולה לחשב את התשובה לפי ערכי התכונות, ולא יכולה להשתמש בשיטה eq או בשיטות אחרות).

חתימת השיטה:

```
def __gt__(self, other)
```

- השיטה difference המקבלת כפרמטר תאריך מסוים, ומחשבת ומחזירה את ההפרש בימים בין התאריך המיוצג על ידי האובייקט עליו מופעלת השיטה, לבין התאריך המיוצג על ידי האובייקט שהועבר כפרמטר. שימו לב שמספר זה צריך להיות תמיד אי שלילי (כלומר, לא משנה מי מהתאריכים קודם לאחר). במידה והאובייקט המתקבל אינו מסוג Date יש להחזיר None.

חתימת השיטה:

```
def difference (self, other)
```

- השיטה __str__ מחזירה מחרוזת תווים המייצגת את התאריך כך: day/month/year בפורמט dd/mm/yyyy. **שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים. לדוגמא:** התאריך 12 במאי 2019 יוחזר כך 12/05/2019
שימו לב **שאין** רווח לפני ואחרי התו /
כמו כן, יש צורך להוסיף 0 אם היום או החודש הוא בן ספרה אחת. כך למשל אם התאריך הוא אחד בפברואר בשנת 2020, המחרוזת שתוחזר תהיה 01/02/2020

חתימת השיטה:

```
def __str__(self)
```

- השיטה tomorrow מחזירה תאריך של היום שלמחרת התאריך המיוצג על ידי האובייקט עליו מופעלת השיטה. **שימו לב שצריך להחזיר תאריך חדש** ולא לשנות את האובייקט עליו מופעלת השיטה. יש להניח שהתאריך עליו מופעלת השיטה אינו 31/12/9999.

לדוגמא,

  - אם התאריך עליו מופעלת השיטה הוא 14/12/2019 אזי השיטה tomorrow תחזיר את התאריך 15/12/2019
  - אם התאריך עליו מופעלת השיטה הוא 28/02/2021 אזי השיטה tomorrow תחזיר את התאריך 01/03/2021

חתימת השיטה:

```
def tomorrow(self)
```