

# מטלת מנחה (ממ"ן) 12

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 3 - 4 נושאי המטלה: שימוש במחלקות נתונות וכתובת מחלקות

מספר השאלות: 2 משקל המטלה: 4 נקודות

סמסטר: 2020 מועד אחרון להגשה: 14.12.2019

(ת)

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים.

בעלי הכלבו (Supermarket) רוצים למחשב את מלאי המצרכים שברשותם.

לצורך כך נגדיר שלוש מחלקות:

המחלקה Date שתייצג תאריך;

המחלקה FoodItem שתייצג מוצר מזון;

המחלקה Stock שתייצג את המלאי בכלבו,

את שתי המחלקות Date ו-FoodItem נממש במטלה זו, ואת המחלקה Stock נממש במטלה הבאה.

## שאלה 1 - 35 נקודות

המחלקה Date מייצגת תאריך:

למחלקה Date יש את התכונות הפרטיות (instance variables) הבאות:

- int \_day – שמייצגת את היום (שלמים בין 1 ל-31);
- int \_month – שמייצגת את החודש (שלמים בין 1 ל-12);
- int \_year – שמייצגת את השנה (שלמים חיוביים בני ארבע ספרות);

למחלקה Date הוגדרו שני בנאים (constructors):

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

```
public Date(int day, int month, int year)
```

אי אפשר להניח שהפרמטרים נכונים ואי אפשר להניח שהתאריך שמתקבל הוא חוקי. שימו לב, אם אחד הפרמטרים (או יותר) אינו חוקי, או שהתאריך אינו חוקי (למשל 30.2.2013), האובייקט שצריך להיווצר הוא של ה-1 בינואר בשנת 2000. לא לשכוח להתייחס לשנים מעוברות בהן בחודש פברואר יש 29 ימים.

- השני - בנאי העתקה המקבל תאריך אחר, ומעתיק את ערכיו.

```
public Date (Date other)
```

## בנוסף הוגדרו במחלקה השיטות הציבוריות:

- פעולות האחזור:

`getDay(), getMonth(), getYear()`.

- הפעולות הקובעות:

`setDay(int dayToSet), setMonth(int monthToSet),  
setYear(int yearToSet)`.

גם בפעולות הקובעות, אם אחד הפרמטרים אינו חוקי או שיווצר תאריך שאינו חוקי, התאריך שבאובייקט **לא ישתנה** בכלל, וישאר כמו שהיה.

- השיטה `equals` המקבלת כפרמטר תאריך מסוים ובודקת אם הוא זהה לתאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה.

- השיטה `before` המקבלת כפרמטר תאריך מסוים ובודקת האם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **קודם** לתאריך שהתקבל כפרמטר.

- השיטה `after` המקבלת כפרמטר תאריך מסוים ובודקת האם התאריך שמיוצג על ידי האובייקט עליו מופעלת השיטה, **מאוחר** מהתאריך שהתקבל כפרמטר. **השיטה `after` חייבת להשתמש אך ורק בשיטה `before`** (היא לא יכולה לחשב את התשובה לפי ערכי התכונות, ולא יכולה להשתמש בשיטה `equals`).

- השיטה `difference` המקבלת כפרמטר תאריך מסוים, ומחשבת מה ההפרש בימים בין התאריך המיוצג על ידי האובייקט עליו מופעלת השיטה, לבין התאריך המיוצג על ידי האובייקט שהועבר כפרמטר. שימו לב שמספר זה תמיד יהיה אי שלילי.

- השיטה `toString` מחזירה מחרוזת תוים המייצגת את התאריך כך: **day/month/year**. שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים.

**לדוגמא:** התאריך 12 במאי 2019 ייווצר 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 שתיווצר תהיה **01/02/2020**

- 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.

**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

- השיטה dayInWeek, מחזירה באיזה יום בשבוע חל התאריך המיוצג על ידי האובייקט עליו מופעלת השיטה. השיטה מחזירה את היום באמצעות מספר שלם בתחום 0 עד 6, כאשר 1 הוא Sunday (יום ראשון), 2 הוא Monday (יום שני) וכך הלאה. המספר 0 הוא Saturday (שבת).

**מציאת היום בשבוע נעשית בעזרת הנוסחה הבאה:**

$$\text{Day} = (D + (26 \times (M+1))/10 + Y + Y/4 + C/4 - 2 \times C) \bmod 7$$

כאשר:

D - הוא מספר היום בחודש.

M - הוא מספר החודש. מספור החודשים מתחיל ממרץ כך: מרץ - 3, אפריל - 4, וכן הלאה עד ינואר - 13, פברואר - 14. **שימו לב:** ינואר ופברואר מטופלים בתור חודשים 13 ו-14 של השנה הקודמת.

Y - הן שתי הספרות האחרונות בשנה.

C - הן שתי הספרות הראשונות בשנה (המאה).

**הערות:**

- החילוק הוא של מספרים שלמים, כך שלמשל 19/4 שווה 4 ולא 4.75.
- הפעולה mod מחזירה את השארית (איך כותבים זאת ב-Java?).
- שימו לב שלפעמים התוצאה היא מספר שלילי. במקרה כזה עליכם להעביר את התוצאה למספר בין 0 ל-6. אחת הדרכים לעשות זאת היא ע"י השיטה floorMod

Math.floorMod (num,7)

**דוגמה:**

נפעיל את הנוסחה על התאריך 15 לינואר 2009.

חודש ינואר מטופל כחודש מספר 13 של השנה הקודמת, **כלומר 2008**.

לכן, D=15, C=20, M=13, Y=8 - בגלל הכלל המיוחד לטיפול בחודשים ינואר ופברואר.

$$\begin{aligned} \text{Day} &= (D + (26 \times (M+1))/10 + Y + Y/4 + C/4 - 2 \times C) \bmod 7 = \\ &= (15 + (26 \times (13+1))/10 + 8 + 8/4 + 20/4 - 2 \times 20) \bmod 7 = \\ &= (15 + 36 + 8 + 2 + 5 - 40) \bmod 7 = \\ &= 26 \bmod 7 = \\ &= 5 \end{aligned}$$

כלומר יום חמישי.

**שימו לב שאנחנו מספקים לכם את השיטה הבאה אשר מחשבת את מספר הימים שחלפו מתחילת הספירה (על פי הלוח הגרגוריאני): כאן אין צורך להגדיר קבועים עבור המספרים בקוד.**

```
// computes the day number since the beginning of the Christian counting of years
private int calculateDate ( int day, int month, int year) {
    if (month < 3) {
        year--;
        month = month + 12;
    }
    return 365 * year + year/4 - year/100 + year/400 + ((month+1) * 306)/10 + (day - 62);
}
```

אתם רשאים להשתמש בשיטה זו אם אתם מוצאים לנכון.

עליכם לכתוב את המחלקה Date לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס **ביחידה 4**, בתת-פרק של מטלה 12.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונות `_month` ו-`_day` ו-`_year`. אין להשתמש במספרים בקוד. יש להוסיף קבועים (final) עבור כל מספר קבוע ולהשתמש בקבוע בקוד. מותר להגדיר קבועים נוספים למחלקה. אתם יכולים להגדיר שיטות פרטיות נוספות על אלו שהוגדרו לעיל, אבל לא שיטות ציבוריות ולא תכונות נוספות.

## שאלה 2 - 65 נקודות

בחנות הכלבו יש מוצרי מזון שונים. לשם כך הוגדרה מחלקה בשם `FoodItem` המייצגת מוצר מזון.

המידע שאנו רוצים לאחסן על כל מוצר הוא :

- שם המוצר (name) מטיפוס String ;
- מספר קטלוגי (catalogueNumber) מטיפוס long ; המספר צריך להיות חיובי.
- כמות המוצר במלאי (quantity) מטיפוס int ; המספר צריך להיות אי-שלילי.
- תאריך הייצור של המוצר (productionDate) ;
- תאריך התפוגה של המוצר (expiryDate) ;
- הטמפרטורה המינימלית בה יכול להיות המוצר (minTemperature) מטיפוס int ;
- הטמפרטורה המקסימלית בה יכול להיות המוצר (maxTemperature) מטיפוס int ;
- מחיר המוצר (price) מטיפוס int ; המספר צריך להיות חיובי.

התכונות שם המוצר, מספר קטלוגי, והטמפרטורות (המינימלית והמקסימלית) של מוצר המזון יהיו immutable, כלומר אי אפשר לשנות אותן. הערך שהתקבל בבנאי נשאר לאורך כל חייו של האובייקט. אין פעולה קובעת לגביו.

למחלקה FoodItem הוגדרו הבנאים הבאים :

- בנאי המקבל שמונה פרמטרים המהווים את פרטי התכונות של האובייקט. שימו לב, התאריכים (ייצור ותפוגה) מתקבלים כאובייקטים ולא כשלשות של מספרים שלמים.  
**אי אפשר להניח כי כל הפרמטרים לבנאי הם חוקיים כשלעצמם: -אבל-** אם תאריך התפוגה של המוצר הוא לפני תאריך הייצור שלו, צריך לשנות את תאריך התפוגה ליום אחד אחרי תאריך הייצור. כמו כן, אם הטמפרטורה המינימלית שהתקבלה כפרמטר גדולה מהטמפרטורה המקסימלית שהתקבלה כפרמטר, צריך להחליף ביניהן. **אם שם המוצר הוא מחרוזת ריקה יש לתת את השם "item".** אם מספר הקטלוגי אינו מספר חיובי בן 4 ספרות יש לתת ערך קטלוגי 9999. אם הכמות שלילית יש לתת ערך 0 ואם המחיר אינו חיובי יש לתת ערך 1. שימו לב להגדיר קבועים עבור כל ערך ברירת מחדל.
- בנאי העתקה המקבל מוצר מזון אחר, ומעתיק את ערכיו.

במחלקה הוגדרו פעולות אחזור (get) וקביעה (set) לפי השמות המקובלים. בפעולות ה-set, אם הפרמטרים לא חוקיים לא ישתנה כלום. שימו לב לעניין ה-immutable.

כמו כן הוגדרו השיטות הבאות :

- השיטה equals המקבלת כפרמטר פריט מזון אחר ומחזירה true אם הם שווים בכל התכונות חוץ מתכונת הכמות, ו-false אחרת.
- השיטה הבוליאנית isFresh המקבלת תאריך מסוים d כפרמטר, ומחזירה true אם בתאריך הזה המוצר טרי (כלומר, נמצא לאחר תאריך הייצור שלו ולפני תאריך התפוגה שלו, כולל). אם לא, היא מחזירה false.
- השיטה toString המחזירה את תוכן האובייקט כמחרוזת תווים בצורה הבאה :

FoodItem: milk CatalogueNumber: 1234 ProductionDate: 14/12/2019 ExpiryDate: 21/12/2019 Quantity: 3

שימו לב, בין שם המוצר לבין המילה "Catalogue Number" יש טאב "\t", כמו גם בין מספר הקטלוגי למילה "ProductionDate" יש טאב "\t", כמו גם בין תאריך הייצור למילה "ExpiryDate" וכמו בין תאריך התפוגה למילה "Quantity". **שימו לב, בהדפסה כאן המחרוזת מתפרשת על שתי שורות, אבל בהדפסת הפלט שלכם היא תהיה על שורה אחת.**

- השיטה olderFoodItem המקבלת כפרמטר מוצר מזון אחר ומחזירה true אם המוצר עליו מופעלת השיטה יוצר לפני המוצר שהתקבל כפרמטר, ו-false אחרת. במילים

**אחרות: מחזירה true** אם המוצר שהתקבל כפרמטר יוצר אחרי המוצר עליו מופעלת השיטה, ו- false אחרת.

- השיטה `howManyItems` המקבלת מספר שלם שמייצג סכום של כסף, ומחזירה כמה פריטים של המוצר ניתן לקנות בסכום זה. שימו לב שצריך לבדוק גם אם יש כמות מספיקה של פריטים במלאי.  
לדוגמא, אם מחירו של המוצר הוא 2 ₪, ויש במלאי 4 פריטים, והסכום שמתקבל כפרמטר לשיטה הוא 10, אמנם אפשר לקנות בסכום זה 5 פריטים, אבל כיוון שיש רק 4 פריטים במלאי, השיטה תחזיר את הערך 4.
- השיטה `isCheaper` המקבלת כפרמטר מוצר מזון אחר ומחזירה **true** אם המוצר עליו מופעלת השיטה הוא זול יותר המוצר שהתקבל כפרמטר, ו- **false** אחרת. במילים אחרות: **מחזירה true** אם המוצר שהתקבל כפרמטר הוא יקר יותר מהמוצר עליו מופעלת השיטה, ו- **false** אחרת.

**עליכם לכתוב את המימוש של המחלקה `FoodItem`, לפי החתימות של השיטות שנמצאות בטבלה להלן. פרטים מדויקים על מקרי קצה אפשר למצוא ב- API באתר הקורס.**

<i>FoodItem (String name, long catalogueNumber, int quantity, Date productionDate, Date expiryDate, int minTemperature, int maxTemperature, int price )</i>	<b>בנאים</b>
<i>FoodItem (FoodItem other)</i>	
<i>long getCatalogueNumber()</i>	<b>שיטות מאחזרות</b>
<i>String getName()</i>	
<i>int getQuantity()</i>	
<i>Date getProductionDate()</i>	
<i>Date getExpiryDate()</i>	
<i>int getMinTemperature()</i>	
<i>int getMaxTemperature()</i>	
<i>int getPrice()</i>	

<i>void setQuantity (int n)</i>	<b>שיטות קובעות</b>  בשיטות הקובעות אם הפרמטר אינו תקין (בכמות, אם הפרמטר הוא שלילי, בתאריכים, אם תאריך הייצור יהיה לאחר השינוי לאחר תאריך התפוגה, ובמחיר, אם הפרמטר אינו חיובי) יש להשאיר את התכונה ללא שינוי.
<i>void setProductionDate (Date d)</i>	
<i>void setExpiryDate (Date d)</i>	
<i>void setPrice (int n)</i>	
<i>boolean equals (FoodItem other)</i>	שיטה המקבלת כפרמטר מוצר מזון אחר ובודקת האם הם שווים
<i>boolean isFresh (Date d)</i>	שיטה המקבלת תאריך מסוים d כפרמטר, ומחזירה true אם בתאריך הזה המוצר טרי. אם לא, היא מחזירה false
<i>String toString()</i>	שיטה המחזירה את תוכן האובייקט כמחרוזת תווים
<i>boolean olderFoodItem (FoodItem other)</i>	שיטה המקבלת כפרמטר מוצר מזון אחר ומחזירה true אם המוצר שעליו מופעלת השיטה יוצר לפני המוצר שהתקבל כפרמטר, ו- false אחרת.
<i>int howManyItems (int amount)</i>	המקבלת מספר שלם שמייצג סכום של כסף, ומחזירה כמה פריטים של המוצר ניתן לקנות בסכום זה. יש לבדוק גם אם יש כמות מספיקה של פריטים במלאי.
<i>boolean isCheaper (FoodItem other)</i>	שיטה המקבלת כפרמטר מוצר מזון אחר ומחזירה true אם המוצר שעליו מופעלת השיטה הוא זול יותר מהמוצר שהתקבל כפרמטר, ו- false אחרת.

שימו לב, אסור להוסיף תכונות פרטיות. מותר להוסיף שיטות פרטיות.  
אין להשתמש במספרים בקוד. יש להוסיף קבועים (final) עבור כל מספר קבוע  
ולהשתמש בקבוע בקוד. מותר להגדיר קבועים נוספים למחלקה.  
בכל השיטות במטלה שמקבלות אובייקט כפרמטר אפשר להניח שמתקבל  
אובייקט שאותחל ואינו שווה ל- null.

## הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

שימו לב ששמנו טסטרים לשתי 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 ללא שגיאות קומפילציה הציון  
במטלה יהיה אפס.

שימו לב לא לבצע aliasing במקומות המועדים.  
עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן  
להשתמש בהערות ה-API שנמצאות באתר.

### הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת יורדו לכם הרבה נקודות!**
3. חובה להריץ את הטסטרים שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרים לא מכסים את כל האפשרויות, ובפרט לא את מקרי הקצה. הם רק בודקים את השמות של השיטות במחלקות. מאד מומלץ להוסיף להם בדיקות. שימו לב שאם הטסטרים לא יעברו קומפילציה מול המחלקות שכתבתם, לא יקבלו נקודות בכלל. אם יש שיטה שאתם מעוניינים לדלג עליה, עלכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטרים יעברו קומפילציה.
4. את התשובות לשאלות יש להגיש בשני קובצי Java הבאים: FoodItem.java, Date.java.
5. ארזו את כל הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

## ב ה צ ל ח ה