

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6
נושא המטלה: לולאות ומערכים
מספר השאלות: 2
משקל המטלה: 5 נקודות
סמסטר: 2020
מועד אחרון להגשה: 28.12.2019

(ת)

חלק א – מערך חד-ממדי

בחלק הראשון של מטלה זו אנו משתמשים במחלקות Date ו- FoodItem שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקות Date ו- FoodItem שכתבתם, או בקבצים Date.class ו- FoodItem.class שאנו שמנו באתר ביחידה 6 בפרק של מטלה 13. אנא קראו את הכתוב במדריך שנמצא ביחידה 1 ב"מדריכי עזר וקישורים" בדף הבית של האתר הקורס. המדריך נקרא:

מדריך עזר קצר לשימוש בקובצי מחלקות (class.) הקיימים בפרוייקט שאתם יוצרים ב-BlueJ קובץ pdf

כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד. שמנו באתר טסטר בסיסי לבדיקה ראשונית של המטלה. חובה להריץ את המטלה מול הטסטר ולבדוק שאין טעויות קומפילציה.

שאלה 1 - להרצה (70%)

המחלקה Stock מייצגת את המלאי בכלבו.

הייצוג נעשה על-ידי מערך ששומר את מוצרי המזון. התכונות במחלקה הן:

- מערך של המוצרים `FoodItem [] _stock`
 - מספר המוצרים שיש בכלבו בפועל `int _noOfItems`
- שימו לב שכל מוצר נספר פעם אחת גם אם יש בו יותר מפריט אחד. כמו כן קיים קבוע מספרי המציין את הגודל המקסימלי של המערך – 100.

המוצרים (כלומר האובייקטים מהמחלקה FoodItem) נמצאים במערך ברצף, ללא "חורים", והמוצרים ממוינים בסדר עולה מהקטן לגדול לפי המספר הקטלוגי. המערך צריך להישאר כך (ממין וללא חורים) לאחר כל פעולה.

הבנאי של המחלקה הוא בנאי ברירת-מחדל (default constructor) היוצר אובייקט שבו מערך בגודל המקסימלי של 100 מוצרי מזון. המערך ריק בהתחלה.

עליכם לממש ב-Java את המחלקה Stock לפי הסעיפים להלן:

1. הגדרת התכונות של המחלקה.
2. בנאי שמאתחל את תכונות המחלקה כך שהמערך יהיה בגודל מקסימלי.
3. השיטה `getNumOfItems` המחזירה את מספר המוצרים במערך (לפי התכונה).
4. השיטה הבוליאנית `addItem` המקבלת מוצר `FoodItem`, ומוסיפה אותו למלאי. אם המוצר קיים כבר במלאי, היא מוסיפה את הכמות שהיתה בתכונה `quantity` ב-`FoodItem` לכמות המוצר במלאי. אם המוצר לא נמצא עדיין במלאי, היא מוסיפה אותו למערך. השיטה מחזירה `true` אם הפעולה הצליחה, ו-`false` אחרת (מתי זה קורה?) **שימו לב כי מוצר בעל אותו שם ואותו ברקוד שיש לו תאריכי ייצור ו/או תפוגה שונים יתפוס מקום חדש במערך ולא יתווסף למוצר הקיים. במקרה זה יש להכניס את המוצר החדש לפני המוצר שכבר קיים במלאי. אם יש מוצר בעל אותו שם ומספר קטלוגי, ניתן להניח באופן אוטומטי שאם קיים שוני בין המוצרים הוא נובע מתאריכי הייצור והתפוגה ולא מתכונה אחרת כגון טמפ' אחסון או מחיר.**
5. השיטה `order` המחזירה מחרוזת תווים המכילה רשימה של שמות המוצרים שצריך להזמין. מוצר ייכנס לרשימת ההזמנה אם הכמות שיש ממנו במלאי קטנה מ-`amount` (שהועבר לשיטה כפרמטר). **השיטה לא תבצע את ההזמנה עצמה**, כלומר לא להוריד מהמלאי את המוצרים, אלא להחזיר רשימה בלבד.
לדוגמא: אם צריך להזמין את המוצרים `milk, bread, cheese` הם יודפסו כך:
`milk, bread, cheese`
בין שמות המוצרים יש פסיק ואחריו רווח אחד.
- שימו לב**, אם במלאי יש כמה מוצרים בעלי אותו שם ומספר קטלוגי, שנבדלים אחד מהשני בתאריכי הייצור והתפוגה או בפרטים אחרים (ולכן אינם יושבים באותו תא במערך), הכמות שיש במלאי היא הסכום של הכמויות של אותם מוצרים. **למשל**, נניח ובמערך יש בתא `k` אובייקט של `milk` עם מספר קטלוגי 1234 עם תאריך ייצור 1/1/2020 וכמות 3, ובתא `k+1` אובייקט של `milk` עם מספר קטלוגי 1234 עם תאריך ייצור 31/12/2020 וכמות 4 (הערכים בשאר השדות לא רלוונטיים כרגע). אם הכמות שהועברה כפרמט `amount` היא 5, המוצר `milk` **לא ייכנס** לרשימה, שכן יש ממנו 7 פריטים. **שימו לב שעל הרשימה להכיל את המוצרים לפי סדר הופעתם ב-Stock.**
6. השיטה `howMany` המקבלת מספר שלם `temp` שמהווה טמפרטורה שיש במקרר מסוים, ומחזירה את מספר המוצרים שיכולים לעבור למקרר זה (כלומר, שהטמפרטורה של המקרר מתאימה לאחסון שלהם). שימו לב, אם במלאי יש 5 יוגורטים שיכולים לעבור למקרר, אז צריך לספור אותם כ-5 ולא כאילו זה מוצר אחד (יוגורט). אם המערך ריק יש להחזיר 0.
7. השיטה `removeAfterDate` המקבלת כפרמטר תאריך `d` ומוחקת מהמלאי את כל מוצרי המזון שתאריך התפוגה שלהם הוא לפני התאריך `d` (כלומר, את כל המוצרים שהם

מקולקלים בתאריך d, כי תאריך התפוגה שלהם כבר עבר). אם אין כאלו, השיטה לא תעשה כלום. שימו לב שצריך למחוק ממש את המוצר מהמערך, ולשים לב שלא יישארו חורים במערך.

8. השיטה mostExpensive המחזירה את המוצר הכי יקר במלאי. אם המערך ריק יש להחזיר null. אם

9. השיטה howManyPieces המחזירה כמה פריטים יש במלאי (כלומר, אם ממוצר מסוים יש 3 פריטים, הוא נספר כ- 3). אם המערך ריק יש להחזיר 0.

10. השיטה toString המחזירה את תוכן האובייקט כמחרוזת תווים בצורה הבאה:

FoodItem: HotDog	Catalogue num:1111	ProductionDate: 14/12/2019	ExpiryDate: 17/12/2019	Quantity: 3
FoodItem: Milk	Catalogue num:1234	ProductionDate: 14/12/2019	ExpiryDate: 17/12/2019	Quantity: 3
FoodItem: Milk	Catalogue num:1234	ProductionDate: 14/12/2019	ExpiryDate: 18/12/2019	Quantity: 2
FoodItem: Honey	Catalogue num:4456	ProductionDate: 14/12/2019	ExpiryDate: 19/12/2019	Quantity: 2

11. השיטה updateStock שמעדכנת את המלאי. השיטה מקבלת כפרמטר מערך של מחרוזות תווים, שמייצג את המוצרים שנמכרו, והיא מעדכנת את המלאי לאחר המכירה. כלומר, מורידה את הכמות בהתאם מכל מוצר ומוצר שבמערך המחרוזות.

לדוגמא, אם מערך המחרוזות הוא:

{"milk", "cheese", "milk", "bread"}

אז השיטה מורידה מהמלאי הקיים:

- 2 milk
- 1 cheese
- 1 bread

אפשר להניח שכל המוצרים הנדרשים נמצאים במלאי ויש כמות מספיקה מהם. אם לאחר העדכון הכמות של המוצר היא 0, צריך למחוק ממש את המוצר מהמערך ולוודא שאין חורים במערך. אם יש מוצר בעל אותו שם המופיע כמה פעמים במלאי – יש להוריד את הכמות תחילה מהמוצר הראשון ורק לאחר מכן לעבור למוצר הבא במערך.

12. השיטה getTempOfStock המחזירה את הטמפרטורה בה צריך להיות המקרר שמכיל את כל מוצרי המזון. אם יש טווח כזה, השיטה תחזיר את הטמפרטורה המינימלית בטווח. אם אין אפשרות למצוא טמפרטורה שמתאימה לכל המוצרים במקרר, יוחזר הערך המקסימלי Integer.MAX_VALUE (המספר המקסימלי האפשרי במספרים שלמים).

לדוגמא, אם יש 3 פריטים:

- פריט 1 שצריך להיות בין טמפרטורה 7 לטמפרטורה 10
- פריט 2 שצריך להיות בין טמפרטורה 6 לטמפרטורה 11
- פריט 3 שצריך להיות בין טמפרטורה 9 לטמפרטורה 13

אזי המקרר יכול להיות בטמפרטורה 9 עד 10 כדי שכל הפריטים יוכלו להיות בו. לכן תוחזר הטמפרטורה 9 (המינימלית ביניהם).

אם נוסף עוד פריט שצריך להיות בין טמפרטורה 14 ל-17, למשל, אין טמפרטורה שהמקרה יהיה בה ויוכל להכיל את כל הפריטים, ולכן השיטה תחזיר Integer.MAX_VALUE. אם המערך ריק, יש להחזיר Integer.MAX_VALUE.

שימו לב לא לבצע aliasing במקומות המועדים.

מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם. אסור להוסיף תכונות.

אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט. המימוש אשר תכתבו צריך להיות בהתאם ל-API אשר נמצא כאן להלן. את הערות ה-API אתם צריכים לכתוב בעצמכם.

Constructor Summary	
<u>Stock</u> ()	
Method Summary	
boolean	<u>addItem</u> (FoodItem newItem)
int	<u>getNumOfItems</u> ()
String	<u>order</u> (int amount)
int	<u>howMany</u> (int temp)
void	<u>removeAfterDate</u> (Date d)
FoodItem	<u>mostExpensive</u> ()
int	<u>howManyPieces</u> ()
void	<u>updateStock</u> (String[] itemsList)
int	<u>getTempOfStock</u> ()

חלק ב – מערך דו-ממדי

שאלה 2 - להרצה (30%)

המחלקה Matrix מייצגת מטריצה = מערך דו-ממדי של מספרים שלמים בטווח 0-255. המטריצה מייצגת תמונה דו-ממדית בשחור-לבן, כאשר המספרים מייצגים את גווני האפור בתמונה 0 הוא לבן ו-255 הוא שחור.

עליכם לממש את המחלקה לפי הפירוט הבא:

תכונות – מותרת רק תכונה אחת שמייצגת את המערך.

מותר להוסיף קבועים כפי רצונכם.

- **בנאים** – עליכם לממש שני בנאים לפי הממשק הבא:

Constructor Summary	
Matrix (int[] array)	Constructs a Matrix from a two-dimensional array; the dimensions as well as the values of this Matrix will be the same as the dimensions and values of the two-dimensional array.
Matrix (int size1, int size2)	Constructs a size1 by size2 Matrix of zeroes.

size1 מייצג את מספר השורות ו-size2 מייצג את מספר העמודות. ניתן להניח כי הפרמטר array המועבר לבנאי הראשון מכיל מטריצה בגודל 1 על 1 לפחות, והערכים במטריצה זו הם בטווח 0-255. ניתן להניח כי הפרמטרים size1 ו-size2 המועברים לבנאי השני הם חיוביים. שימו לב לא לבצע aliasing בבנאי שמקבל מערך.

- **ממשו שיטת toString() למחלקה Matrix.** אלמנטים באותה שורה יופרדו על-ידי tab (המחרוזת "\t"), ושורות יופרדו על-ידי מעבר שורה.

לדוגמה :

אם המטריצה היא זו :

	0	1	2
0	3	8	72
1	4	6	60
2	253	2	1

המחרוזת שתוחזר מהשיטה toString() שתופעל על המטריצה היא :

"3"+"\\t"+"8"+"\\t"+"72"+"\\n"+"4"+"\\t"+"6"+"\\t"+"60"+"\\n"+"253"+"\\t"+"2"+"\\t"+"1"+"\\n"

כלומר, המחרוזת : "3\\t8\\t72\\n4\\t6\\t60\\n253\\t2\\t1\\n"

ולכן היא תודפס כך :

3	8	72
4	6	60
253	2	1

שימו לב לא להוסיף tab בסוף השורות.

- הוסיפו למחלקה Matrix את השיטה public Matrix makeNegative()
המחזירה את תמונת הנגיב של התמונה. כלומר, כל נקודה בתמונה (תא במערך) שהיה שחור הופך להיות לבן, כל תא שהיה לבן הופך להיות שחור, וכך גם כל גווני האפור מתהפכים (למספר המשלים אותם ל-255). ניתן להניח שהמטריצה מכילה מספרים בטווח 0-255.

יש להחזיר מטריצה חדשה ואין לשנות את המטריצה המקורית.

לדוגמה, אם התמונה המקורית היא זו :

	0	1	2	3	4
0	19	124	28	35	38
1	115	22	25	230	31
2	9	21	22	249	230
3	0	6	9	232	255
4	2	5	10	116	129

אז תמונת הנגטיב, שצריכה להיות מוחזרת מהשיטה היא :

	0	1	2	3	4
0	236	131	227	220	217
1	140	233	230	25	224
2	246	234	233	6	25
3	255	249	246	23	0
4	253	250	245	139	126

- הוסיפו למחלקה Matrix את השיטה `public Matrix imageFilterAverage()` המחזירה את התמונה שמתקבלת מהחלקת התמונה. פעולת ההחלקה (שנקראת לפעמים גם הסרת רעש מתמונה או טשטוש), נעשית על-ידי כך שכל תא בתמונה הנוכחית הופך להיות בתמונה החדשה הערך של הממוצע שלו עם שכניו (בתמונה הנוכחית). אנחנו נתייחס לשמונת שכניו הצמודים אליו. שימו לב שלפעמים לתא אין שמונה שכנים, אלא רק חמישה או אפילו רק שלושה (מתי?) שימו לב שתוצאת הממוצע צריכה גם היא להיות מטיפוס `int`.

לדוגמא, אם התמונה המקורית היא זו :

	0	1	2	3	4
0	19	124	28	35	38
1	115	22	25	230	31
2	9	21	22	249	230
3	0	6	9	232	255
4	2	5	10	116	129

אז התמונה המוחלקת, שצריכה להיות מוחזרת מהשיטה היא :

	0	1	2	3	4
0	70	55	77	64	83
1	51	42	84	98	135
2	28	25	90	142	204
3	7	9	74	139	201
4	3	5	63	125	183

- הוסיפו למחלקה Matrix את השיטה `public Matrix rotateClockwise()` המסובבת את התמונה ב- 90 מעלות עם כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לדוגמה, אם התמונה המקורית היא זו :

	0	1	2	3
0	19	124	28	35
1	115	22	25	230
2	19	21	22	249
3	0	16	9	232
4	62	35	10	116

אז התמונה לאחר השיטה היא :

	0	1	2	3	4
0	62	0	19	115	19
1	35	16	21	22	124
2	10	9	22	25	28
3	116	232	249	230	35

- הוסיפו למחלקה את השיטה `public Matrix rotateCounterClockwise()` המסובבת את התמונה ב- 90 מעלות נגד כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לצורך מימוש שיטות אלו, מותר לממש שיטות נוספות (פרטיות), לפי ראות עיניכם.
יש להחזיר מטריצה חדשה ואין לשנות את המטריצה המקורית.

ניתן להניח שהמטריצה מכילה מספרים בטווח 0-255.
דאגו לכך שהקוד יהיה ברור וקריא, וכרגיל, מתועד על-פי כללי javadoc ותיעוד פנימי.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקות שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקות שכתבתם בטסטר זה, ולהגיש אותן רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב- API את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקבצים `Stock.java`, `Matrix.java` עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

בהצלחה