

# מטלת מנחה (ממ"ן) 12

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

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

כתיבת מחלקות

מספר השאלות: 3 משקל המטלה: 3 נקודות

סמסטר: 2025 מועד אחרון להגשה: 3.5.2025

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים. כדי לעמוד על ההבדל בין המימוש לממשק של מחלקה, תתבקשו לכתוב שני מימושים שונים למחלקה המייצגת זמן.

## שאלה 1 - 30 נקודות

המחלקה Time1 מייצגת זמן בשעות, דקות ושניות.

להלן תיאור המחלקה:

למחלקה Time1 יש את התכונות הפרטיות (instance variables) הבאות:

- תכונה מסוג int בשם `_hour` – שמייצגת את השעות (בין 0 ל-23);
- תכונה מסוג int בשם `_minute` – שמייצגת את הדקות (בין 0 ל-59);
- תכונה מסוג int בשם `_second` – שמייצגת את השניות (בין 0 ל-59);

למחלקה Time1 הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שלושה פרמטרים המהווים את ערכי התכונות שיהיו לזמן.

```
public Time1 (int h, int m, int s)
```

אם אחד הפרמטרים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0. שאר הפרמטרים לא צריכים להיות מאותחלים ל-0.

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

```
public Time1 (Time1 other)
```

בנוסף הוגדרו במחלקה השיטות הבאות:

- שיטות האחזור:

- `int getHour()` המחזירה את השעות.
- `int getMinute()` המחזירה את הדקות.
- `int getSecond()` המחזירה את השניות.

- השיטות הקובעות :

- void setHour (int num) המשנה את ערכה של השעה להיות num. אם num הוא לא בתחום 0-23, הערך של \_hour לא משתנה.
- void setMinute (int num) המשנה את ערכה של הדקה להיות num. אם num הוא לא בתחום 0-59, הערך של \_minute לא משתנה.
- void setSecond (int num) המשנה את ערכה של השניה להיות num. אם num הוא לא בתחום 0-59, הערך של \_second לא משתנה.

- השיטה toString() שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm:ss. כך, המחרוזת "07:30:00" מייצגת את הזמן של שבע שעות ושלושים דקות ואפס שניות. **שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים.** יש להקפיד שהשיטה תחזיר מחרוזת בת 8 תווים בדיוק.

public String toString()

- השיטה secFromMidnight שמחזירה כמה שניות עברו מאז חצות הלילה. למשל, אם הזמן המאוחר באובייקט הוא 07:30:20, יוחזר הערך 27020.

public int secFromMidnight()

- boolean equals (Time1 other) – שיטה שמקבלת זמן כפרמטר ומחזירה האם הזמן שעליו הופעלה השיטה והזמן שהתקבל כפרמטר זהים. אם כן, השיטה תחזיר true ואם לא, יוחזר false.

public boolean equals (Time1 other)

- boolean before (Time1 other) - שיטה שמקבלת זמן כפרמטר ומחזירה האם הזמן שעליו הופעלה השיטה הוא לפני הזמן שהתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. למשל, 07:30:00 הוא לפני 13:24:30.

public boolean before (Time1 other)

- boolean after (Time1 other) - שיטה שמקבלת זמן כפרמטר ומחזירה האם הזמן שעליו הופעלה השיטה הוא אחרי הזמן שהתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false.

**השיטה הזו משתמשת אך ורק בשיטה before שהוגדרה לעיל. אסור להשתמש בשום אופרטור אחר או שיטה אחרת.**

public boolean after (Time1 other)

- `int difference (Time1 other)` המקבלת כפרמטר זמן מסוים ומחזירה את ההפרש בשניות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. שימו לב, אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.

`public int difference(Time1 other)`

עליכם לכתוב את המחלקה `Time1` לפי ההגדרות לעיל. הגדרות מדויקות לפי API תמצאו באתר הקורס.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונות `_hour` , `_minute` ו- `_second` מותר להגדיר קבועים נוספים למחלקה. אתם יכולים להגדיר שיטות פרטיות נוספות על אלו שהוגדרו לעיל, אבל לא שיטות ציבוריות ולא תכונות נוספות.

## שאלה 2 - 30 נקודות

גם המחלקה `Time2` מייצגת זמן, אבל היא עושה זאת לפי מספר השניות שעברו מאז חצות הלילה ועד לזמן שמייצג האובייקט.

לדוגמא, אם הזמן שמיוצג על ידי האובייקט הוא: 07:30:20 (כלומר 7 בבוקר, 30 דקות ו- 20 שניות), הוא ייוצג על-ידי הערך 27020 שכן,  $7 * 60 * 60 + 30 * 60 + 20 = 27020$

למחלקה `Time2` יש, אם כן, תכונה פרטית (instance variable) אחת, והיא מספר השניות הזה. התכונה היא מטיפוס `int` והיא נקראת `_secFromMid`

עליכם לכתוב את המימוש של המחלקה `Time2`.

ה-API של שתי המחלקות `Time1` ו-`Time2` זהה לחלוטין! רק הייצוג הפנימי של האובייקטים (התכונות) שונה.

בנאים	<code>Time2 (int h, int m, int s)</code>
אם אחד הפרמטרים שהתקבל בבנאי שמקבל 3 ערכים אינו בתחום הנכון, אז לא משתמשים בערכו כדי לחשב את הערך של התכונה הפרטית. (כלומר משנים את ערכו של כל פרמטר שאינו בתחום ל-0)	<code>Time2 (Time2 other)</code>

<i>int getHour()</i>	<b>שיטות מאחזרות</b>
<i>int getMinute()</i>	
<i>int getSecond()</i>	
<i>void setHour (int num)</i>	<b>שיטות קובעות</b> בשיטות הקובעות אם הפרמטר אינו תקין יש להשאיר את התכונה ללא שינוי.
<i>void setMinute (int num)</i>	
<i>void setSecond (int num)</i>	
<i>int secFromMidnight ()</i>	<b>שיטה</b> המחזירה כמה שניות עברו מאז חצות הלילה
<i>boolean equals(Time2 other)</i>	<b>שיטה</b> הבודקת האם הזמנים שווים
<i>boolean before(Time2 other)</i>	<b>שיטה</b> הבודקת האם הזמן שעליו הופעלה השיטה קודם לזמן שהתקבל כפרמטר
<i>boolean after(Time2 other)</i>	<b>שיטה</b> הבודקת האם הזמן שעליו הופעלה השיטה מאוחר לזמן שהתקבל כפרמטר (השיטה הזו יכולה להשתמש אך ורק בשיטה <i>before</i> ולא בהשוואות נוספות, ולא בשיטות נוספות)
<i>int difference(Time2 other)</i>	<b>שיטה</b> המחזירה את ההפרש בשניות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.
<i>String toString()</i>	<b>שיטה</b> המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm:ss. כך, המחרוזת "07:30:00" מייצגת את הזמן של שבע שעות ושלושים דקות ואפס שניות. שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים. יש להקפיד שהשיטה תחזיר מחרוזת בעלת 8 תווים בדיוק.

**שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונה `_secFromMid`.**

מותר להגדיר קבועים נוספים למחלקה.

במילים אחרות, חתימות השיטות של המחלקה Time1 זהות לחלוטין לאלו של Time2. לבד מהמקרים בהם מתקבל זמן כפרמטר לשיטה, ואז במקום שכתוב Time1 צריך להיות Time2. שימו לב, גם השיטה toString של המחלקה Time2 צריכה להיות לפי זו של המחלקה Time1, כלומר להדפיס את הזמן לפי שעות, דקות ושניות ולא לפי מספר השניות מאז חצות. אין להשתמש בשיטות ובבנאים של המחלקה Time1 במחלקה Time2. מדובר במימושים חלופיים למחלקה של הזמן.

### שאלה 3 - 40 נקודות

המחלקה BusArrival מייצגת הגעה של אוטובוס לתחנה.

הייצוג נעשה על-ידי שלוש תכונות:

- מספר הקו של האוטובוס – מספר שלם מטיפוס int בתחום 1-99 הנקרא `_lineNumber`
- זמן הגעת האוטובוס לתחנה – אובייקט מהמחלקה Time1 הנקרא `_arrivalTime`
- מספר הנוסעים באוטובוס – מספר שלם מטיפוס int בתחום 0-70 בשם `_noOfPassengers`

למחלקה קיים קבוע מספרי המציין את המספר המקסימלי של נוסעים באוטובוס – 70. חשבו אילו קבועים נוספים עליכם להוסיף.

למחלקה BusArrival הוגדרו שלושה בנאים:

1. בנאי המקבל חמישה פרמטרים: מספר שלם המהווה את מספר הקו, מספר שלם המהווה את מספר הנוסעים באוטובוס ושלושה מספרים שלמים המהווים את זמן ההגעה של האוטובוס בשעות (בתחום 0-23), דקות (בתחום 0-59) ושניות (בתחום 0-59). אם אחד הפרמטרים של הזמן או של מספר הנוסעים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0, אם הפרמטר של מספר הקו אינו בתחום, הוא צריך להיות מאותחל ל-1.  
`public BusArrival(int lineNum, int pass, int h, int m, int s)`

2. בנאי המקבל שלושה פרמטרים: מספר שלם המהווה את מספר הקו, מספר שלם המהווה את מספר הנוסעים באוטובוס וזמן ההגעה מתקבל כאובייקט מהמחלקה Time1. אם הפרמטר של מספר הנוסעים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0, אם הפרמטר של מספר הקו אינו בתחום, הוא צריך להיות מאותחל ל-1.  
`public BusArrival(int lineNum, int pass, Time1 t)`

3. בנאי העתקה, המקבל אובייקט מהמחלקה BusArrival ומעתיק את ערכיו.

public BusArrival (BusArrival other)

## השיטות במחלקה הן:

- שיטות האחזור (get) והשיטות הקובעות (set) לפי השמות המקובלים לכל התכונות של המחלקה. בשיטות הקובעות אם הפרמטר אינו תקין יש להשאיר את התכונה ללא שינוי.
- השיטה הבוליאנית equals המקבלת אובייקט אחר מהמחלקה BusArrival ומחזירה true אם הם שווים בערכיהם ו- false אחרת.
- השיטה toString המחזירה ייצוג של האובייקט כמחרוזת תווים. בדיוק לפי התבנית הבאה (ללא רווחים או סימנים אחרים, וכמובן ללא הגרשיים):

"Bus no. 27 arrived at 09:24:10 with 13 passengers"

- השיטה הבוליאנית fuller המקבלת הגעת אוטובוס נוסף כפרמטר ומחזירה true אם האוטובוס עליו מופעלת השיטה עמוס יותר בנוסעים מאשר האוטובוס שהתקבל כפרמטר, ו- false אחרת.
- השיטה הבוליאנית before המקבלת הגעת אוטובוס נוסף כפרמטר ומחזירה true אם האוטובוס עליו מופעלת השיטה הגיע לתחנה לפני האוטובוס שהתקבל כפרמטר, ו- false אחרת.
- השיטה הבוליאנית isFull המחזירה true אם האוטובוס מלא ואין מקום להעלות עליו נוסעים נוספים ו- false אחרת.
- השיטה elapsedTime המקבלת הגעת אוטובוס נוסף כפרמטר ומחזירה את ההפרש בדקות (שלמות) בערך מוחלט בין הגעת האוטובוס עליו מופעלת השיטה לתחנה לבין הגעת האוטובוס שהתקבל כפרמטר. שימו לב שעליכם להחזיר את מספר הדקות ולא את מספר השניות. אם לדוגמא עברו 170 שניות, השיטה צריכה להחזיר את המספר 2 שכן רק שתי דקות שלמות עברו בין האוטובוסים. אתם לא יכולים להניח שהאוטובוס שעליו מופעלת השיטה הגיע לפני האוטובוס שהתקבל כפרמטר. השיטה תמיד תחזיר ערך חיובי.

<i>BusArrival(int lineNum, int pass, int h, int m, int s)</i>	בנאים
<i>BusArrival(int lineNum, int pass, Time t)</i>	
<i>BusArrival(BusArrival other)</i>	
<i>Time getArrivalTime()</i>	שיטות מאחזרות
<i>int getLineNum()</i>	
<i>int getNoOfPass()</i>	

<i>void setArrivalTime(Time1 t)</i>	שיטות קובעות
<i>void setLineNum(int num)</i>	
<i>void setNoOfPass (int num)</i>	
<i>boolean equals(BusArrival other)</i>	שיטה הבודקת האם שני האובייקטים זהים.
<i>String toString()</i>	שיטה המחזירה ייצוג של האובייקט כמחרוזת תווים
<i>boolean fuller (BusArrival other)</i>	שיטה הבודקת אם האוטובוס עמוס יותר מאשר other
<i>boolean before (BusArrival other)</i>	שיטה הבודקת אם האוטובוס הגיע לתחנה לפני other
<i>boolean isFull()</i>	שיטה הבודקת אם האוטובוס מלא עד אפס מקום
<i>int elapsedTime (BusArrival other)</i>	שיטה המחזירה את מספר הדקות השלמות שעברו בין הגעות שני האוטובוסים לתחנה

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

עליכם לכתוב את המחלקה `BusArrival` לפי ההגדרות לעיל (ולפי הכתוב ב-API שבאתר).

אפשר וכדאי, כמובן, להשתמש במחלקה `Time1` ובשיטות שלה בכתיבת המחלקה `BusArrival`. לשם כך אין צורך לייבא את המחלקה `Time1`. מספיק שהיא תהיה באותו פרויקט (תיקיה) של המחלקה `BusArrival`.

שימו לב לא לבצע aliasing במקומות המועדים.

שימו לב:

- עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר.
- בכל המחלקות אין להוסיף תכונות מעבר למתואר במטלה. יש להוסיף קבועים (final). כמו כן מותר להוסיף משתני עזר פנימיים לשיטות למטרת חישובים.
- בכל המחלקות ניתן להניח שכל השיטות שמקבלות אובייקט כפרמטר אינן מקבלות null.

- בכל המחלקות אסור להוסיף תכונות פרטיות, אך מותר להוסיף שיטות פרטיות.
- בכל המחלקות אין להשתמש במספרים בקוד. יש להוסיף קבועים (final) עבור כל מספר קבוע ולהשתמש בקבוע בקוד.
- אין להוסיף הדפסות ביניים במהלך השיטות!

באתר הקורס, ביחד עם מטלה 12 שמנו טסטרס לשלוש המחלקות. הטסטרס בודקים אם השיטות שלכם כתובות נכון (מבחינה תחבירית בלבד!) חובה עליכם להריץ את הטסטרס הללו על המחלקות שכתבתם, ולבדוק שהם עוברים קומפילציה, לפני שאתם מגישים את המטלה. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרס ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרס לא ירוצו ללא שגיאות קומפילציה הציון במטלה יהיה אפס ללא מתן אפשרות לתיקון. כדאי כמובן להוסיף בדיקות לטסטרס שיבדקו שהמחלקות עובדות נכון גם מבחינה לוגית.

## 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 הטסטרס שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרס 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 הטסטרס לא יעברו קומפילציה מול המחלקות שכתבתם, לא יקבלו נקודות בכלל **ללא אפשרות ערעור**. אם יש שיטה שאתם מעוניינים לדלג עליה, עלכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטרס יעברו קומפילציה.
4. **אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.**
5. את התשובות לשאלות יש להגיש בשלושה קובצי Java הבאים: Time1.java, Time2.java, BusArrival.java
6. ארזו את כל הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

## ב ה צ ל ח ה