

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: 3 משקל המטלה: 4 נקודות
סמסטר: 2020 מועד אחרון להגשה: 25.4.2020

(ת)

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

שאלה 1 - 35 נקודות

המחלקה Time1 מייצגת זמן -

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

- `int _hour` – שמייצגת את השעה (בין 0 ל-23);
- `int _minute` – שמייצגת את הדקה (בין 0 ל-59).

למחלקה Time1 הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים (שעה ודקה)

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

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

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

```
public Time1 (Time1 t)
```

בנוסף הוגדרו במחלקה השיטות:

- שיטות האחזור:

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

- 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.

- השיטה toString() שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. **שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים.** יש להקפיד שהשיטה תחזיר מחרוזת בת 5 תווים בדיוק.

```
public String toString()
```

- השיטה minFromMidnight שמחזירה כמה דקות עברו מאז חצות הלילה. למשל, אם הזמן המאוחר באובייקט הוא 07:30, יוחזר הערך 450.

```
public int minFromMidnight()
```

- השיטה equals המקבלת כפרמטר זמן מסוים ובודקת אם הוא זהה לזמן שמיוצג על ידי האובייקט עליו מופעלת השיטה. אם כן, השיטה תחזיר true ואם לא, יוחזר false.

```
public boolean equals (Time1 other)
```

- השיטה before המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה **קודם** בזמן לאובייקט שמתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. למשל, 07:30 הוא לפני 13:24.

```
public boolean before (Time1 other)
```

- השיטה after המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה **מאוחר** בזמן לאובייקט שמתקבל 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)
```

- השיטה difference המקבלת כפרמטר זמן מסוים ומחזירה את ההפרש בדקות 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)
```

- השיטה addMinutes המקבלת כפרמטר מספר שלם num המייצג מספר דקות, ומוסיפה אותו לזמן המיוצג על-ידי האובייקט עליו מופעלת השיטה (this). השיטה מחזירה אובייקט חדש מהמחלקה Time1 המייצג את הזמן החדש. האובייקט עליו מופעלת השיטה addMinutes לא משתנה. אם הפרמטר num הוא שלילי, השיטה מחסירה את מספר הדקות הזה מהאובייקט. שימו לב לדאוג לכך שאם ישנה גלישה ליום הבא (או יותר) **אן אם יש גלישה אחורה בזמן ליום אחד או יותר**, עדיין התכונות צריכות להיות בטווחים של 0-23 ו-59-0. אין צורך לציין באיזושהי צורה את העובדה שעבר יום.

```
public Time1 addMinutes(int num)
```

עליכם לכתוב את המחלקה Time1 לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס ביחידה 4 בתת-פרק העוסק במטלה 12.

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

שאלה 2 - 35 נקודות

המחלקה `Time2` מייצגת זמן, לפי מספר הדקות שעברו מאז חצות הלילה ועד לזמן שמייצג האובייקט.

לדוגמא, אם הזמן שמיוצג על ידי האובייקט הוא: 07:30 (כלומר 7 בבוקר, 30 דקות), הוא ייוצג על-ידי הערך 450 שכן, $450 = 30 + 7 * 60$

למחלקה `Time2` יש, אם כן, תכונה פרטית (instance variable) אחת, והיא מספר הדקות הזה.

`int _minFromMid`

עליכם לכתוב את המימוש של המחלקה `Time2`.

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

<code>Time2 (int h, int m)</code>	בנאים
<code>Time2 (Time2 other)</code>	אם אחד הפרמטרים שהתקבל בבנאי שמקבל 2 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>int getHour()</code>	שיטות מאחזרות
<code>int getMinute()</code>	
<code>void setHour (int num)</code>	שיטות קובעות בשיטות הקובעות אם הפרמטר אינו תקין יש להשאיר את התכונה ללא שינוי
<code>void setMinute (int num)</code>	
<code>int minFromMidnight ()</code>	שיטה המחזירה כמה דקות עברו מאז חצות הלילה
<code>boolean equals(Time2 other)</code>	שיטה הבודקת האם הזמנים שווים
<code>boolean before(Time2 other)</code>	השיטה בודקת האם הזמן שעליו הופעלה השיטה מוקדם מהזמן שהתקבל כפרמטר

<i>boolean after(Time2 other)</i>	השיטה בודקת האם הזמן שעליו הופעלה השיטה מאוחר לזמן שהתקבל כפרמטר (השיטה הזו יכולה להשתמש אך ורק בשיטה <i>before</i> ולא בהשוואות נוספות)
<i>int difference(Time2 other)</i>	שיטה המחזירה את ההפרש בדקות 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>	שיטה המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. שימו לב לדיוק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים. יש להקפיד שהשיטה תחזיר מחרוזת בעלת 5 תווים בדיוק.
<i>Time2 addMinutes(int num)</i>	שיטה המקבלת כפרמטר מספר שלם num המייצג מספר דקות, ומוסיפה אותו לזמן המיוצג על-ידי האובייקט עליו מופעלת השיטה (this). השיטה מחזירה אובייקט חדש מהמחלקה Time2 המייצג את הזמן החדש.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונה `_minFromMid`.

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 ודקות ולא לפי מספר הדקות מאז חצות.

אין להשתמש בשיטות ובבנאים של המחלקה Time1 במחלקה Time2. מדובר במימושים חלופיים למחלקה של הזמן.

שאלה 3 - 30 נקודות

המחלקה Train מייצגת נסיעה של רכבת.

למחלקה Train התכונות הפרטיות (instance variables) הבאות :

- String _destination – יעד הנסיעה.
- Time1 _departure – זמן היציאה של הנסיעה.
- int _duration – משך זמן הנסיעה בדקות.
- int _passengers – מספר הנוסעים בנסיעה.
- int _seats – מספר המושבים ברכבת.
- int _price – מחיר לכרטיס נסיעה לאדם.

למחלקה Train יש שני בנאים :

- בנאי אחד שמקבל כפרמטרים : יעד הנסיעה, שני מספרים שלמים המהווים את זמן היציאה, מספר שלם המייצג את משך זמן הנסיעה בדקות, מספר שלם המייצג את מספר הנוסעים בנסיעה, מספר שלם המייצג את מספר המושבים ברכבת ומספר שלם המייצג מחיר כרטיס נסיעה לאדם.

שימו לב להנחות הבאות :

- אם מספר הנוסעים גדול ממספר המושבים ברכבת, הערך שיינתן לתכונה של מספר הנוסעים יהיה מספר המושבים ולא הפרמטר.
- אם מספר הנוסעים קטן מ-0, ינתן הערך 0.
- אם מספר המושבים קטן מ-0, ינתן הערך 0.
- אם הפרמטר המייצג את משך זמן הנסיעה בדקות קטן מ-0, הערך שיינתן לתכונה של זמן הנסיעה יהיה 0.
- אם הפרמטר המייצג את מחיר כרטיס הנסיעה קטן מ-0, הערך שיינתן לתכונה זו יהיה 0.
- אם המספרים המהווים את זמן היציאה שגויים, הטיפול בכך יתבצע במחלקה Time1.
- בנאי העתקה המקבל נסיעת רכבת אחרת, ומעתיק את ערכיה.

במחלקה הוגדרו פעולות get ו-set לפי השמות המקובלים. ראו פרטים מדויקים ב-API וראו שם גם הנחיות לטיפול במקרי קצה.

כמו כן הוגדרו השיטות הבאות :

- equals שיטה המקבלת נסיעת רכבת אחרת כפרמטר ומחזירה true אם נסיעת הרכבת שעליה השיטה מופעלת ונסיעת הרכבת שהתקבלה כפרמטר זהות. הזהות תיקבע לפי יעד הנסיעה, מספר המושבים וזמן היציאה.

- `getArrivalTime` שיטה המחשבת את זמן ההגעה ליעד של הנסיעה ומחזירה זמן זה (כאובייקט `Time1` מטיפוס).
- `addPassengers` שיטה בוליאנית המקבלת מספר של נוסעים `num`, ומוסיפה אותם לנסיעה, אם יש בה מקום. אם יש, היא מחזירה `true`, אם אין מקום לכולם, היא לא מוסיפה אף אחד, ומחזירה `false`. ניתן להניח שהשיטה מקבלת מספר חיובי.
- `isFull` שיטה בוליאנית המחזירה `true` אם נסיעת הרכבת מלאה (כלומר, כל המושבים תפוסים), ו-`false` אחרת.
- `isCheaper` שיטה המקבלת נסיעת רכבת אחרת כפרמטר ומחזירה `true` אם כרטיס לנסיעה עליה מופעלת השיטה הוא זול יותר מכרטיס לנסיעה שהתקבלה כפרמטר, אחרת יוחזר `false`.
- `totalPrice` שיטה המחשבת את התשלום הכולל שהתקבל מכל נוסעי הנסיעה, ומחזירה ערך זה.
- `arrivesEarlier` שיטה המקבלת נסיעת רכבת אחרת כפרמטר ומחזירה `true` אם הנסיעה עליה מופעלת השיטה מגיעה ליעדה **בזמן מוקדם יותר מאשר** לנצי הנסיעה שהתקבלה כפרמטר, אחרת יוחזר `false`.
- `toString` שיטה המחזירה מחרוזת ובה נתוני הנסיעה הבאים (בלבד), לפי הפורמט הבא:
Train to `_destination` departs at `_departure`. Train is full.
אם הנסיעה אינה מלאה, יודפס `Train is not full`.

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.
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 לשלוש 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 בשלושה קובצי Java הבאים: Time1.java, Time2.java, Train.java
5. ארזו את כל הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

ב ה צ ל ח ה