

מטלת מנחה (ממ"ן) 12

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

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

מספר השאלות: 3 משקל המטלה: 4 נקודות

סמסטר: 2023 מועד אחרון להגשה: 3.12.2022

(ת)

במטלה זו עליכם לכתוב מחלקות עבור מערכת שתשמש סוכנות להשכרת רכב. תתבקשו לממש מחלקה המייצגת מכונית בצי המכוניות של הסוכנות, מחלקה המייצגת תאריך ולבסוף מחלקה המשתמשת בשתי מחלקות אלה על מנת לייצג השכרה של רכב בסוכנות. אנא קראו בעיון את כל ההוראות. תמצאו בהן מידע מועיל שיקל ויבהיר את העבודה.

שאלה 1 - 20 נקודות

המחלקה Car מייצגת מכונית –

לכל רכב בצי המכוניות יש לשמור את הפרטים הבאים: מספר רישוי בן 7 ספרות, דירוג (רמה A, רמה B, רמה C או רמה D), יצרן (למשל "Mazda"), האם לרכב תיבת הילוכים ידנית או אוטומטית, לשם כך עליכם להגדיר למחלקה Car את התכונות הפרטיות (instance variables) הבאות:

- `int _id` – מספר הרישוי של המכונית
- `char _type` – דירוג
- `String _brand` – יצרן
- `boolean _isManual` – האם תיבת ההילוכים ידנית

אין להוסיף תכונות נוספות! מותר (ורצוי) להוסיף קבועים כרצונכם.

עליכם לכתוב שני בנאים (constructors) למחלקה Car:

- האחד - בנאי המקבל ארבעה פרמטרים.

`public Car(int id, char type, String brand, boolean isManual)`
הערות לגבי תקינות הערכים: מספר הרישוי חייב להיות מספר חיובי בן 7 ספרות. אם המספר שהתקבל אינו תקין יש לשמור במקומו את ערך ברירת המחדל 9999999 (ערך זה יש לשמור כקבוע). דירוג הרכב חייב להיות אחד מהערכים A, B, C, D. אם הערך שהתקבל אינו תקין יש לשמור את הערך A. תוכלו להניח שהמחרוזת המייצגת את שם ייצרן הרכב תקינה ואינה null.

- השני - בנאי העתקה המקבל מכונית אחרת, ומעתיק את ערכיה.

`public Car(Car other)`

בנוסף עליכם לכתוב במחלקה את השיטות הבאות:

- שיטות אחזור:

- `int getId()` – מחזירה את מספר הרישוי של המכונית
- `char getType()` – מחזירה את הדירוג של המכונית
- `String getBrand()` – מחזירה את שם הייצרן של המכונית
- `boolean isManual()` – מחזירה האם תיבת ההילוכים ידנית

- שיטות הקובעות:

- `void setId(int id)` – קובעת את מספר הרישוי של המכונית
- `void setType(char type)` – קובעת את הדירוג של המכונית
- `void setBrand(String brand)` – קובעת את שם הייצרן של המכונית
- `void setIsManual(boolean manual)` – קובעת האם תיבת ההילוכים ידנית

בכל השיטות הקובעות אין להניח שהפרמטרים תקינים ויש לבדוק אותם על פי הכללים שתוארו לעיל. אם הפרמטר אינו תקין אין לבצע שינוי כלל.

- השיטה `toString()` שמחזירה את תוכן האובייקט כמחרוזת תווים. לפניכם שלוש דוגמאות לייצוג נכון של מכונית. עליכם לבנות את המחרוזות בדיוק בפורמט זה.

```
id:1234567 type:A brand:Mazda gear>manual
id:8877665 type:D brand:Toyota gear:auto
id:9999999 type:B brand:Renault gear>manual
```

הקפידו על רווחים נכונים (אין רווחים משני צידי תו הנקודתיים, רווח בודד בין חלקי המחרוזת). אין צורך לרדת שורה בסוף המחרוזת.
חתימת השיטה:

```
public String toString()
```

- השיטה `equals` המקבלת כפרמטר מכונית אחרת ובודקת אם היא זהה למכונית שמיוצגת על ידי האובייקט עליו מופעלת השיטה. אם כן, השיטה תחזיר `true` ואם לא, יוחזר `false`. שתי מכוניות נחשבות זהות אם הן בעלות אותו דירוג, אותו יצרן ואותה תיבת הילוכים. (כלומר מספר הרישוי לא רלוונטי להשוואה). תוכלו להניח שהפרמטר אינו `null`.
חתימת השיטה:

```
public boolean equals (Car other)
```

- השיטה `better` המקבלת כפרמטר מכונית אחרת ומחזירה `true` אם המכונית שעליה מופעלת השיטה "טובה" יותר מאשר המכונית האחרת, אחרת `false`. מכונית תיחשב לטובה יותר ממכונית אחרת אם הדירוג שלה גבוה יותר (לדוגמה, דירוג B גבוה יותר מדירוג A). אם לשתיהן אותו הדירוג אז מכונית אוטומטית תיחשב טובה יותר מידנית.
חתימת השיטה:

```
public boolean better (Car other)
```

- השיטה worse המקבלת כפרמטר מכונית אחרת ומחזירה true אם המכונית שעליה מופעלת השיטה "גרועה" יותר מאשר המכונית האחרת, אחרת false (לפי אותם שיקולים שנזכרו לעיל). השיטה הזו משתמשת אך ורק בשיטה better שהוגדרה לעיל. אסור להשתמש בשום אופרטור אחר או שיטה אחרת.
חתימת השיטה :

```
public boolean worse (Car other)
```

עליכם לכתוב את המחלקה Car לפי ההגדרות לעיל.

שאלה 2 - 30 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).
```

בפעולות הקובעות, אם אחד הפרמטרים אינו חוקי או שלאחר ההשמה ייווצר 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.
חתימת השיטה :

```
public boolean equals (Date other)
```

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

`public boolean before (Date other)`

- 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). חתימת השיטה :

`public boolean after (Date 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). חתימת השיטה :

`public int difference (Date other)`

- השיטה toString מחזירה מחרוזת תווים 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 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 :

`public String toString()`

- 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. תוכלו 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 :

`public Date tomorrow()`

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.

במחלקה זו מומלץ לכתוב שיטת עזר בוליאנית פרטית המקבלת שלושה מספרים שלמים המייצגים יום, חודש ושנה ובודקת האם התאריך המיוצג על ידי מספרים אלה הוא חוקי. שיטה זו תאפשר לכתוב את השיטות הנדרשות בצורה פשוטה וקלה (בפרט את השיטה tomorrow).
הגדרות מדויקות לפי API תמצאו באתר הקורס ביחידה 4, בתת-פרק של מטלה 12.

שאלה 3 - 50 נקודות

המחלקה Rent מייצגת השכרה של רכב.

למחלקה Rent התכונות הפרטיות (instance variables) הבאות:

- שם הלקוח – `String _name`
- המכונית המושכרת – `Car _car`
- תאריך קבלת המכונית – `Date _pickDate`
- תאריך החזרת המכונית – `Date _returnDate`

למחלקה Rent יש שני בנאים:

- בנאי אחד שמקבל כפרמטרים: מחרוזת המייצגת את שם הלקוח, אובייקט מסוג Car שמייצג את המכונית המושכרת, תאריך קבלת המכונית ותאריך החזרת המכונית (אובייקטים מסוג Date). תוכלו להניח שכל הפרמטרים אינם null וששם הלקוח תקין. עליכם לוודא שתאריך ההחזרה הוא לפחות יום אחד אחרי תאריך הקבלה. אם לא, קיבעו את תאריך ההחזרה להיות יום אחד אחרי תאריך הקבלה.

```
public Rent (String name, Car car, Date pick, Date ret)
```

- בנאי העתקה המקבל השכרה אחרת, ומעתיק את ערכיה.

```
public Rent (Rent other)
```

עליכם לכתוב את השיטות הבאות:

- שיטות אחזור וקביעה (get ו-set) לפי השמות המקובלים. ראו פרטים מדויקים ב-API וראו שם גם הנחיות לטיפול במקרי קצה.
- equals - שיטה המקבלת השכרה אחרת כפרמטר ומחזירה true אם ההשכרה שעליה השיטה מופעלת וההשכרה שהתקבלה כפרמטר זהות. שתי השכרות נחשבות זהות אם שם הלקוח שלהן זהה, תאריכי הקבלה וההחזרה שלהן זהים והמכוניות שלהן זהות (על פי הזהות שהוגדרה במחלקה Car). חתימת השיטה:

```
public boolean equals(Rent other)
```

- howManyDays - שיטה המחזירה את מספר ימי ההשכרה. לדוגמה, אם תאריך הקבלה הוא 1.2.2022 ותאריך ההחזרה הוא 5.2.2022 אז השיטה תחזיר 4. חתימת השיטה:

```
public int howManyDays()
```

- getPrice - שיטה המחשבת את המחיר הכולל של תקופת ההשכרה. לכל אחת מדרגות הרכב מוגדר מחיר ליום: דרגה A – 100 ₪. דרגה B 150 ₪. דרגה C 180 ₪. דרגה D 240 ₪. בנוסף, על שבוע מלא של השכרה (7 ימים) מקבלים 10% הנחה על המחיר לאותו שבוע. לדוגמה, השכרה של רכב מדרגה A ל-5 ימים תעלה 500 ₪. השכרה של רכב מדרגה A לשבוע תעלה 630 ₪. אבל, השכרה של רכב מדרגה A ל-9 ימים תעלה 830 ₪ (מחיר שבוע בהנחה ועוד ויומיים בתשלום

מלא). השכרה של אותו רכב ל-15 ימים תעלה 1360 ₪. הקפידו על חישוב נכון (כדאי לנסות מספר רב ומגוון של דוגמאות) ושימוש נאות בקבועים.
חתימת השיטה :

```
public int getPrice()
```

- upgrade - מטרתה של שיטה זו לשדרג את המכונית המושכרת ללקוח למכונית "טובה" יותר ולחשב מהי תוספת המחיר שהלקוח יצטרך לשלם עבור השדרוג. השיטה מקבלת אובייקט מסוג Car. אם המכונית שהתקבלה כפרמטר היא אכן טובה יותר (על פי הקריטריונים שהוגדרו במחלקה Car), השיטה תחליף את הערך השמור בתכונה _car למכונית החדשה newCar ותחזיר את תוספת המחיר. אם המכונית שהתקבלה כפרמטר לא טובה יותר אז לא יתבצע שינוי ויוחזר הערך 0. **שימו לב**, העדכון צריך להיות ללא aliasing.
חתימת השיטה :

```
public int upgrade (Car newCar)
```

- overlap - שיטה שנועדה לבדוק אם קיים במערכת רישום כפול של השכרה. השיטה פועלת באופן הבא: השיטה מקבלת כפרמטר השכרה אחרת. אם שם הלקוח או המכונית בפרמטר שונים מהשם או המכונית באובייקט עליו הופעלה השיטה, הרי שהרישום לא כפול והשיטה תחזיר null (זהות של מכוניות נקבעת על פי הזהות שהוגדרה במחלקה Car). אם אכן לפרמטר ולאובייקט עליו הופעלה השיטה יש אותו שם ואותה מכונית השיטה תבדוק האם קיימת חפיפה כלשהי בתאריכי ההשכרה. אם כן, השיטה תחזיר אובייקט חדש מסוג Rent בעל אותו שם ומכונית כמו לאובייקט עליו הופעלה השיטה, אך עם תאריכים המתאימים לתקופה המייצגת את איחוד שתי התקופות. אם לא, השיטה תחזיר null.
להלן דוגמאות (בדוגמאות נניח כי השמות והמכוניות בשני האובייקטים זהים ונתייחס רק לתאריכים):

- אם באובייקט הנוכחי 01/08/2000 – 07/08/2000 ובפרמטר 05/08/2000 – 09/08/2000 אז באובייקט המוחזר 01/08/2000 – 09/08/2000.
- אם באובייקט הנוכחי 01/08/2000 – 07/08/2000 ובפרמטר 08/08/2000 – 09/08/2000 אז אין חפיפה ויוחזר null.
- אם באובייקט הנוכחי 01/08/2000 – 07/08/2000 ובפרמטר 07/08/2000 – 09/08/2000 אז באובייקט המוחזר 01/08/2000 – 09/08/2000.
- אם באובייקט הנוכחי 01/08/2000 – 07/08/2000 ובפרמטר 05/08/2000 – 06/08/2000 אז באובייקט המוחזר 01/08/2000 – 07/08/2000.

חתימת השיטה :

```
public Rent overlap (Rent other)
```

- toString - שיטה המחזירה מחרוזת ובה נתוני ההשכרה לפי הדוגמאות הבאות:

Name:Ruthi From:10/03/2022 To:14/03/2022 Type:A Days:4 Price:400

Name:Rama From:30/10/2022 To:12/11/2022 Type:B Days:13 Price:1845

הקפידו על אותיות רישיות/קטנות, רווחים נכונים (אין רווחים משני צידי תו הנקודתיים, רווח בודד בין חלקי המחרוזת). אין צורך לרדת שורה בסוף המחרוזת.
חתימת השיטה:

```
public String toString()
```

עליכם לכתוב את המחלקה Rent לפי ההגדרות לעיל.

שימו לב, אסור להוסיף תכונות פרטיות.

מותר להוסיף שיטות פרטיות.

אין להשתמש במספרים בקוד (אלא אם מדובר במספרים טריוויאליים, דוגמת 0 או 1). 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 ביחידה 4 בחלק של מטלה 12.

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. מטלה שתוגש עם שגיאות קומפילציה תקבל ציון 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, הציון על המטלה יהיה אפס. 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: Car.java, Date.java, Rent.java.
5. ארזו את כל הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

ב ה צ ל ח ה

