

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 7 **נושאי המטלה:** מחלקות ושיטות, javadoc

מספר השאלות: 3 **משקל המטלה:** 3 נקודות

סמסטר: 2009 **מועד אחרון להגשה:** ...

(ת)

בממ"ן זה אנו רוצים לבדוק את יכולתכם להשתמש במחלקות שאנו מספקים לכם, ולכתוב מחלקות חדשות. הממ"ן מדגים את ההבחנה בין הממשק (interface) והמימוש (implementation) של המחלקות. כאן נתרגל גם את העיקרון של "תכנון לפי חוזה".

אנחנו נספק לכם תכניות מסוג driver שמשתמשות במחלקות שאתם תממשו. התכניות שלנו צריכות לעבוד בלי שנדע בדיוק איך מימשתם את המחלקות הנדרשות. שימו לב שהשיטה main נמצאת רק ב- tester הזה.

אתם תקבלו Application Program Interface (API), המתארים רק את הממשק של המחלקות שאתם צריכים לממש, ולא את פרטי המימוש. אם תממשו את המחלקות בדיוק לפי דרישות ה-API, תכנית ה- tester תעבוד.

אתם יכולים להוסיף שיטות או תכונות (שדות, משתני מופע) פרטיות (private) כרצונכם, אבל אסור לכם להוסיף אף שיטה או תכונה ציבורית (public) שלא מופיעה ב-API.

מימוש תאריך, דרכון ונוסע:

המימוש יכלול שלוש מחלקות: Date, Passport, Traveler.

שאלה 1 - 25%

המחלקה Date מייצגת תאריך המורכב מיום, מחודש ומשנה. במחלקה ישנם שני בנאים וכמה שיטות. אנו מביאים כאן חלקים מהמחלקה Date, בתוספת הסברים קצרים. בשל אורכן, לא הכנסנו כאן את ההערות מצורת API. ראו אותן בקובץ html המתאים באתר. אתם צריכים להשלים את החסר, וליצור את המחלקה Date.

```

public class Date {
    private int _day;
    private int _month;
    private int _year;

    //constructors:
    /**
     * creates a new Date object
     * @param _day the day in the month(1-31)
     * @param _month the month in the year
     * @param _year the year (in 4 digits)
     */

    public Date(int day, int month, int year) {
        // you have to implement
    }

    /**
     * Copy Constructor
     * @param date to be copied
     */
    public Date(Date date){
        // you have to implement
    }

    /** gets the year */
    public int getYear(){
        return _year;
    }

    /** gets the month */
    public int getMonth(){
        // you have to implement
    }

    /** gets the Day */
    public int getDay(){
        // you have to implement
    }

    /** sets the year
     * @param yearToSet the value to be set
     */
    public void setYear(int yearToSet){
        // you have to implement
    }

    /** sets the month
     * @param monthToSet the value to be set
     */
    public void setMonth(int monthToSet){
        // you have to implement
    }
}

```

```

/** sets the day
 * @param dayToSet the value to be set
 */
public void setDay(int dayToSet) {
    // you have to implement
}

/**
 * checks if this date comes before a given date
 * @param date2 the given date
 * @returns true iff this date comes before date2
 */

public boolean before(Date date2) {
    // you have to implement
}

/**
 * @return String that represents this date
 * in the following format:
 * day.month.year (30/9/1917)
 */
public String toString() {
    return _day + "/" + _month + "/" + _year;
}
}

```

כמה הסברים :

- למחלקה Date יש שלושה שדות מטיפוס int המציינים את היום, החודש והשנה של התאריך.
- קיימים שני בנאים למחלקה. האחד מקבל שלושה ערכים שלמים וקובע את התאריך החדש להיות בערכים אלו. **אתם יכולים להניח שהערכים שקיבלתם מהווים תאריך חוקי**; השני הוא בנאי העתקה (copy constructor) המקבל תאריך, ויוצר תאריך חדש שערכיו הם כמו אלו של התאריך שהתקבל כפרמטר.
- השיטות של המחלקה הן:
 - 3 שיטות get לקבלת ערכי day, month ו-year של האובייקט, שיטה אחת עבור כל שדה (כי הרי התכונות האלו מוגדרות כ-private).
 - 3 שיטות set המשנות את ערכי day, month ו-year של האובייקט לפי הערכים שבפרמטרים, שיטה אחת עבור כל שדה.
 - השיטה הבוליאנית before המקבלת תאריך נוסף ומחזירה true אם ורק אם התאריך קודם לתאריך שהועבר אל השיטה כפרמטר.
 - השיטה toString שהיא דרך סטנדרטית ב-Java להציג אובייקט כמחרוזת (אובייקט מהמחלקה String). ראו הסבר מפורט על השיטה toString באתר הקורס בתת-ספריה "חומרי עזר". השיטה כאן מחזירה

מחרוזת המייצגת את התאריך בפורמט הבא: day/month/year, למשל,

15/05/1948

ה-API של המחלקה Date נמצא באתר הקורס ב"קבצים להורדה", בתת-ספריה " "קובצי API".

שאלה 2 - 45%

המחלקה Passport מייצגת דרכון של נוסע. לשם מימושה, עליכם להשתמש במחלקה Date. עליכם לכתוב אותה כך שתממש את הממשק הבא:

תיאור השיטה	חתימת השיטה
בנאי היוצר דרכון חדש על-פי הנתונים המועברים אליו: שם, מספר דרכון ותאריך תפוגה.	public Passport (String name, int number, Date expiryDate)
בנאי העתקה (copy constructor)	public Passport (Passport other)
מחזירה את שמו של בעל הדרכון.	public String getName()
מחזירה את תאריך התפוגה של הדרכון.	public Date getExpiryDate ()
מעדכנת את שמו של בעל הדרכון.	public void setName (String name)
מעדכנת את תאריך התפוגה של הדרכון.	public void setExpiryDate (Date newExpDate)
מחזירה true אם ורק אם הדרכון תקף בתאריך הנתון. שימו לב שהדרכון תקף גם ביום התפוגה שלו.	public boolean isValid (Date dateChecked)
מחזירה מחרוזת המייצגת את הדרכון (על-פי המבנה המתואר להלן).	public String toString()

המחרוזת המייצגת את הדרכון צריכה להיראות כך:

Name: <name>

Pass. num: <number>

Exp date: <expiry date>

את הפרטים המדויקים על המחלקה Passport תוכלו למצוא באתר הקורס ב"קבצים להורדה", בתת-ספריה "קובצי API".

עליכם לממש את המחלקה Passport לפי ה-API המדויק. **אסור להוסיף לה שיטות ציבוריות.**

שאלה 3 - 30%

המחלקה `Traveler` מייצגת נוסע. פרטיו של כל נוסע הם הדרכון שלו והמידע האם הנוסע שילם או לא.

על המחלקה לממש את הממשק הבא (הפעם הבאנו אותו בצורה של API).

Constructor Summary

<code>Traveler(Passport passport, boolean isPaid)</code> creates a new traveler	
--	--

Method Summary

boolean	checkTravel (Date travelDate) Checks whether or not the travel's passport is valid on the given date and if the trip has already been paid for.
boolean	isPaid () Returns true if the traveler has already paid.
void	pay () Pay for the trip.
java.lang.String	toString () Creates a string representing the traveler's details.

להסבר מדויק יותר של השיטות, ראו באתר הקורס.

עליכם לממש את המחלקה `Traveler` לפי ה-API המדויק. **אסור להוסיף לה שיטות ציבוריות.**

לנוחותכם, אתם יכולים לנסות להריץ את התכנית שלהלן. אם המחלקות שכתבתם נכונות, תקבלו את הפלט המופיע לאחריה.

```
public class TestMain {

    public static void main(String [] args){
        Passport passport1 = new Passport("Moshe Cohen", 306914,
                                           new Date(12,10,2003));
        Passport passport2 = new Passport("Lila Cohen ", 306915,
                                           new Date(14,10,2003));

        Date travelDate = new Date(13,10,2003);
        Traveler t1 = new Traveler(passport1,true);
        Traveler t2 = new Traveler(passport2,true);
        if(t1.checkTravel(travelDate))
            System.out.println(t1);
        if(t2.checkTravel(travelDate))
            System.out.println(t2);

    }
} //end of class
```

הפלט המבוקש:

Name: Lila Cohen

Pass. num: 306915

Exp Date 14/10/2003

- הפלט חייב להיות **זהה** לזה המופיע כאן. שימו לב, אנחנו מתכוונים ממש **זהה**. כולל כל סוגי הסוגריים, רווחים וכדו'. זה דורש תשומת לב מיוחדת לשיטה `()toString`.
- מחלקת הבדיקה שכתבנו בודקת את רוב השיטות שאתם צריכים לממש. אם אתם "עוברים" את הבדיקה, זה אומר שסביר להניח שהתכנית שלכם עובדת כמו שהיא צריכה. כדאי שתבדקו בעצמכם את מה שאנו לא בדקנו.
- מכל מקום, אנו בודקים כל שיטה במימוש שלכם. אל תשכחו שאנו בודקים גם את התכנון והסגנון של התכנות שלכם.
- אפשר להניח כי הקלטים לבנאים של `Date`, של `Traveler` ושל `Passport` הם חוקיים ואין בהם שגיאות.
- אל תשכחו לכתוב את הערות ה- `javadoc` לפי ה-API.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן.
3. עליכם להגיש שלושה קובצי Java : `Date.java`, `Passport.java`, `Traveler.java`. לארוז אותם בקובץ zip ולשלוח אותו בלבד.