

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: 22.4.2023

(ת)

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים.

מדינת ישראל רוצה למפות את הערים שבשטחה.

לצורך כך נגדיר שלוש מחלקות:

- המחלקה Point שתייצג קואורדינטה במפה;
- המחלקה Date שתייצג תאריך;
- המחלקה City שתייצג עיר;

שאלה 1 - 20 נקודות

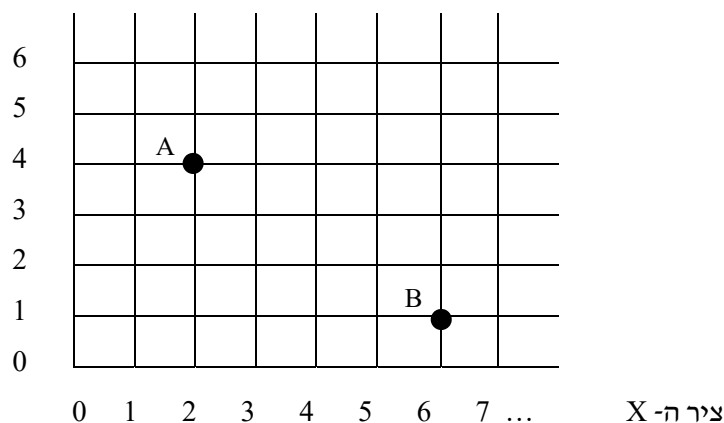
המחלקה Point מייצגת נקודה במישור ברביע הראשון בלבד (כלומר ערכי הצירים אינם יכולים להיות שליליים), לפי מערכת הצירים הקרטזית (Cartesian system).

המחלקה Point מכילה את התכונות הפרטיות (instance variables) הבאות:

- `int _x` – שמייצגת את המיקום על פני ציר ה-X.
- `int _y` – שמייצגת את המיקום על פני ציר ה-Y.

לדוגמה, שתי הנקודות $A = (2,4)$ ו- $B = (6,1)$ מסומנות במרחב:

ציר ה-Y



למחלקה Point הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים, שני מספרים שלמים, המהווים את ערכי התכונות שיהיו לנקודה.

```
public Point(int x, int y)
```

אם אחד הפרמטרים הוא ערך שלילי, יש להציב במקומו את הערך 0.

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

```
public Point (Point other)
```

בנוסף, הוגדרו במחלקה השיטות:

- שיטות האחזור:

○ `int getX()` המחזירה את ערכה של קואורדינטת ה-`x`.

○ `int getY()` המחזירה את ערכה של קואורדינטת ה-`y`.

- השיטות הקובעות:

○ `void setX (int num)` המשנה את ערכה של קואורדינטת ה-`x` להיות `num`.

אם `num` הוא מספר שלילי, הערך של `x` לא ישתנה.

○ `void setY (int num)` המשנה את ערכה של קואורדינטת ה-`y` להיות `num`.

אם `num` הוא מספר שלילי, הערך של `y` לא ישתנה.

- השיטה `toString()` שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המתמטי המקובל - `(x,y)`. כך, המחרוזת `(3,4)` מייצגת את הנקודה שקואורדינטת ה-`x` שלה היא 3 וקואורדינטת ה-`y` שלה היא 4. שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים וללא תווים נוספים.

- `boolean equals (Point other)` – שיטה שמקבלת נקודה נוספת בשם `other` כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה והנקודה `other` שהתקבלה כפרמטר זהות. כלומר, מחזירה `true` אם ערכי הנקודה עליה השיטה מופעלת שווים לערכי הנקודה `other`.

- `boolean isAbove (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מעל לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת מעל לנקודה B)

- `boolean isUnder (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מתחת לנקודה שהתקבלה כפרמטר. השיטה הזו משתמשת אך ורק בשיטה `isAbove` שהוגדרה לעיל ואין להשתמש בפעולות נוספות. אין לגשת לתכונות של הנקודות.

- `boolean isLeft (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת משמאל לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת משמאל לנקודה B)

- `boolean isRight (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מימין לנקודה שהתקבלה כפרמטר. השיטה הזו משתמשת אך ורק בשיטה `isLeft` שהוגדרה לעיל ואין להשתמש בפעולות נוספות. אין לגשת לתכונות של הנקודות.
- `void move (int deltaX, int deltaY)` - המזיזה את הנקודה ב- `deltaX` על ציר ה- `X` וב- `deltaY` על ציר ה- `Y`. אם בעקבות ההזזה הנקודה לא תהיה ברביע הראשון, אין לבצע את ההזזה והנקודה תישאר במקומה.
- `Point middle (Point p)` - שיטה שמקבלת נקודה כפרמטר ומחזירה נקודה חדשה המייצגת את נקודת המרכז בין הנקודה עליה הופעלה השיטה לבין הנקודה שהתקבלה כפרמטר. שימו לב: מדובר בנקודה שהקואורדינטות שלה הן מספרים שלמים. אין צורך לדאוג לנושא של איבוד מידע בשיטה זו.
- לדוגמה - אם הנקודה עליה מופעלת השיטה היא הנקודה (1,3) והנקודה שהועברה כפרמטר היא הנקודה (4,4) השיטה תחזיר את הנקודה (2,3) ולא את הנקודה (2.5, 3.5).
- `double distance (Point p)` - שיטה שמקבלת נקודה כפרמטר ומחזירה את המרחק בין הנקודה שעליה הופעלה השיטה והנקודה שהתקבלה כפרמטר.

תזכורת מתמטית:

בכדי לחשב מרחק בין שתי נקודות - (x_1, y_1) , (x_2, y_2) - השתמשו בנוסחה הבאה:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

על מנת לחשב שורש ריבועי של מספר, ניתן להשתמש בשיטה `Math.sqrt(x)`, שהיא שיטה של Java שנמצאת במחלקה `Math`. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא `Math.sqrt(x)` כאשר במקום הפרמטר `x` כותבים את הביטוי שממנו רוצים להוציא שורש ריבועי.

הפרמטר `x` של השיטה הזו יכול להיות מטיפוס שלם (`int`) או ממשי (`double`). השיטה מחזירה מספר ממשי (גם אם השורש הריבועי של `x` הוא מספר שלם).

עליכם לכתוב את המחלקה `Point` לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס ביחידה 4, בתת-פרק של מטלה 12.

אתם יכולים להגדיר שיטות פרטיות נוספות על אלו שהוגדרו לעיל, אבל לא שיטות ציבוריות ולא תכונות נוספות.

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. שימו לב: על מנת להקל עליכם, אין צורך 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, ויישאר 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,

o אם התאריך עליו מופעלת השיטה הוא 14/12/2019 אזי השיטה tomorrow

תחזיר את התאריך 15/12/2019

o 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.

אתם יכולים להגדיר שיטות פרטיות נוספות על אלו שהוגדרו לעיל, אבל לא שיטות ציבוריות ולא תכונות נוספות.

שאלה 3 - 50 נקודות

המחלקה City מייצגת עיר במדינה.

למחלקה City התכונות הפרטיות (instance variables) הבאות:

- String _cityName – שם העיר
- Date _dateEstablished – תאריך הקמת העיר
- Point _cityCenter – מיקום מרכז העיר
- Point _centralStation – מיקום התחנה המרכזית בעיר
- long _numOfResidents – מספר התושבים בעיר. חייב להיות חיובי או אפס.
- int _numOfNeighborhoods – מספר השכונות בעיר. חייב להיות חיובי ממש.

גם למחלקה City הוגדרו שני בנאים:

- האחד - המקבל פרמטרים עם ערכים לתכונות המתאימות: שם העיר, תאריך הקמת העיר (שלושה מספרים שלמים שמייצגים יום, חודש ושנה). תאריך לא תקין יטופל לפי הכללים של המחלקה Date), מיקום מרכז העיר (שני מספרים שלמים שמייצגים את x ו-y. מיקום לא תקין יטופל לפי הכללים של המחלקה Point), מיקום התחנה המרכזית (שני מספרים שלמים שמייצגים את x ו-y. מיקום לא תקין יטופל לפי הכללים של המחלקה Point), מספר התושבים (אם מספר התושבים שהתקבל הוא שלילי, מספר התושבים יוגדר להיות אפס) ומספר השכונות בעיר (אם מספר השכונות שהתקבל הוא שלילי או אפס, מספר השכונות יוגדר להיות 1).

using !

- השני - בנאי העתקה, המקבל אובייקט מהמחלקה City ומעתיק את ערכיו.

כמו כן, הוגדרו פעולות האחזור (get) והפעולות הקובעות (set) לפי החתימות:

```
String getCityName()
Date getDateEstablished ()
Point getCityCenter()
Point getCentralStation()
long getNumOfResidents()
int getNumOfNeighborhoods()
void setCityName(String cityName)
void setDateEstablished (Date dateEst)
void setCityCenter(Point cityCenter)
void setCentralStation(Point centralStation)
```

`void setNumOfResidents(long numOfResidents)`

`void setNumOfNeighborhoods(int numOfNeighborhoods)`

והשיטה `toString` שמחזירה את נתוני העיר (שם, תאריך הקמת העיר, מיקום מרכז העיר ותחנה מרכזית, מספר תושבים ומספר השכונות) כמחרוזת, לפי הפורמט לדוגמא הבא בדיוק:

City name: Tel Sharon

Date established: 22/04/2023

City center: (100,150)

Central station: (45,36)

Number of residents: 450

Number of neighborhoods: 10

שימו לב – במידה והפרמטרים שהועברו לשיטות הקובעות (set) אינם חוקיים, אין לבצע את

ההשמה.

"" ?

למחלקה City נוסף גם את השיטות הציבוריות:

- `boolean equals (City other)` שיטה המקבלת עיר אחרת כפרמטר ובודקת האם היא זהה לעיר שמיוצגת על ידי האובייקט עליו מופעלת השיטה. שימו לב שהשיטה תחזיר `true` רק אם כל שש התכונות הינן זהות, אחרת יוחזר `false`.
- `boolean addResidents(long residentsUpdate)` שיטה המקבלת מספר תושבים חדשים שהצטרפו לעיר, מוסיפה אותם למספר התושבים הקיימים, ומחזירה `true`. השיטה צריכה לטפל גם במקרה בהם נגרע מספר התושבים (בעקבות הגירה של תושבים מהעיר). שימו לב, שמספר התושבים לא יכול להיות שלילי ולכן אם מספר התושבים שמהגרים מהעיר גדול ממספר התושבים בעיר, השיטה צריכה לאפס את מספר התושבים, ולהחזיר `false`.
- `void moveCentralStation(int deltaX, int deltaY)` שיטה המזיזה את התחנה המרכזית שבעיר למיקום אחר, השיטה מקבלת שני מספרים שלמים `deltaX` ו-`deltaY` המורים בכמה להזיז את התחנה המרכזית בציר ה-X ובציר ה-Y בהתאמה. אם התזוזה מזיזה את התחנה מחוץ לרביע הראשון של מערכת הצירים, התחנה תישאר במקומה ולא תזוז.
- `double distanceBetweenCenterAndStation()` שיטה המחזירה את המרחק בין מרכז העיר ובין התחנה המרכזית.
- `City newCity(String newCityName, int dX, int dY)` שיטה שמקבלת מחרוזת תוים שמהווה שם, ועוד שני מספרים שלמים `dX`, `dY`, ומחזירה עיר חדשה ששמה הוא ערך המחרוזת, תאריך הקמתה הוא ביום למחרת תאריך הקמתה של העיר עליה היא מופעלת, מיקום מרכז העיר הוא במרחק `dX`, `dY` ממרכז העיר עליה היא מופעלת, מיקום התחנה המרכזית בעיר החדשה הוא במרחק `dX`, `dY` מהתחנה המרכזית עליה היא מופעלת, אין

בה תושבים ויש בה שכונה אחת. גם כאן, אם התזוזה מזיזה את מרכז העיר או את התחנה המרכזית מחוץ לרביע הראשון של מערכת הצירים, הנקודה תישאר במקומה ולא תזוז.

- `boolean cityEstablishedBetweenDates(Date date1, Date date2)` שיטה המקבלת שני תאריכים (לא ידוע איזה מהם מוקדם יותר) ומחזירה תשובה האם העיר עליה השיטה מופעלת נוסדה בין שני התאריכים הללו (כולל אותם עצמם).
- `int establishmentDateDiff(City other)` שיטה המחשבת מה ההפרש בימים בין תאריכי הייסוד של שתי הערים – זו שדרכה הופעלה השיטה וזו המועברת כפרמטר. הערך המוחזר צריך תמיד להיות חיובי.

עליכם לכתוב את המחלקה `City` לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס ביחידה 4, בתת-פרק של מטלה 12.

שימו לב, בכל שאלות המטלה :

- אסור להוסיף תכונות פרטיות למחלקות.
- מותר להוסיף שיטות פרטיות אבל לא ציבוריות.
- 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`.
- שימו לב לא לבצע `aliasing` במקומות המועדים.
- הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.
- עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר.

שימו לב ששמנו טסטרנים לשלוש המחלקות באתר הקורס. חובה שטסטרנים אלו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרנים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרנים לא ירוצו בגלל שגיאות קומפילציה הציון במטלה יהיה אפס.

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. עליכם 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: Point.java, Date.java, City.java. **אין להגיש את קובצי ה-API שכתבתם.**
5. **ארזו את שלושת הקבצים בקובץ zip יחיד ושלחו אותו בלבד.**

ב ה צ ל ח ה