

מטלת מנחה (ממ"ן) 12

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 3 - 4 נושאי המטלה: כתיבת מחלקות

מספר השאלות: 3 משקל המטלה: 3 נקודות

סמסטר: 2024 מועד אחרון להגשה: 13.1.2024

(ת)

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים. אנו מעוניינים לבנות מערכת ממוחשבת עבור חברת ניהול דירות להשכרה. לצורך כך נגדיר שלוש מחלקות:

- המחלקה Date שתייצג תאריך;
 - המחלקה Person שתייצג אדם;
 - המחלקה Apartment שתייצג דירה;
- תזכורת – בכל המטלה עליכם להשתמש בקבועים ולא במספרים, כשצריך.

שאלה 1 - 30 נקודות

המחלקה Date מייצגת תאריך:

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

- התכונה `_day` מטיפוס `int` – שמייצגת את היום (שלמים בין 1 ל-31);
- התכונה `_month` מטיפוס `int` – שמייצגת את החודש (שלמים בין 1 ל-12);
- התכונה `_year` מטיפוס `int` – 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)
```

אפשר להניח שהפרמטרים הם מספרים שלמים אבל אי אפשר להניח שהתאריך שמתקבל הוא חוקי. שימו לב, אם אחד הפרמטרים (או יותר) אינו חוקי (למשל, הוא מספר שלילי), או שהתאריך אינו חוקי (למשל 30.2.2013), האובייקט שצריך להיווצר הוא של ה-1 בינואר בשנת 2000. לא לשכוח להתייחס לשנים מעוברות (בלוח הגרגוריאני) בהן בחודש פברואר יש 29 ימים. ראו:

https://he.wikipedia.org/wiki/%D7%A9%D7%A0%D7%94_%D7%9E%D7%A2%D7%95%D7%91%D7%A8%D7%AA

- 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 או בשיטות אחרות).
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). בהמשך תמצאו שיטה בשם calculateDate שאתם יכולים להשתמש בה במימוש השיטה difference.
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 addYearsToDate מקבלת כפרמטר מספר שלם num חיובי ממש (גדול מ-0) המייצג מספר שנים. השיטה צריכה להחזיר את התאריך שיהיה בדיוק כעבור num שנים.

שימו לב למקרה הקצה הזה: כאשר התאריך עליו מופעלת השיטה הוא ביום האחרון של פברואר. מה יהיה תאריך לאחר שנה או שנתיים, או יותר? גם הוא יהיה ביום האחרון של פברואר, אלא שזה יהיה לפי העיבור של אותה שנה. כך למשל, אם התאריך הוא 28.2.2019 והפרמטר num = 1 (כלומר, צריך להוסיף שנה אחת) – התאריך שיוחזר יהיה 29.2.2020, כי בשנת 2020 יש 29 ימים בפברואר. ואם התאריך עליו מופעלת השיטה הוא 29.2.2020 וערך הפרמטר num = 2, אז התאריך שיוחזר יהיה 28.2.2022, כי בשנת 2022 יש 28 ימים בפברואר.

חתימת השיטה:

```
public Date addYearsToDate(int num)
```

לעזרתכם, אנחנו מספקים לכם שתי שיטות פרטיות. אינכם חייבים להשתמש בהן, אבל זה בהחלט מומלץ. בשתי השיטות אנו לא מגדירים קבועים עבור המספרים בקוד.

- האחת – השיטה calculateDate אשר מחשבת את מספר הימים שחלפו מתחילת הספירה (על פי הלוח הגרגוריאני):

```
// 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);
}
```

- **השניה – השיטה הבוליאנית isLeapYear**, המקבלת כפרמטר מספר חיובי שלם בן 4 ספרות המייצג שנה (בין 1000 ל-9999) ומחזירה את הערך **true** אם השנה הזו היא שנה מעוברת, ו- **false** אם לא. אפשר להניח שהפרמטר חוקי (כלומר הוא חיובי שלם בין 1000 ל-9999). אין צורך לבדוק זאת.

```
// checks if the year is a leap year
```

```
private boolean isLeapYear (int y)
{
    return (y%4==0 && y%100!=0) || (y%400==0) ? true : false;
}
```

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.

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 - 20 נקודות

המחלקה Person מייצגת אדם.

למחלקה Person התכונות הפרטיות (instance variables) הבאות:

- **_name** – String – שם האדם
- **_id** – String – מספר הזהות של האדם
- **_dateOfBirth** – Date – תאריך הלידה של האדם

למחלקה Person הוגדרו שני בנאים:

- **האחד** - המקבל פרמטרים עם ערכים לתכונות המתאימות: שם האדם (אם הפרמטר המתקבל הוא מחרוזת ריקה יש לאתחל את התכונה **_name** עם המחרוזת "Someone"), תאריך הלידה (שלושה מספרים שלמים שמייצגים יום, חודש ושנה. תאריך לא תקין יטופל לפי הכללים של המחלקה Date), מספר זהות שהוא **מחרוזת תווים של 9 ספרות בדיוק**.

אפשר להניח שמספר הזהות מכיל **ספרות בלבד**. אין צורך לבדוק זאת. אם מספר הזהות אינו בן 9 ספרות יש לאתחל את התכונה עם הערך "000000000".

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

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

```
String getName()
```

```
String getId()
```

```
Date getDateOfBirth()
```

```
void setName (String name)
```

```
void setId (String id)
```

```
void setDateOfBirth (Date d )
```

שימו לב – בכל השיטות הקובעות, אם הפרמטר שהועבר לשיטות הקובעות (set) אינו חוקי, אין לבצע את ההשמה. ניתן להניח כי הפרמטר המועבר אינו null.

השיטה toString שמחזירה את נתוני האדם (שם, מספר זהות ותאריך הולדת) כמחרוזת, לפי הפורמט לדוגמא הבא בדיוק:

Name: Israel Israeli

ID: 123456789

Date of birth: 22/10/1993

שימו לב שיש **רווח** לאחר הנקודתיים בשלוש השורות.

למחלקה Person נוסף גם את השיטות הציבוריות:

- boolean equals (Person other) שיטה המקבלת אדם אחר כפרמטר ובודקת האם הוא

זהה לאדם שמיוצג על ידי האובייקט עליו מופעלת השיטה. שימו לב שהשיטה תחזיר true רק אם ערכי כל שלוש התכונות זהים, אחרת יוחזר false.

- int compareTo(Person other) שיטה המקבלת אדם אחר כפרמטר. השיטה משווה את

הגילאים של האדם המיוצג על ידי האובייקט עליו מופעלת השיטה מול האדם המיוצג על ידי האובייקט המתקבל כפרמטר. השיטה מחזירה את הערך 1 אם האדם שמיוצג על ידי האובייקט עליו מופעלת השיטה הוא מבוגר יותר מהאדם המתקבל כפרמטר. אחרת, אם האדם המתקבל כפרמטר מבוגר יותר מהאדם עליו מופעלת השיטה, יוחזר הערך -1. אם תאריך הלידה של שניהם זהים, יוחזר הערך 0.

עליכם לכתוב את המחלקה Person לפי ההגדרות לעיל.

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

שאלה 3 - 50 נקודות

המחלקה Apartment מייצגת דירה.

למחלקה Apartment התכונות הפרטיות (instance variables) הבאות :

- `_noOfRooms` – `int` – מספר חדרים בדירה. **חייב להיות שלם חיובי.**
- `_area` – `double` – שטח הדירה במ"ר. **חייב להיות חיובי.**
- `_price` – `double` – מחיר שכירות לחודש בש"ח. **חייב להיות חיובי.**
- `_tenant` – `Person` – שוכר הדירה.
- `_rentalStartDate` – `Date` – תאריך תחילת שכירות.
- `_rentalEndDate` – `Date` – תאריך סיום שכירות **חייב להיות אחרי תאריך תחילת השכירות.**

למחלקה Apartment הוגדרו שני בנאים :

- האחד - המקבל פרמטרים עם ערכים לתכונות המתאימות : מספר חדרים (אם מתקבל ערך שלילי או אפס, יש לאתחל לערך 3), שטח הדירה (אם מתקבל ערך שלילי או אפס, יש לאתחל לערך 80), מחיר השכירות (אם מתקבל ערך שלילי או אפס, יש לאתחל ל-5000), שוכר הדירה (ניתן להניח כי הפרמטר המועבר מוגדר ואינו `null`), תאריך תחילת השכירות (שלושה מספרים שלמים שמייצגים יום, חודש ושנה). תאריך לא תקין יטופל לפי הכללים של המחלקה (`Date`), תאריך סיום השכירות (שלושה מספרים שלמים שמייצגים יום, חודש ושנה). תאריך לא תקין יטופל לפי הכללים של המחלקה (`Date`). אם תאריך סיום השכירות קודם או שווה לתאריך תחילת השכירות, יש להגדיר את תאריך סיום השכירות בדיוק לאחר שנה אחת ממועד תחילת השכירות.

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

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

```
int getNoOfRooms()
double getArea()
double getPrice()
Person getTenant()
Date getRentalStartDate()
Date getRentalEndDate()
void setNoOfRooms(int num)
```

```
void setArea(double area)
void setPrice(double price)
void setTenant(Person p)
void setRentalStartDate(Date d)
void setRentalEndDate(Date d)
```

שימו לב – בכל השיטות הקובעות, אם הפרמטר שהועבר לשיטות הקובעות (set) אינו חוקי, אין לבצע את ההשמה. ניתן להניח כי הפרמטר המועבר אינו null.

כמו כן, לאחר ביצוע של כל שיטה קובעת, תאריך הסיום חייב להישאר לאחר תאריך ההתחלה, אחרת אין לבצע את השיטה.

למחלקה Apartment נוסף גם את השיטות הציבוריות הבאות:

- השיטה toString שמחזירה את נתוני הדירה (מספר חדרים, שטח הדירה, מחיר השכירות, שם שוכר הדירה, תאריך תחילת שכירות ותאריך סיום שכירות) כמחרוזת, לפי הפורמט לדוגמא הבא בדיוק:

Number of rooms: 4

Area: 81.0

Price: 6500.0 NIS

Tenant name: Israel Israeli

Rental start date: 01/07/2022

Rental end date: 01/07/2024

שימו לב שיש רווח לאחר הנקודתיים בכל השורות.

- boolean equals (Apartment other) שיטה המקבלת דירה אחרת כפרמטר ובודקת האם היא זהה לדירה שמיוצגת על ידי האובייקט עליו מופעלת השיטה. שימו לב שהשיטה תחזיר true רק אם כל שש התכונות זהות, אחרת יוחזר false.
- void extendRentalPeriod (int years) שיטה המקבלת מספר שנים להארכת חוזה, מעדכנת בהתאם את תאריך סיום השכירות. אם מספר השנים להארכת חוזה השכירות הוא שלילי או אפס לא יתבצע עדכון כלל. **אפשר להניח שהשנה החדשה לא תהיה לאחר 9999. גם כאן, אם תאריך הסיום הוא ביום האחרון של פברואר, גם תאריך הסיום החדש, לאחר ההארכה, צריך להיות ביום האחרון של פברואר לפי העיבור של אותה שנה.**

- `int daysLeft(Date d)` שיטה המחזירה את מספר הימים שנותרו מהתאריך שהועבר כפרמטר ועד לסיום השכירות. אם התאריך שהועבר כפרמטר הוא לאחר תאריך סיום השכירות יוחזר -1.

- `boolean changeTenant(Date startDate, Person p, double price)` שיטה בוליאנית הבודקת אם אפשר להחליף את השוכר. בהנחה שהשוכר החליט שהוא רוצה לסיים את השכירות, והוא מביא שוכר אחר מועמד להחליפו. השיטה בודקת אם אפשר להחליף את השוכר בשוכר המועמד. היא מקבלת כפרמטרים: תאריך תחילת שכירות, שוכר מועמד ומחיר שכירות. השיטה בודקת אם השוכר המועמד צעיר יותר מהשוכר הנוכחי, אם מחיר השכירות הוא לפחות כפי שהיה (או יותר) ואם תחילת השכירות החדשה היא עד 90 יום לפני מועד סיום השכירות הקיימת. כלומר, שנותרו לכל היותר 90 יום עד לסיום השכירות של הדייר הנוכחי. **אם שלושת התנאים מתקיימים**, יעודכנו פרטי השכירות בפרטי השוכר החדש, מחיר שכירות ומועד תחילת השכירות, ואת מועד סיום השכירות יש לקבוע לאחר שנה מההתחלה, ויוחזר `true`. אחרת, יוחזר `false` ולא יעודכן כלום.

עליכם לכתוב את המחלקה `Apartment` 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`.
- אם אתם משתמשים בפקודת `switch` אתם חייבים להשתמש אך ורק בגרסה של הפקודה כפי שנלמדה בקורס שלנו בהרצאות של ד"ר אמיר גורן. אסור להשתמש בפקודה לפי הגרסאות המתקדמות יותר!
- במטלה זו אסור להשתמש בלולאות ו/או במערכים!
- אין להשתמש בחומר שלא נלמד בקורס, כלומר בהרצאות של ד"ר אמיר גורן.
- שימו לב לא לבצע `aliasing` במקומות המועדים.
- הקפידו להשתמש בשיטות שכבר כתבתם, גם אם הן במחלקות אחרות, ואל תכתבו מחדש קוד חוזרני.

- 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.

שמנו טסטרים לשלושת 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 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 המחלקה לא תעבור קומפילציה עם הטסטר שלה והציון יהיה 0.**
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 : Date.java, Person.java, Apartment.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