

האוניברסיטה הפתוחה

20441  
**מבוא למדעי המחשב**  
**ושפת Java**  
חוברת הקורס – סתיו 2024א

כתבה : תמר וילנר

אוקטובר 2023 – סמסטר סתיו – תשפ"ד

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנטים
ד	1. לוח זמנים ופעילויות
ו	2. תיאור המטלות
ו	2.1 מבנה המטלות
ז	2.2 חומר הלימוד הדרוש לפתרון המטלות
ז	2.3 ניקוד המטלות
י	3. התנאים לקבלת נקודות זכות בקורס
1	ממ"ן 11
7	ממ"ן 12
17	ממ"ן 13
27	ממ"ן 14



## אל הסטודנטים

אנו מקדמים את פניכם בברכה עם הצטרפותכם אל הלומדים בקורס "מבוא למדעי המחשב ושפת Java".

הקורס מבוסס על הרצאותיהם של ד"ר אמיר גורן ותמר וילנר. ההרצאות המצולמות נמצאות באתר הקורס ומלוות במצגות. חוברות השקפים שקיבלתם הן אלו המלוות את סרטי הוידאו שבאתר.

לקורס קיים אתר באינטרנט בו תמצאו את חומרי הלמידה בקורס. בנוסף, באתר הקורס תמצאו חומרים כתובים ומצולמים במהלך הקורס. גם הם חובה ללימוד הקורס. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס.

אתר הקורס הוא למעשה "ביתו" של הקורס. באתר תמצאו הסברים, הפניות והבהרות נוספות. כמו כן, באתר קיימת קבוצת דיון המאפשרת לכם לשאול שאלות את צוות הקורס וגם להתייעץ עם הסטודנטים האחרים בקורס. בקבוצת הדיון יינתן מקום לשאלת שאלות בעניין חומר הלימוד, המטלות, סביבת העבודה ועוד.

אתר הקורס בנוי בשיטת ה"רצפים" (רצפי למידה מקוונים), הרעיון הוא שהתכנים בקורס מחולקים למנות קטנות, המאורגנות באתר הקורס בצורה סדורה על פי רצף למידה המותאם ללוח הזמנים של הסמסטר. מנות התוכן מוצגות בסוגי מדיה מגוונים למשל וידאו, תרגילים אינטראקטיביים, סימולציות ועוד, המעשירות את חוויית הלמידה של הסטודנטים ומאפשרות לתרגל באופן אקטיבי את החומר הנלמד.

בסביבת הלמידה אוֹפֵּל פותח פורמט ייחודי המותאם לפדגוגית רצפי הלמידה. בפורמט הרצפים ממשק המשתמש והעיצוב הויזואלי מדגישים לסטודנטים ולסטודנטיות את רצף הלמידה המומלץ. האתר מוצג לכל סטודנט וסטודנטית במבט אישי בהתאם לקצב ההתקדמות במהלך הסמסטר.

מומלץ להיכנס לאתר לפחות פעמיים בשבוע, לעקוב אחר ההודעות בלוח ההודעות ובקבוצת הדיון. בבעיות טכניות כגון קושי בכניסה לחלקים שונים באתר וכד' אנא פנו אל המוקד בטלפון 09-7782222 או בדואר. אנא עדכנו / בדקו את פרטיכם על-מנת שתיכללו ברשימת התפוצה של דואר אלקטרוני של הקורס.

הלימוד מלווה בפתרון מטלות. משימות הלימוד לכל שבוע והתאריך האחרון למשלוח כל אחת ממטלות הקורס, רשומים ב"לוח זמנים ופעילויות" שבהמשך.

**אתר הקורס הוא חלק בלתי נפרד מהקורס, והסטודנטים מחויבים להיכנס אליו ולהתעדכן בכל מה שקורה בו. זהו המקום בו תמצאו את כל ההודעות החשובות לגבי לימוד הקורס, המטלות והבחינות. לא יישלחו הודעות/הוראות כתובות בדואר רגיל או אלקטרוני. כל ההודעות המתפרסמות באתר, כל השינויים וכל ההסברים מחייבים את כל הסטודנטים.**

**גם המטלות שתכתבו יישלחו אלינו דרך אתר הקורס תחת הכפתור "מערכת שליחת מטלות". הסבר מפורט על השימוש במערכת זו יישלח אליכם בנפרד.**

פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ס בכתובת:

<http://telem.openu.ac.il>

התכניות בקורס "מבוא למדעי המחשב ושפת Java" נכתבות בשפת התכנות Java, ואנו נעבוד בסביבת העבודה BlueJ שפותחה במיוחד להוראת תכנות בשפת Java. BlueJ פותחה כחלק מפרויקט מחקר אוניברסיטאי על הוראת תכנות מונחה-עצמים למתחילים. המערכת פותחה ומתוחזקת על-ידי קבוצת מחקר משותפת של אוניברסיטת Deakin ממלבורן, אוסטרליה, עם מכון Maersk באוניברסיטה של דרום דנמרק ועם האוניברסיטה של Kent ב-Canterbury, אנגליה. הפרויקט נתמך על-ידי חברת Sun Microsystems.

מתוך הנאמר באתר האינטרנט של BlueJ: (כתובת האתר היא [www.bluej.org](http://www.bluej.org))

*The aim of BlueJ is to provide an easy-to-use teaching environment for the Java language that facilitates the teaching of Java to first year students. Special emphasis has been placed on visualization and interaction techniques to create a highly interactive environment that encourages experimentation and exploration.*

כפי שאמרנו, אנו נעבוד בקורס עם סביבה זו, ותמיכה טכנית תינתן לסביבת עבודה זו בלבד. יחד עם זאת, אתם יכולים לעבוד עם כל סביבה אחרת שתמצאו, אבל עליכם להשתמש אך ורק במרכיבי השפה שאנחנו לומדים אותם בקורס (דהיינו בהרצאות המצולמות של ד"ר אמיר גורן של תמר וילנר).

הורדת סביבת העבודה מהאתר של BlueJ היא חינם. הוראות התקנה ל-BlueJ ול-Java Platform נמצאות באתר שלהם, בנוסף, באתר הקורס, ביחידה 1, בתוך "מדריכי עזר וקישורים" תוכלו למצוא מדריך להתקנת Java. וסביבת העבודה BlueJ, שימו לב שמדריך זה הוא קצת ישן (ובוודאי הגרסאות המצוינות בו), ולכן עדיף לעבוד לפי ההסברים והקישורים שיש באתר של BlueJ עצמו. [www.bluej.org](http://www.bluej.org)

באותו מקום, בתוך "מדריכי עזר וקישורים" שביחידה 1 תמצאו סרט על סביבת העבודה BlueJ. בסרט זה תוכלו לראות איך עובדים עם הסביבה, ואיך כותבים ומריצים בה תכניות ב-Java.

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט [www.openu.ac.il/Library](http://www.openu.ac.il/Library).

בשאלות הקשורות ללימודי מדעי המחשב באופן כללי תוכל לפנות ליועצים האקדמיים מתחום מדעי המחשב, על-פי הרשימה המופיעה בידיעון האקדמי.

כדי לקבל ייעוץ בכל הנוגע לקורס זה, תוכלו לפנות אל תמר וילנר, מרכזת ההוראה של הקורס, בדואר אלקטרוני בכתובת [tami@openu.ac.il](mailto:tami@openu.ac.il). **בכל מכתב חובה לכתוב שם מלא ומספר ת"ז, אחרת לא אוכל לטפל בפניה!**

#### **לתשומת לב הסטודנטים הלומדים בחו"ל:**

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.

פרטים נוספים על הקורס נכללים בחוברת הקורס וכן באתר הקורס. מומלץ מאוד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

בחוברת זו תמצאו את לוח הזמנים של הקורס, תנאים לקבלת נקודות זכות בקורס ומטלות.

**בשל המצב הבטחוני, אני כמעט שלא נמצאת במשרד, ולכן אני לא מקיימת שעות ייעוץ טלפוניות בסמסטר זה. הדרך הטובה ביותר לפנות אלי היא באמצעות הדואר האלקטרוני [tami@openu.ac.il](mailto:tami@openu.ac.il), ולא לשכוח לכתוב שם מלא ומספר ת"ז.**

אני מאחלת לכם לימוד פורה ומהנה.

בברכה,

תמר וילנר

מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (20441 / א2024)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה בקבוצות רגילות*	תאריך אחרון למשלוח המטלה
1	08.12.2023-03.12.2023 (ו חנוכה)	יחידות 1 ו-2 – הקדמה ויסודות השפה	מפגש 1	
2	15.12.2023-10.12.2023 (א-ו חנוכה)	יחידות 2 – יסודות השפה ו-3 – שימוש במחלקות נתונות		
3	22.12.2023-17.12.2023	יחידות 3 – שימוש במחלקות נתונות ו-4 כתיבת מחלקות	מפגש 2	ממ"ן 11 23.12.2023
4	29.12.2023-24.12.2023	יחידה 4 – כתיבת מחלקות	מפגש 3	
5	05.01.2024-31.12.2023	יחידה 5 – לולאות		
6	12.01.2024-07.01.2024	יחידה 6 – מערכים	מפגש 4	ממ"ן 12 13.1.2024
7	19.01.2024-14.01.2024	יחידה 9 – יעילות	מפגש 5	
8	26.01.2024-21.01.2024	יחידה 10 – רקורסיה	מפגש 6	

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".



לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה בקבוצות רגילות*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	2.02.2024-28.01.2024	יחידה 11 – רשימות	מפגש 7	ממ"ן 13 3.02.2024
10	11.02.2024-4.02.2024	יחידה 12 – עצים בינריים	מפגש 8	
11	לאחר תום הסמסטר	חזרה לקראת הבחינה		ממ"ן 14 17.02.2024

מועדי בחינות הגמר יפורסמו בנפרד

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

## 2. תיאור המטלות

בקורס זה **ארבע** מטלות שעליכם לפתור ולהגיש במהלך הקורס. יש להגיש את מטלות המנחה (ממ"נים) באמצעות **מערכת המטלות המקוונת שנמצאת באתר הקורס**.  
להלן תמצאו הסבר על אופן הפתרון וכיצד לשלוח את המטלה למנחה. אם שאלה במטלה אינה ברורה לכם, אל תהססו להתקשר אל מרכזת ההוראה או אל אחד המנחים (בשעות הייעוץ הטלפוני שלהם **בלבד**) לצורך קבלת הסבר.

### 2.1 מבנה המטלות

המטלות שמוגשות למנחה הן שאלות הרצה. בשאלה הרצה יש לכתוב תכנית ולהריץ אותה במחשב.  
הקפדה על שמות מחלקות ושיטות (ציבוריות), לפי הנדרש, היא הכרחית. כל חריגה מההגדרות (אפילו החלפה בודדת של אות גדולה בקטנה, למשל) תגרום לבדיקה האוטומטית שלנו להיכשל וכתוצאה מכך לנזק בלתי הפיך בציון.

**עליכם לבדוק שהתכנית שכתבת מבצעת את הנדרש ממנה ללא טעויות, ובפרט שהיא עוברת קומפילציה. תכנית שאינה עוברת קומפילציה לא תיבדק!**

#### תיעוד

בכל תכנית צריך להוסיף תיעוד בתחילת התכנית המסביר את האלגוריתם בו השתמשתם ואת מבנה התכנית. בגוף התכנית צריך להוסיף תיעוד המסביר מהו תפקידו של כל משתנה, מה מבצע כל קטע חשוב בתכנית וכל הסבר נוסף החשוב להבנת מהלך פעולתה של התכנית (את התיעוד יש לכתוב **באנגלית בלבד**). יש להקפיד על בחירת שמות משמעותיים למשתנים (באנגלית).

במשך הקורס, כאשר נגיע ליחידה הרלוונטית, נלמד איך לתעד את התכניות שלנו בעזרת תיעוד שנקרא API. מרגע זה תצטרכו להגיש את כל הממ"נים מתועדים לפי הנחיות ה-API, כפי שיילמד.

הקפידו על אופן שליחת מטלה – **קובץ דחוס מסוג ZIP בלבד** (לא RAR!), המכיל את כל הקבצים הרלוונטיים לפתרון. במידה ומדובר בפתרון שאלה "יבשה", יש לענות עליה בקובץ מסוג DOC או RTF (לא DOCX!) – מי שלא בטוח, שיפנה למנחה מבעוד מועד. במידה ומדובר בפתרון שאלה "להרצה" יש לצרף את הקבצים מסוג JAVA בלבד (לא CLASS או כל קובץ אחר). כאמור, את כל הקבצים הרלוונטיים (מסוג JAVA ו/או RTF/DOC), יש לארוז בקובץ דחוס אחד מסוג ZIP. **חריגה מהוראות אלו תגרור הורדה בציון.**

הסבר מפורט לגבי שליחת הקבצים המתאימים יינתן בכל מטלה בנפרד. המטלות בקורס זה יוגשו בעזרת מערכת שליחת המטלות שבאתר הקורס.

## 2.2 חומר הלימוד הדרוש לפתרון המטלות

בטבלה שלהלן תמצאו מהו חומר הלימוד הנדרש (לפי פרקי הספר) לפתרון כל אחת מהמטלות.

מטלה	חומר הלימוד הנדרש לפתרון
ממ"ן 11	יחידות 1 - 2
ממ"ן 12	יחידות 3 - 4
ממ"ן 13	יחידות 9 - 10
ממ"ן 14	יחידה 11

### שימו לב!

אין להשתמש לפתרון המטלות בידע הנרכש בפרקי לימוד מתקדמים יותר מהפרקים בהם עוסקת המטלה.

## 2.3 ניקוד המטלות

ניתן לצבור עד 10 נקודות. חובה להגיש מטלות במשקל מינימלי של 6 נקודות לפחות.

להלן פירוט הניקוד לכל מטלה:

ממ"ן/ממ"ח	ניקוד
11	1
12	3
13	3
14	3

כדי לעבור את הקורס, צריך (בין היתר) להגיש מטלות במשקל של 6 נקודות לפחות. סטודנט שמגיש את כל המטלות, משקל המטלות שלו הוא 10 נקודות. בהתאם למשקל המטלות, נקבע אחוז הציון של בחינת הגמר בציון הסופי.

חישוב אחוז הציון של הבחינה – אם הוגשו מטלות במשקל 10 נקודות, הבחינה שווה 90% מהציון הסופי של הקורס. אם המטלות שהוגשו הן במשקל 6, הבחינה שווה 94%. וכך גם לכל המספרים ביניהם. לדוגמא, אם הוגשו מטלות 11, 12, 13 המשקל שלהן הוא  $7 = 1 + 3 + 3$  (ללא קשר לציון שהתקבל בכל מטלה), ואז משקל הבחינה בציון הסופי הוא 93%.

## איך מחושב ציון הקורס ?

בכל מטלה, מכפילים את הציון שלה במספר הנקודות שהיא שוקלת. מכפילים גם את ציון הבחינה במשקל שלה (לפי משקל המטלות). מחברים את כל המכפלות האלו, ומחלקים ב-100. זה הציון הסופי בקורס.

### לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:  
אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי. זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.  
**זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.**

לדוגמא, אם הציונים במטלות ובבחינה הם אלו:

מטלה	משקל	ציון
11	1	50
12	3	50
14	3	90
בחינת הגמר	93	75

משקל הבחינה אמור להיות 93%.

הציון אמור להיות:

$$(50 \times 1 + 50 \times 3 + 90 \times 3 + 75 \times 93) / 100 = 74.45 \rightarrow 74$$

שימו לב שבמטלות 11, 12 הציון נמוך מציון הבחינה. אם נוריד את שתייהן, יישארו 3 נקודות. אם נוריד רק את מטלה 11 יישארו 6 נקודות, ומשקל הבחינה יהיה 94%. אם נוריד רק את מטלה 12 יישארו 4 נקודות, וזה לא מספיק. לכן כדאי להוריד רק את מטלה 11 (כלומר להניח כאילו לא הוגשה כלל).

חישוב הציון ייעשה, אפוא, כך:

$$(50 \times 3 + 90 \times 3 + 75 \times 94) / 100 = 74.70 \rightarrow 75$$

כלומר, הציון הסופי הוא 75.

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס.

סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע

בטלפון 09-7782222 או יעדכנו בעצמם באתר שאילתא <http://www.openu.ac.il/sheilta>

**קורסים** → **ציוני מטלות ובחינות** → **הזנת ציון 0 למטלות רשות שלא הוגשו.**

יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהממוצע המשוקלל של המטלות והבחינה יהיה נמוך מ- 60). לכן, אם אתם רוצים להזין ציון 0 במטלה אחת או יותר, בדקו היטב באיזו מטלה כדאי לכם להזין 0, ואל תעשו זאת ליותר מטלות מהדרוש.

**כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.**

### **הכנת המטלות חייבת להיעשות על-ידי כל סטודנט בנפרד.**

**מטלות שלא יבוצעו באופן עצמאי – ייפסלו!!!**

אי אפשר לעשות בזוגות (או בחבורות גדולות יותר). מי שיגיש מטלה שאנחנו נחשוד בה כמועקת (או ככזו שהעתיקו ממנה), יועלה לוועדת משמעת. שימו לב, אפשר להתייעץ זה עם זה במהלך הכנת המטלות, אך ההתייעצות חייבת להיות בעל-פה (ללא כתיבת חומר כלשהו). לאחר ההתייעצות כל אחד חייב לכתוב את המטלה בנפרד.

### **מועדי הגשת המטלות**

בעמוד הראשון של כל מטלה מצוין מועד הגשתה. הממ"ן ייבדק ויוחזר לך תוך שלושה שבועות מהתאריך האחרון להגשת הממ"ן. אם הממ"ן לא יוחזר אליך במועד זה, אנא התקשר עם המנחה לברר סיבת העיכוב.

### **דחייה בהגשת מטלות**

- אנחנו נאשר איחורים רק כאשר יש בקשה מראש לאיחור כזה. בקשה מנומקת לאיחור של עד שבוע יש להפנות למנחה הקבוצה. במקרים חריגים ביותר של דחייה של למעלה משבוע, תצטרכו לפנות למרכזת ההוראה לקבלת אישור כזה. שוב, האישור צריך להתקבל מלכתחילה ולא בדיעבד. שימו לב שאם תאריך ההגשה של המטלה הוא במוצאי שבת, יש להגיש את הבקשה לדחיה עד יום חמישי (כולל). לא ביום שישי ולא בשבת! אלו אינם ימי עבודה!
- מי שיאחר בהגשת המטלה ללא קבלת אישור מראש, יורדו לו 3 נקודות מהציון לכל יום איחור, וגם זה רק עד שבוע. לאחר שבוע המטלה תיבדק למשוב בלבד, ולא תחשב בשקלול הציון.
- לגבי מילואימניקים – אתם יכולים לפנות אל המנחים שלכם ולסכם איתם את מועדי ההגשה למטלות המתאימים לכם ולמנחים. לכל מטלה המוגשת באיחור צרף מכתב/אישור המנמק את סיבת האיחור.

**נזכיר שוב:**

**לבחינת הגמר רשאי לגשת רק סטודנט שצבר 6 נקודות לפחות בהגשת המטלות.**

**3. התנאים לקבלת נקודות זכות בקורס**

- א) צבירת משקל של 6 נקודות לפחות במטלות.
- ב) ציון של 60 נקודות לפחות בבחינת הגמר.
- ג) ציון סופי בקורס של 60 נקודות לפחות.

# מטלת מנחה (ממ"ן) 11

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 1-2 נושאי המטלה: יסודות השפה

מספר השאלות: 2 משקל המטלה: נקודה אחת

סמסטר: 2024 מועד אחרון להגשה: 23.12.2023

(ת)

שימו לב:

- יש להקפיד על שמות המחלקות בדיוק כמו שנכתבו.
- יש לתעד את התכניות בתיעוד פנימי באנגלית בלבד (בתחילת התכנית התיעוד מסביר מה מבצעת התכנית באופן כללי ובמהלך התכניות התיעוד מסביר את הקוד).
- אין להוסיף שיטות מעבר לאלה הנדרשות במטלה במפורש.
- אין להשתמש בחומר מתקדם ובפרט לא בלולאות. כמו כן, אין להשתמש בחומר שלא נלמד בקורס, כלומר בהרצאות של ד"ר אמיר גורן.
- יש להשתמש בקבועים היכן שאפשר.
- יש להקפיד על הזחה (אינדנטציה - עימוד) נכונה, ועל שמות משתנים בעלי משמעות (באנגלית) ולפי המוסכמות בקורס.
- יש להקפיד על פורמט הפלט בדיוק כפי שמצוין בשאלה: איות נכון, אותיות גדולות וקטנות, רווחים, וכו'.
- באתר הקורס תוכלו למצוא קובץ הנחיה לפתרון המטלות התכנותיות. כדאי מאוד לעיין בו ולפעול לפיו. הקובץ נמצא באתר הקורס בתוך "יחידה 1" ב"מדריכי עזר וקישורים" והוא נקרא "הנחיות לכתובת תכניות ומטלות בקורס".
- הגשת המטלה נעשית אך ורק בעזרת מערכת המטלות המקוונת שבאתר הקורס.
- אל תשכחו לשמור את מספר האסמכתא שתקבלו מהמערכת לאחר ההגשה.

## שאלה 1 (50%)

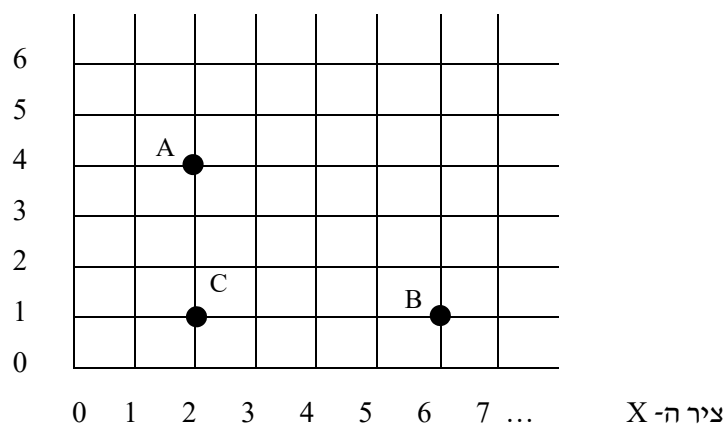
כתבו תכנית המוצאת מבין שלוש נקודות במישור את הנקודה הקרובה ביותר לראשית הצירים. התכנית תקרא מהקלט שהכניס המשתמש שישה מספרים שלמים. כל זוג מספרים מייצג שיעורי (קואורדינטות) של נקודה אחת.

התכנית תחשב את המרחק של כל נקודה לראשית הצירים (כלומר לנקודה  $(0,0)$ ) ותדפיס על הפלט את שיעורי הנקודה שקרובה ביותר לראשית הצירים.

שימו לב שהנקודות לא חייבות להיות ברביע הראשון במישור, כלומר, הקואורדינטות יכולות להיות גם שליליות או 0.

**לדוגמה,** במידה ונקלטו ערכי שלוש הנקודות  $A = (2,4)$ ,  $B = (6,1)$  ו-  $C = (2,1)$  מסומנות במרחב:

ציר ה-Y



ההדפסה תהיה בפורמט הבא:

אם הקלט הוא (הנתונים של הנקודות A, B ו- C שלעיל. הנתונים נקלטים משמאל לימין):

2 4 6 1 2 1

אז הפלט צריך להיות כזה:

The nearest point to the origin is (2,1)

ניתן להניח כי כל זוג ערכים שנקלט מייצג שיעורי הנקודה (הערך הראשון מייצג את שיעור ה-x של הנקודה והערך השני בזוג מייצג את שיעור ה-y של הנקודה).

במקרה בו יש יותר מנקודה אחת הקרובה ביותר לראשית, יש להציג את הנקודה הראשונה שנקלטה מבין הנקודות שכולן הן הקרובות ביותר לראשית.

הקלט לתכנית הוא:

שישה מספרים שלמים המייצגים 3 זוגות ערכים.



קליטת הערכים תיעשה על פי ההנחיות שיוצגו למשתמש כך (שלאחר כל הנחיה יקלטו זוג המספרים המתאים):

Enter first point coordinates:

Enter second point coordinates:

Enter third point coordinates:

לשם קריאה מהקלט השתמשו במחלקה Scanner.

כדי להשתמש בה צריך לכתוב בראשית התכנית את השורה

```
import java.util.Scanner;
```

אפשר למצוא את הממשק של המחלקה Scanner בתוך סעיף 2.6 Interactive Programs מהספר Java Software Solutions שנמצא לאחר הסרטון 2.4, שם מובאות חלק מהשיטות. הסברים על המחלקה והשימוש בה אפשר למצוא באתר הקורס בתוך "יחידה 2" ב"מדריכי עזר וקישורים" בקובץ "מדריך לעבודה עם המחלקה Scanner לקבלת קלט מהמשתמש".

### 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  $\text{Math.sqrt}(x)$ , שהיא שיטה של Java שנמצאת במחלקה Math. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא  $\text{Math.sqrt}(x)$  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).

על מנת לחשב חזקה של מספר, ניתן להשתמש בשיטה  $\text{Math.pow}(a, b)$ , שהיא שיטה של Java שנמצאת במחלקה Math. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא  $\text{Math.pow}(a, b)$  כאשר במקום הפרמטר  $a$  כותבים את הביטוי שאותו רוצים להעלות בחזקה ובמקום הפרמטר  $b$  כותבים את החזקה. הפרמטרים  $a$  ו- $b$  של השיטה הזו יכולים להיות מטיפוס שלם (int) או ממשי (double). השיטה מחזירה מספר ממשי (גם אם  $a$  בחזקת  $b$  הוא מספר שלם).

**התכנית שכתבתם צריכה להיות במחלקה בשם Origin. המחלקה Origin תכיל שיטה אחת בלבד בשם main. אסור לכתוב שיטות נוספות.**

לעזרתכם, כתבנו כאן את תחילת קוד המחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).

אתם חייבים להשתמש בתחילת הקוד שלנו בדיוק כפי שהוא מופיע כאן.

```
import java.util.Scanner;

public class Origin
{
    public static void main (String [] args)
    {
        Scanner scan = new Scanner (System.in);

        System.out.println("Enter first point coordinates:");

        int x1 = scan.nextInt();
        int y1 = scan.nextInt();

        System.out.println ("Enter second point coordinates:");

        int x2 = scan.nextInt();
        int y2 = scan.nextInt();

        System.out.println ("Enter third point coordinates:");

        int x3 = scan.nextInt();
        int y3 = scan.nextInt();

        // כאן עליכם להחשיף...
    } // end of method main
} //end of class Origin
```

שימו לב שהתוצאה של הפרש המרחקים היא לא בהכרח מספר שלם.

## שאלה 2 (50%)

כתבו תכנית הקולטת שלושה מספרים ממשיים  $a, b, c$  המייצגים מקדמי פולינום ממעלה שנייה:

$$ax^2 + bx + c = 0$$

על התכנית לבדוק את האם קיימים שורשי המשוואה בהתאם למקדמים שנקלטו. אם קיימים – התכנית תציג את ערכי שורשיה.

### תזכורת מתמטית

אם ערך הדיסקרימיננטה אי-שלילי (אפס או חיובי) קיימים שורשים לפולינום ממעלה שנייה:

$$\Delta = b^2 - 4ac$$

אם קיימים, חישוב השורשים הוא על פי נוסחת השורשים:

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

כאשר:

עבור  $\Delta < 0$  אין פתרון.

עבור  $\Delta = 0$  קיים פתרון יחיד.

עבור  $\Delta > 0$  קיימים שני פתרונות.

### הקלט לתכנית:

שלושה מספרים ממשיים. אפשר להניח רק שאכן נקלטו שלושה מספרים ממשיים והמקדמים נקלטים על פי סדר המעלה מגבוהה לנמוכה (כלומר המקדם הראשון שיקלט הוא  $a$ , לאחריו המקדם  $b$  ולבסוף המקדם  $c$ ). אפשר להניח ש- $a \neq 0$ , אבל אי אפשר להניח שום דבר נוסף על ערכיהם.

קליטת הערכים תיעשה על פי ההנחיות שיוצגו למשתמש כך (שלאחריה יקלטו שלושת המספרים):

Enter 3 coefficients of the polynomial equation:

### הפלט שהתכנית תפיק:

- עבור הקלט הבא:  $a = 1, b = -2, c = 1$  יתקבל הפלט:  
There is 1 solution.  $X1 = 1.0$ .
- עבור הקלט הבא:  $a = -2.5, b = 1, c = -1.5$  יתקבל הפלט:  
There is no solution.
- עבור הקלט הבא:  $a = 4, b = -5, c = 1$  יתקבל הפלט:  
There are 2 solutions.  $X1 = 1.0, X2 = 0.25$ .

אם ישנם שני פתרונות, עליכם להציג את הפתרון שמשתמש בחיבור לפני זה שמשתמש בחיסור.

התכנית שכתבתם צריכה להיות במחלקה בשם Equation. המחלקה Equation תכיל שיטה אחת בלבד בשם main. אסור לכתוב שיטות נוספות.

גם הפעם השתמשו במחלקה Scanner כדי לקרוא מהקלט.

### בשתי השאלות במטלה זו -

- הקפידו שפלט התוכנית יהיה בדיוק כפי שרשום בהנחיות! אין להוסיף הודעות על מה שנכתב מפורשות בהנחיות ויש לדייק (כולל אותיות גדולות/קטנות וסימני פיסוק) בהודעות המפורטות.
- אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Origin ו-Equation), והכל בשיטה main. אין להוסיף שיטות נוספות.
- אסור להשתמש בלולאות ו/או במערכים!

### הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לשאלה 1 כולל את הקובץ Origin.java.
3. הפתרון לשאלה 2 כולל את הקובץ Equation.java.
4. ארזו את שני הקבצים בקובץ zip (ולא rar) יחיד ושלחו אותו בלבד.
5. אל תשכחו לשמור את מספר האסמכתא שקיבלתם מהמערכת לאחר ההגשה. אם לא קיבלתם מספר אסמכתא, סימן שההגשה לא התקבלה.
6. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורסת את ההגשה הקודמת. אבל עשו זאת אך ורק עד לתאריך ההגשה. אחרי התאריך, ייחשב לכם כאילו הגשתם באיחור, גם אם ההגשה הראשונה היתה בזמן! כמו כן, אם המנחה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

### בהצלחה

# 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](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 בתייעוד 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



# מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 9-10 נושא המטלה: יעילות ורקורסיה

מספר השאלות: 4 משקל המטלה: 3 נקודות

סמסטר: 2024 מועד אחרון להגשה: 3.2.2024

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. אנו ממליצים מאוד לענות עליהן ללא הרצה במחשב (כפי שמקובל בבחינת הגמר) ולאחר מכן להריץ.

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex13 (בדיוק). את התשובות לשאלות על הסיבוכיות כתבו (באנגלית בלבד) כחלק מה-API של השאלה הרלוונטית.

## הערות חשובות לגבי שאלות 1 ו-2:

- השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.
- ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.
- לא לשכוח להתייחס למקרי קצה שונים.
- אסור לשנות את המערכים בשתי השאלות, גם אם הם חוזרים לקדמותם לאחר השיטה!
- כתבו (באנגלית בלבד) כחלק מה-API של השאלה מה סיבוכיות הזמן (Time complexity) וסיבוכיות המקום (Space complexity) של השיטה שכתבתם. הסבירו תשובתכם.
- אל תשכחו לתעד (באנגלית) את מה שכתבתם!

## שאלה 1 - 25 נקודות

כתבו שיטה סטטית המקבלת כפרמטר מערך `arr` המלא במספרים שלמים, (חיוביים ושליילים ללא אפסים).

השיטה צריכה למצוא שלשה של מספרים במערך (לאו דווקא סמוכים זה לזה) שהמכפלה שלהם היא המקסימלית. השיטה תחזיר את המכפלה המקסימלית.

אם המערך בגודל קטן מ-3 השיטה צריכה להחזיר את הערך `Integer.MIN_VALUE`. אפשר להניח שהמערך אינו `null`.

### לדוגמא:

- אם המערך `arr` מכיל את המספרים הבאים:

0	1	2	3	4	5	6	7	8
-8	1	-4	11	9	-15	10	8	-3

אזי תוחזר המכפלה  $-8 * 11 * -15 = 1320$

### חתימת השיטה היא:

```
public static int maxMult3 (int [] arr)
```

מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם? הסבירו תשובתכם.

להזכירכם - השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום.



## שאלה 2 - 25 נקודות

נגדיר : **חציון** (median) הוא המספר האמצעי ברשימה מסודרת של מספרים. כך שמספר המספרים הגדולים ממנו ברשימה שווה למספר המספרים הקטנים ממנו ברשימה. כאשר מספר המספרים ברשימה הוא זוגי, נהוג להגדיר את החציון כממוצע בין שני המספרים האמצעיים ברשימה.

**כך למשל,**

ברשימה {1, 4, 5, 7, 20} החציון הוא 5 - יש ברשימה שני מספרים גדולים ממנו (1, 4) ושני מספרים קטנים ממנו (7, 20).

ברשימה {1, 4, 5, 7, 20, 28} החציון הוא 6 שהוא הממוצע בין 5 ל-7.

**כתבו שיטה המקבלת שני מערכים חד-ממדיים בגודל n מלאים במספרים שלמים, ממוינים בסדר עולה. השיטה צריכה להחזיר את החציון של המספרים הנמצאים בשני המערכים.**

**לדוגמא:**

אם המערכים  $a_1$  ו-  $a_2$  הם אלו :

$$a_1 = \{1, 12, 15, 26, 38\}$$

$$a_2 = \{12, 13, 18, 30, 45\}$$

השיטה תחזיר את הערך 16.

**הסבר :** אחרי מיזוג שני המערכים למערך ממוין אחד נקבל : {1, 12, 12, 13, 15, 18, 26, 30, 38, 45} שני המספרים האמצעיים במערך הם 15 ו-18, והממוצע ביניהם הוא  $16 = (15 + 18) / 2$ , שזה החציון. (16 ולא 16.5 בגלל חלוקה בשלמים).

שימו לב שבגלל שהמערך הממוזג הוא באורך זוגי ( $2n$ ), החציון הוא הממוצע בין שני המספרים האמצעיים.

**אפשר להניח שהמערכים המתקבלים כפרמטרים לשיטה הם באותו גודל, וכן הם ממוינים בסדר עולה. אין צורך לבדוק זאת.**

**רמז, כדאי להשתמש בחיפוש בינרי.**

**חתימת השיטה היא:**

```
public static int findMedian (int[] arr1, int[] arr2)
```

**מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם? הסבירו תשובתכם.**

**להזכירכם - השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום.**

#### הערות חשובות לגבי שאלות 3 ו-4:

- מותר להשתמש בהעמסת-יתר (Overloading)
- אסור להשתמש במשתנים סטטיים (גלובליים)!
- שימו לב להתייחס למקרי קצה.
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

בשאלה 3 מותר להשתמש בשיטות `charAt`, `length` ו- `substring` מהמחלקה `String` ובשיטות `Math.max` ו- `Math.min` מהמחלקה `Math`.

בשאלה 4 מותר לשנות את המערך במהלך השיטה, אבל בסופה הוא צריך לחזור למצבו המקורי.

### שאלה 3 - 25 נקודות

נגדיר:

מחרוזת תווים  $s$  מכילה (include) את המחרוזת  $t$  אם כל התווים של  $t$  נמצאים במחרוזת  $s$  באותו סדר בו הם נמצאים ב- $t$  (אבל יכולים להיות תווים נוספים ביניהם).  
למשל: המחרוזת  $s = "ABCDE"$  מכילה את המחרוזות "BCD" ואת "ABD", אבל לא מכילה את "AXB", "BBC", "ADB".  
שימו לב, כל מחרוזת מכילה את המחרוזת הריקה.

נתונות שתי מחרוזות תווים  $st1$  ו- $st2$ .

עליכם לכתוב שיטה סטטית **רקורסיבית** המקבלת שתי מחרוזות תווים  $st1$  ו- $st2$ , ומחזירה את המחרוזת המינימלית (הקטנה ביותר) המכילה את שתי המחרוזות.  
**לדוגמא,**

- אם הפרמטרים הם  $st1 = "B"$ ,  $st2 = "A"$   
אז המחרוזת הכי קטנה שמכילה את שתי המחרוזות  $st1$  ו- $st2$  היא "AB" (או "BA")  
והשיטה צריכה להחזיר אותה.
- אם הפרמטרים הם  $st1 = "AA"$ ,  $st2 = "A"$   
אז המחרוזת הכי קטנה שמכילה את שתי המחרוזות  $st1$  ו- $st2$  היא "AA" והשיטה צריכה להחזיר אותה.
- אם הפרמטרים הם:  $st1 = "AGGTAB"$ ,  $st2 = "GXTXAYB"$   
אז המחרוזת הכי קטנה שמכילה את שתי המחרוזות  $st1$  ו- $st2$  היא "AGXGTXAYB"  
והשיטה צריכה להחזיר אותה. שימו לב שגם המחרוזת "AGGTXAYB" היא מחרוזת  
הכי קטנה שמכילה את שתי המחרוזות. אפשר להחזיר גם אותה, כמובן.

**חתימת השיטה :**

```
public static String minimalSt(String st1, String st2)
```

**השיטה שתכתבו צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.**

## שאלה 4 - 25 נקודות

נתון מערך דו-ממדי  $mat$ , בגודל  $n \times m$  (n שורות ו-m עמודות) שערכיו הם מספרים שלמים וחיוביים ממש.

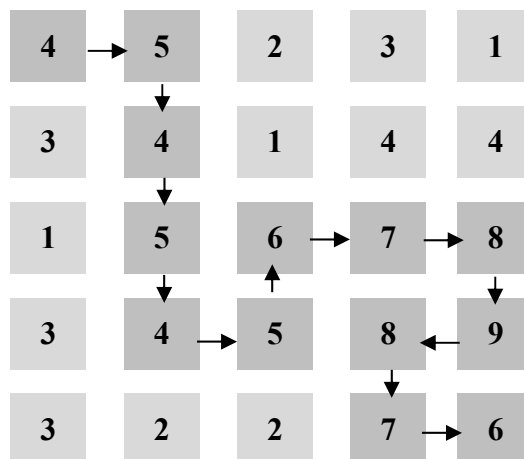
נגדיר מסלול-נחש (snake-path) במערך אם הוא מקיים את התנאים הבאים:

- המסלול מתחיל בתא הראשון במערך – שורה ראשונה ועמודה ראשונה.
- המסלול מסתיים בתא האחרון במערך – שורה אחרונה ועמודה אחרונה.
- אפשר לעבור מתא אחד לשני רק אם הוא שכן שלו מימין, משמאל, למעלה או למטה. לא באלכסון.
- אפשר ללכת מתא לתא רק אם ההפרש בין הערכים שנמצאים בשני התאים הוא מקסימום 1. למשל, מתא שערכו הוא 4 אפשר ללכת לתא שערכו הוא 3 או 4 או 5.
- המסלול לא יכול לחזור לתא בו הוא היה כבר!

עליכם לכתוב שיטה סטטית רקורסיבית המקבלת מערך דו-ממדי, מלא במספרים שלמים. השיטה צריכה להחזיר את אורכו של מסלול הנחש הארוך ביותר שיש במערך. אם אין מסלול כזה השיטה תחזיר את הערך `Integer.MIN_VALUE`

אפשר להניח שהמערך מלא במספרים שלמים חיוביים ממש. אין צורך לבדוק זאת.

לדוגמא, במערך להלן מסומן מסלול-נחש באורך מקסימלי – 13. שימו לב שיש עוד מסלול-נחש אחד באורך 13, והוא: 4-3-4-5-4-5-6-7-8-9-8-7-6



חתימת השיטה היא:

```
public static int maxSnake(int[] [] mat)
```

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

## שימו לב:

בכל השאלות - אל תשכחו לתעד (באנגלית בלבד) את מה שכתבתם!

שימו לב ששמנו טסטר באתר הקורס. חובה שהטסטר ירוץ ללא שגיאות קומפילציה עם המחלקה שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה.

אם הטסטר לא ירוץ ללא שגיאות קומפילציה הציון במטלה יהיה אפס **ללא אפשרות ערעור**.

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות יהיו **בדיוק** כפי שמוגדר בממ"ן.
3. עליכם לתעד (**באנגלית בלבד**) את כל השיטות שאתם כותבים בתיעוד API ובתיעוד פנימי המסביר מה עשיתם בשיטה. בתיעוד זה כתבו גם מה הסיבוכיות של השיטות (בשאלות 1 ו-2).
4. את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex13 (**בדיוק**). ארוז את הקובץ בתוך קובץ zip. אין לשלוח קבצים נוספים.

## ב ה צ ל ח ה

## שאלה לא להגשה

לפניכם שני קטעי הקוד (שאינם קשורים זה לזה):

```
int a =3;
while (a <= n)
    a = a*a;
```

```
public void foo (int n, int m)
{
    int i = m;
    while (i > 100)
        i = i/3;
    for (int k=i ; k>=0; k--)
    {
        for (int j=1; j<n; j*=2)
            System.out.print(k + "\t" + j);
        System.out.println();
    }
}
```

מה סיבוכיות זמן הריצה של קטעי הקוד האלו?

להזכירכם – חוקי הלוגריתמים:

$$\log_a m \times n = \log_a m + \log_a n$$

$$\log_a \frac{m}{n} = \log_a m - \log_a n$$

$$\log_a n^m = m \times \log_a n$$

## שאלה לא להגשה

לפניכם קטע הקוד הבא:

```
public static int foo (int a, int b)
{
    if (a>3)
        return 2 + foo (b-1, a+1);
    if (b<=4)
        return 1 + foo (a-1, b+1);
    return 0;
}
```

לכל אחת מהקריאות הבאות לשיטה foo, ענו אם היא תעצור, ואם כן, מה היא תחזיר.

א. foo (3, 4)

ב. foo (4, 5)

## שאלה לא להגשה

התבוננו בשיטות הבאות :

```
public static void f(int [][] a,
                    int a1, int b1, int a2, int b2)
{
    int temp = a[a1][b1] ;
    a[a1][b1] = a[a2][b2] ;
    a[a2][b2] = temp ;
    if (b1 < a[0].length-1)
        f(a, a1, b1+1, a2, b2-1) ;
    else if (a1+1 < a2-1)
        f(a, a1+1, 0, a2-1, a[0].length-1) ;
}

public static void printArray(int [][] a)
{
    for (int i= 0; i< a.length; i++)
    {
        for (int j= 0; j< a[i].length; j++)
            System.out.print (a[i][j] + "\t");
        System.out.println();
    }
}
```

נניח שנתונה השיטה main הבאה :

```
public static void main (String [] args)
{
    int[][] arr = {{1, 2, 3, 4}, {5, 6, 7, 8}} ;
    f(arr, 0, 0, arr.length-1, arr[0].length-1) ;
    printArray (arr);
}
```

1. מה הפלט שתפיק השיטה ?main

2. כמה קריאות רקורסיביות מתבצעות בזימון

f(arr, 0, 0, arr.length-1, arr[0].length-1) ;





# מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 11 נושא המטלה: רשימות מקושרות

מספר השאלות: 4 משקל המטלה: 3 נקודות

סמסטר: 2024 מועד אחרון להגשה: 17.2.2024

במטלה זו נעסוק ברשימות מקושרות של מספרים שלמים.

בשתי השאלות הראשונות נשתמש ברשימה חד-סטרית. לשם כך נכתוב מחלקה בשם `IntList` המייצגת רשימה מקושרת כזו.

הרשימה תשתמש במחלקה `IntNode` הבאה, המייצגת איבר ברשימה:

```
public class IntNode {
    private int _value;
    private IntNode _next;

    public IntNode(int val, IntNode n) {
        _value = val;
        _next = n;
    }
    public IntNode(int val) {
        _value = val;
        _next = null;
    }

    public int getValue() {
        return _value;
    }
    public IntNode getNext() {
        return _next;
    }

    public void setValue(int v) {
        _value = v;
    }
    public void setNext(IntNode node) {
        _next = node;
    }
}
```

אסור להוסיף למחלקה זו תכונות ו/או שיטות פרטיות או ציבוריות!

## להלן המחלקה IntList:

```
public class IntList
{
    private IntNode _head;

    public IntList( ) {
        _head = null;
    }

    public void addToEnd(int num) {
        // adds num at the end of the list
        IntNode node = new IntNode(num);
        if (_head == null)
            _head = node;
        else {
            IntNode ptr = _head;
            while (ptr.getNext( ) != null)
                ptr = ptr.getNext( );
            ptr.setNext(node);
        }
    }

    public String toString()
    {
        String s = "";
        IntNode temp = _head;
        while (temp != null)
        {
            s = s+ temp.getValue() + " --> ";
            temp = temp.getNext();
        }
        s+= " null";
        return s;
    }

    // כאן ייכנסו השיטות שתכתבו בשאלות הבאות
}
```

## הערות חשובות לגבי שאלות 1 ו-2:

- השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.
- ניתן להשתמש בשיטות עזר פרטיות ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.

- כתבו (באנגלית בלבד) כחלק מה- API של השאלה מה סיבוכיות הזמן (Time complexity) וסיבוכיות המקום (Space complexity) של השיטה שכתבתם. הסבירו תשובתכם.

### שאלה 1 - 25 נקודות

הוסיפו למחלקה `IntList` שיטה בוליאנית בשם `subListSum` המקבלת כפרמטר מספר שלם חיובי **ממש** `num`. השיטה מחזירה `true` אם יש ברשימה תת-רשימה **רצופה** של חוליות שסך הערכים בהן שווה ל-`num`. אם אין תת-רשימה כזו, השיטה תחזיר `false`.

לדוגמא,

אם הרשימה היא  $\{3, 6, 2, 7, 1\}$  (ראש הרשימה הוא 3), והפרמטר `num=15` השיטה תחזיר `true` כי סכום התת-רשימה  $\{6, 2, 7\}$  הוא 15.  
לעומת זאת אם הפרמטר `num = 14`, השיטה תחזיר `false`, כי אין תת-רשימה (רצופה) שסכום ערכיה הוא 14.

הניחו כי ברשימה בשאלה זו יש מספרים חיוביים ממש (ללא אפסים).

חתימת השיטה :

```
public boolean subListSum(int num);
```

### שאלה 2 - 25 נקודות

נתונה רשימה מקושרת שאיבריה הם מספרים שלמים (לא רק חיוביים).

על כל חוליה ברשימה ניתן להביט כמחלקת את הרשימה לשני חלקים :

1. כל החוליות מראש הרשימה עד החוליה הזו (כולל);
2. כל החוליות מהחוליה שאחרי החוליה הזו ועד לסוף הרשימה.

לכל אחד משני החלקים הללו אפשר לחשב את ממוצע הערכים הנמצאים בחוליות השייכות אליו.

הוסיפו למחלקה `IntList` שיטה בשם `averageNode` המחזירה את החוליה המחלקת את הרשימה באופן שבו הערך המוחלט של ההפרש בין ממוצעי שני החלקים יהיה הגדול ביותר האפשרי בהשוואה לכל חלוקה אחרת.

שימו לב: החוליה המוחזרת לא יכולה להיות החוליה האחרונה ברשימה. כלומר, אף אחד מהחלקים שלהם מתחלקת הרשימה לא יכול להיות ריק. אם הרשימה ריקה או בעלת איבר אחד בלבד, השיטה תחזיר `null`.

#### דוגמאות:

- **בהינתן הרשימה הבאה: {5, 7, -2, 10}**

החלוקות האפשריות הן:

1. החלק השמאלי הוא {5} והממוצע שלו 5, החלק הימני הוא {7, -2, 10}

והממוצע שלו  $5 = (7 + (-2) + 10) / 3$ . ההפרש הוא  $5 - 5 = 0$

2. החלק השמאלי הוא {5, 7} והממוצע שלו  $6 = (5 + 7) / 2$ , החלק הימני הוא {-2, 10}

והממוצע שלו  $4 = ((-2) + 10) / 2$ . ההפרש הוא  $6 - 4 = 2$

3. החלק השמאלי הוא {5, 7, -2} והממוצע שלו  $3.3333 = (5 + 7 + (-2)) / 3$ , החלק הימני הוא {10}

והממוצע שלו 10. ההפרש הוא  $10 - 3.333 = 6.666$

ולכן השיטה תחזיר את החוליה שמכילה את הערך -2, שמחלקת את הרשימה כך שההפרש בין הממוצעים הוא מקסימלי. שימו לב שהשיטה לא מחזירה את ההפרש אלא את החוליה.

- **בהינתן הרשימה הבאה: {1, 0, 0, 0, 1}**

החלוקות האפשריות הן:

1. החלק השמאלי הוא {1} והממוצע שלו 1, החלק הימני הוא {0, 0, 0, 1}

והממוצע שלו  $0.25 = (0 + 0 + 0 + 1) / 4$ . ההפרש הוא  $1 - 0.25 = 0.75$

2. החלק השמאלי הוא {1, 0} והממוצע שלו  $0.5 = (1 + 0) / 2$ , החלק הימני הוא {0, 0, 1}

והממוצע שלו  $0.33 = (0 + 0 + 1) / 3$ . ההפרש הוא  $0.5 - 0.33 = 0.17$

3. החלק השמאלי הוא {1, 0, 0} והממוצע שלו  $0.33 = (1 + 0 + 0) / 3$ , החלק הימני הוא {0, 1}

והממוצע שלו  $0.5 = (0 + 1) / 2$ . ההפרש הוא  $0.5 - 0.33 = 0.17$

4. החלק הימני הוא {1, 0, 0, 0} והממוצע שלו  $0.25 = (0 + 0 + 0 + 1) / 4$ , החלק השמאלי הוא {1}

והממוצע שלו 1. ההפרש הוא  $1 - 0.25 = 0.75$

כאן יש שתי חוליות (ה-1 הראשון משמאל, והאפס הראשון מימין) שבהם ההפרש הוא מקסימלי (0.75) במקרה כזה השיטה תחזיר את החוליה שמכילה את החוליה **האחרונה** בה ההפרש הוא מקסימלי והיא ה-0 האחרון.

#### חתימת השיטה :

```
public IntNode averageNode();
```

בשתי השאלות האחרונות נשתמש ברשימה דו-סטריית. לשם כך נכתוב מחלקה בשם `IntListTwo` המייצגת רשימה מקושרת כזו. הרשימה תשתמש במחלקה `IntNodeTwo` הבאה, המייצגת איבר ברשימה:

```
public class IntNodeTwo
{
    private int _num;
    private IntNodeTwo _next, _prev;

    public IntNodeTwo(int n) {
        _num = n;
        _next = null;
        _prev = null;
    }

    public IntNodeTwo(int num, IntNodeTwo n, IntNodeTwo p) {
        _num = num;
        _next = n;
        _prev = p;
    }

    public int getNum() { return _num; }
    public IntNodeTwo getNext() { return _next; }
    public IntNodeTwo getPrev() { return _prev; }
    public void setNum (int n) { _num = n; }
    public void setNext (IntNodeTwo node) { _next = node; }
    public void setPrev (IntNodeTwo node) { _prev = node; }
}
```

אסור להוסיף למחלקה זו תכונות ו/או שיטות פרטיות או ציבוריות!

להלן המחלקה `IntListTwo`:

```
public class IntListTwo
{
    IntNodeTwo _head, _tail;
    // שימו לב שלא להגדיר את התכונות האלו כפרטיות!

    public IntListTwo()
    {
        _head = null;
        _tail = null;
    }

    // כאן ייכנסו השיטות שתכתבו בשאלות הבאות
```

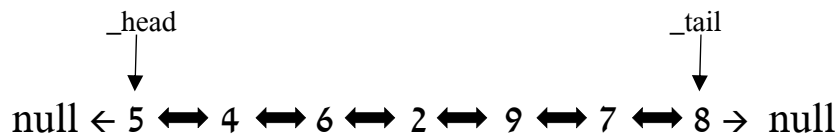
### שאלה 3 - 25 נקודות

1. הוסיפו למחלקה `IntListTwo` שיטה המקבלת כפרמטר מספר שלם ומוסיפה אותו לסוף הרשימה הדו-סטריט.

```
public void addToEnd (int num)
```

**חתימת השיטה:**

2. הוסיפו למחלקה `IntListTwo` את השיטה `toString` המחזירה את הרשימה כמחרוזת תווים מתחילת הרשימה לסופה, לפי הדוגמא הבאה **בדיוק**:  
אם נתונה הרשימה `list` הבאה:



המחרוזת שתוחזר תהיה `{5, 4, 6, 2, 9, 7, 8}`

```
public String toString()
```

**חתימת השיטה:**

שימו לב שהמחרוזת המוחזרת תהיה בדיוק כמו זו שבדוגמא. יש רווח אחד לאחר כל פסיק.  
ללא סימנים נוספים!

3. הוסיפו למחלקה `IntListTwo` את השיטה **הרקורסיבית** `toStringReverse` המחזירה את הרשימה כמחרוזת תווים מסוף הרשימה לתחילתה, לפי הדוגמא הבאה **בדיוק**:  
אם הרשימה `list` היא כמו לעיל, המחרוזת שתוחזר תהיה:  
`{8, 7, 9, 2, 6, 4, 5}`. גם כאן ללא רווחים מיותרים וללא סימנים נוספים.

השיטה `toStringReverse` **חייבת להיות רקורסיבית וללא שימוש בלולאות כלל!**

```
public String toStringReverse()
```

**חתימת השיטה:**

## שאלה 4 - 25 נקודות

נתונה רשימה מקושרת דו-סטריקט שבכל אחת מחוליותיה יש מספר שלם **חיובי**. נגדיר **מסלול חוקי** ברשימה כסדרה של חוליות ברשימה, המתחילה בראש הרשימה ומתקדמת ברשימה מספר צעדים **ימינה או שמאלה** לפי הערך שבחוליה. שימו לב שאי אפשר להתקדם מעבר לגבולות הרשימה. המסלול צריך להסתיים בחוליה האחרונה ברשימה. כתבו שיטה **רקורסיבית** בוליאנית המקבלת רשימה כמתואר לעיל ומחזירה true אם ישנו מסלול חוקי ברשימה, ו-false אחרת.

- עבור הרשימה {2, 4, 1, 6, 4, 2, 4, 3, 5} :

התשובה שתוחזר תהיה true שכן ישנו מסלול שמתחיל באיבר שבראש הרשימה (ערכו 2), הולך שני צעדים ימינה לאיבר שערכו 1, משם צעד אחד שמאלה לאיבר שערכו 4, משם ארבעה צעדים ימינה לאיבר שערכו 2, משם שוב שני צעדים ימינה לאיבר שערכו 3, משם שלושה צעדים שמאלה לאיבר שערכו 4 ומשם ארבעה צעדים ימינה לאיבר שערכו 5 שהוא האחרון ברשימה (הזנב).

- עבור הרשימה {1, 4, 3, 1, 2, 4, 3} :

התשובה שתוחזר תהיה false שכן אין אף מסלול שמתחיל בראש הרשימה ומגיע לזנבה לפי הקפיצות ימינה או שמאלה. אם נתחיל בראש הרשימה, נוזז צעד אחד ימינה לאיבר שערכו 4, משם חייבים לזוז ימינה ארבעה צעדים, כי שמאלה אי אפשר בגלל גבולות הרשימה, וכך מגיעים לאיבר שערכו 4 (שוב) משם חייבים לזוז שמאלה ארבעה צעדים, כי ימינה אי אפשר, (שימו לב שמאיבר זה אין אפשרות להתקדם 4 צעדים ימינה, כיוון שמסתיימת הרשימה) ולכן מגיעים שוב לאיבר השני (שערכו 4) כך שזהו תהליך אינסופי שלא מגיע לסוף הרשימה לעולם.

שימו לב להימנע מרקורסיות אינסופיות כאלו.

**חתימת השיטה היא :**

```
public boolean isWay()
```

**השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות. אפשר לשנות את הרשימה המקורית. אפשר להשתמש בהעמסת-יתר (overloading). אפשר להוסיף למחלקה שיטות פרטיות בלבד, ובלבד שגם הן לא יכילו לולאות.**

## 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 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 1 - 2 יש להגיש בקובץ Java ששמו IntList
5. את התשובות לשאלות 3 - 4 יש להגיש בקובץ Java ששמו IntListTwo
6. שימו לב ששני הקבצים שאתם שולחים חייבים להיות העתק מדויק (copy and paste) של הקבצים מהממ"ן, רק עם התוספות שלכם.
7. שימו לב שהתכונות של ראש וזנב הרשימה בקובץ IntListTwo לא הוגדרו ב-private. זאת לא טעות אלא נועד להקל על בדיקת הממ"ן שלכם. כלומר, באופן יוצא דופן (ובניגוד למה שעליכם לעשות בכל הקשר אחר) בראש המחלקה שלכם (שתועתק מהממ"ן, כמובן) צריך להופיע בדיוק כך :  
  
`;IntNodeTwo _head, _tail`  
  
ולא  
  
`;private IntNodeTwo _head, _tail`
8. אין להגיש את קובצי ה-API שכתבתם, וגם לא את הקבצים IntNode, IntNodeTwo.
9. ארזו את שני הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

## ב ה צ ל ח ה