

האוניברסיטה הפתוחה

20441
מבוא למדעי המחשב
ושפת Java
חוברת הקורס – סתיו 2025א

כתבה : תמר וילנר

אוקטובר 2024 – סמסטר סתיו – תשפ"ה

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
9	ממ"ן 12
21	ממ"ן 13
31	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)

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

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 הספריה באינטרנט 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. **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, 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/ 2025א)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	01.11.2024-29.10.2024	יחידות 1-2	מפגש 1 - הקדמה ויסודות השפה	
2	08.11.2024-03.11.2024	יחידה 2		
3	15.11.2024-10.11.2024	יחידות 3-4	מפגש 2 - מחלקות	ממ"ן 11 16.11.2024
4	22.11.2024-17.11.2024	יחידה 4		
5	29.11.2024-24.11.2024	יחידה 4	מפגש 3 – לולאות ומערכים	
6	06.12.2024-01.12.2024	יחידות 5-6		ממ"ן 12 7.12.2024
7	13.12.2024-08.12.2024	יחידה 9	מפגש 4 - יעילות	
8	20.12.2024-15.12.2024	יחידות 9-10	מפגש 5 – יעילות ורקורסיה	

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

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	27.12.2024-22.12.2024 (ה-ו חנוכה)	יחידה 10	מפגש 6 - רקורסיה	
10	03.01.2025-29.12.2024 (א-ה חנוכה)	יחידה 10		
11	10.01.2025-05.01.2025	יחידות 7-8	מפגש 7 – ירושה ופולימורפיזם	ממ"ן 13 11.1.2025
12	17.01.2025-12.01.2025	יחידה 7-8		
13	24.01.2025-19.01.2025	יחידה 11	מפגש 8 – רשימות מקושרות	
14	31.01.2025-26.01.2025	יחידה 12	מפגש 9 - עצים	ממ"ן 14 1.2.2025
15	03.02.2025-02.02.2025	חזרה	מפגש 10 - חזרה	

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 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

נושאי המטלה: יסודות השפה

חומר הלימוד למטלה: יחידות 1-2

משקל המטלה: 1 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 16.11.2024

סמסטר: 2025א

(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.
- 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 מאד לעיין בו ולפעול לפיו. הקובץ נמצא בלשונית "מדריכי עזר" שנמצאת ביחידה 1, והוא נקרא הנחיות לכתיבת 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%)

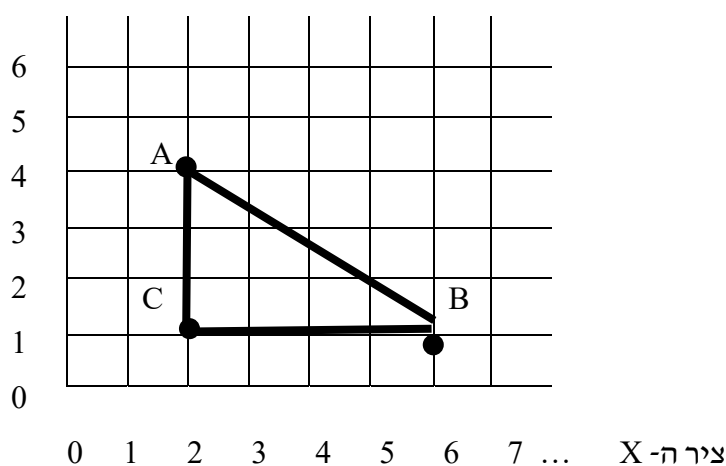
כתבו תכנית המקבלת שלוש נקודות במישור ומוצאת את המרחק הגדול ביותר בין שתי נקודות מהן.

התכנית תקרא מהקלט שהכניס המשתמש שישה מספרים שלמים (חיוביים, שליליים או אפס). כל זוג מספרים מייצג שיעורי נקודה אחת.

התכנית תחשב את המרחק בין כל זוג נקודות ותדפיס את שיעורי הנקודות שהמרחק ביניהן הוא הגדול ביותר.

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:

Max line created by the following points: (2,4), (6,1).

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 מקו אחד שאורכו מקסימלי, יש להציג את זוג הנקודות הראשונות שנבדקו. יש לבדוק את אורכי הקווים הנוצרים בסדר הבא (על פי הדוגמה לעיל): AC, BC, AB.

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 בפרק 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 MaxLine. המחלקה MaxLine תכיל שיטה אחת בלבד בשם main. אסור לכתוב שיטות נוספות.

לעזרתכם, כתבנו כאן חלקים מהמחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).

```
import java.util.Scanner;

public class MaxLine
{
    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 MaxLine
```

שימו לב שהמרחקים בין הנקודות אינם בהכרח מספרים שלמים.

שאלה 2 - להרצה (50%)

משחק "אבן, נייר ומספריים" הוא משחק בו כל אחד מן השחקנים מסתיר יד אחת מאחורי גבו ובוחר לו אחת מתנוחות האצבעות המסמלות את החפץ: אגרוף כאבן, יד פרושה כנייר ושתי אצבעות כמספריים.

המנצח הוא זה שהחפץ שלו גובר על החפץ של רעהו, כאשר כללי הניצחון הם: נייר גובר על אבן (הוא עוטף אותה), אבן גוברת על מספריים (היא שוברת אותם) ומספריים גוברים על נייר (הם גוזרים אותו). אם שני השחקנים מראים אותו חפץ, מתקיים תיקו.

מקור: ויקיפדיה.

קלט:

ערך שם החפץ של כל שחקן. ניתן להניח שערך שם החפץ נקלט כתו באמצעות אות גדולה:

- אבן – R
- נייר – P
- מספריים – S

פלט:

עבור הקלט 'S' עבור השחקן הראשון ו-'R' עבור השחקן השני, יתקבל הפלט:

Player 2 wins.

עבור הקלט 'P' עבור השחקן הראשון ו-'R' עבור השחקן השני, יתקבל הפלט:

Player 1 wins.

עבור הקלט 'R' עבור השחקן הראשון ו-'R' עבור השחקן השני, יתקבל הפלט:

Game ends with a tie.

ניתן להניח שהקלט יהיה חוקי. כלומר תמיד יתקבל 'R', 'P' או 'S'. אין צורך לבדוק זאת.

לעזרתכם, כתבנו כאן חלקים מהמחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).

```
import java.util.Scanner;

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

        System.out.println("Enter first player's object:");

        char player1 = scan.next().charAt(0);

        System.out.println("Enter second player's object:");

        char player2 = scan.next().charAt(0);

        // כאן עליכם להחשיף...

    } // end of method main
} //end of class Game
```

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

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 MaxLine ו-Game), 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 MaxLine.java.
3. הפתרון לשאלה 2 כולל את הקובץ Game.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: שימוש במחלקות נתונות,

כתיבת מחלקות

מספר השאלות: 3 משקל המטלה: 3 נקודות

סמסטר: 2025 מועד אחרון להגשה: 7.12.2024

(ת)

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

מעוניינים לפתח מערכת מידע עבור מרכז מעקב גדילה לתינוקות בקופות חולים.

לצורך כך נגדיר שלוש מחלקות:

- המחלקה Date שמייצגת תאריך
- המחלקה Weight שמייצגת משקל
- המחלקה Baby שמייצגת תינוק

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:

- תכונה בשם `_day` מטיפוס `int` – שמייצגת את היום (שלמים בין 1 ל-31);
- תכונה בשם `_month` מטיפוס `int` – שמייצגת את החודש (שלמים בין 1 ל-12);
- תכונה בשם `_year` מטיפוס `int` – שמייצגת את השנה (שלמים בין 1000 ל-9999);

למחלקה Date הוגדרו שלושה 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 2024. לא לשכוח להתייחס לשנים מעוברות (בלוח הגרגוריאני) בהן בחודש פברואר יש 29 ימים. ראו: [כאן](#).

- השני - בנאי ריק, שלא מקבל פרמטרים ויוצר את התאריך ה-1 בינואר 2024.

```
public Date()
```

- השלישי - בנאי העתקה המקבל תאריך אחר, ומעתיק את ערכיו.

```
public Date (Date other)
```

בנוסף הוגדרו במחלקה השיטות הציבוריות:

- שיטות האחזור:

```
getDay(), getMonth(), getYear().
```

- השיטות הקובעות:

```
setDay(int dayToSet), setMonth(int monthToSet),  
setYear(int yearToSet).
```

בשיטות 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 פרטית 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
בפורמט dd/mm/yyyy. 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()
```

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);
}
```

- **השנייה - השיטה הבוליאנית isLeapYear** אשר מקבלת 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 והבנאים הנדרשים 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 Weight מייצגת משקל:

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

- תכונה בשם `_kilos` מטיפוס `int` – שמייצגת את מספר הקילו (שלם חיובי או אפס)
- תכונה בשם `_grams` מטיפוס `int` – שמייצגת את מספר הגרם (שלם בטווח 0-999)

למחלקה Weight הוגדרו שלושה בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים (מספר קילו, מספר גרם) של המשקל.
`public Weight(int kilos, int grams)`
אם אחד הפרמטרים (או יותר) אינו חוקי, יש לאתחל אותו ל-0.
- השני - בנאי העתקה המקבל משקל אחר, ומעתיק את ערכיו.
`public Weight (Weight other)`
השלישי – בנאי המקבל פרמטר אחד (מספר גרמים כולל של המשקל).
אי אפשר להניח כי ערך הפרמטר חוקי. אם אינו חוקי, יש לאתחל ל-0 ק"ג ו-0 גרם.
- השלישי – בנאי המקבל פרמטר אחד (מספר גרמים כולל של המשקל).
אי אפשר להניח כי ערך הפרמטר חוקי. אם אינו חוקי, יש לאתחל ל-0 ק"ג ו-0 גרם.
`public Weight(int totalGrams)`

בנוסף הוגדרו במחלקה השיטות הציבוריות:

- שיטות האחזור:
`getKilos(), getGrams()`.
- שימו לב שאין שיטות קובעות (set) במחלקה Weight.
- השיטה `equals` המקבלת כפרמטר משקל נוסף ובודקת אם הוא שווה למשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה.

חתימת השיטה:

- `public boolean equals (Weight other)`
השיטה `lighter` המקבלת כפרמטר משקל נוסף ובודקת האם המשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה, קל יותר מהמשקל שהתקבל כפרמטר.

חתימת השיטה:

- `public boolean lighter (Weight other)`
השיטה `heavier` המקבלת כפרמטר משקל נוסף ובודקת האם המשקל שמיוצג על ידי האובייקט עליו מופעלת השיטה, כבד יותר מהמשקל שהתקבל כפרמטר. השיטה `heavier` חייבת להשתמש אך ורק בשיטה `lighter` (היא לא יכולה לחשב את התשובה לפי ערכי התכונות, ולא יכולה להשתמש בשיטה `equals` או בשיטות אחרות).

חתימת השיטה:

- `public boolean heavier (Weight other)`

- השיטה toString מחזירה מחרוזת תווים המייצגת את המשקל כך : kg.grams שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים. לדוגמא: המשקל 3 קילו ו- 55 גרם יוחזר כך : 3.055 . המשקל 4 קילו ו- 5 גרם יוחזר כך : 4.005 . שימו לב, אם יש 0 בסוף ערך הגרמים, הוא לא יופיע במחרוזת. למשל, 4 קילו ו- 70 גרם יוחזר 4.07 ולא 4.070. כך גם 3 קילו ו- 200 גרם יוחזר 3.2, ו- 4 קילו יוחזר 4.0

חתימת השיטה:

```
public String toString()
```

- השיטה add המקבלת תוספת של מספר גרמים grams ומחזירה משקל המהווה תוספת grams ביחס למשקל המיוצג על ידי האובייקט עליו מופעלת השיטה. שימו לב שצריך להחזיר משקל חדש ולא לשנות את האובייקט עליו מופעלת השיטה. שימו לב שמספר הגרמים grams יכול להיות גם שלילי. אם לאחר התוספת (השלילית) המשקל יהיה שלילי, השיטה לא תעשה כלום, ולא תוסיף את הערך grams. האובייקט שיוחזר יהיה עותק של האובייקט המקורי.

לדוגמה,

- אם המשקל עליו מופעלת השיטה הוא 150(g), 3(kg) ו-900 grams אזי השיטה add תחזיר את המשקל 50(g), 4(kg)
- אם המשקל עליו מופעלת השיטה הוא 150(g), 3(kg) ו-250 grams אזי השיטה add תחזיר את המשקל 900(g), 2(kg)
- אם המשקל עליו מופעלת השיטה הוא 150(g), 3(kg) ו-3250 grams אזי השיטה add תחזיר עותק של המשקל המקורי 150(g), 3(kg)

חתימת השיטה:

```
public Weight add (int grams)
```

עליכם לכתוב את המחלקה Weight לפי ההגדרות לעיל.

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

שאלה 3 - 50 נקודות

המחלקה Baby מייצגת תינוק:

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

- תכונה בשם `_firstName` מטיפוס `String` – שמייצגת את השם הפרטי של התינוק
- תכונה בשם `_lastName` מטיפוס `String` – שמייצגת את שם המשפחה של התינוק
- תכונה בשם `_id` מטיפוס `String` – שמייצגת את מספר זהות התינוק
- תכונה בשם `_dateOfBirth` מטיפוס `Date` – שמייצגת את תאריך לידת התינוק
- תכונה בשם `_birthWeight` מטיפוס `Weight` – שמייצגת את משקל התינוק בזמן הלידה
- תכונה בשם `_currentWeight` מטיפוס `Weight` – שמייצגת את משקלו העכשווי של התינוק

למחלקה Baby הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל את הפרמטרים הבאים:
 - שמו הפרטי של התינוק. ניתן להניח כי הערך המתקבל אינו null והמחרוזת אינה ריקה.
 - שם המשפחה של התינוק. ניתן להניח כי הערך המתקבל אינו null והמחרוזת אינה ריקה.
 - מספר זהות התינוק. אם כמות התווים אינה חוקית (כלומר מספר הזהות אינו מורכב מ-9 תווים), יש לאתחל למחרוזת "000000000". ניתן להניח כי התווים מורכבים מספרות בלבד, ואין צורך לבדוק זאת.
 - יום, חודש ושנה של מועד הולדת התינוק. טיפול תקינות ערכי התאריך יטופלו במסגרת הבנאי במחלקה `Date`.
 - משקל התינוק בזמן הלידה בגרמים. טיפול תקינות ערך המשקל יטופל במסגרת הבנאי במחלקה `Weight`.
- שימו לב שהבנאי לא מקבל את משקל התינוק העכשווי כפרמטר. כשנוצר האובייקט, המשקל העכשווי יהיה בדיוק כמו המשקל בזמן הלידה בגרמים.

חתימת הבנאי היא:

```
public Baby(String fName, String lName, String id,
            int day, int month, int year, int birthWeightInGrams)
```

- השני - בנאי העתקה המקבל תינוק אחר, ומעתיק את ערכיו.

חתימת הבנאי היא:

```
public Baby(Baby other)
```

בנוסף הוגדרו במחלקה השיטות הציבוריות:

הוגדרו שיטות האחזור (get) והשיטות הקובעות (set) לפי החתימות הבאות:

- שיטות האחזור:

```
public String getFirstName()  
public String getLastName()  
public String getId()  
public Date getDateOfBirth()  
public Weight getBirthWeight()  
public Weight getCurrentWeight()
```

- השיטה הקובעת:

```
public void setCurrentWeight(Weight weightToSet)
```

שימו לב שיש במחלקה Baby רק שיטה קובעת אחת.

- השיטה toString מחזירה מחרוזת תווים המייצגת את התינוק כך:

Name: Ariel Israeli

Id: 123456789

Date of Birth: 03/08/2024

Birth Weight: 3.005

Current Weight: 3.425

שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים. שימו לב לרווח אחרי התו : (נקודותיים) ולירידות השורה. לאחר השורה האחרונה של המשקל העכשווי יש להוסיף ירידת שורה.

חתימת השיטה:

```
public String toString()
```

- השיטה הבוליאנית equals המקבלת כפרמטר תינוק נוסף ובודקת אם הוא זהה לתינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה. השיטה תחזיר true אם ערכי התינוקות זהים בשם, תאריך הולדתם ומספרי הזהות שלהם. אחרת, תחזיר false.

חתימת השיטה:

```
public boolean equals (Baby other)
```

- השיטה הבוליאנית areTwins המקבלת כפרמטר תינוק נוסף ובודקת אם הוא תאום של התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה. התינוקות הם תאומים אם שם המשפחה שלהם זהה, השם הפרטי שלהם שונה, מספרי ת"ז של התינוקות שונים זה מזה ותאריך הלידה שלהם זהה או שונה ביום אחד בדיוק (כלומר, נולדו באותו יום או בימים עוקבים). אם כל התנאים מתקיימים, השיטה תחזיר true, ואחרת, תחזיר false.

חתימת השיטה:

```
public boolean areTwins (Baby other)
```

- השיטה הבוליאנית heavier המקבלת כפרמטר תינוק נוסף. השיטה תחזיר true אם התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה כבד יותר לפי המשקל העכשווי. אחרת, תחזיר false.

חתימת השיטה:

```
public boolean heavier (Baby other)
```

- השיטה updateCurrentWeight המקבלת כפרמטר grams מספר גרמים לעדכון המשקל העכשווי. השיטה תעדכן את משקל התינוק בתוספת grams גרמים. הערך grams יכול להיות שלילי.

אם לאחר התוספת (השלילית) המשקל העכשווי של התינוק יהיה שלילי, השיטה לא תעשה כלום, ולא תוסיף את הערך grams.

שימו לב להבדל בין השיטה הזו לבין השיטה setCurrentWeight.

חתימת השיטה:

```
public void updateCurrentWeight (int grams)
```

- השיטה הבוליאנית older המקבלת כפרמטר תינוק נוסף. השיטה תחזיר true אם תאריך הולדת התינוק שמיוצג על ידי האובייקט עליו מופעלת השיטה קודם לתאריך הולדת התינוק שהתקבל כפרמטר. אחרת, יוחזר false.

חתימת השיטה:

```
public boolean older (Baby other)
```

- השיטה isWeightInValidRange המקבלת תאריך date, ובודקת אם ההתקדמות של התינוק במשקל מאז שנולד ועד לתאריך date היא תקינה.

משקלה של שיטה זו הוא 15 נקודות.

להלן יוגדרו כללי ההתקדמות התקינה (לפי אתר אינטרנט של אחת מקופות החולים של ישראל):

הכללים הללו הם חלקיים בלבד ואין לראות בהם איזו המלצה או חוות דעת רפואית! כאן כתבנו אותם לצורך תרגול תכנות בלבד!

- בשבוע הראשון לאחר הלידה חלה ירידה טבעית של כעשרה אחוזים במשקל גופם של יילודים. ירידה של למעלה מעשרה אחוזים אינה תקינה. עליה היא תקינה.
- במהלך החודשיים הראשונים לחיים צפויה עליה של כ- 30 גרם במשקל התינוק מדי יום (לאחר השבוע הראשון).
- עד גיל 4 חודשים צפוי התינוק לעלות כ- 750 גרם מדי חודש (לאחר החודשיים הראשונים).
- מגיל 4 חודשים ועד גיל 8 חודשים אמור התינוק לעלות כ- 500 גרם מדי חודש, ולאחר גיל 8 חודשים ועד גיל שנה אמור התינוק לעלות כ- 250 גרם מדי חודש.

לעניין התרגיל שלנו –

- אנחנו לא נעסוק בשבועות או בחודשים, ונתייחס אך ורק לימים.
- אנו נתייחס לשבוע כאל 7 ימים, לחודש כאל 30 ימים (תמיד) ולשנה כאל 365 ימים.
- אם התינוק הוסיף למשקלו יותר ממה שכתוב בכללים, זה תקין. פחות ממה שכתוב, אינו תקין.

השיטה תחזיר ערך שלם בין 1 ל- 4 לפי התוצאה לפי ההנחיות הבאות:

- אם התאריך שניתן כפרמטר הוא קודם לתאריך הלידה של התינוק עליו מופעלת השיטה, השיטה תחזיר את הערך 1.
- אם התאריך תקין, אבל כבר לא רלוונטי (התינוק בן שנה ומעלה), השיטה תחזיר 2.
- אם ההתקדמות לא תקינה לפי הכללים לעיל, השיטה תחזיר 3.
- אם ההתקדמות תקינה לפי הכללים לעיל, השיטה תחזיר 4.

חתימת השיטה:

`public int isWeightInValidRange (Date date)`

דוגמאות לשיטה `isWeightInValidRange`:

- התינוק נולד ב- 1.2.2024 במשקל 3 ק"ג, המשקל הנוכחי 5 ק"ג.
 - אם התאריך בפרמטר הוא 25.1.2024 יוחזר הערך 1 (כי התאריך בפרמטר קודם לתאריך הלידה).
 - אם התאריך בפרמטר הוא 25.2.2025 יוחזר הערך 2 (כי התינוק בן למעלה משנה).

- אם התאריך בפרמטר הוא 1.6.2024 יוחזר הערך 3 (כי התינוק עלה במשקל רק 2 ק"ג בעוד שהיה צריך לעלות לפחות 3 ק"ג בארבעה חודשים).
- אם התאריך בפרמטר הוא 30.3.2024 יוחזר הערך 4 (כי ההתקדמות תקינה – עברו 59 יום מתאריך הלידה עד התאריך בפרמטר, והתינוק היה אמור לרדת במשקל 300 גרם בשבוע הראשון, ואז לעלות 30 גרם בכל יום ב- 52 הימים שנותרו, כלומר משקלו היה אמור להיות 4.260 . משקלו הוא 5 ק"ג, זה יותר ממה שהיה צריך, ולכן ההתקדמות תקינה).

עליכם לכתוב את המחלקה Baby 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 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`, `Weight.java`, `Baby.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: 2025 מועד אחרון להגשה: 11.1.2025

**את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם
Ex13 (בדיוק).**

שימו לב:

מניסיוננו השאלות שבהן מתקשים סטודנטים במבחנים הן בעיקר שתי השאלות הראשונות בהן אתם נדרשים לכתוב קוד: שאלת רקורסיה ושאלת יעילות.

הממ"ן שלפניכם בנוי משאלות שהיו במבחנים בסמסטרים האחרונים. התמודדות עצמאית לחלוטין עם שאלות הממ"ן תהווה הכנה מצוינת למבחן עבורכם. כמובן שאין להסתפק בכך ויש לפתור עוד ועוד שאלות רקורסיה ויעילות לקראת המבחן.

שתי השאלות הראשונות עוסקות בנושא היעילות והשתיים האחרונות ברקורסיה.

הערות 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.

- 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

נתון מערך חד ממדי arr בגודל n ובו מספרים שלמים שונים זה מזה.

נגדיר חציון (median) של המערך arr, הוא המספר במערך arr ש- $n/2$ מהאיברים במערך קטנים ממנו ו- $n/2$ מהאיברים במערך גדולים ממנו. אם מספר התאים במערך הוא זוגי, אנחנו נגדיר שהחציון הוא המספר במערך ש- $n/2$ מהאיברים במערך קטנים ממנו ו- $n/2-1$ מהאיברים במערך גדולים ממנו.

לדוגמא,

- במערך arr1 להלן המספר 1 הוא החציון. שכן האיברים, -3, -5, 0 קטנים ממנו והמספרים 2, 4, 7 גדולים ממנו.

	0	1	2	3	4	5	6
arr1 =	4	-5	-3	1	2	7	0

- במערך arr2 להלן המספר 2 הוא החציון. שכן האיברים, -3, 1, -5, 0 קטנים ממנו והמספרים 4, 7, 9 גדולים ממנו.

	0	1	2	3	4	5	6	7
arr2 =	4	-5	-3	1	2	7	9	0

נגדיר את arr כמערך-מיוחד (special-array) אם מתקיים:

$$arr[0] > arr[1] < arr[2] > arr[3] < arr[4] < \dots > arr[n-2] < arr[n-1]$$

כאשר המערך באורך אי-זוגי (ואז האיבר הלפני-אחרון קטן מהאיבר האחרון);

$$arr[0] > arr[1] < arr[2] > arr[3] < arr[4] < \dots < arr[n-2] > arr[n-1]$$

כאשר המערך באורך זוגי (ואז האיבר הלפני-אחרון גדול מהאיבר האחרון במערך).

כלומר, כל האיברים באינדקסים האי-זוגיים קטנים משני שכניהם, וכל האיברים באינדקסים הזוגיים גדולים משני שכניהם.

אין דרישה ליחס סדר בין איברים שאינם שכנים.

הנה המערכים arr1 ו-arr2 לאחר שסידרנו אותם כך שיהיו מערכים-מיוחדים.

	0	1	2	3	4	5	6
arr1 =	1	-5	7	0	2	-3	4

	0	1	2	3	4	5	6	7
arr2 =	4	-5	7	1	2	-3	9	0

שימו לב שאלו אינם המערכים-המיוחדים היחידים שאפשר ליצור מהמערכים arr1 ו-arr2 לעיל. גם אלו מערכים מיוחדים:

arr1 = {7, -3, 1, -5, 4, 0, 2}

arr2 = {9, -3, 7, 0, 4, 1, 2, -5}

ויש עוד.

כתבו שיטה סטטית המקבלת כפרמטרים מערך חד-ממדי arr מלא במספרים שלמים, שאיבריו שונים זה מזה, ומספר שלם med שהוא החציון של המערך arr. השיטה צריכה לבנות מערך חדש, שהערכים בתאיו הם ערכי המערך arr והם מסודרים כמערך-מיוחד, ולהחזיר את המערך החדש הזה. אין חשיבות לסדר מסוים בין איברים שאינם שכנים.

חתימת השיטה היא:

```
public static int[] specialArr (int[] arr, int med)
```

בשאלה זו אסור לשנות את המערך, גם אם הוא חוזר לקדמותו לאחר השיטה!

הערה חשובה –

שימו לב שאתם צריכים להכניס כפרמטר לשיטה specialArr את החציון של המערך. כאשר אתם בודקים את השיטה שכתבתם, כדאי מאד שתכתבו לעצמכם שיטה שמחשבת את החציון, כדי שבוודאי הפרמטר שתכניסו יהיה נכון. סיבוכיות השיטה הזו אינה חשובה, ואתם לא צריכים להתייחס אליה בסיבוכיות השיטה specialArr. נא לא לשלוח את השיטה שמחשבת את החציון! היא לצרכיכם בלבד.

שאלה 2 - 25 נקודות

כתבו שיטה יעילה המקבלת כפרמטר מערך חד-ממדי `arr` המלא במספרים שלמים לא ממוינים. השיטה מחזירה מהו המספר החיובי הקטן ביותר שלא נמצא במערך. (0 אינו מספר חיובי).

חתימת השיטה היא:

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

דוגמאות:

- עבור המערך `arr` הבא:

0	1	2	3	4	5
1	-3	6	2	0	15

המספר החיובי הקטן ביותר שלא נמצא במערך הוא 3. כי 1 ו-2 נמצאים במערך (בתאים 0 ו-3), אבל 3 לא.

- עבור המערך `arr` הבא: `arr[] = {1, 1, 1, 1}` השיטה תחזיר 2
- עבור המערך `arr` הבא: `arr[] = {1, 2, 3, 4}` השיטה תחזיר 5
- עבור המערך `arr` הבא: `arr[] = {5, -1, 3, 1, 0, -2, 2}` השיטה תחזיר 4
- עבור המערך `arr` הבא: `arr[] = {7, 8, 9, 11, 12, 14}` השיטה תחזיר 1

שימו לב:

- סיבוכיות המקום צריכה להיות קבועה, אחרת יורדו הרבה נקודות.

נקודות.

- בשאלה זו מותר לשנות את המערך במהלך השיטה, ואין צורך להחזיר אותו למצבו המקורי בסיומה.

הערות לגבי שאלות 3 ו-4:

- מותר להשתמש בהעמסת-יתר (Overloading)
- אסור להשתמש במשתנים סטטיים (גלובליים)!
- מותר להשתמש בשיטות Math.max ו-Math.min מהמחלקה Math וכן בקבועים Integer.MIN_VALUE, Integer.MAX_VALUE מהמחלקה Integer.
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

שאלה 3 - 25 נקודות

פלינדרום (Palindrome) הוא מילה, מספר, משפט או כל רצף סמלים אחר שקריאתו מימין לשמאל ומשמאל לימין היא זהה.

לדוגמא: מחרוזות התווים "aba", "1221" הן פלינדרום.

במערך חד-ממדי המכיל מספרים שלמים, נגדיר **רצף פלינדרומי** כסדרה של תאים רצופים במערך המהווים פלינדרום.

נגדיר: רצף כמעט-פלינדרומי (Nearly Palindrome) הוא רצף מספרים במערך חד ממדי כך שאם נשמיט ממנו ערך אחד **לכל היותר** (בתוך הרצף ולא בקצה) נקבל רצף פלינדרומי.

שימו לב שכל רצף פלינדרומי הוא גם רצף כמעט-פלינדרומי, כי לא משמיטים שום ערך ובכל זאת מקבלים רצף פלינדרומי.

לדוגמא, במערך arr הבא:

0	1	2	3	4	5	6	7	8
1	1	4	10	10	4	3	10	10

ישנם שישה רצפים כמעט-פלינדרומיים:

1. בין האינדקסים 0 ל-1 {1, 1} זהו פלינדרום באורך 2 ולכן הוא גם כמעט-פלינדרום
2. בין האינדקסים 3 ל-4 {10, 10} זהו פלינדרום באורך 2 ולכן הוא גם כמעט-פלינדרום
3. בין האינדקסים 7 ל-8 {10, 10} זהו פלינדרום באורך 2 ולכן הוא גם כמעט-פלינדרום
4. בין האינדקסים 2 ל-5 {4, 10, 10, 4} זהו פלינדרום באורך 4 ולכן זה כמעט-פלינדרום

5. בין האינדקסים 4 ל-7 {10, 4, 3, 10} זהו כמעט-פלינדרום שאורכו 4.

כי אם נוריד את 4 או את 3 נקבל פלינדרום.

6. בין האינדקסים 3 ל-8 {10, 10, 4, 3, 10, 10} זהו כמעט-פלינדרום שאורכו 6.

כי אם נוריד את 4 או את 3 נקבל פלינדרום.

ועוד תשעה רצפים כמעט-פלינדרומיים, כל אחד מהם באורך 1, שהם כל אחד מהתאים (כל אחד מהם הוא פלינדרום, ולכן הוא כמעט-פלינדרום).

שימו לב שהרצף בין האינדקסים 2 ל-6 {4, 10, 10, 4, 3} אינו כמעט-פלינדרום כי הערך שהתווסף לפלינדרום {4, 10, 10, 4} נמצא **בקצה הרצף ולא בתוכו**.

כתבו שיטה **סטטית רקורסיבית**, המקבלת כפרמטר מערך חד-ממדי `arr` המכיל מספרים שלמים ומחזירה את אורכו של הרצף **הכמעט-פלינדרומי** הגדול ביותר.

לדוגמא, במערך `arr` שלעיל, הרצף הכמעט-פלינדרומי הגדול ביותר הוא בין האינדקסים 3 ל-8, ואורכו הוא 6. לכן, השיטה צריכה להחזיר את הערך 6.
על המערך `arr = {1, 2, 3, 4}` השיטה תחזיר 1.

חתימת השיטה היא:

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

אסור לשנות את המערך, אפילו לא באופן זמני!

שאלה 4 - 25 נקודות

נתון מערך דו-ממדי ריבועי `mat` המכיל מספרים שלמים. נסמן את מספר השורות והעמודות במערך ב- `n`. המספרים ב- `mat` הם חיוביים ממש בלבד (ללא אפסים).

המסלולים בהם נעבור במערך מתחילים תמיד בתא הראשון $(0,0)$ ועד לתא האחרון שהוא התא $(mat.length-1, mat.length-1)$, כאשר אפשר לעבור מתא (i, j) לכל ארבעת שכניו מימין, משמאל, למעלה ולמטה, אבל לא באלכסון.

אנחנו מעוניינים למצוא את המסלול שהמספר המקסימלי בו הוא המינימלי מבין המספרים המקסימלים במסלולים האחרים האפשריים. במילים אחרות, אם נסתכל על כל המסלולים במערך לפי המתואר לעיל, ובכל מסלול נמצא את האיבר המקסימלי, נרצה להחזיר את האיבר המקסימלי הכי קטן מכולם.

כתבו שיטה סטטית רקורסיבית המקבלת מערך דו-ממדי ריבועי `mat` המלא במספרים שלמים, חיוביים ממש בלבד, ומחזירה את ערכו של המספר המינימלי מבין המספרים המקסימלים בכל המסלולים האפשריים.

דוגמאות:

		נתונים המערכים הבאים:	
		0	1
B =	0	1	3
	1	4	2

		0	1
A =	0	1	2
	1	3	4

בכל אחד מהמערכים A ו-B יש שני מסלולים מהתא הראשון לאחרון:

○ המסלול הראשון עובר בתאים $[0][0] \rightarrow [0][1] \rightarrow [1][1]$

○ המסלול השני עובר בתאים $[0][0] \rightarrow [1][0] \rightarrow [1][1]$

במערך A המספר המקסימלי במסלול הראשון הוא 4, והמספר המקסימלי במסלול

השני הוא 4. לכן יוחזר 4

במערך B המספר המקסימלי במסלול הראשון הוא 3, והמספר המקסימלי במסלול

השני הוא 4. לכן יוחזר 3.

• אם המערכים הם:

	0	1	2	3
0	4	5	8	2
D = 1	3	12	16	7
2	13	1	10	14
3	15	11	9	6

	0	1	2	3
0	4	5	8	2
1	3	12	7	16
2	13	1	10	14
3	15	11	9	6

אז במערך C המספר שיוחזר יהיה 10, שהוא המקסימום במסלול המסומן, והוא המינימלי מבין כל ערכי המקסימום של כל המסלולים האחרים.

אבל אם נחליף את התאים [1][3] שבו יש 16 ואת [1][2] שבו יש 7, (זה המערך D), המספר שיוחזר יהיה 12, שהוא המקסימום במסלול המסומן. שימו לב ש-12 הוא המקסימום גם במסלול הזה: $[0][0] \rightarrow [0][1] \rightarrow [1][1] \rightarrow [2][1] \rightarrow [2][2] \rightarrow [3][2] \rightarrow [3][3]$

	0	1	2	3	4
0	4	1	9	3	25
D = 1	24	23	2	21	5
2	13	12	15	16	22
3	17	11	18	19	20
4	10	14	8	7	6

	0	1	2	3	4
0	4	1	9	3	25
1	24	23	22	21	5
2	13	12	15	16	14
3	17	11	18	19	20
4	10	2	8	7	6

אז במערך C המספר שיוחזר יהיה 21, שהוא המקסימום במסלול המסומן, והוא המינימלי מבין כל ערכי המקסימום של כל המסלולים האחרים. שימו לב ש-21 הוא המקסימום בעוד כמה מסלולים. לדוגמא:

1. $[0][0] \rightarrow [0][1] \rightarrow [0][2] \rightarrow [0][3] \rightarrow [1][3] \rightarrow [2][3] \rightarrow [3][3] \rightarrow [4][3] \rightarrow [4][4]$
 \rightarrow
 2. $[0][0] \rightarrow [0][1] \rightarrow [0][2] \rightarrow [0][3] \rightarrow [1][3] \rightarrow [2][3] \rightarrow [2][2] \rightarrow [3][2] \rightarrow [4][2]$
 \rightarrow
 $\rightarrow [4][3] \rightarrow [4][4]$

אבל אם נחליף את התאים 2, 14, 22 במערך (זה המערך D), המספר שיוחזר יהיה 15, שהוא המקסימום במסלול המסומן.

חתימת השיטה היא:

```
public static int extreme(int [][] m)
```

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

- מותר לשנות את המטריצה במהלך השיטה, אבל המטריצה צריכה לחזור לקדמותה לאחר ביצוע השיטה.
- המטריצה בהכרח ריבועית. כלומר, מספר השורות שווה למספר העמודות.

שימו לב:

- בכל השאלות - אל תשכחו לתעד (באנגלית בלבד) את מה שכתבתם!
- שמנו טסטר למחלקה Ex13 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**.

- אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.

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**). ארזו את הקובץ בתוך קובץ zip. אין לשלוח קבצים נוספים.

מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: רשימות מקושרות

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 1.2.2025

סמסטר: 2025א

ביחידה 6 למדנו על מערכים דו-ממדיים (מטריצות). במטלה זו נעסוק ברשימות מקושרות של מספרים שלמים, כאשר נייצג מטריצה בשימוש ברשימות מקושרות. הרשימות שלנו ייצגו אך ורק מטריצות מלבניות (כלומר, שבכל השורות יש אותו מספר עמודות). לא בהכרח מספר השורות שווה למספר העמודות.

כל תא במטריצה ייוצג על ידי חוליה שיש לה ארבעה מצביעים. אחד לכל "שכן" של התא (למעלה, למטה, מימין ומשמאל). לשם כך מצורף לממ"ן זה קובץ המחלקה `IntNodeMat` המייצגת חוליה ברשימה שהיא תא במערך. המערך ייוצג על ידי רשימה של חוליות כאלו. המחלקה המייצגת את המערך, מצורפת גם היא למטלה זו ונקראת `MatrixList` המייצגת רשימה מקושרת כזו.

פתחו את קובץ המחלקה `IntNodeMat` ועברו עליו. בראש הקובץ נמצאות הגדרות התכונות של המחלקה וכן כמה בנאים ושיטות עזר בהן תוכלו להסתייע בעבודתכם.

פתחו את קובץ המחלקה `MatrixList` ועברו עליו. זהו הקובץ בתוכו תממשו את הפתרון שלכם לממ"ן זה. בראש הקובץ נמצאות הגדרות התכונות של המחלקה וכן כמה בנאים ושיטות עזר בהן תוכלו להסתייע בעבודתכם.

הורידו את שתי המחלקות הללו אל המחשב שלכם ועבדו על הקובץ `MatrixList.java`. אל תמחקו ואל תשנו דבר בקבצים שהעתקתם, רק הוסיפו לקובץ `MatrixList.java` (ורק לו) את הקוד שלכם!

הערות חשובות לגבי כל השאלות בממ"ן זה:

- אסור לכם להעביר את תוכן הרשימה למערך ואז לעבוד על המערך. כך גם לא להעביר את תוכן הרשימה לרשימה אחרת ועבודה עליה. פתרון בגישה כזאת יוביל להורדה של כמעט כל הנקודות על השאלה.
- מותר לשיטות שאתם כותבים לשנות את תוכן או את מבנה הרשימה עליה אתם עובדים. עם זאת כאשר השיטה מסיימת את עבודתה, המבנה ותוכן הרשימה חייבים להיות בדיוק כמו שהיו לפני הקריאה לשיטה.
- אסור להוסיף תכונות למחלקות IntNodeMat ו-MatrixList. קבועים מותר להוסיף.
- אפשר להניח שהפרמטר שמתקבל לכל אחת מהשיטות אינו null. אי אפשר להניח מעבר לכך כל הנחה שהיא לגבי גודל הרשימה. כלומר, יכול להיות שהרשימה תהיה ריקה (כלומר head_ יצביע על null).
- כתבו באנגלית API לכל השיטות הציבוריות שלכם. הוסיפו תיעוד רגיל (באנגלית) בתוך השיטות.
- ניתן להשתמש בשיטות עזר פרטיות ככל הנדרש. לשיטות אלו עליכם לכתוב תיעוד רגיל בלבד, לא API, הן לפני כל שיטה (מה היא עושה) והן בתוכה.
- אין להשתמש בשום מחלקה בג'אווה ובשום מרכיבים של השפה שלא נלמדו במהלך הקורס. אי עמידה בכלל זה יוביל להורדה משמעותית ביותר של נקודות.

למחלקה MatrixList יש תכונה אחת – מצביע על החוליה הראשונה במטריצה (שורה 0 ועמודה 0).
היא מטיפוס IntNodeMat ונקראת _m00.

IntNodeMat _m00;

שאלה 1 - 20 נקודות

במחלקה MatrixList יש שני בנאים:

1. בנאי ריק – היוצר אובייקט שמצביע על מטריצה ריקה (החוליה הראשונה היא null).

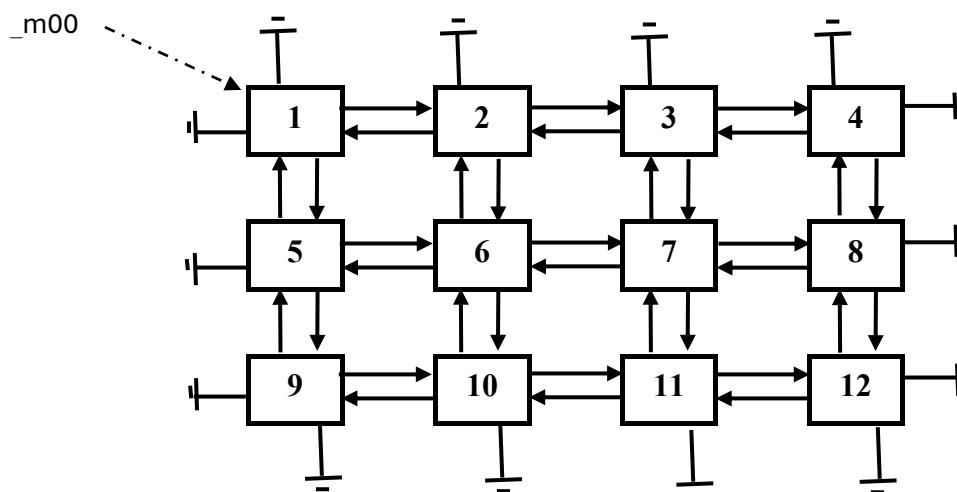
```
public MatrixList()
{
    _m00= null;
}
```

2. בנאי המקבל מטריצה mat ויוצר רשימה המייצגת את המטריצה mat.

לדוגמא, אם המטריצה היא:

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

המטריצה שתיווצר מהבנאי תהיה:



חתימת הבנאי:

```
public MatrixList(int[][]mat)
{
    // את זה עליכם להשלים
}
```

שאלה 2 - 30 נקודות

הוסיפו למחלקה MatrixList את השיטות הבאות:

א. `toString` – המחזירה ייצוג של המטריצה בצורת מחרוזת כאשר בין שורה לשורה במטריצה יש ירידת שורה "`\n`" (לאחר השורה האחרונה יש ירידת שורה), ובין עמודה לעמודה במטריצה יש טאב "`\t`" (לאחר העמודה האחרונה אין טאב).

לדוגמא, אם המטריצה היא זו שלעיל, המחרוזת שתוחזר צריכה להראות כך (בדיוק!):

1	2	3	4
5	6	7	8
9	10	11	12

שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים נוספים וללא תווים נוספים.

אם המטריצה ריקה, המחרוזת שהשיטה תחזיר צריכה להיות המחרוזת הריקה: "".

חתימת השיטה:

```
public String toString()
```

ב. השיטה `getData_i_j`, המקבלת כפרמטרים משתנים `i` ו-`j` המייצגים את השורה והעמודה (בהתאמה), ומחזירה את הערך שיש בתא ה-`[i][j]` במטריצה.

אם אחד מהפרמטרים לא תקין (לדוגמא, יש 3 שורות במטריצה, ו-`i = 4`), השיטה תחזיר את הערך `Integer.MIN_VALUE` (המספר השלם הקטן ביותר האפשרי).

חתימת השיטה:

```
public int getData_i_j(int i, int j)
```

ג. השיטה `setData_i_j`, המקבלת כפרמטרים משתנה שלם `num` ומשתנים `i` ו-`j` המייצגים את השורה והעמודה (בהתאמה), ושמה את הערך `num` בתא ה-`[i][j]` במטריצה.

אם אחד מהפרמטרים לא תקין (לדוגמא, יש 3 שורות במטריצה, ו-`i = 4`), השיטה לא תבצע כלום.

חתימת השיטה:

```
public void setData_i_j(int num, int i, int j)
```


שאלה 3 - 25 נקודות

הוסיפו למחלקה MatrixList את השיטה הבאה:

כתבו שיטה **רקורסיבית** בשם findMax המחזירה את המספר המקסימלי שנמצא ברשימה (שמייצגת את המטריצה).

חתימת השיטה:

```
public int findMax()
```

- השיטה שתכתבו צריכה להיות ללא כל שימוש בלולאות מכל סוג שהוא.
- מותר לכם לכתוב שיטות עזר, אך גם הן חייבות להיות ללא כל שימוש בלולאות מכל סוג שהוא.
- כל שימוש בלולאה בתשובה יוביל להורדה משמעותית ביותר של נקודות.
- אפשר להשתמש בהעמסת-יתר (overloading).
- אין צורך לדאוג ליעילות השיטה הרקורסיבית, ואין צורך לציין מה סיבוכיות הזמן שלה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!

שאלה 4 - 25 נקודות

בשאלה זו אנו מניחים שהמטריצה ממוינת בצורה הבאה:

- כל שורה במטריצה ממוינת בסדר עולה ממש (ללא מספרים שווים באותה שורה)
- כל עמודה במטריצה ממוינת בסדר עולה ממש (ללא מספרים שווים באותה עמודה)

כתבו שיטה **יעילה** בשם howManyX, המקבלת מספר שלם x ומחזירה כמה פעמים הערך x נמצא במטריצה עליה מופעלת השיטה.

חתימת השיטה:

```
public int howManyX(int x)
```

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.

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

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 שלנו והציון יהיה 0.**
3. עליכם להריץ את הטסטר שנמצא באתר הקורס על המחלקה שכתבתם. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקה. מאד מומלץ להוסיף לו בדיקות. שימו לב שאם הטסטר לא יעבור קומפילציה מול המחלקה שכתבתם, הציון על המטלה יהיה אפס. אם יש שיטה שאתם מעוניינים לדלג עליה, עליכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטר יעבור קומפילציה.
4. **גם במטלה זו – אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.**
5. את התשובות לכל השאלות יש לכתוב בקובץ Java ששמו `MatrixList` **שאנחנו שמנו באתר.** אין להוסיף אף קובץ אחר!
6. שימו לב שהקובץ שאתם שולחים חייב להיות אותו הקובץ שאנחנו שמנו באתר, רק עם התוספות שלכם.
7. שימו לב שהתכונה `_m00` בקובץ `MatrixList` לא הוגדרה 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 צריך להופיע בדיוק כך :
`MatrixList _m00;`
ולא
`private MatrixList _m00;`
8. **אין להגיש את קובץ ה-API שנוצר מה-API שכתבתם, וגם לא את הקובץ `IntNodeMat`.**
9. **ארזו את קובץ ה-java ששמו `MatrixList` בקובץ `zip` יחיד ושלחו אותו בלבד.**