

האוניברסיטה הפתוחה

20441
מבוא למדעי המחשב
ושפת Java
חוברת הקורס – סתיו 2022א

כתבה: תמר וילנר

אוקטובר 2021 – סמסטר סתיו – תשפ"ב

פנימי – לא להפצה.

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

תוכן העניינים

א	אל הסטודנט
ה	1. לוח זמנים ופעילויות
ז	2. תיאור המטלות
ז	2.1 מבנה המטלות
ט	2.2 חומר הלימוד הדרוש לפתרון המטלות
ט	2.3 ניקוד המטלות
יב	3. התנאים לקבלת נקודות זכות בקורס
1	מטלת אופל 01
3	ממ"ן 11
9	ממ"ן 12
19	ממ"ן 13
25	מטלת אופל 02
27	ממ"ן 14
33	ממ"ן 15
37	מטלת אופל 03

אל הסטודנט

אנו מקדמים את פניך בברכה עם הצטרפותך 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.
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"ם 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 Mærsk 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

בתוך "הרצאות מצולמות" 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

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!**
בשל מגפת הקורונה, אני לא נמצאת במשרד, ולכן אני לא מקיימת שעות ייעוץ טלפוניות בסמסטר זה.

לתשומת לב הסטודנטים הלומדים בחו"ל:

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.
תוכלו להצטרף לשידור החי ולצפות במפגשים המוקלטים של קבוצת הנחיה מוגברת אחת, לפי מה שמשויך לכם באתר הקורס.

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

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

אני מאחלת לך לימוד פורה ומהנה.

בברכה,

תמר וילנר

מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (2022א/ 20441)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	22.10.2021-17.10.2021	יחידות 1 ו-2 – הקדמה ויסודות השפה	מפגש 1	
2	29.10.2021-24.10.2021	יחידה 2 – יסודות השפה		מטלת אופל 01 30.10.2021
3	05.11.2021-31.10.2021	יחידה 3 – שימוש במחלקות נתונות	מפגש 2	
4	12.11.2021-07.11.2021	יחידה 4 – כתיבת מחלקות		ממ"ן 11 13.11.2021
5	19.11.2021-14.11.2021	יחידה 4 – כתיבת מחלקות	מפגש 3	
6	26.11.2021-21.11.2021	יחידה 5 – לולאות	מפגש 4	ממ"ן 12 27.11.2021
7	03.12.2021-28.11.2021 (ב-1 חנוכה)	יחידה 6 – מערכים		
8	10.12.2021-05.12.2021 (א-ב חנוכה)	יחידה 7 – ירושה	מפגש 5	ממ"ן 13 11.12.2021

* 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	17.12.2021-12.12.2021	יחידה 8 – פולימורפיזם		
10	24.12.2021-19.12.2021	יחידה 9 – יעילות	מפגש 6	מטלת אופל 02 25.12.2021
11	31.12.2021-26.12.2021	יחידות 9 ו-10 – יעילות ורקורסיה		
12	07.01.2022-02.01.2022	יחידה 10 – רקורסיה	מפגש 7	
13	14.01.2022-09.01.2022	יחידה 11 – רשימות	מפגש 8	ממ"ן 14 15.1.2022
14	21.01.2022-16.01.2022	יחידה 12 – מחסנית, תור ועצים בינריים	מפגש 9	ממ"ן 15 22.1.2022
15	28.01.2022-23.01.2022	חזרה לקראת הבחינה	מפגש 10 (3 שעות)	מטלת אופל 03 29.1.2022

מועדי בחינות הגמר יפורסמו בנפרד

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

2. תיאור המטלות

קרא היטב עמודים אלו לפני שתתחיל לענות על השאלות

בקורס זה **שמונה** מטלות שעליך לפתור ולהגיש במהלך הקורס. שלוש מטלות הן מטלות מחשב (ממ"ח), אותן עליך להגיש **במערכת שאלות"א**. הממ"ח נבדק בצורה ממוחשבת ואין לשלוח את פתרון הממ"ח למנחה. השאר הן מטלות מנחה (ממ"ן), אותן עליך להגיש באמצעות **מערכת המטלות המקוונת שנמצאת באתר הקורס**.

להלן תמצא הסבר על אופן הפתרון וכיצד לשלוח את המטלה למנחה. אם שאלה במטלה אינה ברורה לך, אל תהסס להתקשר אל מרכזת ההוראה או אל אחד המנחים (בשעות הייעוץ הטלפוני שלהם **בלבד**) לצורך קבלת הסבר.

2.1 מבנה המטלות

השאלות במטלות שבקורס הן משני סוגים: שאלות "יבשות" ושאלות "הרצה".

להלן הסבר מפורט על אופן הגשת הפתרונות לשאלות:

א. **שאלות "יבשות"** הן שאלות שאינן דורשות הרצת תכניות במחשב. הן נועדו לבדוק את הבנתך בחומר הלימוד.

אופן הגשת שאלה "יבשה":

בשאלה שבה הנך מתבקש לכתוב מה מבצעת תכנית מסוימת, יש לנסח באופן כללי **באמצעות משפט אחד או שניים** (בלבד) **מהי המשימה העיקרית אותה מבצעת התכנית לכל קלט אפשרי**. בשום אופן **אין** לפרט **איך** התכנית מתבצעת, כלומר אין להסביר את מהלך ביצועה. כמו כן, אין להריץ תכניות אלה במחשב אלא ל"הריץ" אותן "על יבשה", ידנית.
זכור! במבחן אין לצדך מחשב!

ב. **שאלת הרצה**: בשאלה זו 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.

אופן הגשת שאלת הרצה :

עליך לבדוק שהיא מבצעת את הנדרש ממנה ללא טעויות. תכנית שאינה רצה נכון לא תיבדק!

תיעוד

בכל תכנית הוסף תיעוד בתחילת התכנית המסביר את האלגוריתם בו השתמשת ואת מבנה התכנית. בגוף התכנית הוסף 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.

המטלות בקורס זה יוגשו בעזרת מערכת שליחת המטלות שבאתר הקורס.

שאלות "יבשות" ייכתבו בעזרת מעבד תמלילים Word.

שאלות הרצה יוגשו כקובצי Java.

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. שוב, חריגה תגרור הורדה בציון.

הסבר מפורט לגבי שליחת הקבצים המתאימים יינתן בכל מטלה בנפרד.

הסברים מפורטים על דרך שליחת המטלות בעזרת המערכת יישלחו במכתב נפרד.

2.2 חומר הלימוד הדרוש לפתרון המטלות

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

מטלה	חומר הלימוד הנדרש לפתרון
מטלת אופל 01	יחידות 1 - 2
ממ"ן 11	יחידות 1 - 2
ממ"ן 12	יחידות 3 - 4
ממ"ן 13	יחידות 5 - 6
מטלת אופל 02	יחידות 7 - 8
ממ"ן 14	יחידות 9 - 10
ממ"ן 15	יחידה 11
מטלת אופל 03	יחידה 12

שימו לב!

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

2.3 ניקוד המטלות

ניתן לצבור עד 25 נקודות. חובה להגיש מטלות במשקל מינימלי של 15 נקודות לפחות.

להלן פירוט הניקוד לכל מטלה:

ממ"ן/ממ"ח	ניקוד
01	1
11	3
12	4
13	5
02	1
14	5
15	5
03	1

כדי לעבור את הקורס, צריך (בין היתר) להגיש מטלות במשקל של 15 נקודות לפחות. סטודנט שמגיש את כל המטלות, משקל המטלות שלו הוא 25 נקודות. בהתאם למשקל המטלות, נקבע אחוז הציון של בחינת הגמר בציון הסופי.

חישוב אחוז הציון של הבחינה – אם הוגשו מטלות במשקל 15 נקודות, הבחינה שווה 85% מהציון הסופי של הקורס. אם המטלות שהוגשו הן במשקל 25, הבחינה שווה 75%. וכך גם לכל המספרים ביניהם. לדוגמא, אם הוגשו מטלות 01, 12, 13, 02, 14, 03 המשקל שלהן הוא:

$1 + 4 + 5 + 1 + 5 + 1 = 17$ (ללא קשר לציון שהתקבל בכל מטלה), ואז משקל הבחינה בציון הסופי הוא 83%.

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 (**עז 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
01	1	70
12	4	68
13	5	80
02	1	100
14	5	90
03	1	85
בחינת הגמר	83	73

שימו לב שבמטלות 01, 12 הציון נמוך מציון הבחינה. אם נוריד את שתיהן, יישארו 12 נקודות. אם נוריד את מטלה 12 יישארו 13 נקודות. לכן אפשר להוריד רק את מטלה 01 (כלומר להניח כאילו לא הוגשה כלל).

חישוב הציון ייעשה, אפוא, כך:

$$(68 \times 4 + 80 \times 5 + 100 \times 1 + 90 \times 5 + 85 \times 1 + 73 \times 84) / 100 = 74.39 \rightarrow 74$$

כלומר, הציון הסופי הוא 74.

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 שאלתא <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 (כשהמוצע המשוקלל של המטלות והבחינה יהיה נמוך מ- 60).

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 כמועתקת (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.
- 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 15 נקודות לפחות בהגשת המטלות.

3. התנאים לקבלת נקודות זכות בקורס

- א) צבירת משקל של 15 נקודות לפחות במטלות.
- ב) ציון של 60 נקודות לפחות בבחינת הגמר.
- ג) ציון סופי בקורס של 60 נקודות לפחות.

מטלת אופל 01

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 1-2 **נושא המטלה:** הקדמה ויסודות השפה

מספר השאלות: 20 **משקל המטלה:** 1 נקודות

סמסטר: 2022 **מועד אחרון להגשה:** 30.10.2021

(ת)

סטודנטים יקרים,
מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.
כל הפרטים הנדרשים נמצאים באתר הקורס.
הכניסה למטלה והגשתה אפשרית מתאריך 27.10 בשעה 10:00 עד לתאריך 30.10 בשעה 23:59.
אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.
לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.
אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה".

מטלת מנחה (ממ"ן) 11

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושאי המטלה: יסודות השפה

חומר הלימוד למטלה: יחידות 1-2

משקל המטלה: 3 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 13.11.2021

סמסטר: 2022א

(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 מאוד לעיין בו ולפעול לפיו. הקובץ נמצא באתר הקורס בתוך "יחידה 1" ב"מדריכי עזר וקישורים" 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 - להרצה (40%)

השימוש במילי-שניות נפוץ במחשבים לצורך מדידת זמן. מילי-שניה היא אלפית השנייה. במטלה זו תכתבו תכנית שממירה ממילי-שניות לפורמט קריא יותר לבני אדם. התכנית תקרא מהקלט שהכניס המשתמש מספר שלם שמייצג מילי-שניות, ותדפיס למסך את מספר הימים, שעות, דקות ושניות ששקולים למספר המילי-שניות שנקלט (יש לעגל כלפי מטה במידת הצורך), בפורמט הבא:

d_days_h:m:s_hours

כאשר במקום d, h, m, s יש להדפיס את מספר הימים, שעות דקות ושניות בהתאמה. (יש לקרוא משמאל לימין).
התו _ מציין רווח, ואין להדפיסו.

דוגמאות:

- עבור הקלט 100000 יודפס 0 days 0:1:40 hours
- עבור הקלט 97490000 יודפס 1 days 3:4:50 hours
- עבור הקלט 97490738 יודפס 1 days 3:4:50 hours

שימו לב – פורמט הפלט צריך להיות בדיוק כפי שהוגדר!

ניתן להניח כי המשתמש יכניס קלט חוקי, כלומר מספר שלם אי שלילי.
ניתן להניח שהקלט גדול מ- 1000 מילי-שניה.

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 "ייחידה 2" ב"מדריכי עזר וקישורים" בקובץ "מדריך לעבודה עם המחלקה Scanner לקבלת קלט מהמשתמש".

התכנית שכתבתם צריכה להיות במחלקה בשם **Milliseconds**.

לעזרתכם, כתבנו כאן חלקים מהמחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).
אל תשכחו להוסיף קבועים לפי הצורך.

להלן תחילת המחלקה **Milliseconds**.

```
import java.util.Scanner;
public class Milliseconds
{
    public static void main (String [] args)
    {
        // כאן צריכים להופיע קבועים מתאימים...

        Scanner scan = new Scanner (System.in);
        System.out.println ("This program reads an integer which " +
            "represents Milliseconds and converts it to days, " +
            "hours, minutes and seconds. ");
        System.out.println ("Please enter the number of Milliseconds");
        long ms = scan.nextLong();

        // כאן עליכם להמשיך...

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

שאלה 2 - להרצה (60%)

רקע:

נאמר על המתמטיקאי והפיסיקאי הידוע אלברט איינשטיין, שנהנה מאד לבלבל את חבריו בחידה המתוארת להלן. גם אתם תוכלו להשתעשע בכך:

קודם כל, כתבו את המספר 1089 על פיסת נייר, קפלו אותה, ותנו לחבר לשמור אותה. את מה שכתבתם אף אחד לא יקרא עד לסוף התהליך.

עכשיו, בקשו מהחבר לכתוב מספר שלם (חיובי) בן שלוש ספרות. הדגישו שהספרה הראשונה והספרה האחרונה חייבות להיות שונות זו מזו. (מספר בן שלוש ספרות ממש, ללא אפסים מובילים). עצמו את עיניכם וסובבו את גבכם בשעה שהחבר עושה זאת, כדי לוודא שאינכם רואים את המספר שכתב.

לאחר שהחבר כתב את המספר, בקשו ממנו להפוך אותו כך שהספרה הראשונה תהיה האחרונה ולהיפך. עתה, בקשו ממנו להחסיר את המספר הקטן מהגדול.

לדוגמא, אם המספר שבחר היה 773, המספר ההפוך הוא 377, וההפרש הוא $773 - 377 = 396$.

לאחר שעשה זאת, בקשו מהחבר להפוך את ההפרש (396 נהיה ל- 693).

לבסוף בקשו ממנו לסכם את ההפרש ואת ההפרש ההפוך. ($396 + 693 = 1089$)

אם עשיתם הכל כשורה, החבר יהיה המום – המספר שכתבתם בתחילה על הפתק, שנמצא מקופל בידיו מכיל את המספר 1089...

עכשיו ניגש למלאכת כתיבת תכנית שמבצעת זאת. התכנית תשחק את משחק איינשטיין כמפורט:

1. הדפיסו הודעה קצרה למשתמש על המשחק והסבירו את הכללים.
2. בקשו מהמשתמש שיכניס מספר בן 3 ספרות כמתואר.
3. קלטו את המספר- אם המספר אינו חיובי בן 3 ספרות, הדפיסו הודעת שגיאה (בפורמט הבא: "The number you entered is not a 3 digit positive number") וסיימו את התכנית. אחרת, המשיכו בתכנית והדפיסו את המספר שהמשתמש הכניס (בפורמט הבא: "User number is: 773").
4. הפכו את המספר והדפיסו את ההפרש בין המספר שהמשתמש הכניס והמספר ההפוך. דאגו שההפרש תמיד יהיה מספר חיובי. (בפורמט הבא: "Difference: 396")
ניתן להשתמש בשיטה `Math.abs(x)`, שהיא שיטה של Java שנמצאת במחלקה `Math` ומחזירה את הערך המוחלט של `x`. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא `Math.abs(x)` כאשר במקום הפרמטר `x` כותבים את הביטוי שרוצים לקבל את הערך המוחלט שלו. במידה והפרמטר `x` של השיטה הזו הינו מטיפוס שלם (`int`), השיטה תחזיר מספר שלם (`int`).
5. הדפיסו את ההפרש ההפוך. (בפורמט הבא: "Reversed difference: 693")
6. בדקו אם הסכום של ההפרש וההפרש ההפוך שווה למספר **הקבוע** 1089, אם כן, הדפיסו `SUCCEEDED`, אחרת הדפיסו `FAILED`.

לעזרתכם, כתבנו כאן פלט שהתקבל מהרצת התכנית עבור המספר 773

Welcome to the Einstein magic game.

Please enter a 3 digit positive number whose first and last digits are different:

773

User number is: 773

Difference: 396

Reversed difference: 693

SUCCEEDED

הערות:

- ניתן להניח שהמשתמש הכניס מספר שלם.
- אם המשתמש הכניס מספר תלת ספרתי שהספרה הראשונה והשנייה בו זהות, עליכם להדפיס הודעת שגיאה.
- כל הדפסה צריכה להיות בשורה נפרדת ומלווה בהודעה מה הודפס (למשל, ההפרש).
- אם ההפרש בין המספר והיפוכו הוא דו ספרתי (למשל, עבור הקלט 384, ההפרש הוא 99), זה בסדר. בתכנית זו ה"היפוך" של 99 הוא 990.
- גם הפעם השתמשו במחלקה Scanner כדי לקרוא מהקלט.

התכנית שכתבתם צריכה להיות במחלקה בשם Einstein.

בממ"ן זה אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Milliseconds ו-Main Einstein), והכל בשיטה main. אין להוסיף שיטות נוספות. במטלה זו 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 Milliseconds.java.
3. הפתרון לשאלה 2 כולל את הקובץ Einstein.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 וכתיבת מחלקות

מספר השאלות: 3

משקל המטלה: 4 נקודות

סמסטר: 2022א

מועד אחרון להגשה: 27.11.2021

(ת)

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

שאלה 1 - 20 נקודות

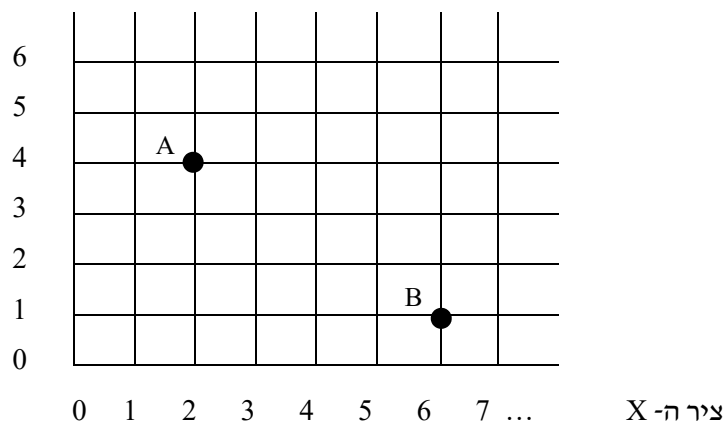
המחלקה Point מייצגת נקודה במישור, לפי מערכת הצירים הקרטזית (Cartesian system).

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

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

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

ציר ה-Y



למחלקה Point הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים המהווים את ערכי התכונות שיהיו לנקודה.
`public Point(int x, int y)`
- השני - בנאי העתקה המקבל נקודה אחרת, ומעתיק את ערכיה.
`public Point(Point other)`

בנוסף, הוגדרו במחלקה השיטות:

- שיטות האחזור:
 - `int getX()` המחזירה את ערכה של קואורדינטת ה-`x`.
 - `int getY()` המחזירה את ערכה של קואורדינטת ה-`y`.
- השיטות הקובעות:
 - `void setX (int num)` המשנה את ערכה של קואורדינטת ה-`x` להיות `num`.
 - `void setY (int num)` המשנה את ערכה של קואורדינטת ה-`y` להיות `num`.
- השיטה `toString()` שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המתמטי המקובל - `(x,y)`. כך, המחרוזת `(3,4)` מייצגת את הנקודה שקואורדינטת ה-`x` שלה היא 3 וקואורדינטת ה-`y` שלה היא 4. שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים וללא תווים נוספים.
- `boolean equals (Point other)` – שיטה שמקבלת נקודה נוספת בשם `other` כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה והנקודה `other` שהתקבלה כפרמטר זהות. כלומר, מחזירה `true` אם ערכי הנקודה עליה השיטה מופעלת שווים לערכי הנקודה `other`.
- `boolean isAbove (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מעל לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת מעל לנקודה B)
- `boolean isUnder (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מתחת לנקודה שהתקבלה כפרמטר. השיטה הזו משתמשת אך ורק בשיטה `isAbove` שהוגדרה לעיל ואין להשתמש בפעולות נוספות. אין לגשת לתכונות של הנקודות.
- `boolean isLeft (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת משמאל לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת משמאל לנקודה B)
- `boolean isRight (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מימין לנקודה שהתקבלה כפרמטר. השיטה הזו משתמשת אך ורק בשיטה `isLeft` שהוגדרה לעיל ואין להשתמש בפעולות נוספות. אין לגשת לתכונות של הנקודות.
- `void move (int deltaX, int deltaY)` – המזיזה את הנקודה ב-`deltaX` על ציר ה-`X` וב-`deltaY` על ציר ה-`Y`.
- `double distance (Point p)` – שיטה שמקבלת נקודה כפרמטר ומחזירה את המרחק בין הנקודה שעליה הופעלה השיטה והנקודה שהתקבלה כפרמטר.

תזכורת מתמטית:

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

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

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

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

עליכם לכתוב את המחלקה `Point` לפי ההגדרות לעיל.

שאלה 2 - 40 נקודות

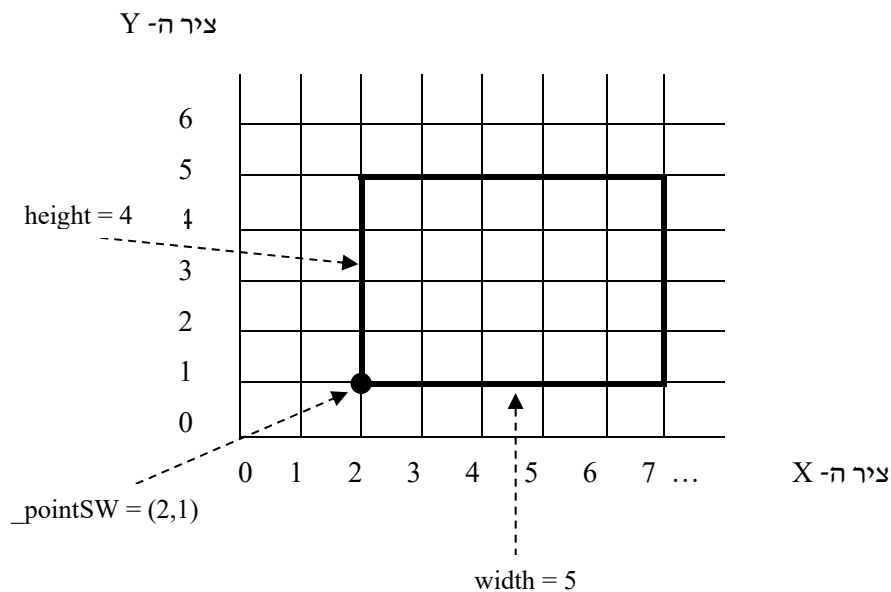
המחלקה `RectangleA` מייצגת מלבן שצלעותיו מקבילות לצירים.

למחלקה `RectangleA` התכונות הפרטיות (instance variables) הבאות:

- `int _width` – רוחב המלבן
- `int _height` – גובה המלבן
- `Point _pointSW` – מיקום הנקודה הדרום-מערבית של המלבן.

הנה דוגמא למלבן:

הנקודה הדרום-מערבית שלו היא במיקום $(2, 1)$, גובהו 4 ורוחבו 5.



למחלקה RectangleA הוגדרו ארבעה בנאים :

- האחד - המקבל שני מספרים שלמים כפרמטרים, הראשון הוא הרוחב של המלבן והשני הוא הגובה. הנקודה הדרום-מערבית תהיה בראשית הצירים (0, 0). האורך והרוחב חייבים להיות שלמים חיוביים ממש. אם מישו מהם אינו כזה, התכונה תאותחל להיות 1.

`public RectangleA(int w, int h)`

- השני - המקבל פרמטרים עם ערכים לתכונות המתאימות: הנקודה הדרום-מערבית תתקבל כפרמטר (נקודה, שאינה null), והרוחב והגובה יתקבלו כפרמטרים. האורך והרוחב חייבים להיות שלמים חיוביים ממש. אם מישו מהם אינו כזה, התכונה תאותחל להיות 1.

`public RectangleA(Point p, int w, int h)`

- השלישי - המקבל שתי נקודות כפרמטרים. הנקודה הראשונה היא הנקודה הדרום-מערבית (sw) והנקודה השנייה היא הנקודה הצפון-מזרחית (ne). אתם יכולים להניח שאכן הנקודה sw נמצאת **ממש** דרומית-מערבית לנקודה ne. בפרט, לשתי הנקודות לא יכולות להיות אותו ערך x או אותו ערך y. אין צורך לבדוק זאת. כמו כן, אפשר להניח ששתי הנקודות המועברות כפרמטרים אינן null.

`public RectangleA(Point sw, Point ne)`

- הרביעי - בנאי העתקה, המקבל אובייקט מהמחלקה RectangleA ומעתיק את ערכיו.

`public RectangleA(RectangleA r)`

כמו כן, הוגדרו שיטות האחזור (get) והשיטות הקובעות (set) לפי השמות המקובלים לכל התכונות של המחלקה, והשיטה toString שמחזירה את נתוני המלבן כמחרוזת תווים.

- `public int getWidth()`
- `public int getHeight()`
- `public Point getPointSW()`
- `public void setWidth(int w)`
- `public void setHeight(int h)`
- `public void setPointSW(Point p)`
- `public String toString()`

הערות:

- בשיטות `setWidth` ו-`setHeight`, אם הפרמטר אינו חיובי ממש, לא ייעשה כלום, והתכונה תישאר כשהיתה.
- בשיטה `toString` המחרוזת המייצגת את המלבן שבאיור לעיל, תיראה כך בדיוק (ללא רווחים נוספים וללא סימנים נוספים) –

Width=5 Height=4 PointSW=(2,1)

למחלקה `RectangleA` נוסף גם את השיטות הבאות:

- שיטה המחזירה את היקף המלבן
`public int getPerimeter()`
- שיטה המחזירה את שטח המלבן
`public int getArea()`
- שיטה המזיזה את המלבן למיקום אחר, השיטה מקבלת שני שלמים `deltaX` ו-`deltaY` המורים בכמה להזיז את המלבן בציר ה-`X` ובציר ה-`Y` בהתאמה.
`public void move(int deltaX, int deltaY)`
- שיטה בוליאנית שמקבלת מלבן כפרמטר ומחזירה האם המלבן שעליו הופעלה השיטה זהה למלבן שהתקבל כפרמטר (זהה מבחינת כל התכונות).
`public boolean equals(RectangleA other)`
- שיטה המחזירה את אורך האלכסון במלבן.
`public double getDiagonalLength()`
- שיטה בוליאנית שמקבלת מלבן כפרמטר ומחזירה האם שטחו של המלבן שעליו מופעלת השיטה גדול משטחו של המלבן שהתקבל כפרמטר.
`public boolean isLarger(RectangleA other)`
- שיטה שמחזירה את הנקודה הצפון-מזרחית של המלבן.
`public Point getPointNE()`
- שיטה המשנה את הצדדים של המלבן, כך שמה שהיה רוחב הופך להיות גובה, והגובה הופך להיות רוחב. הנקודה הדרום-מערבית אינה משתנה.

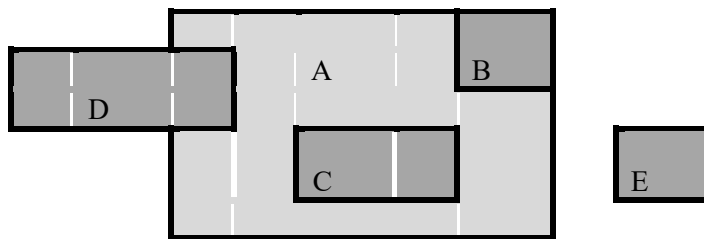
`public void changeSides()`

- שיטה בוליאנית המקבלת מלבן נוסף r ומחזירה `true` אם המלבן עליו מופעלת השיטה נמצא כולו בתוך המלבן r . (גם צלעות משותפות נחשבות "בתוך").

`public boolean isIn (RectangleA r)`

לדוגמא,

המלבן A המצויר להלן, מכיל את המלבנים B ו-C, אבל לא מכיל את המלבנים D ו-E.



- שיטה בוליאנית המקבלת מלבן נוסף r ומחזירה `true` אם יש חפיפה בין המלבנים ו-`false` אחרת.

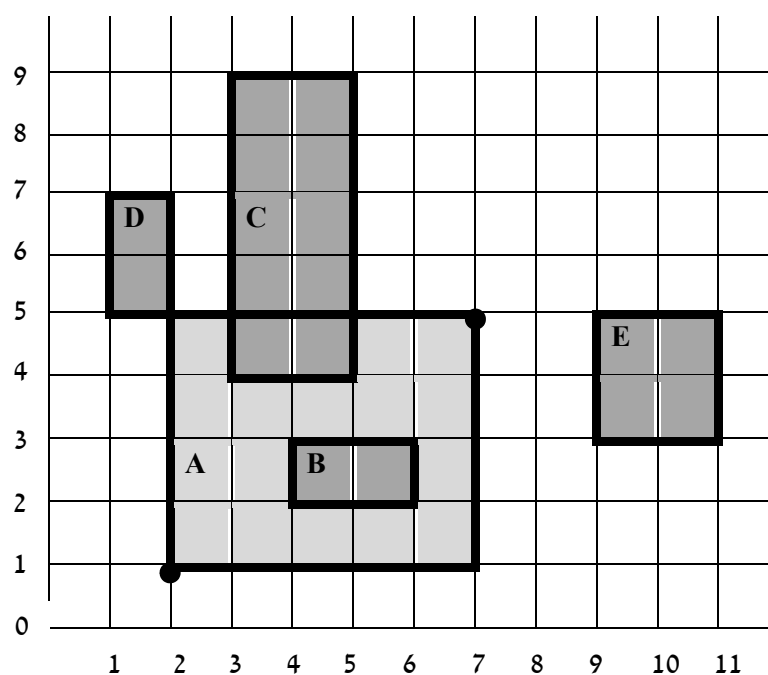
תזכורת –

נאמר שיש חפיפה בין שני מלבנים אם יש נקודות במישור שנמצאות בשני המלבנים.

`public boolean overlap (RectangleA r)`

לדוגמא, נסתכל על האיור שלהלן:

ציר ה-Y



המלבן A המצויר לעיל, הנקודה הדרום-מערבית שלו היא (2,1) והנקודה הצפון-מזרחית שלו היא (7,5). הנקודות מסומנות בגרף בעיגולים בולטים.

לכל המלבנים הבאים יש חפיפה עם המלבן A:

- B שהנקודה הדרום-מערבית שלו היא (4,2) והנקודה הצפון-מזרחית שלו היא (6,3)
- C שהנקודה הדרום-מערבית שלו היא (3,4) והנקודה הצפון-מזרחית שלו היא (5,9)
- D שהנקודה הדרום-מערבית שלו היא (1,5) והנקודה הצפון-מזרחית שלו היא (2,7)

שימו לב שהמלבן A ו-D חופפים רק בנקודה אחת שהיא (2,5) ואילו המלבן

- E שהנקודה הדרום-מערבית שלו היא (9,3) והנקודה הצפון-מזרחית שלו היא (11,5) אינו חופף באף נקודה למלבן A.

עליכם לכתוב את המחלקה RectangleA לפי ההגדרות לעיל (ולפי הכתוב ב-API שבאתר).

שימו לב לא לבצע aliasing במקומות המועדים.

שאלה 3 - 40 נקודות

המחלקה RectangleB מייצגת מלבן שצלעותיו מקבילות לצירים.

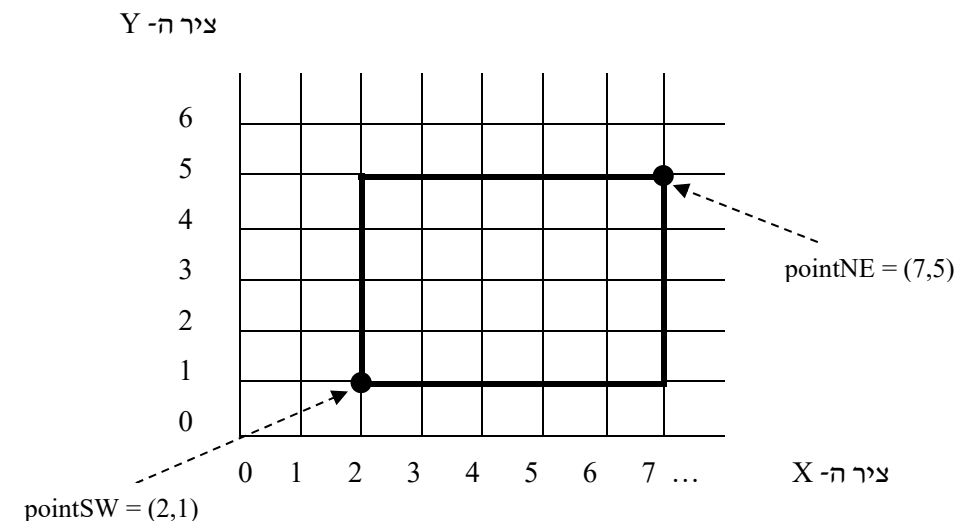
למחלקה RectangleB התכונות הפרטיות (instance variables) הבאות:

- Point _pointSW – מיקום הנקודה הדרום-מערבית של המלבן.
- Point _pointNE – מיקום הנקודה הצפון-מזרחית של המלבן.

הנה דוגמא למלבן שהוצג בשאלה 2:

הנקודה הדרום-מערבית שלו היא במיקום (2, 1),

הנקודה הצפון-מזרחית שלו היא במיקום (7, 5)



למחלקה RectangleB הוגדרו ארבעה בנאים :

- האחד - המקבל שני מספרים שלמים כפרמטרים, הראשון הוא הרוחב של המלבן והשני הוא הגובה. הנקודה הדרום-מערבית תהיה בראשית הצירים (0, 0). האורך והרוחב חייבים להיות שלמים חיוביים ממש. אם מישו מהם אינו כזה, התכונה תאותחל להיות 1.

```
public RectangleB(int w, int h)
```

- השני - המקבל פרמטרים עם ערכים לתכונות המתאימות: הנקודה הדרום-מערבית תתקבל כפרמטר (נקודה, שאינה null), והרוחב והגובה יתקבלו כפרמטרים. האורך והרוחב חייבים להיות שלמים חיוביים ממש. אם מישו מהם אינו כזה, התכונה תאותחל להיות 1.

```
public RectangleB(Point p, int w, int h)
```

- השלישי - המקבל שתי נקודות כפרמטרים. הנקודה הראשונה היא הנקודה הדרום-מערבית (sw) והנקודה השנייה היא הנקודה הצפון-מזרחית (ne). אתם יכולים להניח שאכן הנקודה sw נמצאת **ממש** דרומית-מערבית לנקודה ne. אין צורך לבדוק זאת. כמו כן, אפשר להניח ששתי הנקודות המועברות כפרמטרים אינן null.

```
public RectangleB(Point sw, Point ne)
```

- הרביעי - בנאי העתקה, המקבל אובייקט מהמחלקה RectangleB ומעתיק את ערכיו.

```
public RectangleB(RectangleB r)
```

השיטות של המחלקה RectangleB זהות לחלוטין לאלו של RectangleA.

לבד מהמקרים בהם מתקבל מלבן כפרמטר לשיטה, ואז במקום שכתוב RectangleA צריך להיות RectangleB.

במחלקה RectangleB אין להשתמש במחלקה RectangleA או בשיטות שנכתבו בה, ולהיפך! אלה שתי מחלקות נפרדות ובלתי תלויות לחלוטין!

שימו לב, גם השיטה toString של המחלקה RectangleB צריכה להיות לפי זו של המחלקה RectangleA, כלומר להדפיס את הנקודה הדרום-מערבית, את הרוחב והגובה.

הגדרות מדויקות לפי API תמצאו באתר הקורס בתת-ספריה "קובצי API".

הנחיות חשובות:

1. בכל השיטות במטלה שמקבלות אובייקט כפרמטר אפשר להניח שמתקבל אובייקט שאותחל ואינו שווה ל- `null`.
 2. עליכם להימנע מביצוע `aliasing` בשיטות ובבנאים.
 3. אם במהלך כתיבת השיטות המבוקשות אתם רוצים להשתמש בשיטות עזר נוספות, הן חייבות להיות `private`.
 4. כדי להימנע משכפול קוד במחלקות `RectangleA` ו- `RectangleB`, יש להשתמש לפי הצורך בשיטות הקיימות במחלקות `Point`.
 5. עליכם לתעד היטב את המחלקות `Point`, `RectangleA` ו- `RectangleB` שכתבתם בשיטת ה-API, כפי שהודגמה בהרצאות. כדי ליצור את קובץ ה- `html` שמכיל את ה-API, עליכם לעבור למצב של `documentation` או `interface` (תלוי איזו גרסה של `BlueJ` מותקנת אצלכם) בכפתור העליון בצד ימין (ללחוץ על החץ), במסך של המחלקה. כשתעברו למצב התיעוד ייווצר בו בזמן קובץ ובו תיעוד ה-API של המחלקה, בשם `RectangleA.html`, `RectangleB.html` ו- `Point.html`. הקובץ הזה נמצא בתוך התת-תיקיה `doc` שנמצאת בתוך התיקיה המכילה את הפרויקט שלכם.
- עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר. שימו לב שאתם לא צריכים לשלוח את קובצי ה- `html`, אבל בהחלט עליכם להכין אותם.
6. שימו לב ששמנו טסטרס לשלוש המחלקות באתר הקורס. חובה לוודא שטסטרס אילו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה לשיטה ובתוך גוף השיטה החזירו ערך סתמי כדי שהטסטרס ירוצו עם המחלקות ללא שגיאות קומפילציה. מי שיגיש מטלה שלא עוברת קומפילציה הציון במטלה שלו יהיה אפס!

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. זכרו כי 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.
לכן, הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת יורדו לכם הרבה נקודות!**
3. עליכם להריץ את הטסטרים שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרים לא מכסים את כל האפשרויות, ובפרט לא את מקרי הקצה. הם רק בודקים את השמות של השיטות במחלקות. מאד מומלץ להוסיף להם בדיקות.
4. עליכם להגיש את הקבצים Point.java, RectangleA.java ו-RectangleB.java **בלבד**.
הגשה של קבצים נוספים תגרע מהציון. באחריותכם להגיש את הקבצים הנכונים.
5. עטפו את שלושת הקבצים בקובץ zip יחיד ושלחו. **אין לשלוח קבצים נוספים.**

ב ה צ ל ח ה

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6 נושא המטלה: לולאות ומערכים

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2022 א מועד אחרון להגשה: 11.12.2021

במטלה זו נשחק קצת במשחק סודוקו Sudoku.
לשם כך נגדיר 2 מחלקות: Square3x3 ו-Sudoku

שאלה 1 (25%)

כתבו את המחלקה Square3x3 המייצגת מערך דו-ממדי ריבועי של מספרים שלמים בגודל 3×3 . למחלקה הממשק הבא (בקיצור):

הבנאים:

Constructor Summary	
<u>Square3x3()</u>	Constructor for objects of class Square3X3. Constructs and initializes a 2-dimensional array of the size 3X3, with the values of -1 in each cell.
<u>Square3x3(int[][] array)</u>	Constructs a 2-dimensional array of the size 3X3, whose values are taken from the given array. If the given array's size is bigger than 3X3, only the first 3X3 cells are taken. If the given array is smaller, the rest of the cells are initialized to -1. Note that the given array may be non-symmetrical, or may even have rows of different lengths (!!). Make sure to initialize cells to -1 if and only if the corresponding cell in the given array does not exist. You may assume the array is NOT null.
<u>Square3x3(Square3x3 other)</u>	Copy constructor. Constructs a 2-dimensional array of the size 3X3, whose values are taken from the array of the given Square3x3 object.

השיטות שעליכם לממש במחלקה Square3X3 הן:

Method Summary										
int	<u>getCell</u> (int row, int col) Returns the value in the (row, col) cell. If the row and/or col are out of the array bounds, returns -1. Legal values for row/col are 0,1,2.									
void	<u>setXY</u> (int row, int col, int value) Sets the cell (row, col) in the array to the given value. If the row and/or col are out of the array bounds – does nothing. Legal values for row/col are 0,1,2.									
String	<u>toString</u> () Returns a String representation of the array. For example: The square <table border="1"><tr><td>4</td><td>2</td><td>4</td></tr><tr><td>8</td><td>3</td><td>5</td></tr><tr><td>7</td><td>6</td><td>2</td></tr></table> will be represented like this: 4 2 4 8 3 5 7 6 2	4	2	4	8	3	5	7	6	2
4	2	4								
8	3	5								
7	6	2								

שימו לב! בשיטה toString יש להדפיס את הספרות בדיוק (!!) באופן המצוין. בכל שורה, יופיעו שלוש ספרות, כאשר כל שתי ספרות יופרדו באמצעות טאב ("t").

כלומר, בדוגמה הנתונה, שלוש השורות יודפסו באמצעות שלוש המחרוזות הבאות:

"4\t2\t4\n"

"8\t3\t5\n"

"7\t6\t2\n"

למען הסר ספק, אין להוסיף סימני פיסוק נוספים!

למחלקה זו יש תכונה פרטית אחת בלבד – מערך דו מימדי של שלמים בגודל 3x3.

שאלה 2 (25%)

הוסיפו למחלקה Square3x3 את השיטות הבאות, וממשו אותן :

- שיטה בוליאנית בשם allThere. שיטה זו תבדוק האם במערך של המחלקה נמצאים כל המספרים מ-1 עד 9. היא תחזיר true אם כן, ו-false אחרת. חתימת השיטה היא :

```
public boolean allThere()
```

- שיטה בשם whosThereRow וחתימתה היא :

```
public void whosThereRow (int row, boolean[] values )
```

אופן פעולת השיטה: השיטה מקבלת מספר שורה (מבין הערכים 0,1,2) ומערך בוליאני בגודל 10. עבור כל מספר בין 1-9 המופיע בשורה המתאימה במערך הדו-ממדי של המחלקה השיטה תציב ערך true בתא המתאים במערך values. שאר התאים במערך ישארו ללא שינוי. לדוגמה, אם באחד התאים בשורה המתאימה הופיע המספר 7, השיטה תציב true בתא בעל אינדקס 7 במערך. שימו לב שהתא בעל האינדקס 0 במערך values אינו בשימוש.

- שיטה בשם whosThereCol וחתימתה היא :

```
public void whosThereCol (int col, boolean[] values )
```

השיטה עובדת כמו השיטה הקודמת, רק שהיא עושה את הבדיקה עבור עמודה נתונה, ולא עבור שורה.

שאלה 3 (20%)

כתבו מחלקה בשם Sudoku.

מחלקה זו תייצג לוח של המשחק סודוקו כמערך דו-ממדי ריבועי בגודל 3×3 של אובייקטים מסוג Square3x3. שימו לב, הכוונה היא לא לבנות מערך של int מגודל 9×9 , אלא מערך בגודל של 3×3 שכל אחד מאיבריו הוא אובייקט של המחלקה Square3x3 ! **מערך דו-ממדי זה הוא התכונה היחידה של המחלקה.**

למחלקה יהיה 2 בנאים : בנאי ריק שיאתחל את כל הלוח ל -1, ובנאי שיקבל כפרמטר מערך 3×3 של אובייקטים של המחלקה Square3x3 וייצר על פיו את הלוח.

חתימות הבנאים :

```
public Sudoku()
```

```
public Sudoku(Square3x3[][] square3x3Array)
```

שאלה 4 (30%)

ממשו את השיטה isValid של המחלקה Sudoku. שיטה זו תבדוק האם לוח הסודוקו הוא חוקי. אם כן – תחזיר true. אם הלוח לא חוקי, אין צורך להדפיס למה הוא לא חוקי – אלא להחזיר false בלבד.

חתימת השיטה:

```
public Boolean isValid()
```

לצורך מימוש השיטה יש להשתמש בשיטות שכתבתם בשאלה 3. מומלץ גם לכתוב שיטות עזר פרטיות שישמשו אתכם.

תזכורת – לוח סודוקו הוא חוקי אם בכל שורה מופיעים כל המספרים מ-1 עד 9, וגם בכל עמודה מופיעים כל המספרים מ-1 עד 9, וגם בכל אחד מתשעת הריבועים המרכיבים את הלוח מופיעים כל המספרים מ-1 עד 9. לדוגמא, הלוח הבא אינו חוקי:

1	9	8	9	2	3	4	5	7
7	3	2	6	5	4	1	8	9
5	6	4	8	1	7	2	3	6
3	1	9	2	8	5	7	6	4
8	4	7	1	3	6	5	9	2
2	5	6	4	7	9	8	1	3
4	8	5	3	6	2	9	7	1
6	2	1	7	9	8	3	4	5
9	7	3	5	4	1	6	2	8

כיוון שבשורה הראשונה שמסומנת המספר 9 מופיע פעמיים, ובשורה השנייה המסומנת, המספר 6 מופיע פעמיים. השיטה תחזיר שהלוח אינו חוקי כיוון שיש בשורה הראשונה מספר שמופיע פעמיים (אין צורך "לגלות" את כל הטעויות בלוח, מספיק למצוא את הראשונה ולהדפיס).

אם נחליף בין התא השני בשורה הראשונה (9) עם התא השני בשורה השלישית (6) נקבל לוח חוקי (אתם מוזמנים לבדוק).

שימו לב לא לבצע aliasing במקומות המועדים.

מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.
אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקות שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקות שכתבתם בטסטר זה, ולהגיש אותן רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב API - את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקבצים Square3x3.java, Sudoku.java עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

בהצלחה

מטלת אופל 02

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 7 – 8

נושא המטלה: ירושה ופולימורפיזם

מספר השאלות: 20

משקל המטלה: 1 נקודות

סמסטר: 2022א

מועד אחרון להגשה: 25.12.2021

(ת)

סטודנטים יקרים,

מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.

כל הפרטים הנדרשים נמצאים באתר הקורס.

הכניסה למטלה והגשתה אפשרית מתאריך 22.12 בשעה 10:00 עד לתאריך 25.12 בשעה 23:59.

אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.

לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.

אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה."

מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

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

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2022 מועד אחרון להגשה: 15.1.2022

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. אנו ממליצים מאד, בשלב הראשון, לענות עליהן ללא הרצה במחשב (כפי שמקובל בבחינת הגמר).

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex14.java (בדיוק).
את התשובות לשאלות על הסיבוכיות כתבו כחלק מה-API.

שאלה 1- 25 נקודות

נתון מערך חד-ממדי a המלא במספרים שלמים.

כתבו שיטה סטטית, המקבלת כפרמטר מערך כזה, ושני מספרים שלמים x ו-y. השיטה צריכה להחזיר את המרחק המינימלי מבחינת תאים בין שני המספרים האלו במערך.
אם אחד מהמספרים (או שניהם) לא נמצא במערך, יוחזר הערך Integer.MAX_VALUE (המספר השלם המקסימלי האפשרי).

חתימת השיטה היא:

```
public static int findMinDiff (int[] a, int x, int y)
```

לדוגמא,

אם המערך a הוא זה:

0	1	2	3	4	5	6	7	8	9
1	3	5	4	8	2	4	3	6	5

• אם המספרים x ו-y הם: $x = 3, y = 2$

המספר 3 נמצא במערך באינדקסים 1 ו-7

המספר 2 נמצא במערך באינדקס 5

ולכן המרחק המינימלי בין 3 ל-2 במערך הוא 2 (בין האינדקס 5 לאינדקס 7).

השיטה תחזיר 2

- אם המספרים x ו- y הם: $x = 2, y = 5$
השיטה תחזיר 3 (המרחק בין האינדקס 2 לאינדקס 5)
- אם המספרים x ו- y הם: $x = 2, y = 7$
השיטה תחזיר Integer.MAX_VALUE (כי המספר 7 לא נמצא במערך)

שימו לב:

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.

אל תשכחו לתעד את מה שכתבתם!

שאלה 2- 25 נקודות

בשאלה זו נתייחס למערכים דו-ממדיים ריבועיים, כלומר, מספר השורות והעמודות שווה (נניח שהוא שווה ל- n).

לצורך השאלה נניח כי n הוא חזקה שלמה של 2.

עבור מערך כזה, נגדיר חלוקה פנימית שלו לארבעה רובעים בגודל $n/2 \times n/2$, ממוספרים מ-1 עד 4 באופן הבא:

1	2
4	3

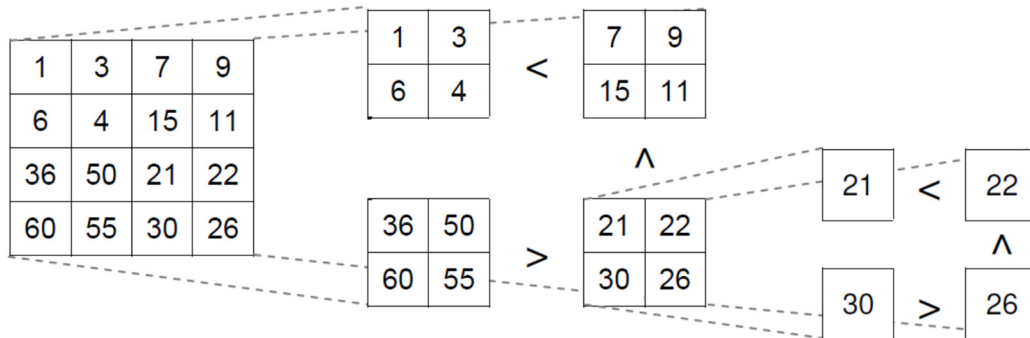
נאמר שהמערך הוא **סיבובי** (circular) אם כל האיברים ברובע 1 קטנים ממש מכל אלו שברובע 2, אלו שברובע 2 קטנים ממש מכל אלו שברובע 3, ואלו שברובע 3 קטנים ממש מכל אלו שברובע 4. למשל, המערך הבא הוא סיבובי:

1	5
9	7

לשם הנוחות, נגדיר גם כל מערך בגודל 1×1 כמערך סיבובי.

נאמר שמערך בגודל $n \times n$ הוא **ממוין-סיבובי** (circular sorted) אם הוא סיבובי, ארבעת הרובעים שלו סיבוביים, וכן הלאה עד לרובעים בגודל 1×1 .

לדוגמא, המערך הבא ממוין-סיבובית :



עליכם לכתוב שיטה סטטית בוליאנית לחיפוש במערך ממוין-סיבובית. השיטה מקבלת כפרמטרים מערך דו-ממדי `mat` שהוא ממוין-סיבובית, וערך לחיפוש `num`. אם הערך `num` נמצא במערך `mat`, השיטה תחזיר `true` ותדפיס את מספר השורה ומספר העמודה שבהם נמצא המספר `num`. אם הערך `num` לא נמצא במערך `mat`, השיטה תחזיר `false` ולא יודפס כלום.

חתימת השיטה היא:

```
public static boolean search (int [][] mat, int num)
```

לדוגמא,

אם המערך `mat` הוא המערך המצויר לעיל, והמספר `num` הוא 22, השיטה תחזיר את הערך `true`, ויודפסו השורות הבאות:

```
row = 2
```

```
col = 3
```

אם המערך `mat` הוא המערך המצויר לעיל, והמספר `num` הוא 23, השיטה תחזיר את הערך `false`, ולא יודפס כלום.

אתם יכולים להניח שהמערך `mat` אינו `null` והוא ממוין-סיבובית. אינכם צריכים לבדוק זאת.

שימו לב:

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. אל תשכחו לתעד את מה שכתבתם!

שאלה 3- 25 נקודות

כתבו שיטה סטטית רקורסיבית בוליאנית המקבלת מערך של מספרים שלמים `arr`. השיטה צריכה להחזיר `true` אם אפשר לחלק את איברי המערך לשתי קבוצות שונות **שוות בגודלן** (כלומר מספר האיברים בהם זהה) כך **שסכום האיברים בשתי הקבוצות שווה**. אם אי אפשר לעשות זאת, השיטה תחזיר `false`.

חתימת השיטה היא:

```
public static boolean equalSplit (int[] arr)
```

דוגמאות:

- עבור המערך הבא:

0	1	2	3	4	5
-3	5	12	14	-9	13

השיטה תחזיר `true` כי $-3 + 5 + 14 = 12 + -9 + 13$ וגם בכל קבוצה יש 3 איברים

- עבור המערך הבא:

0	1	2	3	4	5
-3	5	-12	14	-9	13

השיטה תחזיר `false` כי אמנם $-3 + 5 + 14 + -12 = -9 + 13$ אבל מספר האיברים בשתי הקבוצות שונה (באחת יש ארבעה איברים ובשנייה רק שניים).

- עבור המערך הבא:

0	1	2	3	4
-3	5	-12	14	-9

השיטה תחזיר `false` כי אין חלוקה של איברי המערך לשתי קבוצות שוות בגודלן.

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

אפשר להשתמש בהעמסת-יתר (overloading). אפשר להניח שהמערך אינו `null` ואינו ריק.

אין לשנות את תוכן המערך (אפילו לא זמנית), ולא להשתמש במערך עזר.

אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!

אל תשכחו לתעד את מה שכתבתם!

שאלה 4- 25 נקודות

נגדיר **מספרים מיוחדים** (Special numbers) על-ידי ההסבר והדוגמא הבאה :

נקח את קבוצת המספרים השלמים החיוביים :

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20...

עכשיו נמחק כל מספר שני בקבוצה, ונישאר עם הקבוצה הבאה :

1, ~~2~~, 3, ~~4~~, 5, ~~6~~, 7, ~~8~~, 9, ~~10~~, 11, ~~12~~, 13, ~~14~~, 15, ~~16~~, 17, ~~18~~, 19, ~~20~~...

כלומר הקבוצה שנשארה היא :

1, 3, 5, 7, 9, 11, 13, 15, 17, 19...

עכשיו נמחק כל מספר שלישי בקבוצה שנשארה, ונישאר עם הקבוצה הבאה :

1, 3, ~~5~~, 7, 9, ~~11~~, 13, 15, ~~17~~, 19...

כלומר הקבוצה שנשארה היא :

1, 3, 7, 9, 13, 15, 19...

וכך נמשיך בתהליך ללא סוף.

כל מספר שלא נמחק במהלך התהליך נקרא "מספר מיוחד".

לכן, תחילתה של הקבוצה של המספרים המיוחדים היא :

1, 3, 7, 13...

כתבו שיטה סטטית בוליאנית רקורסיבית, המקבלת כפרמטר מספר שלם n ומחזירה true אם המספר הוא מספר מיוחד ו- false אחרת.

חתימת השיטה היא :

```
public static boolean isSpecial (int n)
```

רמז אפשרי לפתרון –

שימו לב שאם נסתכל על מספר כלשהו, לאחר כל "סיבוב" של מחיקות, המיקום שלו בסדרה הוא אחר. הסתכלו, למשל, על המספר 7. בהתחלה המיקום שלו הוא כמובן, 7. לאחר מחיקת כל מספר שני, הוא הופך להיות המספר הרביעי בסדרה. לאחר מחיקת כל מספר שלישי מהסדרה, הוא הופך להיות המספר השלישי בסדרה. גם לאחר מחיקת כל מספר רביעי בסדרה (וכך הלאה), הוא יישאר המספר השלישי בסדרה. והוא אכן מספר מיוחד.

חשבו איך לחשב את **המיקום** הבא של המספר, כשאתם יודעים מה המיקום הנוכחי, ואיזה מספר הוא זה שיימחק (כל שני, כל שלישי, כל רביעי וכד').

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

אפשר להשתמש בהעמסת יתר (overloading).

שימו לב:

**בשאלות 3 ו-4 אין צורך לדאוג ליעילות השיטה שתכתבו!
אסור להשתמש במשתנים גלובליים שהוגדרו מחוץ לשיטות עצמן!
בכל השאלות - אל תשכחו לתעד את מה שכתבתם!**

**שימו לב ששמנו טסטר באתר הקורס. חובה שהטסטר ירוץ עם המחלקה שכתבתם
ללא שגיאות קומפילציה. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך
סתמי כדי שהטסטר ירוצו עם המחלקה ללא שגיאות קומפילציה.
אם בהרצת הטסטר יתגלו 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 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 Ex14.java (**בדיוק**). ארוזו אותו בתוך קובץ zip. אין לשלוח קבצים נוספים.

בהצלחה

מטלת מנחה (ממ"ן) 15

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: רשימה מקושרת

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 5 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 22.1.2022

סמסטר: 2022א

(ת)

כזכור, במטלה 12 הגדרנו מחלקה Point המייצגת נקודה במישור (על ידי קואורדינטות x ו-y) ומחלקה RectangleA המייצגת מלבן המקביל לצירים (על ידי הנקודה הדרום-מערבית של המלבן וכן הרוחב והגובה של המלבן).

במטלה זו נשתמש בשתי מחלקות אלו, וניצור רשימה מקושרת של מלבנים.

כדי לעשות זאת, עליכם להגדיר שתי מחלקות RectNode ו-RectList.

חובה עליכם להשתמש במחלקות של Point ו-RectangleA ששמנו באתר במטלה 15. שימו לב שהקבצים שיש באתר הם קובצי class ולא הקוד ב-Java. אי אפשר לפתוח אותם אלא להשתמש בהם בלבד. לצידם יש קובצי html שהם ה-API של המחלקות Point ו-RectangleA.

שאלה 1 - להרצה (10%)

המחלקה RectNode תייצג צומת ברשימה מקושרת שיש בו מלבן.

לכל אובייקט במחלקה יש שני שדות:

1. RectangleA _rect // המלבן
2. RectNode _next // מצביע לאיבר הבא

למחלקה זו עליכם להגדיר שלושה בנאים:

1. public RectNode (RectangleA r)
בנאי המקבל מלבן, שדה ה-next יאותחל ל-null.
2. public RectNode (RectangleA r, RectNode n)
בנאי המקבל מלבן ואיבר נוסף מטיפוס RectNode, ומאתחל את התכונות לפי הפרמטרים.
3. public RectNode (RectNode r)
בנאי העתקה.

השיטות במחלקה RectNode הן:

- public RectangleA getRect() - שיטה המחזירה עותק של המלבן שבאיבר.
- public RectNode getNext() - שיטה המחזירה מצביע לאיבר הבא.

- `public void setRect(RectangleA r)` - שיטה המקבלת מלבן ומעדכנת את תכונת המלבן שבאיבר.
- `public void setNext(RectNode next)` - שיטה המקבלת מצביע ומעדכנת את תכונת המצביע לאיבר הבא.

שימו לב: הבנאים והשיטות במחלקה זו מקבלים ומחזירים אובייקטים (מצביעים). לפיכך יש לפעול בזהירות ולקבוע מתי `aliasing` הוא נכון ומתי הוא טעות. כאשר עליכם להעתיק או להחזיר אובייקט מסוג מלבן יש להיזהר מ-`aliasing`. כאשר נרצה להחזיר או להעתיק מצביע לאיבר הבא ברשימה `aliasing` אינו טעות.

שאלה 2 - להרצה (90%)

המחלקה `RectList` תייצג רשימה מקושרת של מלבנים. הייצוג נעשה על-ידי רשימה ששומרת את רשימת המלבנים.

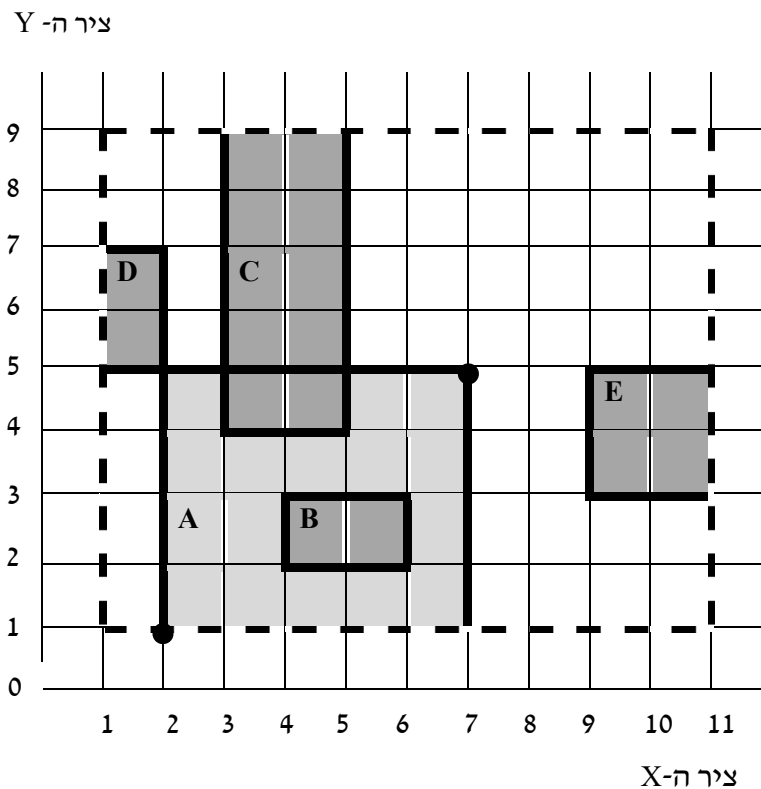
במחלקה זו מותר להגדיר אך ורק תכונה פרטית אחת, ראש הרשימה `_head`, שיצביע לתחילת הרשימה לאובייקט מסוג `RectNode`. אין להוסיף תכונות מעבר לתכונה זו.

עליכם לממש ב-Java את המחלקה `RectList` לפי הסעיפים להלן:

1. הגדרת התכונה של המחלקה.
2. בנאי שיוצר רשימה ריקה - מאתחל את ראש הרשימה להיות `null`.
3. שיטה `addRect` המקבלת כפרמטר מלבן ומוסיפה אותו **לסוף הרשימה**. אם המלבן הזה כבר נמצא ברשימה, השיטה לא תעשה כלום.
4. שיטה `howManyWithPoint` המקבלת כפרמטר נקודה `p` (אובייקט מהמחלקה `Point`) ומחזירה את מספר המלבנים ברשימה שהנקודה הדרום-מערבית שלהם היא `p`. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) או שאין מלבנים ברשימה שהנקודה `p` היא הנקודה הדרום-מערבית שלהם, השיטה תחזיר 0.
5. שיטה `longestDiagonal` המחזירה את אורך האלכסון הארוך ביותר בין כל המלבנים. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר 0.
6. שיטה `mostLeftRect` המחזירה העתק של הנקודה הדרום-מערבית של המלבן הכי שמאלי ברשימה. מלבן אחד יותר שמאלי ממלבן אחר אם הנקודה הדרום-מערבית שלו נמצאת משמאל לנקודה הדרום-מערבית של המלבן האחר. אם יש יותר ממלבן אחד שהם השמאליים ביותר, השיטה מחזירה את המלבן הראשון מביניהם בו נתקלה. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר `null`.
7. שיטה `highestRect` המחזירה העתק של הנקודה הצפון-מזרחית של המלבן הכי גבוה ברשימה. מלבן אחד יותר גבוה ממלבן אחר אם הנקודה הצפון-מזרחית שלו גבוהה

יותר מזו של האחר. אם יש יותר ממלבן אחד בגובה הגבוה ביותר, השיטה מחזירה את המלבן הראשון מביניהם בו נתקלה. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר null.

8. שיטה `minimalContainer` המחזירה **מלבן** חדש (אובייקט מהמחלקה `RectangleA`) שהוא המלבן בעל השטח המינימלי המכיל את כל המלבנים שברשימה. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר null.
לדוגמא, נסתכל על האיור שלהלן:



המלבנים המצוירים באיור הם:

- A שהנקודה הדרום-מערבית שלו היא (2,1) והנקודה הצפון-מזרחית שלו היא (7,5)
- B שהנקודה הדרום-מערבית שלו היא (4,2) והנקודה הצפון-מזרחית שלו היא (6,3)
- C שהנקודה הדרום-מערבית שלו היא (3,4) והנקודה הצפון-מזרחית שלו היא (5,9)
- D שהנקודה הדרום-מערבית שלו היא (1,5) והנקודה הצפון-מזרחית שלו היא (2,7)
- E שהנקודה הדרום-מערבית שלו היא (9,3) והנקודה הצפון-מזרחית שלו היא (11,5)

המלבן הקטן ביותר שמכיל את כל חמשת המלבנים הוא זה שהנקודה הדרום-מערבית שלו היא (1,1) והנקודה הצפון-מזרחית שלו היא (11,9). הגבולות החיצוניים שלו מסומנים בקו מקווקו באיור.

9. השיטה toString המחזירה מחרוזת תווים המייצגת את רשימת המלבנים. המחרוזת צריכה להיות בפורמט הבא: שימו לב שאין רווחים במחרוזת שמייצגת את הנקודה.

The list has 3 rectangles.

1. Width=5 Height=4 PointSW=(2,1)
2. Width=4 Height=7 PointSW=(2,1)
3. Width=5 Height=4 PointSW=(3,1)

אם אין מלבנים ברשימה, השיטה תחזיר מחרוזת בפורמט הבא:

The list has 0 rectangles.

שימו לב לא לבצע aliasing במקומות המועדים (מלבד המקומות בהם במפורש נדרש אחרת). מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.

אתם צריכים כמובן לכתוב API לשתי המחלקות.
שימו לב לכל מקרי השגיאה האפשריים!

כתבו כהערה ב-API מה סיבוכיות הזמן וסיבוכיות המקום של כל שיטה שכתבתם.
הקפידו על יעילות השיטות שכתבתם!

שימו לב:

1. אסור להשתמש במחלקות מוכנות כבר של Java.
2. מותר ורצוי להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים, לדוגמא Math.sqrt.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות והמחלקות יהיו בדיוק לפי הוראות הממ"ן.
3. את התשובות לשאלות יש להגיש בשני קובצי Java הבאים: RectNode.java, ו- RectList.java ארוזים יחד בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.

מטלת אופל 03

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 12 **נושא המטלה:** מחסנית, תור, עצים בינריים

מספר השאלות: 20 **משקל המטלה:** 1 נקודות

סמסטר: 2022 **מועד אחרון להגשה:** 29.1.2022

(ת)

סטודנטים יקרים,
מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.
כל הפרטים הנדרשים נמצאים באתר הקורס.
הכניסה למטלה והגשתה אפשרית מתאריך 26.1 בשעה 10:00 עד לתאריך 29.1 בשעה 23:59.
אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.
לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.
אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה".