

20441
מבוא למדעי המחשב
ושפת Java
חוברת הקורס – סתיו 2021א

כתבה: תמר וילנר

אוקטובר 2020 – סמסטר סתיו – תשפ"א

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
יא	3. התנאים לקבלת נקודות זכות בקורס
1	ממ"ח 01
10	ממ"ן 11
17	ממ"ן 12
27	ממ"ן 13
32	ממ"ח 02
45	ממ"ן 14
50	ממ"ן 15
54	ממ"ח 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. חלק מחומר הלימוד נמצא גם בספר הלימוד Java Software Solutions מאת Lewis & Loftus. בחלק מהיחידות הסרטים הם בנוסף למה שכתוב בספר, ובחלק מהיחידות הסרטים הם חומר הלימוד בקורס, כיוון שבספר לא מרחיבים על חלק מהנושאים. ספר זה מכסה את הנושאים הנלמדים בקורס, אך אין חובה לקוראו. הספר ניתן כעזרה נוספת וכמקור נוסף אליו ניתן לפנות בהגדרות השפה. חוברות השקפים שקיבלתם הן אלו המלוות את סרטי הוידאו שבאתר.

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

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 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 (2021א/ 20441)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה בקבוצות רגילות*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	23.10.2020-18.10.2020	יחידות 1 ו-2 – הקדמה ויסודות השפה בספר – פרקים 1 ו-2	מפגש 1	
2	30.10.2020-25.10.2020	יחידה 2 – יסודות השפה בספר – פרקים 2 ו-5		ממ"ן 01 31.10.2020
3	06.11.2020-01.11.2020	יחידה 3 – שימוש במחלקות נתונות בספר – פרק 3	מפגש 2	
4	13.11.2020-08.11.2020	יחידה 4 – כתיבת מחלקות בספר – פרק 4		ממ"ן 11 14.11.2020
5	20.11.2020-15.11.2020	יחידה 4 – כתיבת מחלקות בספר – פרק 4	מפגש 3	
6	27.11.2020-22.11.2020	יחידה 5 – לולאות בספר – פרקים 5 ו-6	מפגש 4	ממ"ן 12 28.11.2020
7	04.12.2020-29.11.2020	יחידה 6 – מערכים בספר – פרק 8		
8	11.12.2020-06.12.2020 (ו חנוכה)	יחידה 7 – ירושה בספר – פרק 9	מפגש 5	ממ"ן 13 12.12.2020

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

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה בקבוצות רגילות*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	18.12.2020-13.12.2020 (א-ו חנוכה)	יחידה 8 – פולימורפיזם בספר – פרק 10		
10	25.12.2020-20.12.2020	יחידה 9 – יעילות בספר – פרק 10	מפגש 6	ממ"ן 02 26.12.2020
11	01.01.2021-27.12.2020	יחידות 9 ו-10 – יעילות ורקורסיה בספר – פרק 12		
12	08.01.2021-03.01.2021	יחידה 10 – רקורסיה בספר – פרק 12	מפגש 7	
13	15.01.2021-10.01.2021	יחידה 11 – רשימות בספר – פרק 13	מפגש 8	ממ"ן 14 16.1.2021
14	22.01.2021-17.01.2021	יחידה 12 – מחסנית, תור ועצים בינריים בספר – פרק 12	מפגש 9	ממ"ן 15 23.1.2021
15	29.01.2021-24.01.2021 לאחר תום הסמסטר	חזרה לקראת הבחינה	מפגש 10 (3 שעות)	ממ"ן 03 30.1.2021

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
ממ"ח 01	יחידות 1 - 2
ממ"ן 11	יחידות 1 - 2
ממ"ן 12	יחידות 3 - 4
ממ"ן 13	יחידות 5 - 6
ממ"ח 02	יחידות 7 - 8
ממ"ן 14	יחידות 9 - 10
ממ"ן 15	יחידה 11
ממ"ח 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:

$17 = 1 + 5 + 1 + 5 + 4 + 1$ (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 שתייהן, 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.

כיוון שחובה להגיש מטלות במשקל של לפחות 15 נקודות, אם מישהו לא הגיש את המשקל הדרוש, הוא יכול לבקש שזין לו ציון 0 במטלה אחת או שתיים (**ולא יותר**). כלומר, המשקל של המטלה ייחשב, אבל הציון יושפע מהאפס, ותצטרכו להוציא ציון גבוה יותר בבחינה כדי לעבור את הקורס. בקשות כאלו עליכם להפנות אל מרכזת ההוראה בדואר אלקטרוני לכתובת tami@openu.ac.il. בבקשה עליכם לכתוב את שמכם המלא ואת מספר תעודת הזהות שלכם, ולציין באיזה מטלות תרצו לקבל ציון 0 (שוב, **שתי מטלות לכל היותר**).

הכנת המטלות חייבת להיעשות על-ידי כל סטודנט בנפרד.

מטלות שלא יבוצעו באופן עצמאי – ייפסלו!!!

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

מועדי הגשת המטלות

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

דחייה בהגשת מטלות

- אנחנו נאשר איחורים רק כאשר יש בקשה מראש לאיחור כזה. בקשה מנומקת לאיחור של עד שבוע יש להפנות למנחה הקבוצה. במקרים חריגים ביותר של דחייה של למעלה משבוע, תצטרכו לפנות למרכזת ההוראה לקבלת אישור כזה. שוב, האישור צריך להתקבל מלכתחילה ולא בדיעבד.
- 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

- צבירת משקל של 15 נקודות לפחות במטלות.
- ציון של 60 נקודות לפחות בבחינת הגמר.
- ציון סופי בקורס של 60 נקודות לפחות.

מטלת מחשב (ממ"ח) 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: 2021 מועד אחרון להגשה: 31.10.2020

(ת)

את התשובות לממ"ח יש לשלוח באמצעות מערכת שאילת"א

בכתובת <http://www.openu.ac.il/sheilta/>

הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח למנחה!

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

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת הממ"ח. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את הממ"ח.

מטלות מחשב – ממ"ח

- הממ"ח הוא "מבחן רב-ברירה" (מבחן אמריקאי) הנבדק באמצעות מחשב.
- יש להקפיד לשלוח את התשובות לממ"ח במועד שנקבע. אל תקדימו במשלוח התשובות יותר משבוע לפני התאריך הנקוב בלוח הזמנים לאותו ממ"ח.
- בתוך שלושה שבועות מהתאריך האחרון, המצוין בלוח הזמנים, תקבלו לבתיכם הודעה שתכלול:
- התשובות הנכונות לממ"ח לעומת תשובותיכם.
 - הערות (אם תהיינה כאלה) המתייחסות לתשובותיכם.
 - ציונכם בממ"ח ומשקלו של ממ"ח זה בחישוב הציון הסופי בקורס.

הנחיות לפתרון הממ"ח

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

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."

משלוח הממ"ח

יש לשלוח את התשובות לממ"ח באמצעות מערכת **שאלתא** (שירותים אינטראקטיביים לסטודנטים באמצעות תקשורת ואינטרנט). הסבר על המערכת ניתן למצוא בחוברת הקורס וכן באתר האו"פ באינטרנט בכתובת: www.openu.ac.il/sheilta במערכת ניתן לראות את תוצאות בדיקת הממ"ח מיד עם פרסומן.

הוראות למילוי תשובות ומשלוח ממ"ח באמצעות מערכת שאלתא

1. היכנס למערכת שאלתא. (הכניסה היא מאתר הבית של האו"פ בכתובת www.openu.ac.il/sheilta באמצעות שם המשתמש והסיסמה שנשלחה אליך).
2. היכנס לתפריט קורסים.
3. בדף הקורסים, בחר בפירוט הקורס המבוקש.
4. בפירוט הקורס, היכנס לקישור מטלת מחשב.
5. בחר בממ"ח שברצונך לשלוח ע"י הקלקה על הכפתור שמימין לממ"ח ולחץ על הזנת תשובות.
6. הזן את התשובות לכל השאלות. (לבחירת התשובה לחץ על החץ שבכל תיבה).
7. שלח את תשובותיך על-ידי לחיצה על לחצן שלח.
8. בתפריט פניות תוכל לראות את פרטי הממ"ח ששלחת.

ערעור על ציון בממ"ח

ערעור על ציון שקיבלת בממ"ח יוגש למרכז ההישגים הלימודיים תוך שבוע מיום קבלת תוצאות הממ"ח, ובצירוף ההודעה על הציון שקיבלת מהמחשב (או צילומה). אין ערעור נוסף על ההחלטה בערעור זה.

שאלה 1. מהו אלגוריתם?

- א. הרצת התוכנית עם אוסף נתוני בדיקה לצורך איתור שגיאות.
 - ב. המרת הקוד שכתב המתכנת לקוד מכונה שהמחשב יכול לבצע.
 - ג. רצף של פעולות הנועד לפתור בעיה חישובית.
 - ד. חלוקת אובייקטים לקבוצות על-מנת לשייכם למחלקות משותפות.
- ה. מונח זה מתאר את תהליך פיתוח התוכנה במלואו, משלב כתיבת הקוד ועד הרצת התוכנית המלאה.

שאלה 2.

אילו מהמשפטים הבאים איננו נכון בשפת java?

- א. מילים שמורות בשפה אינן יכולות לשמש כמזהים (identifiers).
- ב. הקומפילר מתרגם את התכנית ל-bytecode שאינו תלוי בפלטפורמה.
- ג. השיטה הראשית main יכולה להופיע אך ורק במחלקה ששמה Main.
- ד. תכנית ב-java אינה חייבת לכלול פקודת פלט.
- ה. משתנים שונים יכולים לשמור ערכים זהים.

שאלה 3.

מהי שגיאת ריצה?

- א. שגיאה שהתקבלה בגלל מילה בתכנית שכתובה בצורה לא נכונה
- ב. שגיאה שהתקבלה בגלל שהריצו את התכנית במחשב איטי מדי
- ג. שגיאת תחביר של השפה שגרמה לכך שאי אפשר להריץ את התכנית
- ד. שגיאה שקרתה במהלך ריצת התכנית

השאלות 5 - 4 מתייחסות לשיטה main שלהלן:

```
public static void main(String [ ] args)
{
    System.out.print("Here");
    System.out.println("There " + "Everywhere");
    System.out.println("But not" + "in Texas");
}
```

שאלה 4.

התכנית תדפיס את המילה "Here" ואז תדפיס:

- א. "There Everywhere" בשורה שאחרי "Here"
- ב. "There" בשורה שאחרי "Here" ואז "Everywhere" בשורה שאחרי "There"
- ג. "ThereEverywhere" באותה שורה של "Here"
- ד. "ThereEverywhere" בשורה שאחרי "Here"
- ה. "There Everywhere" באותה שורה של "Here"

שאלה 5. כמה שורות יהיו בפלט התכנית בסך-הכל:

- א. 1
- ב. 2
- ג. 3
- ד. 4
- ה. 5

שאלה 6. טקסט הכתוב כהערה בתכנית

- א. יכול להיות כל דבר שהמתכנת רוצה לכתוב.
- ב. נבדק על ידי המהדר (compiler) לשם הדיוק
- ג. חייב להופיע בשורה הראשונה של התכנית
- ד. מודפס כאשר התכנית רצה

שאלה 7. נניח ש-x ו-y הם משתנים מטיפוס int. איזו שורה מבין השורות הבאות מכילה הערה תקנית (גם אם לא מדויקת)?

- א. `int num = x + y; /* num holds the product of x and y */`
- ב. `int num = x + y; / / * num holds the sum of x and y */`
- ג. `int num = x + y; /** num holds the sum of x and y [comment] //`
- ד. `int num = x + y; /* num holds the product of x and y */`

שאלה 8. איזה מבין המזהים (identifiers) הבאים אינו חוקי בשפת Java?

- א. 25a
- ב. _25a
- ג. a25
- ד. _a25
- ה. a25_

שאלה 9. ברצוננו להצהיר על משתנה בשם t אשר יכיל תו בודד ויאוחל לאות t. כיצד ניתן לעשות זאת?

- א. `char t;`
- ב. `char t = t;`
- ג. `char t = 't'`
- ד. `char c = 't'`
- ה. `String t = (char) "t"`
- ו. לא ניתן לבצע זאת.

שאלה 10. מה מהבאים הוא תו (char) שמייצג את הספרה 0?

- א. "0"
- ב. 0
- ג. String "zero"
- ד. char 0
- ה. '0'
- ו. (char) 0

שאלה 11. בתכנית מוגדרים המשתנים int a = 3 וגם int b = 5. מה מבין קטעי הקוד הבאים ידפיס על המסך את הפלט: The sum: 8

- א. System.out.print ("The sum: ", a+b);
- ב. System.out.print ("The sum: " + a+b);
- ג. System.out.print ("The sum: " + (a+b));
- ד. System.out.print ("The sum: " + "a+b");
- ה. System.out.print ("The sum: ", 'a'+'b');

שאלה 12. בהינתן המשתנה int a, ברצוננו לבדוק האם הערך השמור ב-a הוא שלם חיובי חד-ספרתי. אילו מבין התנאים הבאים יבדוק זאת?

- א. if (a<10)
- ב. if (a>0 && a<10)
- ג. if (a>=0 || a<=9)
- ד. if (a>-1 && a<10)
- ה. if (a.length()==1)
- ו. if (a.length==1)

שאלה 13. בהינתן המשתנים int x = 0, int a = 1 ו- int b = -1. מה יהיה ערכו של המשתנה x לאחר ביצוע קטע הקוד שלהלן?

```
if (a > 0)
    if (b > 0)
        x = x + 5;
    else
        if (a > 5)
            x = x + 4;
        else
            x = x + 3;
else
    x = x + 2;
```

א. 0

ב. 2

ג. 3

ד. 4

ה. 5

שאלה 14. אם נתונה הפקודה שלהלן בתכנית:

```
System.out.println ( /* 4 + 7 */ - 5 + 9 /* 8 */ / 2 ); /* + 4 */ - 10;
```

מה יהיה הפלט שיודפס בעקבות הרצתה?

א. 3

ב. -1

ג. 5

ד. -8

ה. -2

שאלה 15. נתון קטע הקוד הבא:

```
int a = 10;
```

```
System.out.println(a++);
```

מה הערך שיודפס ומה הערך של המשתנה a בסופו של הקטע?

א. הערך שיודפס יהיה 10 והערך של a יהיה 11

ב. הערך שיודפס יהיה 11 והערך של a יהיה 11

ג. הערך שיודפס יהיה 11 והערך של a יהיה 10

ד. הערך שיודפס יהיה 10 והערך של a יהיה 10

ה. אי אפשר לדעת מה הערך שיודפס והערך של a יהיה 11

ו. אי אפשר לדעת מה הערך של a והערך שיודפס יהיה 11

שאלה 16. נתון משתנה שלם num מוצהר ומאותחל. להלן קטע קוד (מספרי השורות אינם חלק מהקוד, כמובן).

```
1. int a=num%100/10; System.out.println(a);
2. int b=(num- num/100*100) /10; System.out.println(b);
3. int c= num/10%10; System.out.println(c);
4. int d=num%1000/100; System.out.println(d);
```

- א. עבור כל ערך של המשתנה num יודפס אותו הערך בכל אחת מהשורות 1-4
- ב. יתכן ערך התחלתי של המשתנה num שעבורו יודפס אותו הערך בכל אחת מהפקודות 1-4
- ג. עבור כל ערך של המשתנה num יודפס אותו הערך בכל אחת מהשורות 1-3
- ד. אי אפשר לדעת אם עבור כל ערך של num יודפס אותו הערך בכל אחת מהפקודות 1-4
- ה. אף אחת מהתשובות אינה נכונה
- ו. יש יותר מתשובה אחת א-ד שהיא נכונה

שאלה 17. בהנחה שנתונים שני משתנים שלמים f ו-s, שניהם מוצהרים ומאותחלים. אנחנו מעוניינים להדפיס הודעה הכוללת את המחרוזת "first is ", לאחריה הערך של f, אחר כך המחרוזת " second = " ואז את ערכו של s. הכל צריך להיות מודפס בשורה אחת ועם ריווח מתאים, ואז לעבור לשורה הבאה. איזה מבין קטעי הקוד הבאים יבצע זאת?

- א. `System.out.print("first is " + f + " second = " + s);`
- ב. `System.out.println("first is" + f + "second=" + s)`
- ג. `System.out.println(first is + " f " + second = + " s ");`
- ד. `System.out.print("first is " + f + " second = " + s + "\n");`

שאלה 18. בהנחה שנתונים שלושה משתנים שלמים num1, num2, num3, שלושתם מוצהרים ומאותחלים, וכן נתון משתנה ממשי avg, ומוצהר avg, אנחנו מעוניינים לחשב את הממוצע הממשי של שלושת המשתנים השלמים ולהציב אותו לתוך avg. איזה מבין המשפטים הבאים לא יבצע זאת?

- א. `avg = (double) ((num1+num2+num3)/3)`
- ב. `avg = (num1+num2+num3)/3.0`
- ג. `avg = (double)(num1+num2+num3)/3`
- ד. `avg = (num1+num2+num3)/(double)3`
- ה. `avg = ((double)num1+num2+num3)/3`

שאלה 19. נניח ש-a ו-b הם משתנים מטיפוס int וערכיהם: a = 1, b = 2, ו-x ו-y הם משתנים מטיפוס char שערכיהם x = 'x', y = 'A'. התייחסו לארבעת התנאים הבאים, וענו אילו מהטענות להלן נכונה?

- Condition 1: `(a < b && a > 0)`
- Condition 2: `(x != y || a != 1)`
- Condition 3: `!(true && false)`
- Condition 4: `(a > b || x == 'A' || y != 'A')`

- א. כל ארבעת התנאים הם true.
- ב. רק condition2 הוא true.
- ג. רק condition2 ו-condition4 הם true.
- ד. condition2, condition3 ו-condition4 הם true ו-condition1 הוא false.
- ה. כל ארבעת התנאים הם false.
- ו. אף אחת מהטענות א-ה לעיל אינה נכונה.

שאלה 20. בהנחה שנתונים שלושה משתנים שלמים a, b, c, שלושתם מוצהרים ומאותחלים, אנחנו מעוניינים לבדוק האם קיימים ביניהם שניים (או שלושה) ערכים זהים. קטע הקוד הבא אמור לבצע זאת. (מספרי השורות אינם חלק מהקוד, כמובן)

```

1. if (a==b)
2.     if (b==c)
3.         System.out.println ("All three are the same.");
4.     else
5.         System.out.println ("a and b are the same.");
6. else
7.     if (b==c)
8.         System.out.println ("b and c are the same.");
9.     else
10.        if (a==c)
11.            System.out.println ("a and c are the same.");
12.        else
13.            System.out.println ("3 different values.");

```

- מה מהמשפטים הבאים נכון?
- א. קטע הקוד לא יעבור קומפילציה בכלל.
- ב. קטע הקוד תקין.
- ג. קטע הקוד שגוי. השינוי היחיד צריך להיות בשורה 2, צריך לכתוב
if (a==c && b==c)
- ד. קטע הקוד שגוי. השינוי היחיד צריך להיות בשורה 7, צריך לכתוב
if (a!=c && b==c)
- ה. קטע הקוד שגוי ונדרשים מספר שינויים על מנת לתקנו.
- ו. אף אחת מהתשובות לעיל אינן נכונות.

השאלות הבאות חשובות לנו מאד בפיתוח התכנים של הקורס, אך לא ישפיעו בשום אופן על הציון שלכם, לא בממ"ח ולא בקורס בכלל.

שאלה 21. התרשמותי מהממ"ח הנוכחי :

- א. קל מאד לפתרון
- ב. רמה סבירה
- ג. קשה לפתרון
- ד. קשה מאד לפתרון

שאלה 22. יכולתי לענות על הממ"ח בהצלחה :

- א. רק אחרי שקראתי את הפרקים המתאימים בספר Java Software Solutions
- ב. רק אחרי שצפיתי בהרצאות המוקלטות של ד"ר אמיר גורן באתר (יחידות 1 – 2)
- ג. רק אחרי שצפיתי במפגשי ההנחיה המוקלטים (אופק/זום)
- ד. רק אחרי שקראתי בספר וגם צפיתי בהרצאות
- ה. יש לי ידע מוקדם ולא הייתי צריך לקרוא בספר או לצפות בהרצאות המוקלטות כדי לענות על הממ"ח.
- ו. אין לי ידע מוקדם ובכל זאת לא הייתי צריך לקרוא בספר או לצפות בהרצאות המוקלטות כדי לענות על הממ"ח.

את התשובות לממ"ח יש לשלוח באמצעות מערכת שאילת"א

בכתובת <http://www.openu.ac.il/sheilta/>

הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון

הממ"ח למנחה!

שימו לב שהממ"ח אכן נקלט בשאילתא,

ושמרו את האסמכתא שקיבלתם כתוצאה

מהשליחה.

אם לא קיבלתם אישור – כאילו לא שלחתם!

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: 14.11.2020

סמסטר: 2021א

(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" והוא נקרא guideline.pdf
- 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%)

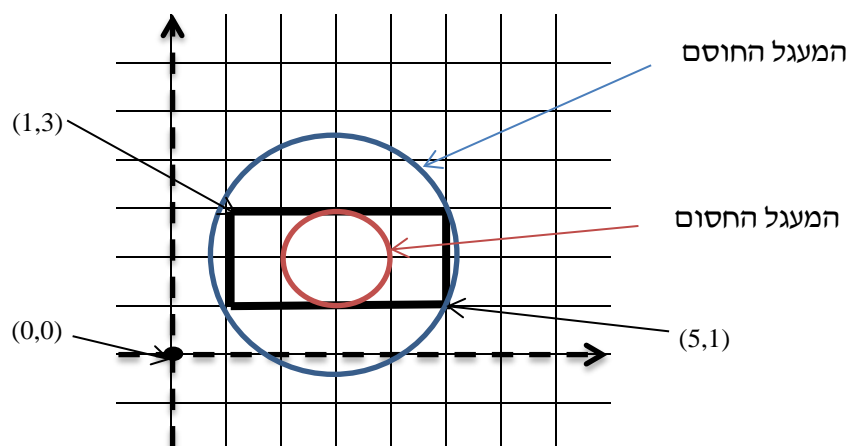
כתבו תכנית הקוראת מהמשתמש ארבעה מספרים שלמים (לתוך משתנים מטיפוס int). שני המספרים הראשונים יצינו קואורדינטת x ו- קואורדינטת y של הפינה השמאלית העליונה של מלבן (במישור); שני המספרים האחרונים יצינו קואורדינטת x ו- קואורדינטת y של הפינה הימנית התחתונה של המלבן (במישור). ניתן להניח כי רוחב המלבן גדול מגובהו (כלומר הפער בין ערכי x גדול מהפער בין ערכי y). אנו מתייחסים למלבן שצלעותיו מקבילות לצירים. **בהנחה שהנקודה השנייה נמצאת מימין ומתחת לנקודה הראשונה.**

על תכניתכם לחשב ולהציג את רדיוס (radius), שטח (area) והיקף (perimeter, circumference) המעגל החסום (incircle, inscribed circle) במלבן, ואת רדיוס, שטח והיקף המעגל החוסם (excircle, circumscribed circle) את המלבן.

לדוגמה: אם המשתמש מעוניין לתאר את המלבן שפינתו השמאלית העליונה מצויה בנקודה (1, 3) ופינתו הימנית התחתונה מצויה בנקודה (5, 1) אזי הוא יזין את הקלט:

1 3

5 1



הפלט עבור קלט זה יהיה **בדיוק** כזה שלהלן: (לגבי מספר הספרות שאחרי הנקודה העשרונית – זה לא משנה. יכולות להיות יותר ספרות או פחות. אל תתייחסו לזה!)

Incircle: radius = 1, area = 3.1415, perimeter = 6.283

Excircle: radius = 2.23607, area = 15.7075, perimeter = 14.0492

שימו לב שאין צורך להסתבך בחישובים מיותרים. אינכם צריכים לצייר את המעגלים, ולא לחשב את נקודת המרכז, וכד', אלא רק לחשב את הקוטר (ומכאן את הרדיוס), ואז לחשב את השטח וההיקף של המעגלים.

לשם קריאה מהקלט השתמשו במחלקה Scanner.

import java.util.Scanner;

את השורה

אפשר למצוא את הממשק של המחלקה Scanner בספר, שם מובאות חלק מהשיטות.

הסברים על המחלקה והשימוש בה אפשר למצוא באתר הקורס בספרית ה"קבצים להורדה", בתת-ספריה "מדריכי עזר".

בשאלה זו אתם צריכים לכתוב את כל התכנית בשיטה אחת - main.

אל תשכחו להוסיף קבועים לפי הצורך.

התכנית שתכתבו צריכה להיות במחלקה בשם Circle.

את הקלט תקראו כארבעה מספרים שלמים (int) עם רווחים ביניהם, ללא פסיקים או סימנים אחרים. בתחילה תבקשו מהמשתמש שיכניס שני מספרים המייצגים את הנקודה השמאלית העליונה ואז תקראו אותם, ואחר כך תבקשו מהמשתמש שיכניס את שני המספרים שמייצגים את הנקודה הימנית התחתונה, ותקראו אותם.

שימו לב, בשאלה זו אתם יכולים להניח שהקלט שניתן הוא תקין בכל מובן שהוא. המשתמש יקליד ארבעה מספרים שלמים, ואין צורך לבדוק אם אכן הקואורדינטות מתאימות.

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

```
import java.util.Scanner;

public class Circle
{
    public static void main (String [] args)
    {
        Scanner scan = new Scanner (System.in);
        System.out.println ("This program calculates the areas " +
            "and the perimeters of the excircle and the incircle " +
            "of a given rectangle ");
        System.out.print ("Please enter the two coordinates of the " +
            "left-upper point of the rectangle");
        int leftUpX = scan.nextInt();
        int leftUpY = scan.nextInt();

        // כאן עליכם להמשיך ...
    } // end of method main
} //end of class Circle
```

תזכורת מתמטית –

- הנוסחה לחישוב היקף של עיגול היא $2 \times \pi \times r$ כאשר r הוא הרדיוס של המעגל.
- הנוסחה לחישוב שטח של עיגול היא $\pi \times r^2$ כאשר r הוא הרדיוס של המעגל.
- כדי לחשב מרחק בין שתי נקודות - (x_1, y_1) , (x_2, y_2) - השתמשו במשפט פיתגורס:

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

- 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).
- כדי לחשב מספר a בחזקת b , ניתן להשתמש בשיטה `Math.pow(a, b)`.
- כדי להשתמש בערך הקבוע פִּי (π) `Math.PI`, ניתן להשתמש בקבוע `Math.PI`.

אפשר למצוא את הממשק של המחלקה `Math` בספר `Java Software Solutions`, שם מובאות חלק מהשיטות.

שאלה 2 - להרצה (60%)

מדידת טמפרטורות נעשית ביחידות מידה שונות.

יחידות המידה הנפוצות הן צלזיוס (Celsius), פרנהייט (Fahrenheit) וקלווין (Kelvin).

לפניכם טבלת המרות חלקית בין הסולמות (השלימו בעצמכם את השאר):

מ... / אל...	צלזיוס (C)	פרנהייט (F)	קלווין (K)
צלזיוס (C)			$K = C + 273.15$
פרנהייט (F)	$C = \frac{5}{9}(F - 32)$		
קלווין (K)		$F = \frac{9}{5}(K - 273.15) + 32$	

עליכם לכתוב תכנית להמרה בין הסולמות השונים.

קלט:

אחת משלוש אותיות (C, F, K) וטמפרטורה (מספר ממשי) בסולם מתאים.

פלט:

שלוש הטמפרטורות בסולמות השונים (לפי הסדר: צלזיוס, פרנהייט וקלווין) בשורות נפרדות, כשיש רווח אחד בין הטמפרטורה לבין סימון הסולם. ראו את הדוגמאות להלן.

דוגמאות:

- אם הקלט הוא: F 100
הפלט יהיה:

37.78 C
100.0 F
310.93 K

- אם הקלט הוא: C 100
הפלט יהיה:

100.0 C
212 F
373.15 K

יש להגדיר את כל המספרים הקבועים במשתני final. אין להשתמש במספרים עצמם בחישובים.

גם כאן, לשם קריאה מהקלט השתמשו במחלקה Scanner.

הפעם עליכם לקרוא מהקלט תו בודד (char) ואחריו מספר ממשי (כשביניהם יש רווח)

כדי לקרוא תו בודד מהקלט עליכם להשתמש בשיטה next מהמחלקה Scanner. השיטה הזו קוראת מילה (עד הרווח), ואז להתייחס לאות הראשונה במילה על ידי השיטה charAt מהמחלקה String.

כך תבצעו את הקריאה של תו בודד :

בהנחה שהגדרנו את האובייקט scan מהמחלקה Scanner

```
String word = scan.next();
```

```
char c = word.charAt(0);
```

ואז, אם הקלט היה F 100.0 (משמאל לימין), אחרי שתי הפקודות האלו במשתנה c יהיה התו 'F'.

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

בממ"ן זה אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Circle ו- Temperature בהתאמה), והכל בשיטה main. אין להוסיף שיטות נוספות.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לשאלה 1 כולל את הקובץ Circle.java.
3. הפתרון לשאלה 2 כולל את הקובץ Temperature.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.

ב ה צ ל ח ה

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: 2021א

מועד אחרון להגשה: 28.11.2020

(ת)

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים. תתבקשו לממש מחלקות שונות המייצגות נקודה ומקטע במישור. כדי לעמוד על ההבדל בין המימוש לממשק של מחלקה, תתבקשו לכתוב שני מימושים שונים למחלקה המייצגת מקטע.

שאלה 1 - 20 נקודות

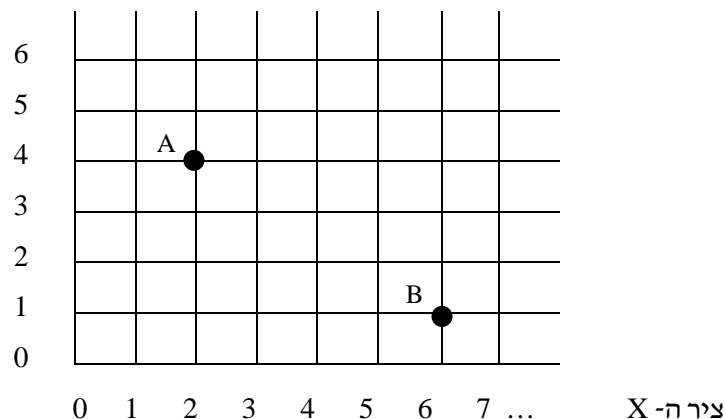
בהרצאות הקורס של ד"ר אמיר גורן, הוגדרה המחלקה Point שמייצגת נקודה במישור, לפי מערכת הצירים הקרטזית (Cartesian system) -

המחלקה Point שהוגדרה בהרצאות הכילה את התכונות הפרטיות (instance variables) הבאות:

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

לדוגמא, הנה מסומנות שתי הנקודות $A = (2.0, 4.0)$ ו- $B = (6.0, 1.0)$ במרחב:

ציר ה-`Y`



בשאלה זו עליכם לכתוב מחדש את המחלקה Point. הפעם המימוש שלה יהיה לפי המערכת הפולרית (Polar system).

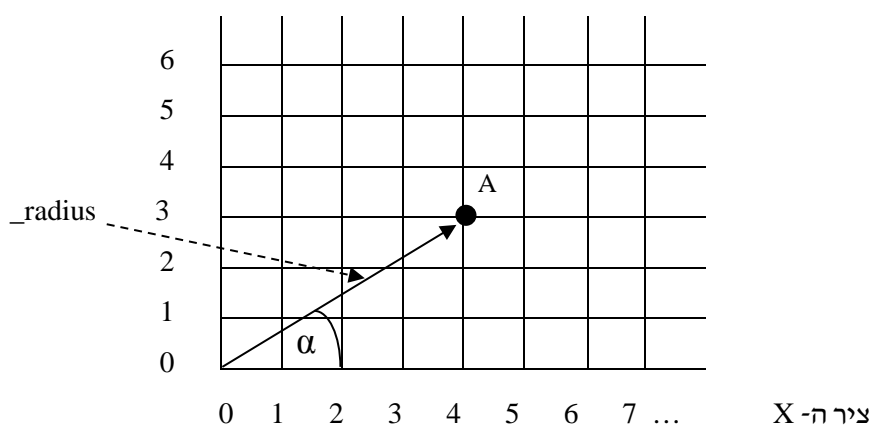
המחלקה Point תייצג נקודה במישור ברביע הראשון בלבד.

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

- `double _radius` – שמייצגת את אורך הוקטור מראשית הצירים עד הנקודה;
- `double _alpha` – שמייצגת את הזווית במעלות של הוקטור עם ציר ה- x .

לדוגמא, הנה מסומנת הנקודה A (שנמצאת בקואורדינטות (4.0, 3.0) במרחב: כאן אורך הוקטור שמחבר את הנקודה (0.0, 0.0) עם (4.0, 3.0) הוא 5.0 וערכה של הזווית α (alpha) הוא 36.87 מעלות (וברדיאנים - 0.64)

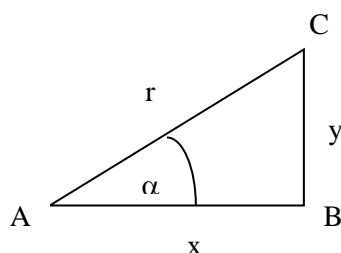
ציר ה-Y



תזכורת מתמטית קצרה

לדוגמא, נתון המשולש ישר הזווית הבא:

קדקודי המשולש הם A, B ו-C. צלעות המשולש הן x , y ו- r . (ראו תרשים להלן).



$$\sin \alpha = y/r$$

$$\cos \alpha = x/r$$

$$\tan \alpha = y/x$$

- מכאן, כאשר נתונים אורכי הצלעות x ו- y , אפשר לחשב את הזווית α על-ידי שימוש

בנוסחה ההפוכה \arctan . כלומר, $\alpha = \arctan\left(\frac{y}{x}\right)$ ערך הזווית בראדיאנים.

אם ערך ה- x הוא אפס אזי הזווית תהיה 90 מעלות.

(אם ערך ה- x הוא אפס וגם ערך ה- y הוא אפס אזי הזווית תהיה 90 מעלות.)

- כאשר נתונים אורכי הצלעות x ו- y , אפשר לחשב את אורך הצלע r (היתר במשולש ישר-זווית) על-ידי שימוש במשפט פיתגורס $r = \sqrt{x^2 + y^2}$

כזכור, 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).

במחלקה `Math` תוכלו למצוא גם את השיטות `atan`, `tan`, `cos`, `sin` (הוא `arctan`), וגם את הקבוע `Math.PI` שהוא Π .

אפשר למצוא את ה-API של המחלקה `Math` בכתובת

<http://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

שימו לב:

המחלקה `Math` מתייחסת לזוויות ברדיאנים (`radians`) ולא במעלות (`degrees`). לכן, עליכם לעשות את השינויים המתאימים, ללא שימוש בשיטות `toRadians` או `toDegrees` של המחלקה.

להזכירכם:

$$\Pi = 3.14159... = \text{Math.PI} = 180^\circ$$

$$\text{Math.PI} / 2 = 90^\circ \text{ וכן הלאה}...$$

שוב, כיון שאנחנו מתייחסים במטלה זו רק לנקודות ברביע הראשון של מערכת הצירים, הזוויות האפשריות הן רק מ- 0° עד 90° , כלומר מ- 0 עד $\text{PI}/2$.

עליכם לכתוב את המחלקה `Point` (לפי המערכת הפולרית) לפי התיאור הבא:

שימו לב – ההתייחסות לנקודה היא לפי הפרמטרים של המערכת הקרטזית, כלומר הקואורדינטה בציר ה- x והקואורדינטה בציר ה- y , אולם המימוש הפנימי הוא לפי המערכת הפולרית.

לכן כל השיטות במחלקה בכלל לא יקבלו פרמטרים המתייחסים לתכונות לפי המערכת הפולרית. במימוש השיטות עליכם לדאוג להמרה הזו.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונות `_radius` ו-`_alpha`.

למחלקה Point הוגדרו שני **בנאים** (constructors) :

- האחד - בנאי המקבל שני פרמטרים המהווים את ערכי התכונות שיהיו לנקודה.
`public Point(double x, double y)`
אם אחד הפרמטרים שהתקבל הוא שלילי, הוא צריך להיות מאותחל ל-0.
- השני - בנאי העתקה המקבל נקודה אחרת, ומעתיק את ערכיה.
`public Point (Point other)`

בנוסף, הוגדרו במחלקה השיטות:

- שיטות **האחזור** :
 - `double getX()` המחזירה את ערכה של קואורדינטת ה- x.
 - `double getY()` המחזירה את ערכה של קואורדינטת ה- y.
- השיטות **הקובעות** :
 - `void setX (double num)` המשנה את ערכה של קואורדינטת ה- x להיות num.
אם num הוא מספר שלילי, הערך של x לא משתנה.
 - `void setY (double num)` המשנה את ערכה של קואורדינטת ה- y להיות num.
אם num הוא מספר שלילי, הערך של y לא משתנה.
- השיטה `toString` שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המתמטי המקובל - (x,y). כך, המחרוזת (3.0,4.0) מייצגת את הנקודה שקואורדינטת ה- x שלה היא 3.0 וקואורדינטת ה- y שלה היא 4.0. **שימו לב לדייק במחרוזת לפי הכתוב כאן.** ללא רווחים וללא תווים נוספים.
כדי להימנע מאי דיוקים קטנים בחישובים של ממשיים , לדוגמא קבלת תוצאה 3.9999999994 במקום 4.0 השתמש בנוסחה:
`Math.round(d*10000)/(double)10000` כאשר d מכיל את המספר הממשי שמעוניינים לעגל.
פעולת העיגול `Math.round(x)` מקבלת מספר ממשי x ומחזירה מספר שלם לפי כללי העיגול המקובלים.

יש לבצע עיגול (כפי שניתן לראות בדוגמאות לעיל, 4 ספרות אחרי הנקודה) רק בשיטה `toString` ולא בשום מקום אחר.

- `boolean equals (Point other)` – שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה והנקודה שהתקבלה כפרמטר זהות.
- `boolean isAbove (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת מעל לנקודה שהתקבלה כפרמטר. (באזור 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.
- `boolean isLeft (Point other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת משמאל לנקודה שהתקבלה כפרמטר. (באזור 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.
- `double distance (Point p)` - שיטה שמקבלת נקודה כפרמטר ומחזירה את המרחק בין הנקודה שעליה הופעלה והנקודה שהתקבלה כפרמטר. לעזרתכם, הנוסחה לחישוב מרחק בין הנקודה (x_1, y_1) , (x_2, y_2) היא $\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$
- `void move (double dx, double dy)` - המזיזה את הנקודה ב- `dx` על ציר ה- `X` וב- `dy` על ציר ה- `Y`. אם התזוזה מזיזה את הנקודה מחוץ לרביע הראשון של מערכת הצירים, הנקודה תישאר במקומה ולא תזוז.

עליכם לכתוב את המחלקה `Point` לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס בספר הדיגיטלי של יחידות 3-4, בתת-פרק של מטלה 12.

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

מוותר להשתמש אך ורק בשיטות הבאות מהמחלקה `Math`:

`cos`, `sin`, `atan`, `round`, `pow`, `sqrt` וכן בקבוע `PI`.

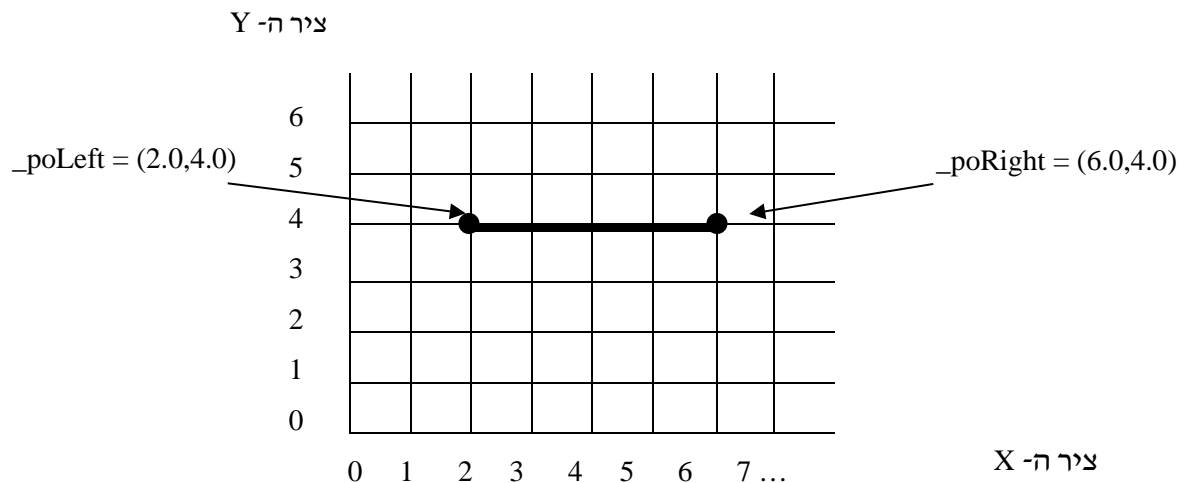
שאלה 2 - 40 נקודות

המחלקה Segment1 מייצגת מקטע במישור המקביל לציר ה- x .

למחלקה Segment1 יש את התכונות הפרטיות (instance variables) הבאות:

- Point _poLeft – שמייצגת את הנקודה השמאלית של המקטע;
- Point _poRight – שמייצגת את הנקודה הימנית של המקטע;

לדוגמא, הנה מסומן המקטע המחבר את שתי הנקודות $_poLeft = (2.0, 4.0)$ ו- $_poRight = (6.0, 4.0)$ במרחב:



שימו לב, כל המקטעים מקבילים לציר ה- X .

הנקודות $_poLeft$ ו- $_poRight$ יכולות להיות אותה נקודה. במקרה כזה אורכו של המקטע הוא 0.

אי אפשר להוסיף תכונות פרטיות למחלקה זו.

למחלקה Segment1 הוגדרו שלושה **בנאים** (constructors):

- בנאי המקבל שני פרמטרים המהווים את הנקודות השמאלית והימנית של המקטע.
`public Segment1 (Point left, Point right)`
אפשר להניח שהנקודה השמאלית left אכן שמאלית לנקודה הימנית right ואין צורך לבדוק זאת.
אי אפשר להניח שהנקודות אכן יוצרות קטע מקביל לציר ה- X . אם אכן הקטע לא מקביל, יש לקבוע את קואורדינטת ה- y של הנקודה $_poRight$ לפי קואורדינטת ה- y של הנקודה $_poLeft$.
לדוגמא, אם הנקודה left היא (3.0, 4.0) והנקודה right היא (5.0, 6.0) אזי הבנאי ייצור מקטע שהנקודה $_poLeft$ שלו תהיה (3.0, 4.0) והנקודה $_poRight$ שלו תהיה (5.0, 4.0).

- בנאי המקבל ארבעה פרמטרים שהם מספרים ממשיים. שני הראשונים הם קואורדינטות ה- x וה- y של הנקודה השמאלית של המקטע, השלישי והרביעי הם קואורדינטות ה- x וה- y של הנקודה הימנית של המקטע.

גם כאן :

אפשר להניח שהנקודה השמאלית left אכן שמאלית לנקודה הימנית right ואין צורך לבדוק זאת.

אי אפשר להניח שהנקודות אכן יוצרות קטע מקביל לציר ה- X. אם אכן הקטע לא מקביל, יש לשנות את קואורדינטת ה- y של הנקודה poRight לפי קואורדינטת ה- y של הנקודה poLeft.

```
public Segment1(double leftX ,double leftY,
                double rightX ,double rightY)
```

- בנאי העתקה המקבל מקטע אחר, ומעתיק את ערכיו.
- ```
public Segment1 (Segment1 other)
```

### בנוסף הוגדרו במחלקה השיטות:

- שיטות האחזור :
  - Point getPoLeft() המחזירה את הנקודה השמאלית של המקטע.
  - Point getPoRight() המחזירה את הנקודה הימנית של המקטע.
  - double getLength() המחזירה את אורך המקטע.
- השיטה toString שמחזירה את תוכן האובייקט כמחרוזת תווים כאשר משמאל לימין תוצג הנקודה השמאלית, לאחריה שלושה מקפים ולאחריה הנקודה הימנית. כך, המחרוזת המייצגת את המקטע שהנקודה השמאלית שלו היא (3.0, 4.0) והימנית היא (5.0, 4.0) תראה כך :
 

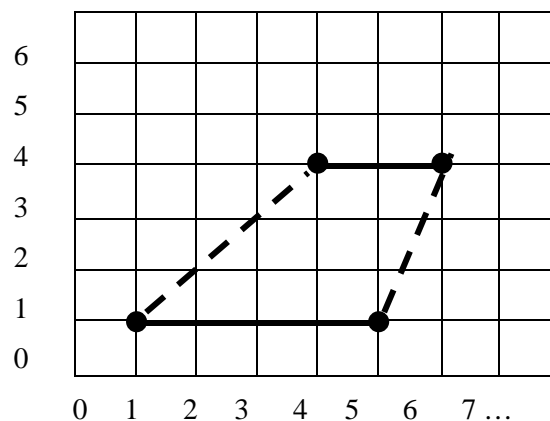
(3.0,4.0)---(5.0,4.0)
- שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים וללא תווים נוספים.
- boolean equals (Segment1 other) – שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה והמקטע שהתקבל כפרמטר זהים.
- boolean isAbove (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא מעל למקטע שהתקבל כפרמטר.
- boolean isUnder (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא מתחת למקטע שהתקבל כפרמטר. השיטה הזו משתמשת אך ורק בשיטה isAbove שהוגדרה לעיל.
- boolean isLeft (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא כולו משמאל למקטע שהתקבל כפרמטר.

שימו לב, השיטה תחזיר true רק אם כל המקטע שעליו הופעלה השיטה נמצא ממש משמאל לכל המקטע שהתקבל כפרמטר. (בלי נקודות השקה).

- boolean isRight (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא **כולו מימין** למקטע שהתקבל כפרמטר. בלי נקודות השקה.
- void moveHorizontal (double delta) - שיטה שמקבלת מספר ממשי delta כפרמטר ומזיזה את המקטע ב-delta על ציר ה-X.
- void moveVertical (double delta) - שיטה שמקבלת מספר ממשי delta כפרמטר ומזיזה את המקטע ב-delta על ציר ה-Y.
- בשתי שיטות ה-move, במידה ולאחר ההזזה אחד מקצות המקטע (או שניהם) יוצא מתחום הרביע הראשון, השיטה צריכה להשאיר את המקטע ללא שינוי.
- void changeSize (double delta) - שיטה שמקבלת מספר ממשי delta כפרמטר ומגדילה או מקטינה את אורך המקטע ב-delta. הנקודה השמאלית לא משתנה, אלא רק הנקודה הימנית. שימו לב, אם השינוי גורם לכך שהנקודה הימנית תהיה משמאל לנקודה הימנית, השינוי לא מתבצע בכלל, והמקטע נשאר כשהיה.
- boolean pointOnSegment (Point p) - שיטה המקבלת כפרמטר נקודה p ומחזירה האם הנקודה נמצאת על המקטע (גם בקצוות).
- public boolean isBigger (Segment1 other) - שיטה המקבלת כפרמטר מקטע other ומחזירה האם המקטע שעליו הופעלה השיטה ארוך יותר מהמקטע שהתקבל כפרמטר.
- public double overlap (Segment1 other) - שיטה המקבלת כפרמטר מקטע other ומחזירה את אורך החפיפה בין המקטע שעליו הופעלה השיטה ובין המקטע שהתקבל כפרמטר (אם יש כזה). אם אין חפיפה, יוחזר 0.
- public double trapezePerimeter (Segment1 other) - שיטה המקבלת כפרמטר מקטע other ומחזירה את היקף הטרפז הכלוא בין שני המקטעים.

לדוגמא, באיור הבא,

ציר ה-Y



ציר ה-X

אורך החפיפה בין המקטעים הוא 1 (הקטע בין הקואורדינטה 4 לקואורדינטה 5 על ציר ה-X). הטרפז הכלוא בין שני המקטעים הוא זה המסומן באיור.

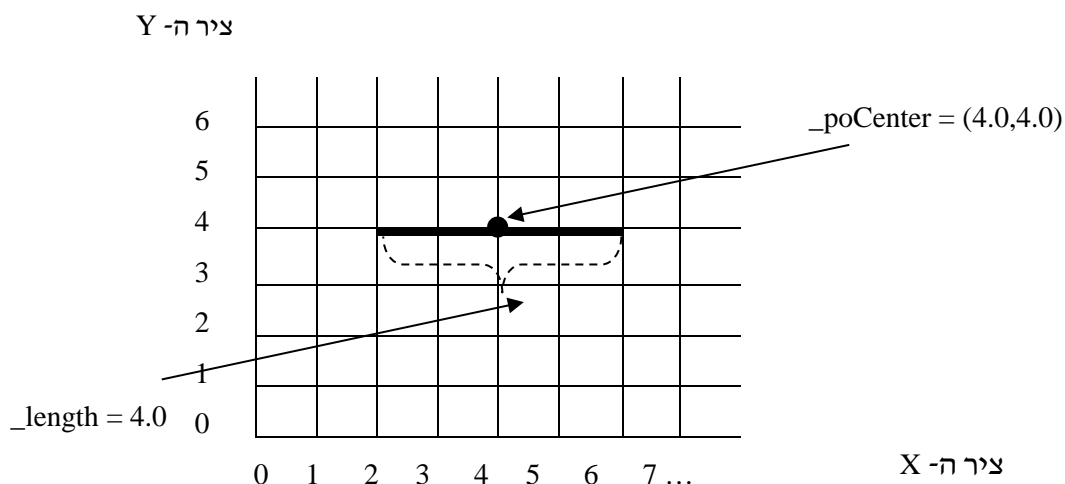
### שאלה 3 - 40 נקודות

המחלקה Segment2 מייצגת מקטע במישור המקביל לציר ה-X.

למחלקה Segment2 יש את התכונות הפרטיות (instance variables) הבאות:

- Point \_poCenter – שמייצגת את הנקודה האמצעית של המקטע;
- double \_length – שמייצגת את אורכו של המקטע;

כך למשל, המקטע שמיוצג במחלקה Segment1 על-ידי הנקודות (2.0,4.0) - השמאלית ו- (6.0, 4.0) – הימנית ייוצג במחלקה Segment2 על-ידי הנקודה המרכזית (4.0, 4.0) והאורך 4.0.



עליכם לכתוב מימוש למחלקה Segment2, כך שהיא תבצע בדיוק את אותן שיטות שמבצעת המחלקה Segment1. למרות שהייצוג הפנימי של האובייקטים (התכונות) שונה. על השיטות הכתובות עבור מחלקות Segment1 ו-Segment2 להיות זהות מבחינת שם ופונקציונליות. עם זאת, שימו לב ששיטות מקבילות בשתי המחלקות אינן מקבלות בהכרח את אותם הפרמטרים (ראו את ה-API המדויק באתר).

בנוסף, קיים למחלקה Segment2 בנאי נוסף. הבנאי מקבל כפרמטרים נקודה אחת (המרכזית) ומספר המהווה את אורך המקטע, ויוצר מהם אובייקט מהמחלקה Segment2.

**שימו לב, אסור להוסיף תכונות פרטיות.  
מותר להוסיף שיטות פרטיות.**



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 ששמנו טסטרים לשלושת המחלקות באתר הקורס. חובה שטסטרים אילו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, תכתבו חתימה ותחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרים לא ירוצו ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס. שימו לב לא לבצע aliasing במקומות המועדים.

עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר.

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת יורדו לכם הרבה נקודות!**
3. עליכם להריץ את הטסטרים שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרים לא מכסים את כל האפשרויות, ובפרט לא את מקרי הקצה. הם רק בודקים את השמות של השיטות במחלקות. מאד מומלץ להוסיף להם בדיקות
4. את התשובות לשאלות יש להגיש בשלושה קובצי Java הבאים : Segment1.java, Point.java, Segment2.java
5. ארוז את כל 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 5 – 6 נושא המטלה: לולאות ומערכים

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: 2021 א מועד אחרון להגשה: 12.12.2020

## במטלה זו אנו משתמשים במחלקה Point שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקה Point שכתבתם או בקובץ Point.class שיהיה באתר בספר הדיגיטלי של יחידות 5 – 6 בצמוד למטלה 13. נשים את הקובץ הזה באתר רק אחרי ההגשה של מטלה 12.

אנא קראו את הכתוב במדריך creating\_a\_project\_and\_using\_existing\_classes שנמצא בלשונית

"מדריכי עזר" במשאבי הלמידה בדף הבית של האתר הקורס. כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד.

שמנו באתר טסטר בסיסי לבדיקה ראשונית של המטלה. חובה להריץ את המטלה מול הטסטר ולבדוק שאין טעויות קומפילציה.

## שאלה 1 – להרצה (100%)

מצולע קמור הוא מצולע שכל זוויותיו הפנימיות קטנות מ-180 מעלות. במצולע קמור הקו המחבר כל שתי נקודות מתוך המצולע עובר רק בתוך המצולע.

### המחלקה Polygon מייצגת מצולע קמור במישור.

הייצוג נעשה על-ידי מערך ששומר את רשימת הקודקודים (vertices) של המצולע לפי סדר הופעתם במצולע. אין חשיבות מי הקודקוד הראשון. כל קדקוד מיוצג על-ידי נקודה במישור. התכונות במחלקה הן:

- מערך של הקודקודים `Point [] _vertices`
- מספר הקודקודים במצולע `int _noOfVertices`

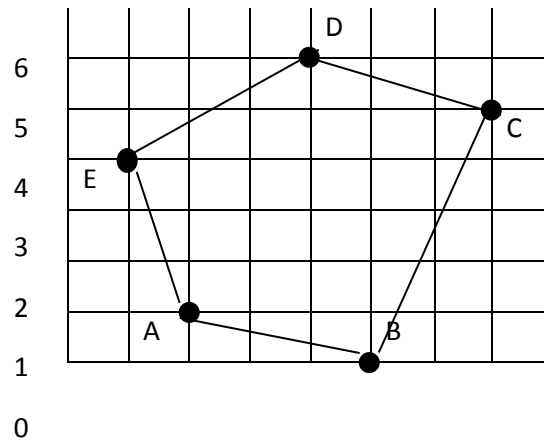
כמו כן קיים קבוע מספרי המציין את המספר המקסימלי של הקודקודים במצולע - 10.

אין להוסיף תכונות מעבר לתכונות אילו. לא פרטיות ולא ציבוריות.

הקודקודים) כלומר האובייקטים מהמחלקה (Point נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

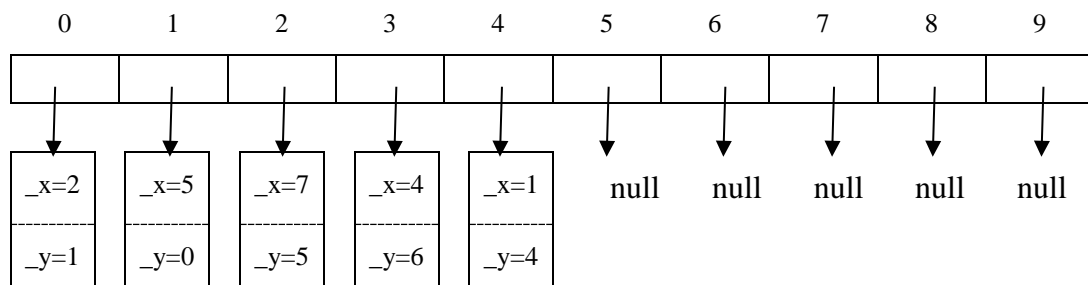
הנה דוגמא למצולע קמור. השמות שהצמדנו לנקודות הם רק לשם התייחסות בהמשך.

ציר ה-Y



ציר ה-X

המערך המייצג את הקודקודים של המצולע הוא :



כאשר `_noOfVertices = 5`

### עליכם לממש ב Java - את המחלקה Polygon לפי הסעיפים להלן:

- הגדרת התכונות של המחלקה.
- בנאי שמאתחל את תכונות המחלקה כך שהמערך יהיה בגודל מקסימלי.
- שיטה בוליאנית `addVertex` שמוסיפה קודקוד למצולע. היא מקבלת כפרמטרים שני מספרים ממשיים `x` ו `y` - שמייצגים את הקואורדינטות של הקודקוד הנוסף, ומכניסה נקודה עם תכונות אלו למערך הקודקודים במקום הראשון האפשרי במערך. השיטה תחזיר `true` אם ההוספה התבצעה כראוי, כלומר נמצא מקום במערך עבור הקודקוד הנוסף ו `false` - אם המערך כבר מלא. ניתן להניח את ההנחות הבאות (כלומר, אין צורך לבדוק זאת):
  - הקודקוד החדש שנוסף לא מקלקל את היות המצולע מצולע קמור.
  - הקודקוד אינו קיים עדיין במצולע.

• הוספת הקודקודים על ידי השיטה addVertex נעשית תמיד לפי סדר הופעתם במצולע.

4. שיטה highestVertex המחזירה העתק של את הקודקוד שנמצא הכי גבוה במצולע. אם יש יותר מאחד בגובה הגבוה ביותר, היא מחזירה את הראשון בו נתקלה. אם אין קודקודים בפוליגון (כלומר המערך הוא ריק) השיטה תחזיר null.

5. השיטה toString המחזירה מחרוזת תווים המייצגת את המצולע. המחרוזת צריכה להיות **בדיוק** בפורמט הבא: שימו לב שאין רווחים במחרוזת של הקודקודים.

The polygon has 5 vertices:

((2.0,1.0),(5.0,0.0),(7.0,5.0),(4.0,6.0),(1.0,4.0))

אם אין קודקודים השיטה תחזיר מחרוזת **בדיוק** בפורמט הבא (כולל הנקודה):

The polygon has 0 vertices.

6. שיטה calcPerimeter המחזירה מספר ממשי (double) המייצג את היקף המצולע. אם מספר הקודקודים הוא 2 יוחזר אורך הקטע (לא הלוך וחזור). אם מספר הקודקודים הוא 1 או 0 יוחזר 0.

7. שיטה calcArea המחזירה מספר ממשי (double) המייצג את שטח המצולע. כדי לחשב את שטח המצולע, צריך לסכום את שטחי המשולשים המכסים את שטח המצולע. בדוגמא לעיל, למשל, צריך לסכום את שטחי המשולשים הבאים A-C-D, A-D-E, A-B-C.

לשם חישוב שטח המשולש, ניתן להשתמש בנוסחת Heron הקובעת כי שטח המשולש שווה לשורש הריבועי של  $s(s-a)(s-b)(s-c)$  כאשר  $a, b$  ו- $c$  הם אורכי שלושת הצלעות של המשולש, ו- $s$  הוא מחצית היקפו.

ניתן להוסיף שיטה לחישוב שטח משולש. שימו לב שהשיטה לחישוב שטח משולש צריכה להיות פרטית ולא ציבורית.

אם מספר הקודקודים קטן מ-3 יוחזר 0.

8. שיטה isBigger בוליאנית המקבלת מצולע אחר, ומחזירה true אם המצולע שעליו מופעלת השיטה גדול בשטחו מהמצולע המועבר כפרמטר ואחרת מחזירה false. ניתן להניח שהפרמטר אינו null.

9. שיטה findVertex המקבלת נקודה כפרמטר ומחזירה את המיקום שלה במערך, אם היא נמצאת. אם לא, יוחזר -1 (מינוס). לדוגמא, אם השיטה תופעל עם הפרמטר (5,0) היא תחזיר את הערך 1. שימו לב שהקודקוד הראשון נמצא במיקום 0 במערך. ניתן להניח שהפרמטר אינו null.

10. שיטה getNextVertex המקבלת נקודה כפרמטר, ומחזירה העתק של הנקודה המייצגת את הקודקוד הבא במצולע. אם הנקודה שהתקבלה אינה קודקוד במצולע, השיטה תחזיר null. אם הנקודה היא האיבר האחרון במערך, יוחזר העתק של הנקודה הראשונה. אם הנקודה היא הנקודה היחידה במערך יוחזר העתק של הנקודה עצמה. ניתן להניח שהפרמטר אינו null.

11. שיטה `getBoundingBox` המחזירה את המלבן (כפוליגון) (המקביל לצירים) החוסם את המצולע. אם מספר הקודקודים קטן מ-3 יוחזר `null`. הקודקוד הראשון במערך המייצג את המלבן צריך להיות הקודקוד השמאלי התחתון, ושאר הקודקודים יופיעו בסדר הפוך מכיוון השעון (כלומר הקודקוד השני יהיה הימני התחתון, השלישי הימני העליון והרביעי השמאלי העליון).

לפניכם רשימת החתימות של הבנאי ושיטות המחלקה :

|                                                           |                                                                                             |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <code>public Polygon()</code>                             | בנאי                                                                                        |
| <code>public boolean addVertex(double x, double y)</code> | שיטה שמוסיפה קודקוד למצולע                                                                  |
| <code>public Point highestVertex()</code>                 | שיטה שמחזירה העתק של את הקודקוד שנמצא הכי גבוה במצולע                                       |
| <code>public String toString()</code>                     | שיטה שמחזירה מחרוזת תווים המייצגת את המצולע                                                 |
| <code>public double calcPerimeter ()</code>               | שיטה שמחזירה את היקף המצולע                                                                 |
| <code>public double calcArea()</code>                     | שיטה שמחזירה את שטח המצולע                                                                  |
| <code>public boolean isBigger(Polygon other)</code>       | שיטה שמקבלת מצולע אחר, ובודקת אם המצולע שעליו מופעלת השיטה גדול בשטחו מהמצולע המועבר כפרמטר |
| <code>public int findVertex(Point p)</code>               | שיטה שמקבלת נקודה כפרמטר ומחזירה את המיקום שלה במערך                                        |
| <code>public Point getNextVertex(Point p)</code>          | שיטה שמקבלת נקודה כפרמטר, ומחזירה העתק של הנקודה המייצגת את הקודקוד הבא במצולע              |
| <code>public Polygon getBoundingBox()</code>              | שיטה שמחזירה את המלבן (כפוליגון) החוסם את המצולע                                            |

**שימו לב לא לבצע aliasing במקומות המועדים.**

מוותר 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 והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקה שכתבתם בטסטר זה, ולהגיש אותה 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 הקובץ , Polygon.java עטפו אותו בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

## **בהצלחה**

# מטלת מחשב (ממ"ח) 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: 2021 מועד אחרון להגשה: 26.12.2020

(ת)

את התשובות לממ"ח יש לשלוח באמצעות מערכת שאלת"א

בכתובת <http://www.openu.ac.il/sheilta/>

הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח למנחה!

## שאלה 1

נתונה חתימת השיטה הבאה:

```
public double myFunc(double x, int y)
```

אילו מהחתימות הבאות אינה מהווה העמסה חוקית של השיטה myFunc?

א. `public int myFunc()`

ב. `public int myFunc(double y, char x)`

ג. `public void myFunc(int x)`

ד. `public double myFunc(double a, int b)`

ה. כל החתימות בסעיפים א – ד מהוות העמסה חוקית של myFunc

ו. יש יותר מחתימה אחת בסעיפים א – ד שאינה מהווה העמסה חוקית של myFunc

## שאלה 2

נתונה המחלקה הבאה:

```
public class A {
 public int method1(double x){ ... }
}
```

עבור המחלקה B שירשת מ-A. אילו מחתימות השיטות הבאות שיכתבו במחלקה B מהווה

דריסה (overriding) של השיטה method1?

א. `public int method1(double y)`

ב. `private int method1(char x)`

ג. `public void method1()`

ד. `public int method1(int y)`

ה. אף חתימה אינה מהווה דריסה של השיטה method1 מהמחלקה A

### שאלה 3

נניח שיש בפרויקט חמש מחלקות – Mammal, Dalmatian, Poodle, Cat, Dog, מה מבנה המחלקות הסביר ביותר (אפשר להניח שמעבר לתיאור בכל משפט אין קשרי ירושה נוספים בין המחלקות ואם לא כתוב ממי יורשת מחלקה, היא יורשת מהמחלקה Object)?

- א. המחלקות Dalmatian, Poodle, Cat, Dog כולן יורשות מהמחלקה Mammal
- ב. המחלקות Dalmatian, Poodle יורשות מהמחלקה Dog.
- ג. המחלקה Mammal יורשת מהמחלקות Dalmatian, Poodle, Cat, Dog
- ד. המחלקה Cat יורשת מהמחלקה Mammal והמחלקה Dog יורשת מהמחלקה Poodle.
- ה. המחלקות Dalmatian, Poodle יורשות מהמחלקה Dog והמחלקות Cat, Dog יורשות מהמחלקה Mammal

### שאלה 4

נתונה המחלקה הבאה :

```
public class Mmh02 {
 private int _num;
 private static int _count = 0;

 public Mmh02() {
 _count++;
 _num = _count;
 }

 public void printNow() {
 System.out.println (_num + " " + _count);
 }
}
```

כמה אובייקטים יש לייצר מהמחלקה Mmh02 ועל איזה אובייקט יש להפעיל את השיטה printNow() כך שהפלט יהיה 53?

- א. יש לייצר 5 אובייקטים ולהפעיל את השיטה על האובייקט ה-3 שנוצר
- ב. יש לייצר 5 אובייקטים ולהפעיל את השיטה על האובייקט הראשון שנוצר
- ג. יש לייצר 53 אובייקטים ולהפעיל את השיטה על האובייקט ה-53 שנוצר
- ד. יש לייצר 3 אובייקטים ולהפעיל את השיטה על האובייקט ה-3 שנוצר
- ה. תשובות א-ד אינן נכונות



## שאלה 5

נתונה המחלקה A, והמחלקה B שיורשת מ-A, ונתון המשפט הבא שעובר קומפילציה ורץ בצורה תקינה:

```
A a1 = new B();
```

אילו מהמשפטים הבאים נכון בוודאות?

א. המחלקה A היא מחלקה אבסטרקטית.

ב. לא ניתן לכתוב את המשפט:

```
Object obj = a1;
```

מכיוון שדרושה המרה מפורשת (casting).

ג. לא ניתן לכתוב את המשפט:

```
A a2 = a1;
```

מכיוון שדרושה המרה מפורשת (casting).

ד. לא ניתן לכתוב את המשפט:

```
B b1 = a1;
```

מכיוון שדרושה המרה מפורשת (casting).

ה. לא ניתן לכתוב את המשפט:

```
B b1 = a1;
```

מכיוון שתגרם בוודאות שגיאת ריצה.

## שאלה 6

אילו מהשיטות/בנאים הבאים לא מוגדרים במחלקה Object?

א. `public String toString()`

ב. `public void println(String s)`

ג. `public boolean equals(Object obj)`

ד. `public Object()`

## שאלה 7

נתון קטע הקוד הבא :

```
public class Person
{
 private String _name;
}

public class Student extends Person
{
 private int[] _grades;
 private int _id;
 public static int _studentsNum;

 public Student (Student student)
 {
 super(student._name);
 }
}
```

אילו מבין המשפטים הבאים נכון?

- א. רק שינוי המאפיין `_name` ל-`protected` יביא לשגיאת ריצה.
- ב. רק שינוי המאפיין `_name` ל-`public` יביא לשגיאת ריצה.
- ג. רק שינוי המאפיין `_name` ל-`default` יביא לשגיאת ריצה.
- ד. התכנית לא תעבור קומפילציה.
- ה. רק שינוי המאפיין `_name` ל-`protected` יביא לשגיאת קומפילציה.
- ו. רק שינוי המאפיין `_name` ל-`public` יביא לשגיאת קומפילציה.
- ז. רק שינוי המאפיין `_name` ל-`default` יביא לשגיאת קומפילציה.
- ח. הרצת התכנית תביא לשגיאת קומפילציה.

## הכתוב להלן מתייחס לשאלות 8 - 14

נתונות המחלקות AA , BB ו-Driver הבאות (בקבצים שונים כמובן):

```
public class AA {
 private int _val=0;
 public AA(){
 _val=5;
 }

 public AA(int val){
 _val=val;
 }

 public int getVal(){
 return _val;
 }

 public void setVal(int val){
 _val=val;
 }

 public String toString(){
 return "val=" + _val ;
 }
}

//-----

public class BB extends AA {
 private String _st;
 public BB (){
 _st="bb";
 }

 public BB(String st, int val){
 super(val);
 _st=st;
 }

 public String getSt(){
 return _st;
 }
}

// שימו לב, בעמוד הבא יש שלוש שיטות שונות בשם equals של המחלקה BB
```

```

public boolean equals (Object ob) // שיטה 1
{
 if ((ob != null) && (ob instanceof BB))
 {
 if (_st.equals(((BB)ob)._st) &&
 (getVal() == ((BB)ob).getVal()))
 return true;
 }
 return false;
}

public boolean equals (AA ob) // שיטה 2
{
 if ((ob != null) && (ob instanceof BB))
 {
 if (_st.equals(((BB)ob)._st) &&
 (getVal() == ((BB)ob).getVal()))
 return true;
 }
 return false;
}

public boolean equals (BB ob) // שיטה 3
{
 if (ob != null)
 {
 if (_st.equals(((BB)ob)._st) &&
 (getVal() == ((BB)ob).getVal()))
 return true;
 }
 return false;
}
}

```

המחלקה Driver שלהלן נמצאת גם היא באותו פרויקט.

```
public class Driver
{
 public static void main (String [] args)
 {
 AA a1 = new AA();
 AA a2 = new BB();
 AA a3 = new AA();
 AA a4 = new BB();
 BB b1 = new BB();
 BB b2 = new BB();

 // כאן יוכנסו השורות שבשאלות להלן
 }
}
```

השאלות בלתי תלויות אחת בשניה.

## שאלה 8

בעקבות הפעלת השורה `System.out.println(a3.equals(a1));`

|    |                 |         |       |
|----|-----------------|---------|-------|
| א. | נקראה השיטה 1   | ויודפס: | true  |
| ב. | נקראה השיטה 2   | ויודפס: | true  |
| ג. | נקראה השיטה 3   | ויודפס: | true  |
| ד. | נקראה שיטה אחרת | ויודפס: | true  |
| ה. | נקראה השיטה 1   | ויודפס: | false |
| ו. | נקראה השיטה 2   | ויודפס: | false |
| ז. | נקראה השיטה 3   | ויודפס: | false |
| ח. | נקראה שיטה אחרת | ויודפס: | false |

## שאלה 9

בעקבות הפעלת השורה `System.out.println(a4.equals(a2));`

|    |                        |         |              |
|----|------------------------|---------|--------------|
| א. | נקראה השיטה 1          | ויודפס: | <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: | <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: | <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: | <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: | <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: | <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>false</b> |

## שאלה 10

בעקבות הפעלת השורה `System.out.println(a1.equals(a2));`

|    |                        |         |              |
|----|------------------------|---------|--------------|
| א. | נקראה השיטה 1          | ויודפס: | <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: | <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: | <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: | <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: | <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: | <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>false</b> |

## שאלה 11

בעקבות הפעלת השורה `System.out.println(a2.equals(b1));`

|    |                        |         |              |
|----|------------------------|---------|--------------|
| א. | נקראה השיטה 1          | ויודפס: | <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: | <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: | <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: | <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: | <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: | <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: | <b>false</b> |

## שאלה 12

בעקבות הפעלת השורה `System.out.println(b1.equals(a1));`

|    |                        |                      |
|----|------------------------|----------------------|
| א. | נקראה השיטה 1          | ויודפס: <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>false</b> |

## שאלה 13

בעקבות הפעלת השורה `System.out.println(b2.equals(b1));`

|    |                        |                      |
|----|------------------------|----------------------|
| א. | נקראה השיטה 1          | ויודפס: <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>false</b> |

## שאלה 14

בעקבות הפעלת השורה `System.out.println(b1.equals(a4));`

|    |                        |                      |
|----|------------------------|----------------------|
| א. | נקראה השיטה 1          | ויודפס: <b>true</b>  |
| ב. | נקראה השיטה 2          | ויודפס: <b>true</b>  |
| ג. | נקראה השיטה 3          | ויודפס: <b>true</b>  |
| ד. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>true</b>  |
| ה. | נקראה השיטה 1          | ויודפס: <b>false</b> |
| ו. | נקראה השיטה 2          | ויודפס: <b>false</b> |
| ז. | נקראה השיטה 3          | ויודפס: <b>false</b> |
| ח. | נקראה שיטה <b>אחרת</b> | ויודפס: <b>false</b> |

## שאלה 15

מה נוכל להגיד בוודאות על קטע התכנית הבא, אשר רץ ללא שגיאה ?  
אין להניח שנעשה שימוש במוסכמות כלשהן לגבי השמות (naming conventions):

```
a x = new b(t,s);
h.g(n,d);
```

- א. השיטה g חייבת להיות ציבורית (public)
- ב. t הוא בהכרח משתנה מטיפוס פשוט (כמו int, double וכד')
- ג. b יכולה להיות מחלקה אבסטרקטית
- ד. h הוא בהכרח אובייקט
- ה. d יכול להיות שם של שיטה
- ו. b היא בהכרח מחלקה שיורשת מ a (גם אם לא ישירות)
- ז. יתכן ש a היא מחלקה שיורשת מ b
- ח. אף תשובה אינה נכונה
- ט. יש יותר מתשובה אחת נכונה

## שאלה 16

לפניכם המחלקות A, B. כל אחת בקובץ נפרד, כמובן.

```
public class A
{
}
public class B extends A
{
 public B()
 {
 System.out.println ("B constructor");
 }
}
```

איזו מהטענות הבאות אינה נכונה?

- א. הבנאי (constructor) של B יקרא לבנאי הריק של A.
- ב. הבנאי הריק של Object לא ייקרא כי ל-A אין בנאי מפורש שיקרא לו.
- ג. הבנאי הריק של Object ייקרא לאחר הדפסת המחרוזת "B constructor".
- ד. קודם ייקרא הבנאי הריק של A ואז תודפס המחרוזת "B constructor".
- ה. בחר בסעיף הזה אם הטענות ב ו-ג אינן נכונות.



### נתונה המחלקה המופשטת הבאה:

```
public abstract class A {
 public abstract boolean f(int x);
}
```

בכל אחת מהשאלות 17-19 נתונה מחלקה היורשת מהמחלקה A.

#### שאלה 17

```
public abstract class B extends A {
 public boolean f(int x) {
 return x == 2;
 }
}
```

איזה מהמשפטים הבאים הינו נכון?

- א. המחלקה לא תעבור קומפילציה כיוון שאין לה בנאי ריק.
- ב. המחלקה לא תעבור קומפילציה כיוון שהיא מופשטת ולכן היא לא יכולה לדרוס שיטה מופשטת של המחלקה המורשה.
- ג. המחלקה חוקית ותעבור קומפילציה.
- ד. המחלקה תעבור קומפילציה אבל הרצת השיטה f תגרום לשגיאת ריצה.
- ה. אף משפט בסעיפים א-ד אינו נכון

#### שאלה 18

```
public class B extends A {
 public boolean f(int x) {
 Object obj = new A();
 return this.equals(obj);
 }
}
```

איזה מהמשפטים הבאים הינו נכון?

- א. המחלקה חוקית ותעבור קומפילציה.
- ב. המחלקה לא תעבור קומפילציה כיוון של-A אין בנאי ריק.
- ג. המחלקה לא תעבור קומפילציה כיוון שלמחלקה B אין שיטה בשם equals.
- ד. המחלקה לא תעבור קומפילציה כיוון שלא ניתן ליצור אובייקטים מסוג A.
- ה. אף משפט בסעיפים א-ד אינו נכון

```
public class B extends A {
 public boolean f(double x) {
 return x == 2.0;
 }
}
```

איזה מהמשפטים הבאים הינו נכון?

- א. המחלקה לא תעבור קומפילציה כיוון שהיא לא דורסת את השיטה המופשטת של A.
- ב. המחלקה חוקית ותעבור קומפילציה.
- ג. המחלקה לא תעבור קומפילציה כיוון שאין לה בנאי ריק.
- ד. המחלקה לא תעבור קומפילציה כיוון שהשיטה f אינה מחזירה ערך בוליאני.
- ה. אף משפט בסעיפים א-ד אינו נכון

## שאלה 20

תהא A מחלקה אבסטרקטית (abstract class). איזו מהטענות הבאות נכונה?

- א. כל שיטה במחלקה A חייבת להיות מוגדרת כשיטה אבסטרקטית בעצמה
- ב. למחלקה A אין בנאי
- ג. כל מחלקה שירשת מ-A חייבת בכל מקרה לממש את כל השיטות האבסטרקטיות של A
- ד. יכול להיות שבמחלקה A אין אף שיטה אבסטרקטית
- ה. כל הטענות לעיל אינן נכונות

**את התשובות לממ"ח יש לשלוח באמצעות מערכת שאילת"א**

**בכתובת <http://www.openu.ac.il/sheilta/>**

**הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח למנחה!  
שימו לב שהממ"ח אכן נקלט בשאילתא, ושמרו את האסמכתא שקיבלתם כתוצאה  
מהשליחה. אם לא קיבלתם אישור – כאילו לא שלחתם!**

# 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: 2021 מועד אחרון להגשה: 16.1.2021

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 (להגשה)

נתון מערך מלא במספרים שלמים, שבו כל מספר מופיע פעמיים ברצף פרט למספר אחד שמופיע רק פעם אחת. המערך אינו ממוין. לדוגמא, המערכים הבאים מקיימים את התנאי:

|     |   |   |    |    |    |    |    |   |   |
|-----|---|---|----|----|----|----|----|---|---|
|     | 0 | 1 | 2  | 3  | 4  | 5  | 6  | 7 | 8 |
| a = | 6 | 6 | 18 | 18 | -4 | -4 | 12 | 9 | 9 |

|     |   |   |    |    |   |   |   |   |    |    |    |    |    |
|-----|---|---|----|----|---|---|---|---|----|----|----|----|----|
|     | 0 | 1 | 2  | 3  | 4 | 5 | 6 | 7 | 8  | 9  | 10 | 11 | 12 |
| b = | 8 | 8 | -7 | -7 | 3 | 3 | 0 | 0 | 10 | 10 | 5  | 5  | 4  |

|     |   |
|-----|---|
|     | 0 |
| c = | 5 |

כתבו שיטה סטטית שמקבלת כפרמטר מערך שמקיים את התנאי הנ"ל, ומחזירה את המספר שמופיע במערך רק פעם אחת. לדוגמא, במערכים לעיל:

- המספר הבודד במערך a הוא 12 שנמצא באינדקס 6
- המספר הבודד במערך b הוא 4 שנמצא באינדקס 12
- המספר הבודד במערך c הוא 5 שנמצא באינדקס 0

אתם יכולים להניח שהמערך אינו ריק ושהוא מקיים את התנאי, אין צורך לבדוק זאת.

חתימת השיטה היא :

```
public static int findSingle (int [] a)
```

**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 של השיטה שכתבתם? הסבירו תשובתכם.  
אל תשכחו לתעד את מה שכתבתם!

## שאלה 2- 25 נקודות (להגשה)

בהינתן מערך של מספרים שלמים חיוביים ומספר שלם וחיובי  $x$  מצא את גודלו של תת המערך הקטן ביותר שסכומו גדול יותר מהמספר  $x$ .  
לדוגמא :

עבור המערך :  $arr[] = \{1, 4, 45, 6, 0, 19\}$  והמספר  $x = 51$  יוחזר הערך 3 – כיוון שתת המערך המינימלי שסכומו גדול מ-51 הוא  $\{4, 45, 6\}$

עבור המערך :  $arr[] = \{1, 10, 5, 2, 7\}$  והמספר  $x = 9$  יוחזר הערך 1 כיוון שתת המערך המינימלי הוא  $\{10\}$

עבור המערך :  $arr[] = \{1, 11, 100, 1, 0, 200, 3, 2, 1, 250\}$  והמספר  $x = 280$  יוחזר הערך 4 כיוון שתת המערך המינימלי הוא  $\{100, 1, 0, 200\}$

עבור המערך :  $arr[] = \{45, 6, 0\}$  והמספר  $x = 51$  יוחזר הערך 2 כיוון שתת המערך המינימלי הוא  $\{45, 6\}$

אם המערך ריק או שאין תת מערך שסכומו גדול יותר מהערך  $x$  יוחזר הערך -1  
חתימת השיטה היא :

```
public static int smallestSubSum(int arr[], int x)
```

**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 של השיטה שכתבתם? הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!

### שאלה 3- 25 נקודות (להגשה)

כתבו שיטה סטטית רקורסיבית המקבלת כפרמטר מספר שלם חיובי num ומחזירה את מספר הפתרונות למשוואה

$$x_1 + x_2 + x_3 = \text{num}$$

כאשר נתון ששלושת ה-  $x$  הם מספרים שלמים וחייבים בין 1 ל-10. השיטה צריכה גם להדפיס את הפתרונות האלו, כל פתרון בשורה נפרדת. סדר ההדפסה לא משנה.

לדוגמא,

- אם  $\text{num} = 3$  אזי השיטה תחזיר את הערך 1 שכן יש רק פתרון אחד למשוואה  $x_1 + x_2 + x_3 = 3$  והוא כאשר  $x_1 = 1, x_2 = 1, x_3 = 1$ . לכן השיטה תדפיס  $1+1+1$
- אם  $\text{num} = 5$  אזי השיטה תחזיר את הערך 6 ותדפיס את הפתרונות הבאים:  
 $1 + 1 + 3$   
 $1 + 2 + 2$   
 $1 + 3 + 1$   
 $2 + 1 + 2$   
 $2 + 2 + 1$   
 $3 + 1 + 1$

- אם  $\text{num} < 3$  או  $\text{num} > 30$  השיטה תחזיר 0.

חתימת השיטה:

```
public static int solutions(int num)
```

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

אפשר להשתמש בהעמסת-יתר (overloading). אין צורך לדאוג ליעילות השיטה.

## שאלה 4-25 נקודות (להגשה)

נתון מערך דו-ממדי ריבועי (מספר השורות שווה למספר העמודות) שמכיל ערכים בוליאניים – true/false.

נגדיר : איזור true במערך (true region), כאוסף מקסימלי של תאים סמוכים שכולם בעלי ערך true. תאים הממוקמים באלכסון זה לזה לא נחשבים לסמוכים.

למשל, עבור המערך מימין (כאן אנחנו מסמנים את הערך true כ-1 ואת הערך false כ-0), קיימים 3 איזורי true והם מסומנים במערך משמאל :

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

עליכם לכתוב שיטה רקורסיבית המקבלת כפרמטר מטריצה ריבועית בוליאנית ומחזירה כמה איזורי true שונים קיימים במטריצה. אם לא קיימים איזורי true יוחזר 0. שימו לב שאיזור true מורכב לפחות מתא אחד.

חתימת השיטה היא :

```
public static int cntTrueReg (boolean[][]mat)
```

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) וכן מותר לשנות את המערך.

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 שטסטר 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.

## 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

חומר הלימוד למטלה: יחידה 11 נושא המטלה: רשימות מקושרות

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: א2021 מועד אחרון להגשה: 23.1.2021

בתרגיל זה תתמודדו עם בעיה פרקטית – אנו יודעים שהמספרים אותם ניתן לשמור בתאי זיכרון מסוג `int` או `long` מוגבלים בגודלם (בגלל גודל הזיכרון). אולם לפעמים נצטרך להשתמש באפליקציות מסוימות במספרים גדולים מאוד, כדוגמת המספר הבא:

233,128,765,555,234

מספר זה בן 15 ספרות לא יכול להישמר בתא זיכרון בודד מסוג `long` בגלל גודלו הרב. דרך אחת להתמודד עם שמירת מספרים גדולים מאוד היא באמצעות רשימה מקושרת. נוכל להגדיר רשימה מקושרת שכל איבר בה יכיל ספרה אחת מהמספר, כך שהמספר עצמו לא ישמר כמספר אלא כאוסף ספרותיו אחת אחרי השנייה. בתרגיל זה תגדירו רשימה מקושרת לשמירת מספרים גדולים מאוד, ותגדירו כמה פעולות שימושיות על הרשימה.

**אנו מתייחסים במטלה זו למספרים אי-שליליים בלבד (כלומר חיוביים ואפס).**  
**שימו לב שהרשימה המקושרת חייבת להיות חד-סטריית.**

## שאלה 1 (100%)

הגדירו את המחלקה `BigInteger`. מחלקה זו תייצג רשימה מקושרת של ספרות, שייצגו מספר גדול מאוד. תוכלו להשתמש במחלקה `IntNode` שהוצגה בהרצאות כדי להגדיר את החוליה הבסיסית של הרשימה. כמו כן, תוכלו להתבסס על ההגדרה של המחלקה `IntList` שהוצגה אף היא בהרצאות, כדי לבנות את המחלקה `BigInteger`.

שימו לב – השקיעו מחשבה בייצוג המספר. כיוון שמדובר ברשימה מקושרת יש לכם שתי אפשרויות – לייצג את המספר מהסוף להתחלה (כלומר, האיבר הראשון ברשימה יכיל את ספרת האחדות במספר) או מההתחלה לסוף (כלומר, האיבר הראשון ברשימה יכיל את הספרה השמאלית ביותר במספר). חשבו היטב איזה ייצוג עדיף מבחינת המעברים על הרשימה (רמז – קחו בחשבון שהפעולות החשבוניות על מספרים שלמים מתחילות בד"כ בחישוב על הספרות הימניות יותר...).

אתם רשאים להחליט על המבנה הפנימי של המחלקה בעצמכם, אולם הממשק של המחלקה חייב להכיל רק (ואך ורק) את השיטות הבאות:



## בנאים

כתבו למחלקה שלושה בנאים :

1. בנאי ריק שמאתחל את הרשימה למספר 0. כלומר, רשימה שמכילה איבר אחד שספרתו היא 0.
2. בנאי שמקבל כפרמטר מספר מסוג long ומאחסן אותו בפורמט של רשימה.
3. בנאי העתקה.

## שיטות

מאחר והמחלקה מייצגת מספרים שלמים, נרצה לבצע בקלות פעולות חשבון שונות על המספרים. לכן עליכם להגדיר את השיטות הבאות :

- `public String toString()`

השיטה צריכה להחזיר כמחרוזת תווים את המספר שמיוצג ברשימה. המחרוזת תכיל את המספר בצורה הרגילה שלו, כלומר, הספרה המשמעותית ביותר תופיע ראשונה, ואחריה שאר הספרות.

חובה לממש את השיטה בזמן  $O(n)$  כאשר  $n$  הוא מספר הספרות במספר. אסור ליצור רשימות חדשות, מערכים או להפוך את הרשימה למספר. ממשו את השיטה הזאת ברקורסיה.

- `public int compareTo (BigInteger other)`

שיטת השוואה בין שני מספרים גדולים. אם המספר עליו מופעלת השיטה קטן מהמספר שהתקבל כפרמטר, יוחזר הערך -1, אם המספר עליו מופעלת השיטה גדול מהמספר שהתקבל כפרמטר, יוחזר הערך 1, אם המספרים שווים יוחזר הערך 0.

- `public BigInteger addBigInteger (BigInteger other)`

שיטת החיבור של שני מספרים גדולים. החיבור מחזיר כתוצאה אובייקט חדש מסוג `BigInteger`.

- `public BigInteger addLong (long num)`

שיטת החיבור של אובייקט מסוג מספר גדול ועוד מספר `long`. החיבור מחזיר כתוצאה אובייקט חדש מסוג `BigInteger`.

- `public BigInteger subtractBigInteger (BigInteger other)`

שיטת החיסור של שני מספרים גדולים. החיסור הוא של המספר הגדול מהשניים פחות המספר הקטן. מחזיר כתוצאה אובייקט חדש מסוג `BigInteger`. אי אפשר להניח מי מהשניים גדול. יש לבדוק זאת (כדי שהתוצאה לא תהיה מספר שלילי).

- `public BigInteger multBigInteger (BigInteger other)`

שיטת הכפל של שני מספרים גדולים. הכפל מחזיר כתוצאה אובייקט חדש מסוג `BigInteger`.

הערה כללית לגבי השיטות חיבור, חיסור וכפל. השיטות לא ישנו את האובייקט עליו הן הופעלו ולא את האובייקט שהועבר כפרמטר, אלא יצרו `BigInteger` חדש שיכיל את פתרון התרגיל.

כל הפעולות חייבות להיות יעילות ככל שניתן מבחינת המעברים על הרשימה (כלומר – אם אפשר להסתדר רק עם מעבר אחד על הרשימה, עשו זאת. אין טעם להשקיע במעברים מיותרים על הרשימות).

הערה חשובה – בכל השיטות בתרגיל **אסור** להפוך את המספר שמיוצג על-ידי הרשימה למספר אמיתי, לבצע עליו את הפעולות ואז להפוך אותו חזרה לרשימה (מה הסיבה?). כמו כן **אסור** להשתמש במערכים בתרגיל זה.

אנחנו כתבנו את הממשק הפומבי ועליו לא ניתן להוסיף שיטות, אולם ניתן להוסיף שיטות פרטיות לנוחות וקריאות המחלקה.

שימו לב לכל מקרי השגיאה האפשריים!

כתבו (כחלק מהתיעוד) 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 `IntNode`, `IntList` בחוברת השקפים.
3. עליכם לכתוב תיעוד API ותיעוד פנימי מלא.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות יהיו בדיוק לפי הוראות הממ"ן.
3. עליכם להגיש את הקובץ `BigInteger` עטפו אותו בקובץ `zip` ושלחו. אין לשלוח קבצים נוספים.

## ב ה צ ל ח ה

# מטלת מחשב (ממ"ח) 03

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 12 נושא המטלה: תורים, מחסניות, עצים בינריים

מספר השאלות: 20 משקל המטלה: 1 נקודות

סמסטר: 2021 מועד אחרון להגשה: 30.1.2021

(ת)

את התשובות לממ"ח יש לשלוח באמצעות מערכת שאלת"א

בכתובת <http://www.openu.ac.il/sheilta/>

הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח למנחה!

יש לשמור על מספר האסמכתא שאתם מקבלים מהמחשב!

בשאלות מטלה זו, יש להעזר בממשקים למחלקות Queue, Stack, Node  
הנתונים בהרצאות הקורס.

שאלה 1. איזה מהמשפטים הבאים נכון?

- א. LIFO הוא מושג שמתאר התנהגות של תור – האחרון שנכנס הוא הראשון שיוצא.
- ב. LIFO הוא מושג שמתאר התנהגות של מחסנית – האחרון שנכנס הוא הראשון שיוצא.
- ג. LIFO הוא מושג שמתאר התנהגות של תור – הראשון שנכנס הוא הראשון שיוצא.
- ד. LIFO הוא מושג שמתאר התנהגות של מחסנית – הראשון שנכנס הוא הראשון שיוצא.
- ה. אף אחד מהמשפטים א – ד אינו נכון.
- ו. יש יותר ממשפט אחד נכון במשפטים א – ד.

שאלה 2. הניחו שיש תור q (מהמחלקה Queue) שמכיל את המספרים 1,2,3,4,5,6 בסדר זה

(המספר 1 בראש התור). הניחו שנתונה s מחסנית ריקה ולא ניתן לייצר עוד מחסניות ותורים, וניתן לבצע רק שלושה סוגים של פעולות:

- i. להוציא איבר מ-q (dequeue) ולהדפיס אותו
- ii. להוציא איבר מ-q (dequeue) ולשים אותו ב-s (push)
- iii. להוציא איבר מ-s (pop) ולהדפיס אותו.

איזה מהפלטים הבאים (משמאל לימין) אינו אפשרי בשימוש אך ורק בפעולות אלה?

א. 123456

ב. 345612

ג. 234561

ד. 125643

ה. 654321

ו. יש יותר מפלט אחד בין א – ה שאינו אפשרי

**שאלה 3.** מה מבצעת השיטה הבאה כשהיא מופעלת על מחסנית s1 כלשהי:

```
public void something()
{
 Stack s = new Stack();
 while (!empty())
 {
 int num = pop();
 System.out.print (num + "\t");
 s.push(num);
 }
 System.out.println();
 while (!s.empty())
 {
 int num = s.pop();
 push(num);
 }
}
```

א. מדפיסה את הערכים שבמחסנית s1 מזנב המחסנית אל ראשה כל ערך מודפס פעמיים

ב. מדפיסה את הערכים שבמחסנית s1 מראש המחסנית אל זנבה כל ערך מודפס פעמיים

ג. מדפיסה את הערכים שבמחסנית s1 מזנב המחסנית אל ראשה

ד. מדפיסה את הערכים שבמחסנית s1 מראש המחסנית אל זנבה

ה. אף אחת מהתשובות א – ד אינה נכונה

ו. יש יותר מתשובה אחת נכונה לעיל

**שאלה 4.** נניח ש-s היא מחסנית מהמחלקה Stack. בהינתן השיטה what הבאה:

```
public static boolean what (Stack s)
{
 if (s.isEmpty())
 return true;
 int sum = s.pop();
 while (!s.isEmpty())
 {
 int temp = s.pop();
 if (temp != sum)
 return false;
 sum += temp;
 }
 return true;
}
```

להלן נתונות מחסניות שונות - (הערכים במחסנית כתובים משמאל לימין, בשמאל ראש המחסנית):

s1. 1 1 2 4 8 16 32 64

s2. 1 1 2 4 8 16 32 64

s3. 4 4 4 4

s4. 5 10 20

s5. 1 1 2 3 5 8 13 21

s6. 1 1 2 3 5 8 13 21

אם נפעיל את השיטה what לעיל על כל אחת מהמחסניות, על אילו מתוך המחסניות השיטה תחזיר true?

א. s1 ו-s2

ב. s1, s2, s5 ו-s6

ג. רק s2

ד. s2, s4 ו-s6

ה. s2 ו-s4

ו. s2 ו-s6

ז. אף אחת מהתשובות א-ו לעיל אינה נכונה

## השיטות הבאות מתייחסות לשאלות 5 - 6

```

public static int secret(int num)
{
 int temp, dig;
 if (num<10)
 return num;
 else
 {
 temp = secret (num/10);
 dig = num%10;
 if (dig<temp)
 return dig;
 else return temp;
 }
}

public static boolean check(Stack s, int m)
{
 boolean fine;
 int x = s.pop();
 if (secret(x) != m)
 {
 s.push(x);
 return false;
 }
 else
 if (s.isEmpty())
 {
 s.push(x);
 return true;
 }
 else
 {
 fine = check(s, m);
 s.push(x);
 return fine;
 }
}

```

### שאלה 5. איזה מהמשפטים הבאים אינו נכון?

- א. אם נפעיל את השיטה secret על המספר 2222 יוחזר המספר 2
- ב. אם נפעיל את השיטה secret על המספר 1234 יוחזר המספר 1
- ג. אם נפעיל את השיטה secret על מספר שלילי, הערך שיוחזר יהיה בעל משמעות אחרת מזה שיוחזר אם המספר הוא חיובי
- ד. אם נפעיל את השיטה secret פעם על המספר 5 ופעם על המספר 7586 נקבל אותה תוצאה.
- ה. אם נפעיל את השיטה secret פעם על המספר 7345 ופעם על המספר 5437 נקבל אותה תוצאה
- ו. אם נפעיל את השיטה secret פעם על המספר 45 ופעם על המספר 298 נקבל אותה תוצאה
- ז. יש יותר ממשפט אחד לעיל שאינו נכון

## שאלה 6. איזה מהמשפטים הבאים אינו נכון?

- א. אם נפעיל את השיטה check לעיל על המחסנית הבאה (הערכים משמאל לימין. ראש המחסנית משמאל) 248 5535 28 829 והמספר 2 יוחזר הערך false.
- ב. כדי לקבל ערך אחר מהערך שהוחזר בטענה א, מספיק לשנות מספר אחד במחסנית.
- ג. אם נפעיל את השיטה check לעיל על המחסנית הבאה (הערכים משמאל לימין. ראש המחסנית משמאל) 48 64 455 849 והמספר 4 יוחזר הערך true.
- ד. בחר בתשובה זו אם כל המשפטים א – ג נכונים.

## הכתוב להלן מתייחס לשאלות 7 – 8 המופיעות בהמשך.

נתונות שתי השיטות הסטטיות הבאות:

```
public static int check (Queue q, int x)
{
 if (q.isEmpty() || x!= (q.getHead()).getValue())
 return 1;
 x = q.dequeue();
 return 1+ check (q, x);
}

public static int check1(Queue q)
{
 int y;
 if (q.isEmpty())
 y = 0;
 else
 {
 int x = q.dequeue();
 int z = x * check (q, x);
 y = 1 + check1(q);
 q.enqueue(z);
 }
 return y;
}
```

בהינתן התור q הבא (הערכים משמאל לימין. 2 ראש התור, 5 זנב התור)

2 3 1 1 2 2 2 4 4 4 5

**שאלה 7.** הערך המוחזר מהשיטה check1 כשהיא מופעלת על התור q הוא :

א. 11

ב. 5

ג. 6

ד. בחר בתשובה זו אם אף אחת מהתשובות א – ג אינה נכונה.

**שאלה 8.** לאחר הפעלת השיטה check1 על התור q שלעיל, התור יראה כך (משמאל לימין – ראש התור בשמאל וזנבו בימין) :

א. 2 3 1 1 2 2 2 4 4 5

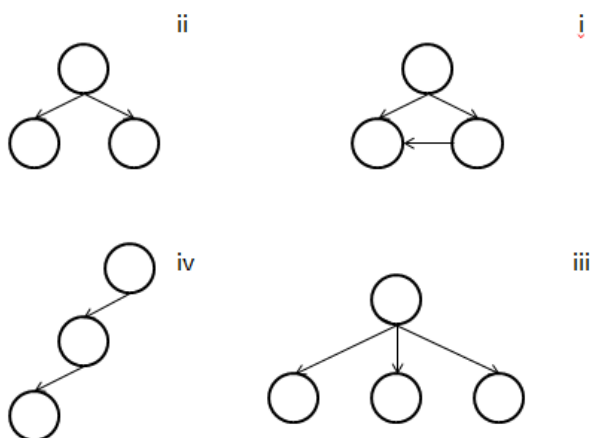
ב. 5 4 4 4 2 2 2 1 1 3 2

ג. 2 3 2 6 12 5

ד. 5 12 6 2 3 2

ה. בחר בתשובה זו אם אף אחת מהתשובות א – ד אינה נכונה.

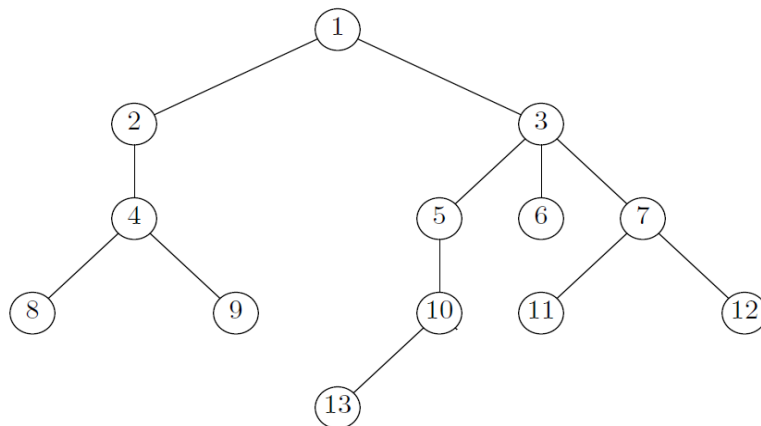
**שאלה 9.** לפניכם ארבעה תרשימים כאשר כל עיגול מייצג צומת מסוג Node. קבעו עבור כל אחד מהם את התיאור המתאים ביותר :



- |            |             |           |             |
|------------|-------------|-----------|-------------|
| א. i לא עץ | ii עץ בינרי | iii עץ    | iv עץ בינרי |
| ב. i עץ    | ii עץ בינרי | iii לא עץ | iv עץ       |
| ג. i לא עץ | ii עץ בינרי | iii לא עץ | iv לא עץ    |
| ד. i עץ    | ii עץ בינרי | iii עץ    | iv עץ       |
- ה. אף אחת מהתשובות א – ד אינה נכונה



**שאלה 10.** נתון עץ ששורשו הוא הצומת 1 :



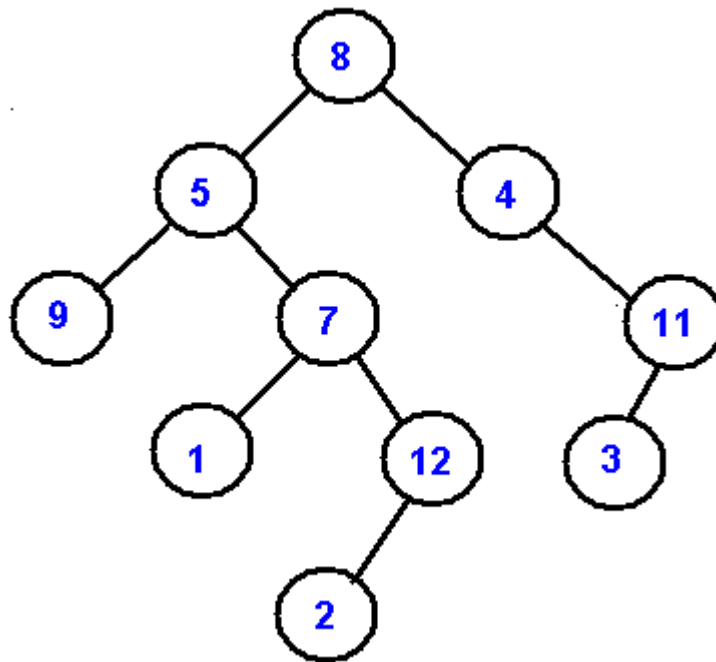
נתונות הטענות הבאות :

1. השורש של העץ הוא 1 :
2. העלים של העץ הם : 13, 12, 11, 9, 8, 6 :
3. הצמתים הפנימיים של העץ הם : 1, 2, 3, 4, 5, 7, 10 :
4. האבא של הצומת 7 הוא 3
5. 12 הוא בן של 3
6. 5 הוא אב קדמון של 9
7. 3 הוא אב קדמון של 13
8. 2 הוא אבא של 8 ו-9

סמנו את התשובה הנכונה :

- א. רק הטענה 6 אינה נכונה
- ב. רק הטענות 1, 2, 3 ו-4 נכונות
- ג. רק הטענות 1, 2, 3, 4 ו-7 נכונות
- ד. כל הטענות 1 – 8 נכונות
- ה. אף אחת מהתשובות א – ד אינה נכונה

שאלה 11. נתון העץ הבינארי הבא



נתונות הטענות הבאות :

1. סדר הביקור בצמתים של העץ לאחר סיור InOrder הוא 9, 5, 1, 7, 2, 12, 8, 4, 3, 11
2. סדר הביקור בצמתים של העץ לאחר סיור InOrder הוא 8, 5, 9, 7, 1, 12, 2, 4, 11, 3
3. סדר הביקור בצמתים של העץ לאחר סיור PostOrder הוא 9, 1, 2, 12, 7, 5, 3, 11, 4, 8
4. סדר הביקור בצמתים של העץ לאחר סיור PostOrder הוא 3, 11, 4, 2, 12, 1, 7, 9, 5, 8
5. סדר הביקור בצמתים של העץ לאחר סיור PreOrder הוא 8, 5, 4, 9, 7, 11, 1, 12, 3, 2
6. סדר הביקור בצמתים של העץ לאחר סיור PreOrder הוא 8, 5, 9, 7, 1, 12, 2, 4, 11, 3

סמנו את התשובה הנכונה :

- א. רק הטענות 1 ו-4 נכונות
- ב. הטענות 1, 3, 6 נכונות
- ג. הטענות 2, 3, 6 נכונות
- ד. הטענות 1, 4, 5 נכונות
- ה. אף תשובה מסעיפים א-ד אינה נכונה

**שאלה 12.** לאחר הפעלת סיור PreOrder ו InOrder על עץ בינארי כלשהו התקבלו התוצאות הבאות (משמאל לימין):

PreOrder : 1 2 3 4 5 6 7 8 9 10

InOrder : 3 4 2 1 5 7 8 6 10 9

אם נפעיל סיור PostOrder על העץ הזה נקבל את התוצאה הבאה.

סמנו את התשובה הנכונה :

א. 10 9 8 7 6 5 4 3 2 1

ב. 9 10 6 8 7 5 1 2 4 3

ג. 1 5 6 9 10 7 8 2 3 4

ד. 4 3 2 8 7 10 9 6 5 1

ה. אי אפשר לדעת מה היה העץ המקורי על סמך תוצאת סיור ה-PreOrder וה- InOrder

ולכן לא ניתן לדעת מה תוצאת סיור ה- PostOrder על העץ.

ו. אפשר לדעת מה היה העץ המקורי, אבל אף אחת מהתשובות א – ד אינה נכונה.

**שאלה 13.** לאחר הפעלת סיור PostOrder על עץ חיפוש בינארי כלשהו התקבלה התוצאה הבאה (משמאל לימין):

PostOrder : 1 4 3 2 5 7 8 9 6 10

אם נפעיל סיור PreOrder על העץ הזה נקבל את התוצאה הבאה.

סמנו את התשובה הנכונה :

א. 7 8 9 4 3 1 2 5 6 10

ב. 10 6 9 8 7 5 2 3 4 1

ג. 7 8 10 9 6 4 5 2 3 1

ד. 10 9 8 7 6 5 4 3 2 1

ה. 10 6 5 2 1 3 4 9 8 7

ו. אי אפשר לדעת מה היה העץ המקורי על סמך תוצאת סיור ה-PostOrder ולכן לא ניתן

לדעת מה תוצאת סיור ה- PreOrder על העץ.

ז. אפשר לדעת מה היה העץ המקורי, אבל אף אחת מהתשובות א – ה אינה נכונה.

### הקטע הבא (ובו קטעי מחלקות) מתייחס לשאלות 14-17

עץ טרנארי (Ternary tree) הוא עץ שיש בו לכל היותר שלושה בנים לצומת: בן שמאלי, בן אמצעי ובן ימני.

נניח שהמחלקה TerNode שלהלן מממשת צומת בעץ טרנארי.

```
public class TerNode
{
 private int _number;
 private TerNode _leftSon, _middleSon, _rightSon;

 public TerNode(int number) {
 _number = number;
 _leftSon = null;
 _middleSon = null;
 _rightSon = null;
 }

 public int getNumber() {return _number;}
 public TerNode getLeftSon() {return _leftSon;}
 public TerNode getMiddleSon() {return _middleSon;}
 public TerNode getRightSon() {return _rightSon;}

 public void setNumber(int number) {_number = number;}
 public void setLeftSon(TerNode node) {_leftSon = node;}
 public void setMiddleSon(TerNode node){_middleSon = node;}
 public void setRightSon(TerNode node){_rightSon = node;}
}
```

המחלקה TernaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ טרנארי.

בין השיטות נתונות השיטות what ו-something הבאות:

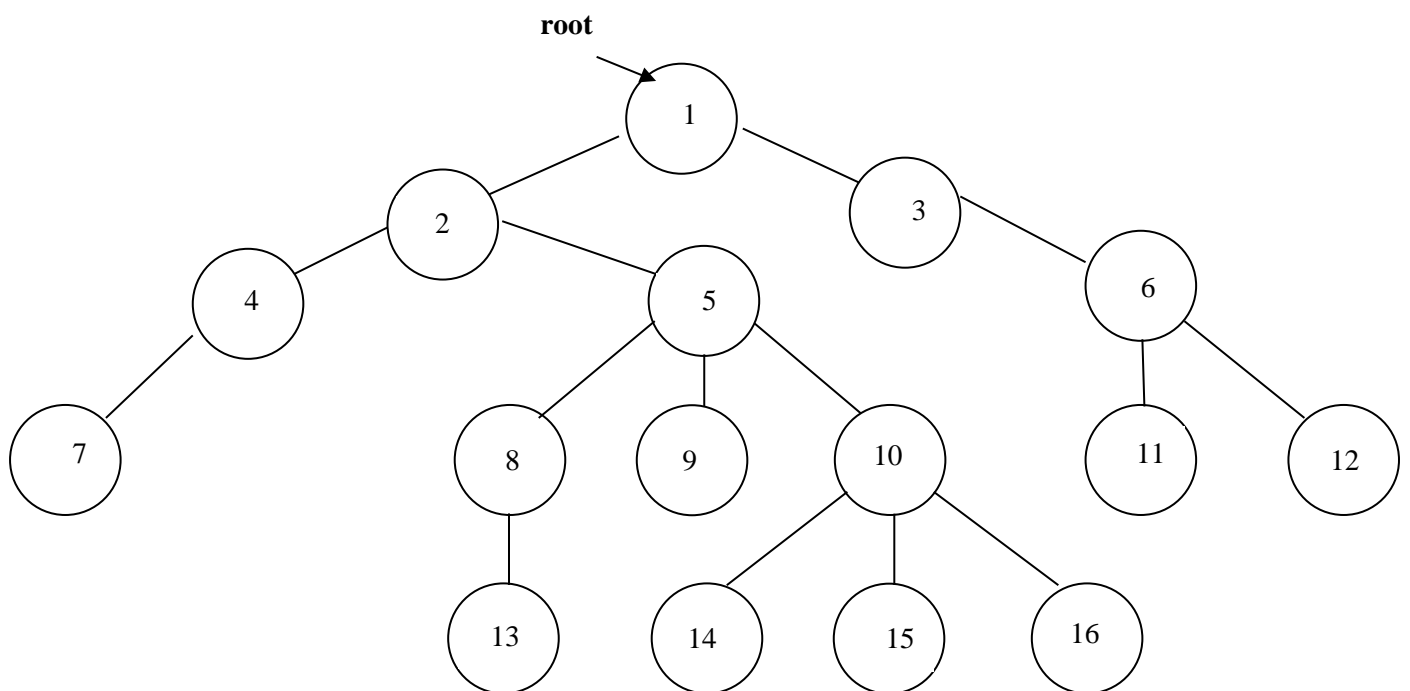
```
public static int what (TerNode t)
{
 if (t == null)
 return 0;
 int temp = 0;
 if (t.getLeftSon() != null) temp++;
 if (t.getMiddleSon() != null) temp++;
 if (t.getRightSon() != null) temp++;
 return temp;
}
```

```

public static boolean something (TerNode t)
{
 if (t == null)
 return true;
 if (what(t) == 3)
 return false;
 return something (t.getLeftSon())
 && something (t.getMiddleSon())
 && something (t.getRightSon());
}

```

נתון העץ הטרינארי הבא, ששורשו הוא root



#### שאלה 14.

- מה מבצעת השיטה what באופן כללי, בהינתן לה צומת t של עץ טרינארי?
- השיטה מחזירה את מספר הבנים המקסימלי של צומת בעץ ששורשו t.
  - השיטה מקבלת צומת t ומחזירה את מספר צאצאיו (כולל בניו הישירים).
  - השיטה מקבלת צומת t ומחזירה את מספר צאצאיו (לא כולל בניו הישירים).
  - השיטה מקבלת צומת t ומחזירה את מספר בניו (הישירים).
  - אף אחת מהתשובות א – ד אינה נכונה.

### שאלה 15.

בהינתן העץ הטרנארי לעיל ששורשו root, איזה ערך תחזיר השיטה something בעקבות הקריאה

`?BinaryTree.something(root)`

- א. 3
- ב. 1
- ג. true
- ד. false

### שאלה 16.

אם נרצה שהשיטה something תחזיר תשובה אחרת מזו שהוחזרה בשאלה 13, אילו שינויים **מינימליים** עלינו לבצע בעץ לעיל? (כלומר, שינויים שיגעו במינימום צמתים)

- א. יש למחוק את הצמתים 9 ו-15 מהעץ
- ב. יש למחוק את הצמתים 8 ו-15 מהעץ
- ג. יש למחוק את הצומת 10 (ובניו) מהעץ
- ד. יש להוסיף צומת כבן של הצומת 6
- ה. אף אחת מהתשובות א – ד אינה נכונה

### שאלה 17.

מה מבצעת השיטה something באופן כללי, בהינתן לה צומת t של עץ טרנארי?

- א. השיטה בודקת מה מספר הבנים של כל צומת בעץ ששורשו t, ומחזירה true אם הוא 3 ו-false אם הוא קטן מ-3.
- ב. השיטה בודקת מה מספר הבנים של כל צומת בעץ ששורשו t, ומחזירה false אם הוא 3 ו-true אם הוא גדול מ-3.
- ג. השיטה בודקת אם הצומת t הוא עלה ומחזירה true אם כן, ו-false אם לא.
- ד. השיטה בודקת אם הצומת t הוא שורש של עץ בינרי ומחזירה true אם כן ו-false אם לא.
- ה. אף אחת מהתשובות א – ד אינה נכונה

## השאלות 18 - 19 להלן מתייחסות לקטע הבא.

נתונות שלוש שיטות המתייחסות למחלקות המייצגות רשימה מקושרת ועץ בינרי. השיטה insert(p,x) מקבלת כפרמטרים מצביע לאיבר p ברשימה ומספר שלם x ומכניסה את המספר x כאיבר אחרי האיבר p. אם p הוא null, x ייכנס לראש הרשימה.

```
public static IntList what1(Node t)
{
 IntList list=new IntList();
 what2(t, list);
 return list;
}

public static void what2(Node t, IntList list)
{
 if(t!=null)
 {
 int x=t.getInfo();
 if (what3(t.getLeft(),x) && what3(t.getRight(),x))
 list.insert(null,x);
 what2(t.getLeft(),list);
 what2(t.getRight(),list);
 }
}

public static boolean what3(Node t, int x)
{
 if(t==null)
 return true;
 if(t.getInfo()==x)
 return false;
 return what3(t.getLeft(), x) && what3(t.getRight(),x);
}
```

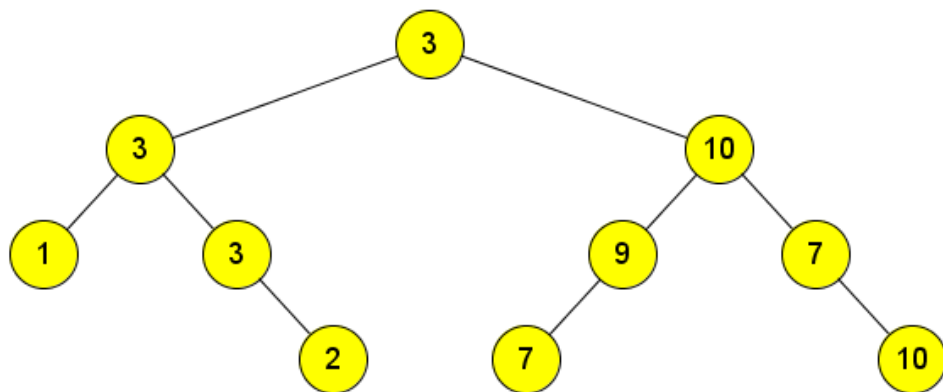
### שאלה 18.

מה מבצעת השיטה what3 לעיל בהינתן לה צומת t ומספר שלם x?

- מחזירה true אם המספר x נמצא בצומת t ו- false אחרת.
- מחזירה false אם המספר x נמצא בצומת t ו- true אחרת.
- מחזירה true אם המספר x נמצא בעץ ששורשו t ו- false אחרת.
- מחזירה false אם המספר x נמצא בעץ ששורשו t ו- true אחרת.
- אף אחת מהתשובות א – ד אינה נכונה

### שאלה 19.

בהינתן העץ t שלהלן ,



מה תהיה תוצאת הזימון `System.out.println(what1(t))`? (בהנחה שהדפסת רשימה מדפיסה את האיברים שבה משמאל לימין (האיבר שעומד בראש הרשימה הוא השמאלי ביותר).

- [10,7,7,9,10,2,3,1,3,3]
- [10,7,7,9,2,3,1]
- [3,3,13,2,10,9,7,7,10]
- [1,3,2,9,7,7,10]
- [1,3,3,2,3,7,9,10,7,10]
- [10,7,9,7,3,2,1]
- אף אחת מהתשובות א – ו אינה נכונה



```
public static boolean secret(Node t)
{
 if (t.getLeft()==null && t.getRight()==null)
 return true;
 if (t.getLeft()==null || t.getRight()==null)
 return false;
 if (t.getLeft().getInfo()>0)
 return false;
 if (t.getRight().getInfo()<0)
 return false;
 return secret(t.getLeft()) && secret (t.getRight());
}
```

נתונות שתי הטענות הבאות:

1. **קיים** עץ  $t$  המכיל רק ערכים חיוביים (גדולים מ-0), כך שהשיטה  $\text{secret}(t)$  תחזיר true.
2. **לא קיים** עץ  $t$  שלא מכיל ערכים חיוביים, כך שפעולה  $\text{secret}(t)$  תחזיר true.

הנחות לשתי הטענות:

1. ב- $t$  יש **לפחות** שני צמתים.
2. צומת לא יכול להכיל את הערך 0.

סמנו את התשובה הנכונה:

- א. שתי הטענות לא נכונות.
- ב. טענה 1 נכונה וטענה 2 לא נכונה.
- ג. טענה 1 לא נכונה וטענה 2 נכונה.
- ד. שתי הטענות נכונות.

**את התשובות לממ"ח יש לשלוח באמצעות מערכת שאילת"א**

**בכתובת <http://www.openu.ac.il/sheilta/>**

**הממ"ח נבדק בצורה ממוחשבת. אין לשלוח את פתרון הממ"ח**

**למנחה!**

**שימו לב שהממ"ח אכן נקלט בשאילתא,**

**ושמרו את האסמכתא שקיבלתם כתוצאה מהשליחה.**

**אם לא קיבלתם אישור – כאילו לא שלחתם!**

## **בהצלחה גם בבחינות הגמר**