

האוניברסיטה הפתוחה

20441

מבוא למדעי המחשב

ושפת Java

חוברת הקורס – אביב 2017ב

כתבה: תמר וילנר

מרץ 2017 – סמסטר אביב – תשע"ז

פנימי – לא להפצה.

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

תוכן העניינים

א	אל הסטודנט
ב	1. לוח זמנים ופעילויות
ד	2. תיאור המטלות
ד	2.1 מבנה המטלות
ו	2.2 חומר הלימוד הדרוש לפתרון המטלות
ו	2.3 ניקוד המטלות
ח	3. התנאים לקבלת נקודות זכות בקורס
1	ממ"ח 01
9	ממ"ן 11
15	ממ"ן 12
25	ממ"ן 13
31	ממ"ח 02
43	ממ"ן 14
51	ממ"ן 15
57	ממ"ח 03

אל הסטודנט

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס "מבוא למדעי המחשב ושפת Java".

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

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, אותם מפרסם/מת מרכז/ת ההוראה. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:

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

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, תוכל לפנות אל תמר וילנר בימי ד' בין השעות 11:00-13:00 בטלפון: 02-6773323. ניתן לפנות גם בדואר אלקטרוני tami@openu.ac.il פגישה יש לתאם מראש.

לתשומת לב הסטודנטים הלומדים בחו"ל:

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

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

אני מאחלת לך לימוד פורה ומהנה.

בברכה,

תמר וילנר

מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (20441 / ב'2017)

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח ממ"ן (למנחה)
1	24.3.2017-19.3.2017	יחידות 1 ו-2 – הקדמה ויסודות השפה בספר – פרקים 1 ו-2	מפגש 1	
2	31.3.2017-26.3.2017	יחידה 2 – יסודות השפה בספר – פרקים 2 ו-5		ממ"ח 01 1.4.2017
3	7.4.2017-2.4.2017	יחידה 3 – שימוש במחלקות נתונות בספר – פרק 3		ממ"ן 11 8.4.2017
4	14.4.2017-9.4.2017 (ב ערב פסח) (ג-ו פסח)	יחידה 4 – כתיבת מחלקות בספר – פרק 4	מפגש 2	
5	21.4.2017-16.4.2017 (א-ב פסח)	יחידה 4 – כתיבת מחלקות בספר – פרק 4		
6	28.4.2017-23.4.2017 (ב יום הזכרון לשואה)	יחידה 5 – לולאות בספר – פרקים 5 ו-6	מפגש 3	ממ"ן 12 29.4.2017
7	5.5.2017-30.4.2017 (ב יום הזיכרון) (ג יום העצמאות)	יחידה 6 – מערכים בספר – פרק 8		
8	12.5.2017-7.5.2017	יחידה 7 – ירושה בספר – פרק 9	מפגש 4	ממ"ן 13 13.5.2017
9	19.5.2017-14.5.2017 (א ל"ג בעומר)	יחידה 8 – פולימורפיזם בספר – פרק 10		

* 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)
10	26.5.2017-21.5.2017 (ג יום ירושלים)	יחידה 9 – יעילות בספר – פרק 10	מפגש 5	ממ"ח 02 27.5.2017
11	2.6.2017-28.5.2017 (ד שבועות)	יחידות 9 ו-10 – יעילות ורקורסיה בספר – פרק 12	מפגש 6	
12	9.6.2017-4.6.2017	יחידה 10 – רקורסיה בספר – פרק 12	מפגש 7	
13	16.6.2017-11.6.2017	יחידה 11 – רשימות בספר – פרק 13	מפגש 8	ממ"ן 14 17.6.2017
14	23.6.2017-18.6.2017	יחידה 12 – מחסנית, תור ועצים בינריים בספר – פרק 12		ממ"ן 15 24.6.2017
15	30.6.2017-25.6.2017	יחידה 20 וחזרה	מפגש 9	ממ"ח 03 1.7.2017

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 שאילת"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 בקובצי 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 לפתרונה
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 30 נקודות. חובה להגיש מטלות במשקל מינימלי של 20 נקודות לפחות.

להלן פירוט הניקוד לכל מטלה:

ממ"ן/ממ"ח	ניקוד
01	2
11	3
12	4
13	5
02	3
14	5
15	5
03	3

כדי לעבור את הקורס, צריך (בין היתר) להגיש מטלות במשקל של 20 נקודות לפחות. סטודנט שמגיש את כל המטלות, משקל המטלות שלו הוא 30 נקודות. בהתאם למשקל המטלות, נקבע אחוז הציון של בחינת הגמר בציון הסופי.

חישוב אחוז הציון של הבחינה – אם הוגשו מטלות במשקל 20 נקודות, הבחינה שווה 80% מהציון הסופי של הקורס. אם המטלות שהוגשו הן במשקל 30, הבחינה שווה 70%. וכך גם לכל המספרים ביניהם. לדוגמא, אם הוגשו מטלות 01, 12, 13, 02, 14, 03 המשקל שלהן הוא:

$22 = 3 + 5 + 3 + 5 + 4 + 2$ (ללא קשר לציון שהתקבל בכל מטלה), ואז משקל הבחינה בציון הסופי הוא 78%.

איך מחושב ציון הקורס ?

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

לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי.

זאת בתנאי שמטלות אלה אינן חלק מדרישות החובה בקורס ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.

זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

לדוגמא, אם הציונים במטלות ובבחינה הם אלו:

מטלה	משקל	ציון
01	2	70
12	4	68
13	5	80
02	3	100
14	5	90
03	3	85
בחינת הגמר	78	73

שימו לב שבמטלות 01, 12 הציון נמוך מציון הבחינה. אם נוריד את שתייהן, יישארו 16 נקודות. אם נוריד את מטלה 12 יישארו 18 נקודות. לכן אפשר להוריד רק את מטלה 01 (כלומר להניח כאילו לא הוגשה כלל).

חישוב הציון ייעשה, אפוא, כך:

$$75 \rightarrow 75.17 = (68 \times 4 + 80 \times 5 + 100 \times 3 + 90 \times 5 + 85 \times 3 + 73 \times 80) / 100$$

כלומר, הציון הסופי הוא 75.

כיוון שחובה להגיש מטלות במשקל של לפחות 20 נקודות, אם מישהו לא הגיש את המשקל הדרוש, הוא יכול לבקש שנוזן 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 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 צריך מכתב/אישור המנמק את סיבת האיחור.

נזכיר שוב:

לבחינת הגמר רשאי לגשת רק סטודנט שצבר 20 נקודות לפחות בהגשת המטלות.

3. התנאים לקבלת נקודות זכות בקורס

- א) צבירת משקל של 20 נקודות לפחות במטלות.
- ב) ציון של 60 נקודות לפחות בבחינת הגמר.
- ג) 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: 2 נקודות

סמסטר: 2017 מועד אחרון להגשה: 1.4.2017

(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.

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 תגלה, 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 ב**פירוט 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. סיבית (bit) היא...

- א. מתקן אלקטרוני שמשתמשים בו במחשבים
- ב. מילה נרדפת לבית (byte)
- ג. היא 2
- ד. ספרה בינרית - או אחד או אפס
- ה. התשובות לעיל (א-ד) שגויות.

שאלה 2. 3K משמעותו שלושת אלפים בתים. איך אי אפשר לסמן מאתיים מיליון בתים?

א. 200M

ב. 200 megabytes

ג. 200 GB

ד. 200 MB

שאלה 3. יש לכתוב תכנית שמכילה מחלקה בשם Shalom ומדפיסה את המשפט Shalom, world על המסך. אילו מהתכניות הבאות תבצע את הדרוש?

```
א public class shalom {  
    public static void main (String [] args) {  
        System.out.print ("Shalom, world");  
    }  
}
```

```
ב public class Shalom {  
    public static void main (String [] args) {  
        System.out.print ("Shalom, world");  
    }  
}
```

```
ג public static void main (String [] args) {  
    System.out.print ("Shalom, world");  
}
```

```
ד public class Shalom {  
    public static void main (String [] args) {  
        system.out.print ("Shalom, world");  
    }  
}
```

```
ה public class Shalom {  
    public static void main (String [] args) {  
        System.out.print ("Hello, world");  
    }  
}
```

שאלה 4. הטקסט הכתוב בהערה 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 5. אלו מבין השורות הבאות מכילה הערה תקנית (גם אם לא מדויקת)?

- א. `int twoPi = 2*3.14159; /* holds the value of two times pi */`
- ב. `int twoPi = 3.14159; / / *holds the value of 6 /*`
- ג. `double twoPi = 2*3.14159;/**/ holds the value of two time pi`
`[comment] //`
- ד. `int twoPi = 3.14159; /* holds the value of two times pi */`

שאלה 6. אלו מבין המזהים (identifiers) הבאים אינו חוקי בשפת Java?

- א. `5Abc`
- ב. `_32`
- ג. `_`
- ד. `Love`
- ה. `Number72`

שאלה 7. בתכנות מונחה-עצמים (Object-Oriented programming) מחלקה היא:

- א. המושג שמשמשים בו עבור תכנית
- ב. קבוצת משתמשים שיש להם אובייקטים משותפים.
- ג. פרוצדורה לחישוב בתוך תכנית
- ד. מודל או תבנית שממנו נוצרים אובייקטים.

שאלה 8. מה מהבאים הוא מחרוזת תווים שמכילה בדיוק 4 סימני שאלה?

- א. `????`
- ב. `System.out.println ("????");`
- ג. `String "????"`
- ד. `"????"`
- ה. `'????'`

שאלה 9. מה מהבאים הוא מחרוזת תווים שמכילה בדיוק תו אחד בודד (כל תו אפשרי)?

- א. "v"
- ב. v
- ג. 'v'
- ד. String "v"
- ה. String v

שאלה 10. מה מבין קטעי הקוד הבאים ידפיס על המסך את המחרוזת Cohn, Arthur ?

- א. System.out.print ("Cohen, Arthur");
- ב. System.out.print ("Cohn, Arthur");
- ג. System.Out.print ("Cohn, Arthur");
- ד. System.out.print ("Arthur, Cohn");
- ה. System.out.print ("Cohn Arthur");

שאלה 11. לפני שמשתמשים במשתנה (variable), הוא צריך להיות:

- א. מוצהר (declared)
- ב. מיובא (imported)
- ג. מחושב (evaluated)
- ד. בעל ערך (assigned)

שאלה 12. איזו מהפקודות הבאות מזהירה על משתנה שלם שנקרא NumOfCentimeters?

- א. static int NumOfCentimeters
- ב. int NumOfCentimeters
- ג. int NumOfCentimeters;
- ד. private int NumOfCentimeter =6;
- ה. int NumOfSentimeters;

שאלה 13. איזו מהפקודות הבאות **אינה** מזהירה על משתנה שלם בשם area ומאתחלת אותו לערך 10?

- א. `int area;`
`{`
`area = 10;`
`}`
- ב. `long area = (10);`
- ג. `int area = 10;`
- ד. `int area = (int)10;`
- ה. `int Area = 10;`

שאלה 14. אם נתונה הפקודה שלהלן בתכנית, `System.out.println (/* 5 + 3 */ - 9 + 6/*8*//2); /*+4*/ -10;` מה יהיה הפלט שיודפס בעקבות הרצתה

- א. 3
- ב. -7
- ג. 5
- ד. -6
- ה. -8

שאלה 15. מה מהבאים הוא תו (char) שמייצג את הספרה 2?

- א. `"2"`
- ב. `2`
- ג. `String "two"`
- ד. `char 2`
- ה. `'2'`
- ו. `(char) 2`

שאלה 16. מה מהביטויים החשבוניים הבאים **אינו** מייצג את הממוצע של הערכים 14 ו-30?

- א. 22
- ב. `14+30/2`
- ג. `(int) (14+30)/(2)`
- ד. `44/2`
- ה. `14/2+30/2`

שאלה 17. נתון קטע הקוד הבא: `"int v = 10; System.out.println(v--);"` מה

הערך שיודפס ומה הערך של המשתנה `v` בסופו של הקטע?

- א. הערך שיודפס יהיה 9 והערך של `v` יהיה 10
- ב. הערך שיודפס יהיה 10 והערך של `v` יהיה 10
- ג. הערך שיודפס יהיה 10 והערך של `v` יהיה 9
- ד. הערך שיודפס יהיה 9 והערך של `v` יהיה 9
- ה. אי אפשר לדעת מה הערך שיודפס והערך של `v` יהיה 10

שאלה 18. 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.print("first is " + f + " second = " + s + "\n");`
- ג. `System.out.println("first is" + f + "second=" + s);`
- ד. `System.out.println(first is + " f " + second = + " s ");`

שאלה 19. בהנחה שנתונים שלושה משתנים שלמים `num1`, `num2`, `num3`, שלושתם מוצהרים

ומאותחלים, וכן נתון משתנה **ממשי** 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 20. עיקר תפקידו של מהדר (compiler) :

- א. מתרגם קוד מוכן להרצה לקוד בשפת מכונה
- ב. בודק את הלוגיקה של התכנית
- ג. בודק את התחביר של התכנית
- ד. מתחזק אוסף של תכניות.
- ה. מתרגם קוד מקור לקוד מוכן להרצה

השאלות הבאות חשובות לנו מאוד 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 מאוד לפתרון
- ב. רמה סבירה
- ג. קשה לפתרון
- ד. קשה מאוד 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

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

מספר השאלות: 2 משקל המטלה: 3 נקודות

סמסטר: 2017 מועד אחרון להגשה: 8.4.2017

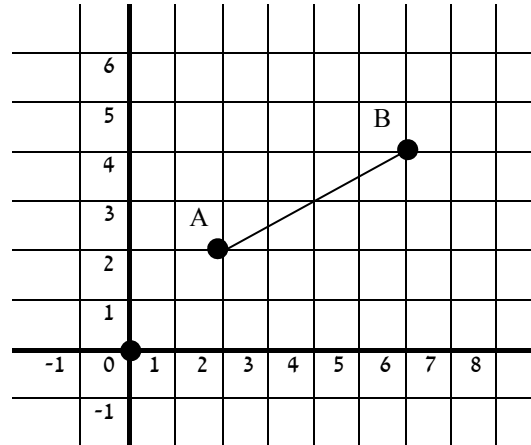
(ת)

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 לחישוב אורך של ישר במישור.
התכנית תקרא מהקלט שהכניס המשתמש ארבעה מספרים שלמים כאשר כל זוג מייצג קואורדינטות של נקודה במישור.



הקואורדינטות של הנקודות A ו- B בתרשים לעיל הן

$$A = (2, 2), B = (6, 4)$$

הקואורדינטה השמאלית בסוגריים היא x והימנית היא y.
לאחר הקריאה, התכנית צריכה לחשב את אורכו של הישר הנמתח בין שתי הנקודות, ולהדפיס את האורך.
ההדפסה תהיה בפורמט הבא :
אם הקלט הוא (הנקודות A ו- B שלעיל) :

2 2 6 4

אז הפלט צריך להיות כזה :

The length of the line between the points (2,2) and (6,4) is 4.47213595499958.

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 בספר בעמוד 114, שם מובאות חלק מהשיטות.
הסברים על המחלקה והשימוש בה אפשר למצוא באתר הקורס בספרית ה"קבצים להורדה",
בתת-ספריה "מדריכי עזר".

כדי 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 $\text{Math.sqrt}(x)$ כאשר במקום הפרמטר x כותבים את הביטוי שממנו רוצים להוציא שורש ריבועי. הפרמטר x של השיטה הזו יכול להיות מטיפוס שלם (int) או ממשי (double). השיטה מחזירה מספר ממשי (גם אם השורש הריבועי של x הוא מספר שלם).

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

התכנית שכתבתם צריכה להיות במחלקה בשם Line.

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

```
import java.util.Scanner;
public class Line
{
    public static void main (String [] args)
    {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Please enter 4 integers ");
        System.out.println ("Please enter x1:");
        int x1 = scan.nextInt();
        System.out.println ("Please enter y1:");
        int y1 = scan.nextInt();
        // כאן עליכם להמשיך...
    } // end of method main
} //end of class Line
```

בשאלה זו אתם יכולים להניח שהקלט שניתן הוא תקין ושהוכנסו מספרים שלמים חיוביים (כולל אפסים). אין צורך לבדוק זאת.

שאלה 2 - להרצה (60%)

עליכם לכתוב תכנית שתקלוט מהמשתמש קדקודים של שני משולשים, ותדפיס הודעה אם המשולשים האלו חופפים (congruent) או שלא.

מתוך ויקיפדיה:

משולשים חופפים הם זוג משולשים שניתן להזיז, לסובב או לשקף אותם כך שהם יתלכדו זה עם זה, כלומר שלוש הצלעות שלהם ושלוש הזוויות שלהם שוות בהתאמה. אינטואיטיבית, שני משולשים חופפים הם בעצם שני עותקים שונים של אותו משולש.

אחד ממשפטי החפיפה: שני משולשים השווים זה לזה באורכי צלעותיהם הם חופפים ("צלע-צלע-צלע")

עליכם לכתוב תכנית שקולטת מהמשתמש שישה זוגות של מספרים ממשיים. שלושת הזוגות הראשונים הם הקואורדינטות של שלוש הנקודות המהוות את קדקודי המשולש הראשון, ושלושת הזוגות השניים הם הקואורדינטות של שלוש הנקודות המהוות את קדקודי המשולש השני. התכנית צריכה לחשב את אורכי הצלעות של שני המשולשים, ואז לבדוק אם יש חפיפה בין המשולשים או שאין.

ההדפסה של הפלט חייבת להיות בדיוק לפי הפורמט הבא:

The first triangle is (x11, y11) (x12, y12) (x13, y13).

Its lengths are a1, b1, c1.

The second triangle is (x21, y21) (x22, y22) (x23, y23).

Its lengths are a2, b2, c2.

ואז במקרה שהמשולשים חופפים יודפס:

The triangles are congruent.

או במקרה שהמשולשים לא חופפים:

The triangles are not congruent.

הערה: מה שכתוב כאן בגופן נטוי לא צריך להיות נטוי בפלט שלכם. כאן כתבנו זאת רק כדי להדגיש את המשתנים.

הנחות:

- אתם יכולים להניח שהמשתמש אכן הכניס שישה זוגות של מספרים ממשיים ולא אותיות או תווים אחרים.
- אתם יכולים להניח שהמספרים שהוכנסו אכן מהווים קדקודים של שני משולשים.
- אתם יכולים להניח שששת המספרים הראשונים הם קדקודים של משולש אחד והשישה הבאים הם קדקודים של משולש שני. אין ערבוב.

התכנית שכתבתם צריכה להיות במחלקה בשם Congruent.

בממ"ן זה אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Line ו- Congruent), והכל בשיטה main. אין להוסיף שיטות נוספות. כמו כן, אסור להשתמש בלולאות במטלה זו.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
 2. הפתרון לשאלה 1 כולל את הקובץ Line.java.
 3. הפתרון לשאלה 2 כולל את הקובץ Congruent.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: 2017 מועד אחרון להגשה: 29.4.2017

(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

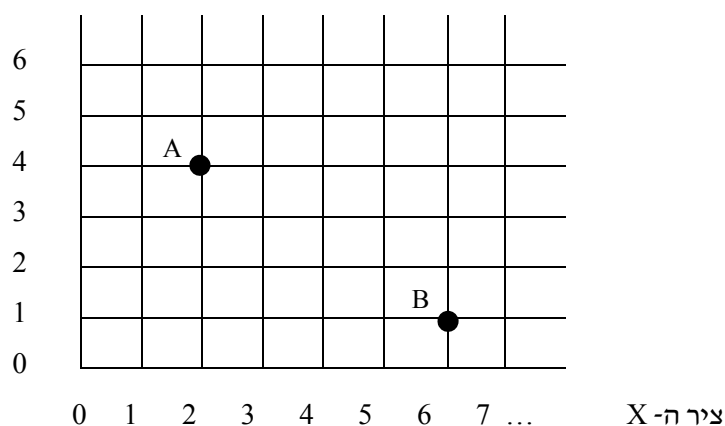
בהרצאות הקורס של ד"ר אמיר גורן, הוגדרה המחלקה Point שמייצגת נקודה במישור, לפי מערכת הצירים הקרטזית (Cartesian system) -

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 $A = (2,4)$ ו- $B = (6,1)$ במרחב:

ציר ה-`Y`



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

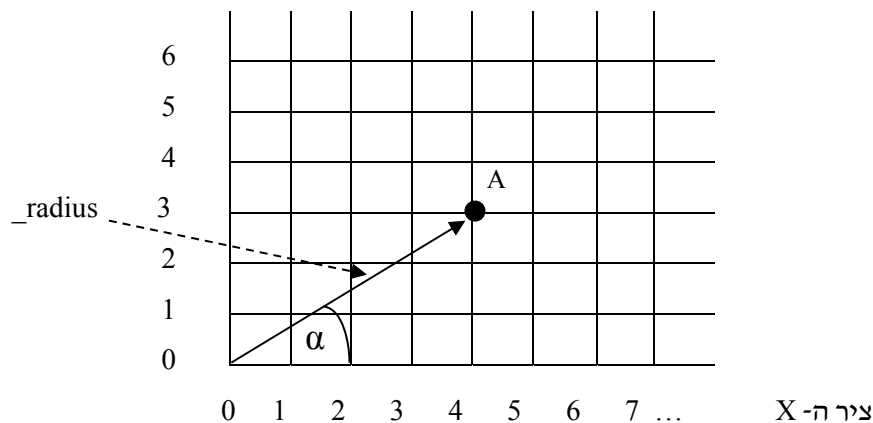
המחלקה Point תייצג נקודה במישור ברביע הראשון בלבד.

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

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

לדוגמא, הנה מסומנת הנקודה A (שנמצאת בקואורדינטות (4,3) במרחב: כאן אורך הוקטור שמחבר את הנקודה (0,0) עם (4,3) הוא 5.0 וערכה של הזווית α (alpha) הוא 36.87 מעלות (וברדיאנים - 0.64)

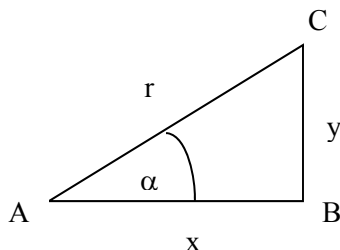
ציר ה-Y



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).



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

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

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

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

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

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

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.

כאשר נתונים אורכי הצלעות x ו- y , אפשר לחשב את אורך הצלע r (היתר במשולש ישר-

$$\text{זווית) על-ידי שימוש במשפט פיתגורס } r = \sqrt{x^2 + y^2}$$

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).

בהמרה של ערך ממשי לשלם השתמש בפעולת עיגול `Math.round(x)` המקבלת מספר ממשי `x` ומחזירה מספר שלם לפי כללי העיגול המקובלים.

במחלקה `Math` תוכלו למצוא גם את השיטות `sin`, `cos`, `tan`, `atan` (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:

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

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

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(int x, int 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**:
 - `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. אם num הוא מספר שלילי, הערך של x לא משתנה.
 - `void setY (int 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) . כלומר, להדפיס את הנקודה לפי מערכת הצירים הקרטזית. 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.
- `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. (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.
- `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.
- `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 (int dx, int 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.

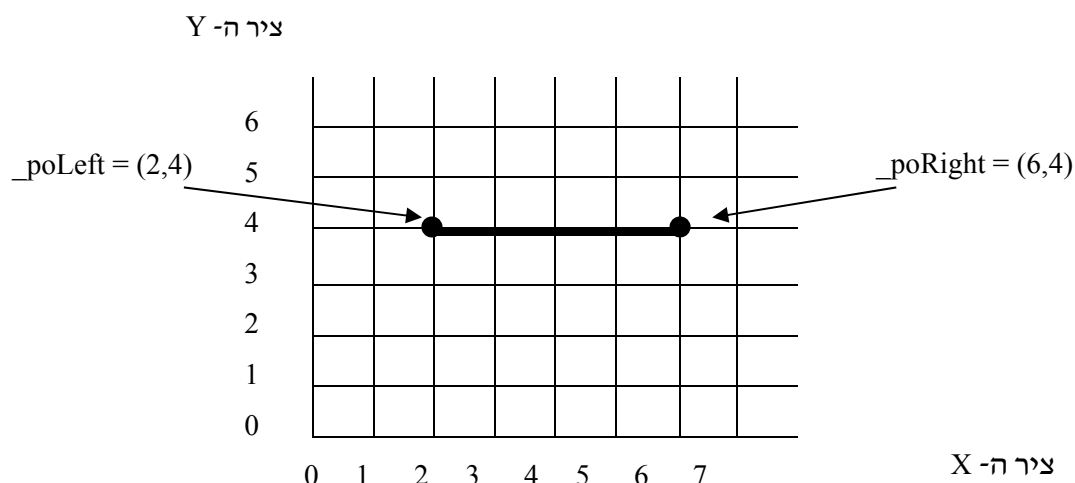
שאלה 2 - 40 נקודות

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 = (2,4)` ו- `_poRight = (6,4)` 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 לשנות 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)`

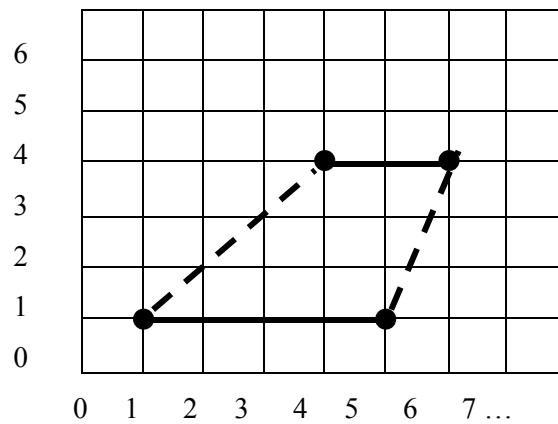
בנוסף הוגדרו במחלקה השיטות:

- 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.
- `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.

- `other (Segment1 other) public double overlap` - 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.
- `other (Segment1 other) public double trapezePerimeter` - שיטה המקבלת כפרמטר מקטע `other` ומחזירה את היקף הטרפז הכלוא בין שני המקטעים. לדוגמא, באיור הבא,

ציר ה-Y



ציר ה-X

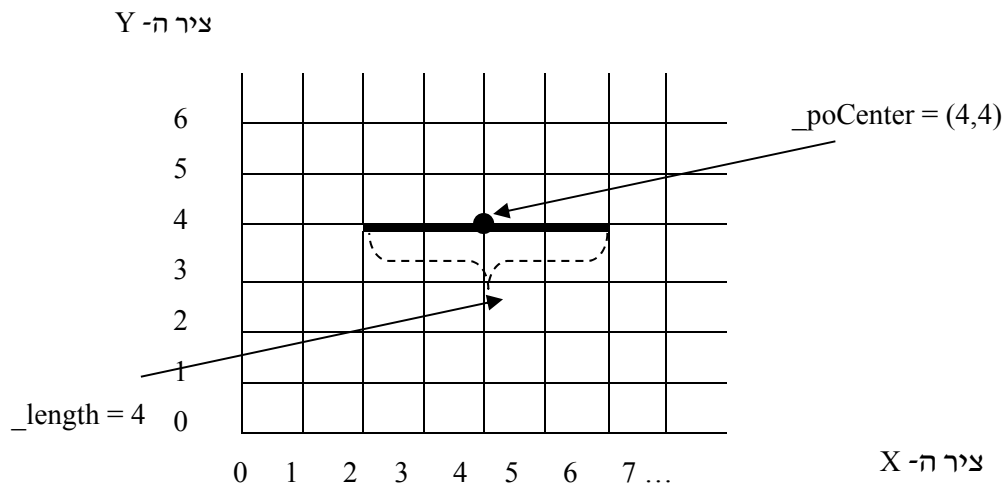
אורך החפיפה בין המקטעים הוא 1 (הקטע בין הקואורדינטה 4 לקואורדינטה 5 על ציר ה-X). הטרפז הכלוא בין שני המקטעים הוא זה המסומן באיור.

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 (2,4) - השמאלית ו- (6, 4) – הימנית ייוצג במחלקה Segment2 על-ידי הנקודה המרכזית (4, 4) והאורך 4.



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.

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. ארזו 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: 2017 מועד אחרון להגשה: 13.5.2017

(ת)

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 הקדקודים (vertices) של המצולע לפי סדר הופעתם במצולע. אין חשיבות מי הקדקוד הראשון. כל קדקוד מיוצג על-ידי נקודה במישור. התכונות במחלקה הן:

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

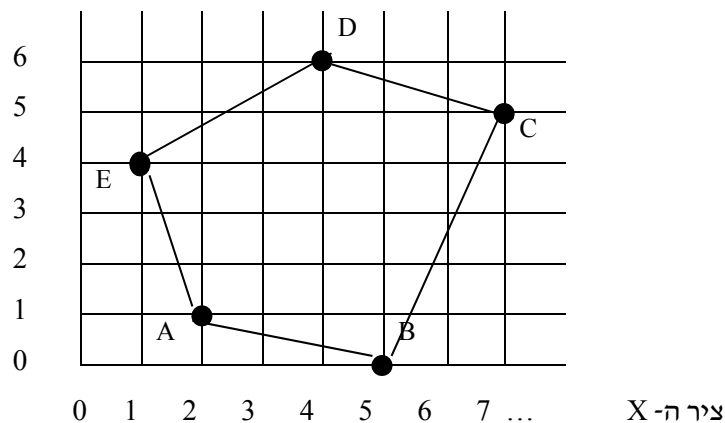
כמו כן קיים קבוע מספרי המציין את המספר המקסימלי של הקדקודים במצולע - 10.

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

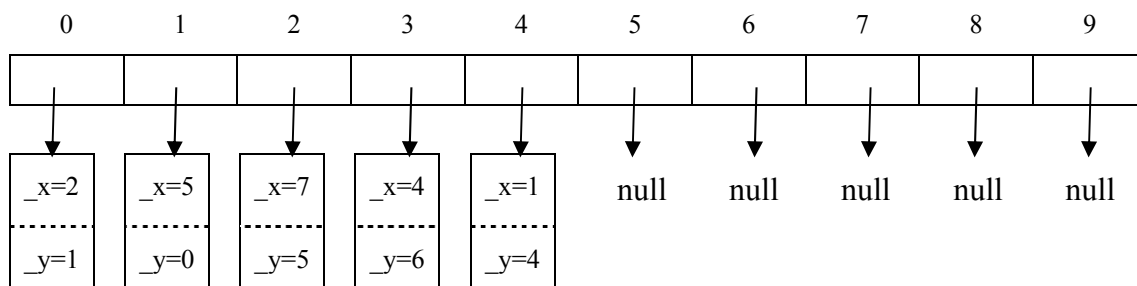
הקדקודים (כלומר האובייקטים מהמחלקה Point) נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

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

ציר ה-Y



המערך המייצג את הקדקודים 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 קדקוד למצולע. היא מקבלת כפרמטרים שני שלמים `x` ו-`y` שמייצגים את הקואורדינטות של הקדקוד הנוסף, ומכניסה נקודה עם תכונות אלו למערך הקדקודים במקום הראשון האפשרי במערך. השיטה תחזיר `true` אם ההוספה התבצעה כראוי, כלומר נמצא מקום במערך עבור הקדקוד הנוסף ו-`false` אם המערך כבר מלא.
- ניתן להניח את ההנחות הבאות (כלומר, אין צורך לבדוק זאת):
- הקדקוד החדש שנוסף לא מקלקל את היות המצולע מצולע קמור.
 - הקדקוד אינו קיים עדיין במצולע.

- הוספת הקדקודים על ידי השיטה `addVertex` נעשית תמיד לפי סדר הופעתם במצולע.
- 4. שיטה `highestVertex` המחזירה העתק של את הקדקוד שנמצא הכי גבוה במצולע. אם יש יותר מאחד בגובה הגבוה ביותר, היא מחזירה את הראשון בו נתקלה. אם אין קדקודים 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 הקדקודים.
The polygon has 5 vertices:
((2,1), (5,0), (7,5), (4,6), (1,4))
אם אין קדקודים 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 הקדקודים הוא 2 יוחזר אורך הקטע (לא הלוך וחזור). אם מספר הקדקודים 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-B-C, A-C-D, A-D-E.
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, c הם אורכי שלוש 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 הקדקודים קטן מ-3 יוחזר 0.
- 8. שיטה בוליאנית `isBigger` 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 שהקדקוד הראשון נמצא במיקום 0 במערך.
- 10. שיטה `getNextVertex`, המקבלת נקודה כפרמטר, ומחזירה העתק של **הנקודה** המייצגת את הקדקוד הבא במצולע. אם הנקודה שהתקבלה אינה קדקוד במצולע, השיטה תחזיר `null`. אם הנקודה היא האיבר האחרון במערך, תוחזר העתק של הנקודה הראשונה. אם הנקודה היא הנקודה היחידה במערך תוחזר העתק של הנקודה עצמה.

11. שיטה `getBoundingBox` המחזירה את המלבן (כפוליגון) (המקביל לצירים) החוסם את המצולע. אם מספר הקדקודים קטן מ-3 יוחזר `null`. הקדקוד הראשון במערך המייצג את המלבן צריך להיות הקדקוד השמאלי התחתון, ושאר הקדקודים יופיעו בסדר הפוך מכיוון השעון (כלומר הקדקוד 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(int x, int y)</code>	שיטה שמוסיפה קדקוד למצולע
<code>public Point highestVertex()</code>	שיטה שמחזירה העתק של את הקדקוד 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 הקדקוד הבא במצולע
<code>public Polygon getBoundingBox()</code>	שיטה שמחזירה את המלבן (כפוליגון) החוסם את המצולע

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 והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלה 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. 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: 3 נקודות

סמסטר: 2017 מועד אחרון להגשה: 27.5.2017

(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!

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)`

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)`

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 שירשת 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).

שאלה 6

אילו מבין המשפטים הבאים אינו נכון?

א. שיטות סטטיות יכולות לגשת למשתנים סטטיים ולשיטות סטטיות.

ב. שיטות לא-סטטיות יכולות לגשת למשתנים סטטיים.

ג. שיטות סטטיות יכולות לגשת למשתנים לא סטטיים ולשיטות לא סטטיות.

ד. שיטות לא סטטיות יכולות לגשת לשיטות סטטיות.

שאלה 7

היינו רוצים להגדיר שמאפיין של מחלקה יהיה נגיש בכל מחלקה יורשת, ללא תלות בחבילה בה נמצאת המחלקה. אילו הרשאות גישה ישיגו מטרה זו?

א. package access (default access) ו protected בלבד

ב. package access (default access) בלבד

ג. public בלבד

ד. private ו protected בלבד

ה. public ו protected בלבד

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

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

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 10

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

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 11

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

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 12

בעקבות הפעלת השורה `System.out.println(b1.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 13

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

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 14

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

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 15

מהו הסדר הנכון של הרשאות גישה – מהמצומצם (משמאל) למתירני ביותר (מימין)?

- א. private, package visibility (default access), protected, public
- ב. private, protected, package visibility (default access), public
- ג. private, public, protected, package visibility (default access)
- ד. private, protected, public, package visibility (default access)

שאלה 16

במחלקה A הוגדרה ומומשה שיטה f(). גם במחלקה B שירשת מהמחלקה A הגדירו ומימשו שיטה בשם f(). אפשר להניח ששתי השיטות לא מקבלות פרמטרים ושיש להן את אותו ערך החזר. באיזה מקרה המחלקה B לא תעבור קומפילציה.

- א. אם הרשאת הגישה לשיטה f שבמחלקה A היא public והרשאת הגישה לשיטה f שבמחלקה B היא public.
- ב. אם הרשאת הגישה לשיטה f שבמחלקה A היא private והרשאת הגישה לשיטה f שבמחלקה B היא protected.
- ג. אם הרשאת הגישה לשיטה f שבמחלקה A היא protected והרשאת הגישה לשיטה f שבמחלקה B היא private.
- ד. אם הרשאת הגישה לשיטה f שבמחלקה A היא private והרשאת הגישה לשיטה f שבמחלקה B היא private.

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.

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.

שאלה 19

```
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.

שאלה 20

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

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: 2017 מועד אחרון להגשה: 17.6.2017

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. חלק מהשאלות הן לתרגול בלבד ולא להגשה. אנו ממליצים מאד 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)

נתאר את בעיית מציאת "בור" במערך דו-ממדי ריבועי:

קלט: מערך דו-ממדי ריבועי בגודל $n \times n$ המלא באפסים ואחדים בלבד.

נגדיר ש- k הוא בור (sink) אם בשורה ה- k ית כל הערכים הם 0, ובעמודה ה- k ית כל

הערכים הם 1 (חוץ מהאיבר $[k][k]$ עצמו שהוא 0).

פלט: האם קיים מספר k המהווה בור במערך? אם כן, יש להחזיר את ערכו אחרת יש להחזיר -1.

לדוגמא: במערך A הוא "בור", ובמערך B אין בור.

B						A					
0	1	0	0	0	1	0	1	0	1	1	0
1	0	0	1	1	1	1	0	1	1	0	0
0	0	0	0	0	0	0	0	1	0	0	1
1	1	1	1	1	1	0	0	0	0	0	0
0	1	0	1	0	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1	0	1	1	1

כתבו שיטה יעילה הפותרת את הבעיה. השיטה תחזיר את המספר k המהווה בור במערך, אם קיים אחד כזה, ו-1 אם לא קיים בור במערך. כתבו והסבירו מה סיבוכיות השיטה שכתבתם.

חתימת השיטה תהיה :

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

שימו לב,

השיטה צריכה להיות יעילה ככל הניתן.

שיטה שתעבוד בסיבוכיות גבוהה מזו הנדרשת (במקום או בזמן) לא תקבל את מירב הנקודות. פתרון נכון שיהיה בסיבוכיות $O(n^2)$, יזכה את כותבו ב- 10 נקודות בלבד.

אל תשכחו לתעד את מה שכתבתם!

שאלה 2 - 25 נקודות (להגשה)

נתונה מטריצה mat מגודל $n \times n$ המיוצגת על ידי מערך דו מימדי. המטריצה מכילה מספרים שלמים. אנו מניחים כי $n = 2^k$, עבור k טבעי כלשהו. נאמר כי המטריצה ממוינת אם בחלוקה של המטריצה ל- 4 רבעים (לפי הציר), כל איבר ברביע הראשון קטן או שווה לכל איבר ברביע השני, כל איבר ברביע השני קטן או שווה לכל איבר ברביע השלישי, וכל איבר ברביע השלישי קטן או שווה מכל איבר ברביעי. כמו להלן:

1 2

3 4

כמו כן התכונה הנ"ל מתקיימת רקורסיבית על כל אחד מהרבעים.

למשל, המטריצה הבאה ממוינת:

-4	-2	5	9
2	5	12	13
13	20	25	25
22	24	49	57

כתבו שיטה עם החתימה הבאה:

```
public static boolean find(int [][] mat, int x)
```

המחזירה true אם ורק אם הערך x נמצא במטריצה הממוינת mat (נתון שהמטריצה אכן ממוינת ואין צורך לבדוק זאת).

השיטה צריכה להיות יעילה ככל הניתן.

פתרון נכון שאינו מספיק יעיל יקבל ניקוד חלקי בלבד. פתרון מסדר גודל ריבועי $O(n^2)$ לא יתקבל בכלל.

אין להקצות זיכרון נוסף בגודל שאינו קבוע.
ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.
שימו לב, השיטה לא חייבת להיות רקורסיבית!

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

אל תשכחו לתעד את מה שכתבתם!

שאלה 3-25 נקודות (להגשה)

תהי נתונה קבוצה S של מספרים טבעיים (שלמים גדולים מ-0), ויהי n מספר טבעי כלשהו. נאמר ש- n הוא סכום מתוך S אם ניתן לבטא את S כסכום של מספרים (עם או בלי חזרות) מתוך S .

דוגמאות:

- תהי $S = \{4, 5\}$, אזי למשל 13 הוא סכום מתוך S שכן $4 + 4 + 5 = 13$, אבל 6 אינו סכום מתוך S , שכן לא ניתן לבטא את 6 כסכום של מספרים מתוך S .
- תהי $S = \{4, 9, 3\}$, אזי למשל 15 הוא סכום מתוך S שכן $9 + 3 + 3 = 15$, אבל 5 אינו סכום מתוך S , שכן לא ניתן לבטא את 5 כסכום של מספרים מתוך S .

שימו לב: 0 הוא סכום מתוך כל קבוצה S , שכן ניתן לבטא את 0 כסכום של קבוצה ריקה של מספרים. כמו כן, לכל מספר טבעי n השייך לקבוצה S , n הוא סכום מתוך S , כן ניתן לבטא את n כסכום של עצמו בלבד.

כתבו שיטה סטטית רקורסיבית בוליאנית המקבלת כפרמטרים מערך s מלא במספרים טבעיים שונים זה מזה, המהווים את איברי הקבוצה, ומספר טבעי n , ומחזירה true אם n הוא סכום מתוך s . אחרת, השיטה תחזיר false.
המערך לא ממוין.

חתימת השיטה היא:

```
public static boolean isSumOf(int [] s, int n)
```

אתם יכולים להניח כי הפרמטרים תקינים. המערך s אינו null ואינו ריק, וכל האיברים שבו הם טבעיים ושונים זה מזה, והמספר n הוא טבעי.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. ניתן להשתמש בשיטות עזר, אך גם הן לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).

אל תשכחו לתעד את מה שכתבתם!

שאלה 4-25 נקודות (להגשה)

נתון מערך דו-ממדי ריבועי המכיל בתוכו מספרים שלמים. נגדיר מסלול (path) במערך המתחיל בתא $[x1][y1]$ ומסתיים בתא $[x2][y2]$ כך: סדרה של תאים במערך באופן שמתקיים כי:

- התא הראשון בסדרה הוא $[x1][y1]$
- התא האחרון בסדרה הוא $[x2][y2]$
- המעבר מתא הוא רק לשכניו כאשר שכניו של תא הם התאים הצמודים אליו שמעליו, מתחתיו, מימינו ומשמאלו. במקרה ורק חלק מהשכנים קיימים (כאשר התא $[i][j]$ נמצא בשולי המערך), יש להתחשב רק בשכנים הקיימים.
- כל תא במערך מופיע לכל היותר פעם יחידה במסלול.

לדוגמא, במערך בן חמש שורות וחמש עמודות קיימים המסלולים הבאים בין התא $[0][1]$ לתא $[2][4]$:

- $[0][1] - [0][2] - [0][3] - [1][3] - [1][4] - [2][4]$
 - $[0][1] - [1][1] - [2][1] - [2][0] - [3][0] - [3][1] - [3][2] - [3][3] - [3][4] - [2][4]$
- וקיימים, כמובן, עוד מסלולים רבים.

כתבו שיטה סטטית **רקורסיבית** המקבלת כפרמטר מערך דו-ממדי `mat` מלא במספרים שלמים חיוביים, וזוג תאים במערך המצויים מעל לאלכסון הראשי (כלומר, האלכסון `mat[i][i]`). השיטה צריכה להחזיר את מספר המסלולים שאינם חוצים את האלכסון הראשי (אך הם עשויים להכיל תאים המצויים באלכסון זה) הקיימים בין שני התאים. **בדוגמא לעיל**, המסלול הראשון אינו חוצה את האלכסון הראשי, אך המסלול השני כן חוצה, ולכן הוא לא ייספר.

חתימת השיטה:

```
public static int numPaths (int[] [] mat,
                           int x1, int y1, int x2, int y2)
```

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).

אסור להשתמש במשתנים סטטיים (גלובליים)!

אסור להשתמש במערך עזר.

ערכי המטריצה לאחר הרצת השיטה numPaths צריכים להיות תואמים לערכי המטריצה לפני הרצת השיטה.

שימו לב:

בשאלות 3 ו-4 אין צורך לדאוג ליעילות השיטה שתכתבו!

שאלה לא להגשה

לפניכם שני קטעי הקוד (שאינם קשורים זה לזה):

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

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

מה סיבוכיות זמן הריצה של קטעי הקוד האלו?

להזכירכם – חוקי הלוגריתמים:

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

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

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

שאלה לא להגשה

התבוננו בשיטות הבאות :

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

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

נניח שנתונה השיטה main הבאה :

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

1. מה הפלט שתפיק השיטה main?

2. כמה קריאות רקורסיביות מתבצעות בזימון

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

שאלה לא להגשה

לפניכם קטע הקוד הבא:

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

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

א. foo (3, 4)

ב. foo (4, 5)

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: רשימות מקושרות

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 5 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 24.6.2017

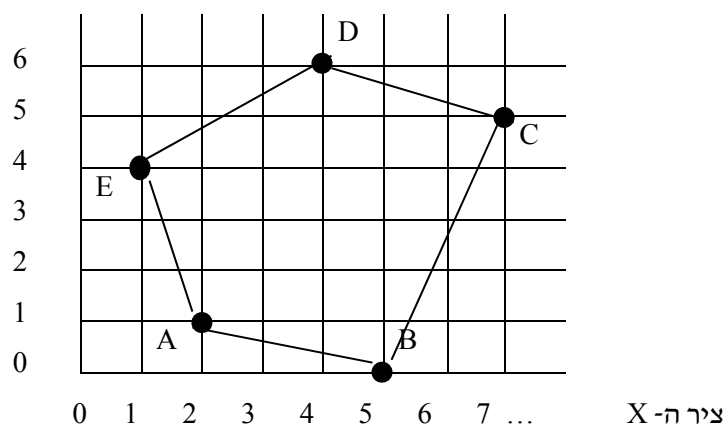
סמסטר: 2017

כזכור, במטלה 12 הגדרנו מחלקה Point המייצגת נקודה במישור ובמטלה 13 הגדרנו מחלקה Polygon המייצגת מצולע קמור על ידי שימוש מערך חד-ממדי של נקודות מהמחלקה Point. במטלה זו נייצג מצולע קמור על ידי שימוש ברשימה מקושרת.

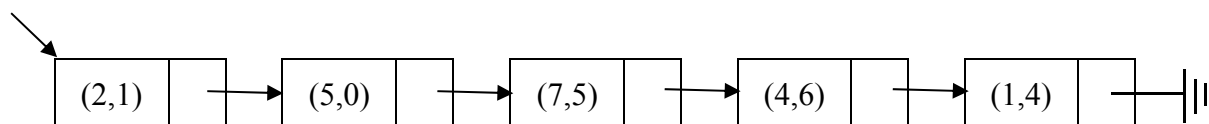
הייצוג נעשה על-ידי רשימה מקושרת ששומרת את רשימת הקדקודים (vertices) של המצולע לפי סדר הופעתם במצולע. אין חשיבות מי הקדקוד הראשון. כל קדקוד מיוצג על-ידי נקודה במישור.

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

ציר ה-Y



הרשימה המייצגת את הקדקודים של המצולע היא:



כדי לעשות זאת, עליכם להגדיר שתי מחלקות PointNode ו-Polygon.

שאלה 1 - להרצה (10%)

המחלקה **PointNode** תייצג קדקוד אחד במצולע. המחלקה **Point** היא זו שכתבתם במטלה 12 (או שאנחנו סיפקנו במטלה 13)

לכל אובייקט במחלקה יש שני שדות:

```
1. Point _point // הנקודה במישור
2. PointNode _next // מצביע לאיבר הבא
```

למחלקה זו עליכם להגדיר שלושה בנאים:

```
1. public PointNode (Point p)
    בנאי המקבל נקודה, שדה ה-_next יאותחל ל-null.
```

```
2. public PointNode (Point p, PointNode n)
    בנאי המקבל נקודה ואיבר נוסף מטיפוס PointNode, ומאתחל את התכונות לפי הפרמטרים.
```

```
3. public PointNode (PointNode p)
    • בנאי העתקה. שימו לב שפה aliasing הוא לא טעות. יש להעתיק את המידע (next) עצמו ולא עותק של המצביע.
```

השיטות במחלקה **PointNode** הן:

- `public Point getPoint()` - שיטה המחזירה עותק של הנקודה שבקדקוד.
- `public PointNode getNext()` - שיטה המחזירה מצביע לאיבר הבא. שימו לב שפה **aliasing** הוא לא טעות. יש להחזיר את המצביע **next** ולא עותק של המצביע.
- `public void setPoint(Point p)` - שיטה המקבלת נקודה ומעדכנת את תכונת הנקודה שבקדקוד.
- `public void setNext(PointNode next)` - שיטה המקבלת מצביע ומעדכנת את תכונת המצביע לאיבר הבא. שימו לב שפה **aliasing** הוא לא טעות. יש לעדכן את המידע (**next**) עצמו ולא עותק.

שאלה 2 - להרצה (90%)

המחלקה **Polygon** מייצגת מצולע קמור במישור. שימו לב שזו בדיוק אותה המחלקה שכתבתם במטלה 13 אלא שהפעם המימוש הוא על ידי רשימה מקושרת.

הייצוג נעשה על-ידי רשימה ששומרת את רשימת הקדקודים (**vertices**) של המצולע לפי סדר הופעתם במצולע. אין חשיבות מי הקדקוד הראשון. כל קדקוד מיוצג על-ידי נקודה במישור.

במחלקה זו מותר להגדיר אך ורק תכונה פרטית אחת, ראש הרשימה, שתצביע להתחלת הרשימה. אין להוסיף תכונות מעבר לתכונה זו.

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

1. הגדרת התכונה של המחלקה.
 2. בנאי שיוצר מצולע ריק - מאתחל את ראש הרשימה להיות null.
 3. שיטה בוליאנית addVertex שמוסיפה קדקוד למצולע. היא מקבלת כפרמטרים נקודה p, ומספר שלם pos המסמן לאיזה מקום ברשימה תיכנס הנקודה החדשה p. אם הוספת הקדקוד הצליחה, השיטה תחזיר true, אם לא – השיטה תחזיר false. עליכם לחשוב מתי השיטה עלולה שלא להצליח להוסיף קדקוד לרשימה. **אפשר להניח שהקדקוד החדש שנוסף לא מקלקל את היות המצולע מצולע קמור, ואין צורך לבדוק זאת. (זה לא יגרום להחזרת false).** כמו כן, ניתן להניח שכאשר מייצגים מצולע קמור ברשימה, מוסיפים את הקדקודים על ידי השיטה addVertex לפי סדר הופעתם במצולע.
 4. שיטה highestVertex המחזירה העתק של את הקדקוד שנמצא הכי גבוה במצולע. אם יש יותר מאחד בגובה הגבוה ביותר, היא מחזירה את הראשון בו נתקלה. אם אין קדקודים במצולע (כלומר הרשימה ריקה) השיטה תחזיר null.
 5. השיטה toString המחזירה מחרוזת תווים המייצגת את המצולע. המחרוזת צריכה להיות בפורמט הבא: שימו לב שאין רווחים במחרוזת של הקדקודים.
The polygon has 5 vertices:
((2,1) , (5,0) , (7,5) , (4,6) , (1,4))
אם אין קדקודים השיטה תחזיר מחרוזת בפורמט הבא:
The polygon has 0 vertices.
 6. שיטה calcPerimeter המחזירה מספר ממשי (double) המייצג את היקף המצולע. אם מספר הקדקודים הוא 2 יוחזר אורך הקטע (לא הלך וחזר). אם מספר הקדקודים 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-B-C, A-C-D, A-D-E. 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, c הם אורכי שלוש הצלעות של המשולש, ו s הוא מחצית היקפו. אם מספר הקדקודים קטן מ-3 יוחזר 0.
- שימו לב שהשיטה לחישוב שטח משולש צריכה להיות פרטית ולא ציבורית.**

8. שיטה בוליאנית isBigger 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 ברשימה, אם היא נמצאת. אם לא, יוחזר 1-. לדוגמא, אם השיטה תופעל עם הפרמטר (5,0) היא תחזיר את הערך 2. (כי זהו האיבר השני ברשימה).

10. שיטה getNextVertex, המקבלת נקודה כפרמטר, ומחזירה העתק של הנקודה המייצגת את הקדקוד הבא במצולע. אם הנקודה שהתקבלה אינה קדקוד במצולע, השיטה תחזיר null. אם הנקודה היא האיבר האחרון ברשימה, יוחזר העתק של הנקודה הראשונה. אם הנקודה היא הנקודה היחידה ברשימה יוחזר העתק של הנקודה עצמה.

11. שיטה getBoundingBox המחזירה את המלבן (כפוליגון) (המקביל לצירים) החוסם את המצולע. אם מספר הקדקודים קטן מ-3 יוחזר null.

לפניכם רשימת החתימות של הבנאי ושיטות המחלקה:

public Polygon()	בנאי
public boolean addVertex(Point p, int pos)	שיטה שמוסיפה קדקוד למצולע
public Point highestVertex()	שיטה שמחזירה העתק של את הקדקוד שנמצא הכי גבוה במצולע
public String toString()	שיטה שמחזירה מחרוזת תווים המייצגת את המצולע
public double calcPerimeter ()	שיטה שמחזירה את היקף המצולע
public double calcArea()	שיטה שמחזירה את שטח המצולע
public boolean isBigger(Polygon other)	שיטה שמקבלת מצולע אחר, ובודקת אם המצולע שעליו מופעלת השיטה גדול בשטחו מהמצולע המועבר כפרמטר
public int findVertex(Point p)	שיטה שמקבלת נקודה כפרמטר ומחזירה את המיקום שלה ברשימה
public Point getNextVertex(Point p)	שיטה שמקבלת נקודה כפרמטר, ומחזירה העתק של הנקודה המייצגת את הקדקוד הבא במצולע
public Polygon getBoundingBox()	שיטה שמחזירה את המלבן המקביל לצירים (כפוליגון) החוסם את המצולע

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.
3. שימו לב לא לכתוב קוד מיותר (שכבר נכתב) אלא להשתמש במחלקות המתאימות.

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: PointNode.java, Polygon.java ארוזים יחד בתוך קובץ zip יחיד. 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: 3 נקודות

סמסטר: 2017 מועד אחרון להגשה: 1.7.2017

(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!

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

שאלה 1. איזה מהמשפטים הבאים נכון?

- א. אם נממש את המחלקה Stack בעזרת רשימה מקושרת, הדרך היעילה ביותר לממש את הפעולה `empty()` היא ב $O(n)$.
- ב. אין הבדל ממשי בין המחלקות Stack ו Queue וניתן לבחור להשתמש בכל אחת מהן באופן אקראי.
- ג. LIFO הוא מושג שמתאר התנהגות של תור – האחרון שנכנס הוא הראשון שיוצא.
- ד. LIFO הוא מושג שמתאר התנהגות של מחסנית – האחרון שנכנס הוא הראשון שיוצא.

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. 654321

ג. 135246

ד. 234561

ה. 125643

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();
        System.out.print (num + "\t");
        push(num);
    }
}
```

א. מדפיסה את הערכים שבמחסנית s1 מזנב המחסנית אל ראשה

ב. מדפיסה את הערכים שבמחסנית s1 מראש המחסנית אל זנבה

ג. מדפיסה את הערכים שבמחסנית s1 מראש המחסנית אל זנבה ובחזרה מהזנב אל הראש

ד. מדפיסה את הערכים שבמחסנית s1 מזנב המחסנית אל ראשה ובחזרה מהראש אל הזנב

שאלה 4.

נניח ש $s1$ ו- $s2$ הן שתי מחסניות מהמחלקה Stack כאשר $s1$ מלאה ב- n איברים (אפשר להניח ש n מספר גדול) ו- $s2$ היא מחסנית ריקה. x הוא משתנה מסוג `int`. ידוע גם שקיימת שיטה בשם `length` שמקבלת כפרמטר אובייקט מהמחלקה Stack ומחזירה את אורך המחסנית.

בהינתן קטע הקוד הבא:

```
int len = length(s1) - 2;
for (int i = 0; i < len; i++){
    x = s1.pop();
    s2.push(x);
}

len = length(s1) - 2;
for (int i = 1; i <= len; i++){
    x = s2.pop();
    s1.push(x);
}
```

לאחר הרצת קטע הקוד שלעיל, מה מהמשפטים הבאים נכון?

- א. $s1$ לא תשתנה, הערך של x יהיה שווה לערך ה-3 מהתחתית של המחסנית $s2$
- ב. $s1$ לא תשתנה והערך של x יהיה זהה לערך שבראש המחסנית $s1$
- ג. $s1$ תכלול שני איברים והערך של x יהיה זהה לערך שבראש המחסנית $s1$
- ד. $s1$ לא תשתנה והערך של x יהיה זהה לערך שבראש המחסנית $s2$
- ה. $s1$ תכלול שני איברים והערך של x יהיה זהה לערך בתחתית המחסנית $s1$
- ו. $s1$ תכלול שני איברים והערך של x יהיה זהה לערך שבראש המחסנית $s2$

```

public void secret(){
    _head = secret(this, new Stack(),null)._head;
}

private Stack secret(Stack source, Stack dest, Stack s){
    if (s==null)
        s = new Stack();

    if (!source.empty()){
        int temp = source.pop();
        while (!source.empty()){
            if (source.top() > temp){
                dest.push(temp);
                temp = source.pop();
            }
            else{
                dest.push(source.pop());
            }
        }
        s.push(temp);
        return secret(dest,source,s);
    }
    else {
        return s;
    }
}

```

בהנחה שהמחסנית עליה מתבצעת השיטה secret הכילה את האיברים הבאים:
(משמאל לימין, בשמאל ראש המחסנית)

6 3 4 7 2 5

_head הוא ראש המחסנית.

איך תיראה המחסנית לאחר ביצוע השיטה secret ? (משמאל לימין, בשמאל ראש המחסנית)

- | | | | | | | |
|----|---|---|---|---|---|---|
| א. | 6 | 7 | 3 | 4 | 2 | 5 |
| ב. | 7 | 6 | 5 | 4 | 3 | 2 |
| ג. | 2 | 3 | 4 | 5 | 6 | 7 |
| ד. | 5 | 6 | 2 | 7 | 4 | 3 |

הכתוב להלן מתייחס לשאלות 6 – 8 המופיעות בהמשך.

נתונה המחלקה `DataQueue` הבאה, המממשת תור. איברי התור הם אובייקטים מהמחלקה `TreeNode`, המכילים ערכים מטיפוס `int`:

```
public class DataQueue
{
    private TreeNode _rear, _front;

    public DataQueue() {...}
    // Constructs an empty queue

    public boolean empty() {...}
    // Returns true iff the queue is empty

    public void enqueue(TreeNode newItem) {...}
    // Adds the new item to the end of the queue

    public TreeNode retrieve() {...}
    // Removes oldest item from the queue and returns it.
    // Returns null if queue is empty.
}
```

בנוסף, נתונות שתי השיטות הסטטיות הבאות:

```
public static boolean mystery (DataQueue q1, DataQueue q2)
{
    if (q1.empty() && q2.empty())
        return true;
    if (q1.empty() || q2.empty())
        return false;
    if (!(q1.retrieve().equals( q2.retrieve())))
        return false;
    return mystery (q1, q2);
}

public static void secret (DataQueue q)
{
    if (!q.empty())
    {
        TreeNode node = q.retrieve();
        secret (q);
        q.enqueue (node);
    }
}
```

השיטה `equals` המופיעה בשיטה `mystery` שייכת למחלקה `TreeNode` והיא מחזירה `true` אם האובייקט שעליו מופעלת השיטה שווה בערכיו לאובייקט המועבר לה כפרמטר.

שאלה 6.

אם התור q_2 מכיל את האיברים הבאים (משמאל לימין – האיבר השמאלי הוא ראש התור והימני הוא זנב התור) $\{5, 4, 1, 6, 3, 2\}$

אם נריץ את רצף הפקודות

```
secret (q2);  
System.out.println (mystery(q1, q2));
```

כדי שיודפס על הפלט הערך $true$, התור q_1 צריך להיות:

א. $\{5, 4, 1, 6, 3, 2\}$

ב. $\{4, 5, 6, 1, 2, 3\}$

ג. $\{4, 1, 6, 3, 2, 5\}$

ד. $\{2, 3, 6, 1, 4, 5\}$

ה. $\{5, 5, 5, 2, 2, 2\}$

ו. אף תור q_1 לא יחזיר ערך $true$ בקריאה לעיל.

שאלה 7.

כאשר השיטה `mystery` מקבלת כפרמטרים שני תורים q_1 ו- q_2 , היא:

א. בודקת אם התורים q_1 ו- q_2 מכילים אותם איברים (הסדר לא חשוב)

ב. בודקת אם התורים q_1 ו- q_2 מכילים אותם איברים (הסדר כן חשוב)

ג. בודקת אם לתורים q_1 ו- q_2 יש אותו מספר איברים.

ד. בודקת אם התורים q_1 ו- q_2 מכילים אותו איבר בראש התור ובזנב התור.

שאלה 8.

כאשר השיטה `secret` מקבלת כפרמטר תור q , היא:

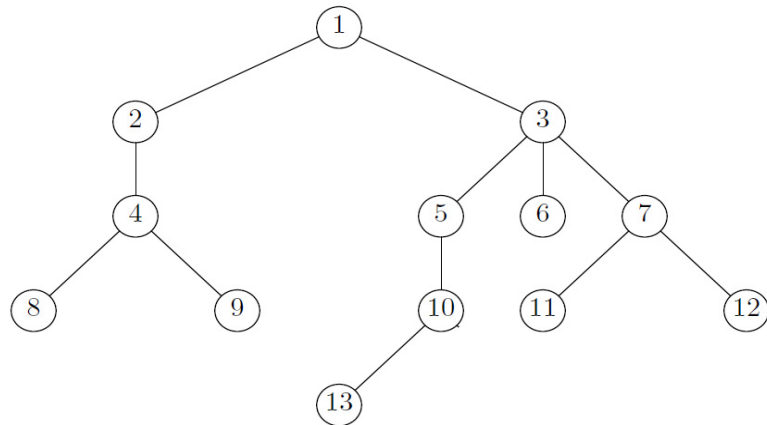
א. מחזירה את התור כמו שהוא.

ב. מחזירה את התור כשבראשו נמצא האיבר המקורי ואחריו שאר האיברים בסדר כלשהו

ג. הופכת את הסדר בין איברי התור.

ד. אי אפשר לדעת מה עושה השיטה באופן כללי.

שאלה 9. נתון עץ ששורשו הוא הצומת 1 :



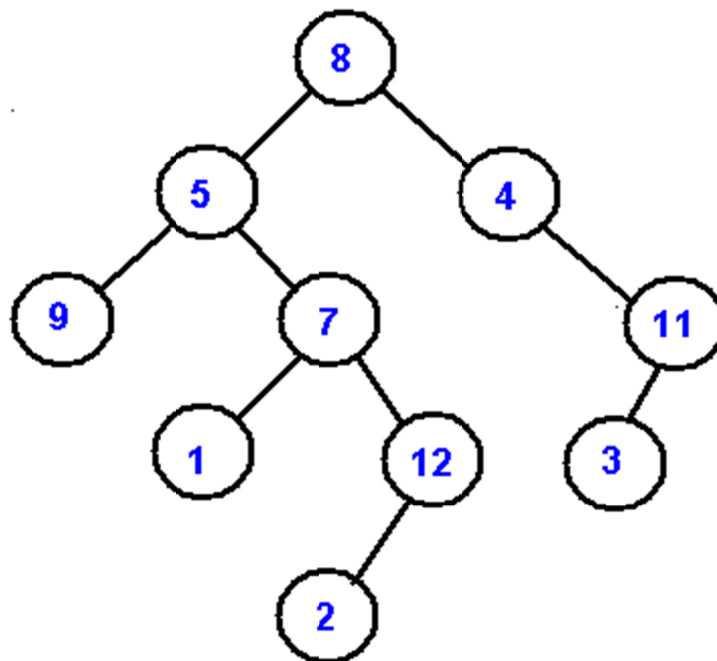
נתונות הטענות הבאות :

1. השורש של העץ הוא : 1
2. העלים של העץ הם : 13, 12, 11, 9, 8, 6
3. הצמתים הפנימיים של העץ הם : 10, 7, 5, 4, 3, 2, 1
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. אף תשובה אינה נכונה
- ה. כל הטענות נכונות

שאלה 10. נתון העץ הבינרי הבא



נתונות הטענות הבאות :

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

סמנו את התשובה הנכונה :

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

שאלה 11. לאחר הפעלת סיור PreOrder ו InOrder על עץ בינארי כלשהו התקבלו התוצאות הבאות:

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

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

אם נפעיל סיור PostOrder על העץ הזה נקבל את התוצאה הבאה.
סמנו את התשובה הנכונה :

א. 9 7 6 3 10 8 5 4 2 1

ב. 4 10 8 5 2 6 9 7 3 1

ג. 10 9 8 7 6 5 4 3 2 1

ד. 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.

שאלה 12. לאחר הפעלת סיור PostOrder על עץ חיפוש בינארי כלשהו התקבלה התוצאה הבאה:

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

אם נפעיל סיור PreOrder על העץ הזה נקבל את התוצאה הבאה.
סמנו את התשובה הנכונה :

א. 7 6 4 2 1 3 5 8 10 9

ב. 7 8 10 9 6 4 5 2 3 1

ג. 10 9 8 7 6 5 4 3 2 1

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

קטע הקוד הבא מתיחס לשאלות 13 ו-14.

נתון עץ בינארי כלשהו T בעל n צמתים שמכילים מספרים שלמים חיוביים בלבד
כמו כן נתון קטע הקוד הבא :

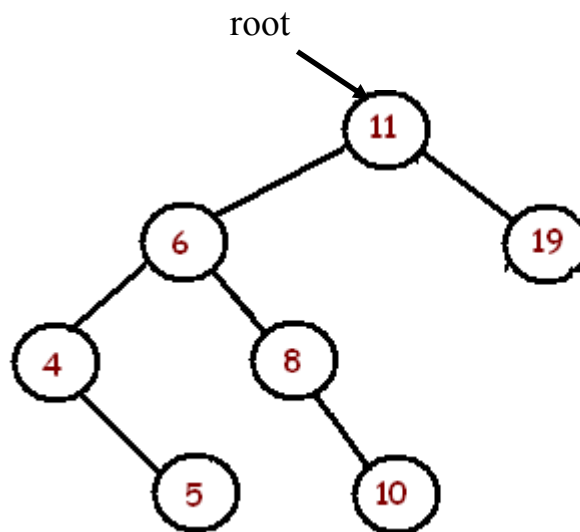
```
public int what(Node root)
{
    if(root == null)
        return 0;
    int x = what(root.getLeft());
    int y = what(root.getRight());
    return root.getValue() + f(x,y);
}

private int f(int x,int y)
{
    return y>x? y:x;
}
```

השיטה what מקבלת כפרמטר מצביע לשורש העץ T .

שאלה 13.

מה תחזיר השיטה what לעיל בהינתן לה העץ הבינארי הבא ששורשו root?



סמנו את התשובה הנכונה

- א. 19
- ב. 21
- ג. 30
- ד. 35
- ה. 63

שאלה 14.

סמנו את התשובה הנכונה :

- א. השיטה what מחזירה את המסלול הארוך ביותר בעץ
- ב. השיטה what מאתרת את המסלול מהשורש לעלה אשר סכום הצמתים לאורכו הוא מקסימלי ומחזירה את הסכום
- ג. השיטה what מחזירה את הסכום של האיברים בעץ לאורך המסלול הארוך ביותר
- ד. התשובות ב ו- ג נכונות

השאלות 15 - 17 להלן מתייחסות לקטע הבא.

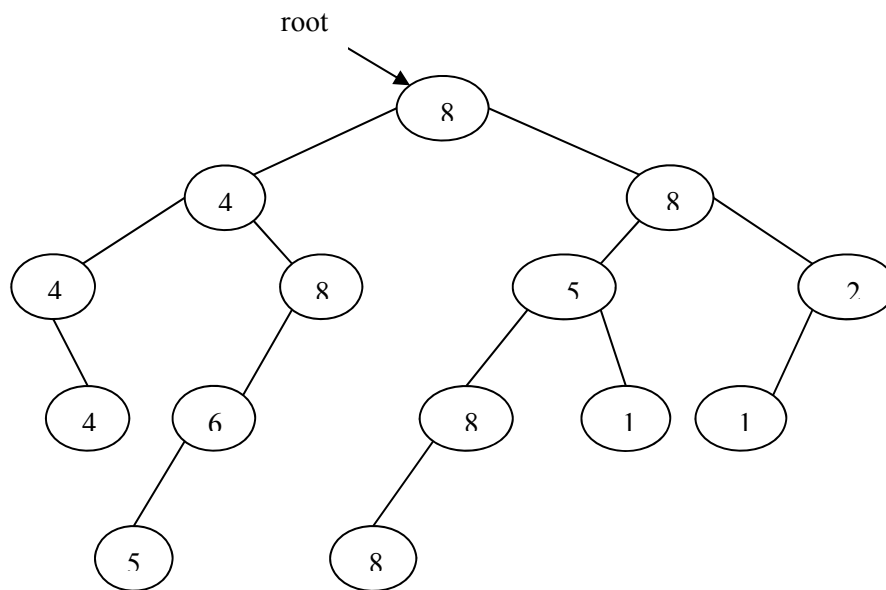
המחלקה Node מממשת צומת בעץ בינרי (לפי ההרצאה ביחידה 12).

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי. התבוננו בשיטה הבאה וענו על השאלות שאחריה:

```
public static boolean f (Node t)
{
    if (t == null)
        return true;
    if (t.getLeftSon() == null && t.getRightSon() == null)
        return true;
    if (t.getLeftSon() == null)
        return ((t.getNumber()==t.getRightSon().getNumber()) &&
                f(t.getRightSon()));
    if (t.getRightSon() == null)
        return ((t.getNumber()==t.getLeftSon().getNumber()) &&
                f(t.getLeftSon()));
    return (((t.getNumber()== t.getRightSon().getNumber()) &&
            f(t.getRightSon()))
            ||
            ((t.getNumber()== t.getLeftSon().getNumber()) &&
            f(t.getLeftSon())));
}
```

שימו לב, אכן יש בשיטה הרבה סוגריים והביטויים הלוגיים מורכבים, אך אין בה טעות. בדקו היטב מה מוחזר בכל אחד מהתנאים.

בהינתן העץ הבינרי הבא:



שאלה 15.

מה יודפס כתוצאה מהפקודה:

```
System.out.println (BinaryTree.f(root));
```

false .N

true ב.

שאלה 16.

מהם השינויים המינימליים שעלינו לבצע בעץ הנתון לעיל כדי שהשיטה f תדפיס תוצאה אחרת מזו שהודפסה בסעיף 1. (שינויים בעץ ולא בשיטה).

א. לשנות את הצומת שערכו 5 (שהוא בן שמאלי של 8) לערך 8

ב. לשנות את השורש ל- 4

ג. לבצע את אחת מהטענות א או ב – לא משנה מה

ד. לבצע את שתי הטענות א וגם ב

שאלה 17.

מה מבצעת השיטה f באופן כללי, כאשר היא מקבלת שורש לעץ בינרי?

- השיטה f מחזירה את הסכום של האיברים בעץ לאורך המסלול הארוך ביותר
- השיטה f מאתרת את המסלול מהשורש לעלה אשר סכום הצמתים לאורכו הוא מקסימלי ומחזירה את הסכום
- השיטה f מחזירה את המסלול הארוך ביותר בעץ
- השיטה f בודקת אם יש מסלול מהשורש עד לאחד העלים כך שבכל הצמתים במסלול זה יש אותו ערך
- אף אחת מהתשובות לעיל (א – ד) אינה נכונה

שאלה 18. בהינתן עץ חיפוש בינרי, איזה מהמשפטים הבאים נכון?

- שורש העץ הוא בהכרח האיבר המינימלי בעץ
- אם רוצים להוסיף איבר מסוים x לעץ יתכן ויש יותר ממקום אחד שמתאים לו בעץ כך שעדיין העץ ימשיך להיות עץ חיפוש בינרי
- אם נתקדם בעץ כל פעם שמאלה נגיע לאיבר המקסימלי בעץ
- בהנתן רשימת ערכים x_1, x_2, \dots, x_n , סדר הכנסת האיברים לעץ משפיע על מבנה העץ. כלומר, אם נכניס את האיברים בסדר שונה, מבנה העץ יהיה שונה.

שאלה 19. איזה משפט מהמשפטים הבאים אינו נכון?

- בבסיס בינרי 10 ועוד 10 שווה 100.
- 8 ביט זה גודל מספיק כדי לייצג מספר בין 0 ל – 127.
- 8 ביט זה גודל מספיק כדי לייצג מספר בין 0 ל – 511.
- 8 ביט נקראים בייט (byte).

שאלה 20. מה ערך המספר הבינרי 001010111 בבסיס עשרוני (10) ?

- 78
- 80
- 109
- 128

ה. אף אחת מהתשובות א-ד אינה נכונה

ב ה צ ל ח ה