

האוניברסיטה הפתוחה

20441
מבוא למדעי המחשב
ושפת Java
חוברת הקורס – סתיו 2018א

כתבה: תמר וילנר

אוקטובר 2017 – סמסטר סתיו – תשע"ח

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
11	ממ"ן 11
17	ממ"ן 12
25	ממ"ן 13
29	ממ"ח 02
41	ממ"ן 14
47	ממ"ן 15
53	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/ 2018א)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	20.10.2017-17.10.2017	יחידות 1 ו-2 – הקדמה ויסודות השפה בספר – פרקים 1 ו-2	מפגש 1	
2	27.10.2017-22.10.2017	יחידה 2 – יסודות השפה בספר – פרקים 2 ו-5		ממ"ן 01 28.10.2017
3	3.11.2017-29.10.2017	יחידה 3 – שימוש במחלקות נתונות בספר – פרק 3		
4	10.11.2017-5.11.2017	יחידה 4 – כתיבת מחלקות בספר – פרק 4	מפגש 2	ממ"ן 11 11.11.2017
5	17.11.2017-12.11.2017	יחידה 4 – כתיבת מחלקות בספר – פרק 4		
6	24.11.2017-19.11.2017	יחידה 5 – לולאות בספר – פרקים 5 ו-6	מפגש 3	ממ"ן 12 25.11.2017
7	1.12.2017-26.11.2017	יחידה 6 – מערכים בספר – פרק 8		
8	8.12.2017-3.12.2017	יחידה 7 – ירושה בספר – פרק 9	מפגש 4	ממ"ן 13 9.12.2017

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

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
9	15.12.2017-10.12.2017 (ד-ו חנוכה)	יחידה 8 – פולימורפיזם – בספר פרק 10		
10	22.12.2017-17.12.2017 (א-ד חנוכה)	יחידה 9 – יעילות – בספר פרק 10	מפגש 5	ממ"ן 02 23.12.2017
11	29.12.2017-24.12.2017	יחידות 9 ו-10 – יעילות ורקורסיה – בספר פרק 12	מפגש 6	
12	5.1.2018-31.12.2017	יחידה 10 – רקורסיה – בספר פרק 12	מפגש 7	
13	12.1.2018-7.1.2018	יחידה 11 – רשימות – בספר פרק 13	מפגש 8	ממ"ן 14 13.1.2018
14	19.1.2018-14.1.2018	יחידה 12 – מחסנית, תור ועצים בינריים – בספר פרק 12		ממ"ן 15 20.1.2018
15	29.1.2018-21.1.2018	יחידה 20 וחזרה	מפגש 9	ממ"ן 03 29.1.2018

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 שאלת"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:

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

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 כמועתקת (או ככזו שהעתיקו ממנה), יועלה לוועדת משמעת. שימו לב, אפשר להתייעץ עם זה במהלך הכנת המטלות, אך ההתייעצות חייבת להיות בעל-פה (ללא כתיבת חומר כלשהו). לאחר ההתייעצות כל אחד חייב לכתוב את המטלה בנפרד.

מועדי הגשת המטלות

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

דחייה בהגשת מטלות

- אנחנו נאשר איחורים רק כאשר יש בקשה מראש לאיחור כזה. בקשה מנומקת לאיחור של עד שבוע יש להפנות למנחה הקבוצה. במקרים חריגים ביותר של דחייה של למעלה משבוע, תצטרכו לפנות למרכזת ההוראה לקבלת אישור כזה. שוב, האישור צריך להתקבל מלכתחילה ולא בדיעבד.
- מי שיאחר בהגשת המטלה ללא קבלת אישור מראש, יורדו לו 3 נקודות מהציון לכל יום איחור, וגם זה רק עד שבוע. לאחר שבוע המטלה תיבדק למשוב בלבד, ולא תחשב בשקלול הציון.
- לגבי מילואימניקים – אתם יכולים לפנות אל המנחים שלכם ולסכם איתם את מועדי ההגשה למטלות המתאימים לכם ולמנחים. לכל מטלה המוגשת באיחור 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: 2018 מועד אחרון להגשה: 28.10.2017

(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. 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 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 **שאילתא** (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 שאילתא

1. היכנס למערכת שאילתא. (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. מהי מחלקה (class)?

- א. מינוח מקובל לשפת תכנות מונחית-עצמים
- ב. מינוח המשמש לתאור מופשט של קבוצת אובייקטים בעלי תכונות ופעולות משותפים
- ג. מושג המתייחס לקבוצה של אובייקטים שנוצרה בזמן נתון בכדי לבצע פעולה חישובית אחת
- ד. אף אחת מהתשובות א-ג איננה נכונה

שאלה 2.

אילו מהתיאורים הבאים הוא המתאים ביותר לתאור של המחלקה "סטודנט"

- א. למחלקה סטודנט יש תכונות (attributes) כגון "שם", "כתובת מגורים", "שכר לימוד", "מחלקה" ופעולות (methods) כגון "שנה", "תעודת זהות" ו"תאריך לידה"
- ב. למחלקה סטודנט יש פעולות (methods) כגון "שם", "כתובת מגורים", "שכר לימוד", "מחלקה" ותכונות (attributes) כגון "העבר מחלקה", "הורד שכר לימוד" ו"עדכן כתובת מגורים"
- ג. למחלקה סטודנט יש תכונות (attributes) כגון "שם", "כתובת מגורים", "שכר לימוד", "מחלקה" ופעולות (methods) כגון "העבר מחלקה", "הורד שכר לימוד" ו"עדכן כתובת מגורים"
- ד. למחלקה סטודנט יש פעולות (methods) כגון "שם", "כתובת מגורים", "שכר לימוד", "מחלקה" ואין לה תכונות כלל.

שאלה 3.

אילו מהמשפטים הבאים איננו נכון?

- א. שגיאת קומפילציה גורמת לשגיאת ריצה
- ב. שגיאה מילונית ושגיאה תחבירית הן שגיאות קומפילציה
- ג. כל אובייקט הוא מופע של מחלקה
- ד. שגיאה באלגוריתם גורמת לשגיאת ריצה או שגיאת מטרה
- ה. שגיאת ריצה עלולה לגרום לשגיאת מטרה

שאלה 4.

מהו אלגוריתם?

- א. הרצת התוכנית עם אוסף נתוני בדיקה לצורך איתור שגיאות
- ב. המרת הקוד שכתב התוכניתן 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 5.

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

שאלה 6. יש לכתוב תכנית שמכילה מחלקה בשם GoodMorning ומדפיסה את המשפט

Good morning, people על המסך. אילו מהתכניות הבאות תבצע את הדרוש?

```
א public class GoodMorning {  
    public static void main (String [] args) {  
        System.out.print ("Good morning, people");  
    }  
}
```

```
ב public class Good Morning {  
    public static void main (String [] args) {  
        System.out.print ("Good morning, people");  
    }  
}
```

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

```
ד public class GoodMorning {  
    public static void main (String [] args) {  
        system.out.print ("Good morning, people");  
    }  
}
```

```
ה public class goodMorning {  
    public static void main (String [] args) {  
        System.out.print ("Good morning, people");  
    }  
}
```

שאלה 7. הטקסט הכתוב בהערה בתכנית

א. יכול להיות כל דבר שהמתכנת רוצה לכתוב.

ב. נבדק על ידי המהדר (compiler) לשם הדיק

ג. חייב להופיע בשורה הראשונה של התכנית

ד. מודפס כאשר התכנית רצה

שאלה 8. אלו מבין השורות הבאות מכילה הערה תקינה (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 9. אלו מבין המזהים (identifiers) הבאים אינו חוקי בשפת Java?

א. `37abc`
ב. `_37abc`
ג. `Abc37`
ד. `abc37`
ה. `_`

שאלה 10. מה מהבאים הוא מחרוזת תווים שמכילה בדיוק תו אחד בודד (כל תו אפשרי)?

א. `3`
ב. `'3'`
ג. `"3"`
ד. `String "3"`
ה. `String 3`

שאלה 11. מה מהבאים הוא תו (char) שמייצג את הספרה 3?

א. `"3"`
ב. `3`
ג. `String "three"`
ד. `char 3`
ה. `'3'`
ו. `(char) 3`

שאלה 12. מה מבין קטעי הקוד הבאים ידפיס על המסך את המחרוזת Israel, Israeli ?

א. `System.out.print ("Israel Israeli");`
ב. `System.out.print ("Israeli, Israeli");`
ג. `System.Out.print ("Israel, Israel");`
ד. `System.out.print ("Israel, Israeli");`
ה. `System.out.print ("Israel, israeli");`

שאלה 13. איזו מהפקודות הבאות מצהירה על משתנה שלם שנקרא numOfBooks?

- א. `static int numOfBooks`
- ב. `int numOfBooks`
- ג. `int numOfBooks;`
- ד. `private int numOBooks =6;`
- ה. `int NumOfBooks;`

שאלה 14. איזו מהפקודות הבאות אינה מצהירה על משתנה שלם בשם size ומאתחלת אותו

לערך 10?

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

שאלה 15. 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 16. נתון קטע הקוד הבא: `int v = 7; System.out.println(v++);` מה

הערך שיודפס ומה הערך של המשתנה v בסופו של הקטע?

- א. הערך שיודפס יהיה 7 והערך של v יהיה 7
- ב. הערך שיודפס יהיה 8 והערך של v יהיה 8
- ג. הערך שיודפס יהיה 7 והערך של v יהיה 8
- ד. הערך שיודפס יהיה 8 והערך של v יהיה 7
- ה. אי אפשר לדעת מה הערך שיודפס והערך של v יהיה 8
- ו. אי אפשר לדעת מה הערך של v והערך שיודפס יהיה 8

שאלה 17. נתון קטע הקוד הבא: `int v = 7; System.out.println(++v);` מה

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

- א. הערך שיודפס יהיה 7 והערך של `v` יהיה 7
- ב. הערך שיודפס יהיה 8 והערך של `v` יהיה 8
- ג. הערך שיודפס יהיה 7 והערך של `v` יהיה 8
- ד. הערך שיודפס יהיה 8 והערך של `v` יהיה 7
- ה. אי אפשר לדעת מה הערך שיודפס והערך של `v` יהיה 8
- ו. אי אפשר לדעת מה הערך של `v` והערך שיודפס יהיה 8

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.println("first is" + f + "second=" + s);`
- ג. `System.out.println(first is + " f " + second = + " s`
- ד. `System.out.print("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. בהנחה שנתונים שלושה משתנים **שלמים** a, b, c, שלושתם מוצהרים ומאותחלים, אנחנו מעוניינים להדפיס את המקסימום ביניהם. קטע הקוד הבא אמור לבצע זאת. מספרי השורות אינם חלק מהקוד, כמובן.

```
1.         if (a<b)
2.             if (b>c)
3.                 System.out.println ("max = " + b);
4.             else
5.                 System.out.println ("max = " + c);
6.         else
7.             if (b>c)
8.                 System.out.println ("max = " + a);
9.             else
10.                System.out.println ("max = " + c);
```

מה מהמשפטים הבאים נכון?

- א. קטע הקוד לא יעבור קומפילציה בכלל.
- ב. קטע הקוד תקין.
- ג. קטע הקוד שגוי והשינוי צריך להיות בשורה 2, צריך לכתוב `if (a>c)`
- ד. קטע הקוד שגוי והשינוי צריך להיות בשורה 7, צריך לכתוב `if (a>c)`
- ה. קטע הקוד שגוי והשינוי צריך להיות בשורה 10, צריך לכתוב `"max = " + b`
- ו. אף אחת מהתשובות לעיל אינן נכונות

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: 2018 מועד אחרון להגשה: 11.11.2017

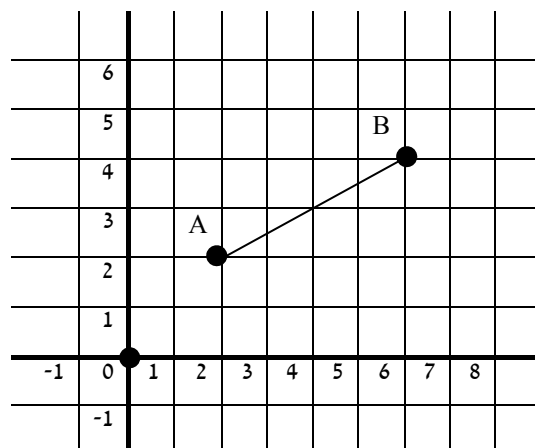
(ת)

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 `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 `Math.pow(a, b)`, שהיא שיטה של Java שנמצאת במחלקה `Math`. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא `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`.

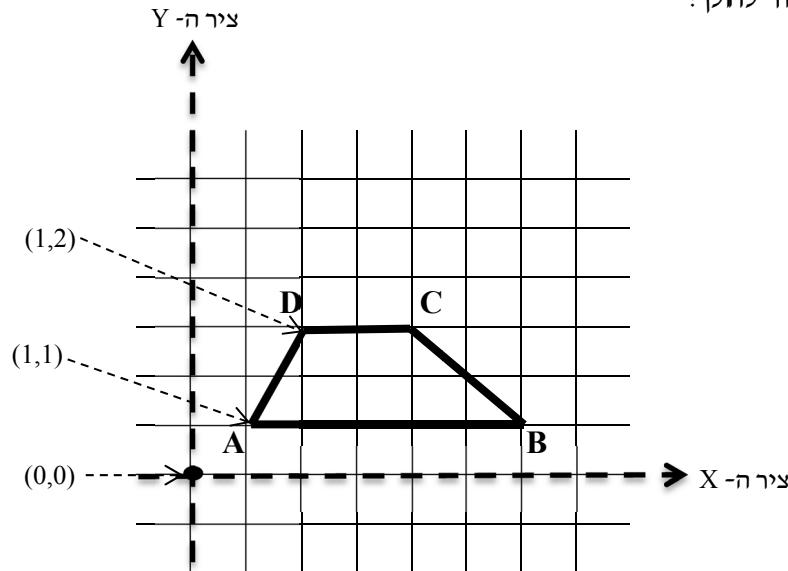
לעזרתכם, כתבנו כאן חלקים מהמחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).

```
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%)

כתבו תכנית לחשוב שטח והיקף של טרפז (trapezoid). **טרפז** הוא מרובע אשר לו שתי צלעות נגדיות מקבילות (אין שום תנאי על הצלעות האחרות). בשאלה זו נתייחס רק לטרפזים שהצלעות המקבילות שלו מקבילות גם לציר ה-X. ראו את האיור להלן:



תחילה עליכם לקלוט את נתוני הבסיסים של הטרפז. (הישרים בין A ל-B ובין C ל-D). כדי לעשות זאת עליכם לקלוט לכל בסיס את הנקודה השמאלית שלו ואת אורכו. משתמש שירצה להזין את נתוני הטרפז שבאיור לעיל יצטרך להזין כך:

1 1 5

1 2 2

כלומר, הנקודה השמאלית של בסיס אחד (A-B) היא (1,1) ואורכו הוא 5
הנקודה השמאלית של בסיס שני (D-C) היא (1,2) ואורכו הוא 2

כתבו תכנית הקוראת מהמשתמש שישה מספרים שלמים (לתוך משתנים מטיפוס int). שני המספרים הראשונים יצינו קואורדינטת x ו- קואורדינטת y של הנקודה השמאלית של בסיס אחד של הטרפז; המספר השלישי יציין את אורך הבסיס הזה; שני המספרים הבאים יצינו קואורדינטת x ו- קואורדינטת y של הנקודה השמאלית של בסיס שני של הטרפז; המספר השישי יציין את אורך הבסיס הזה. אין חשיבות לסדר בין הבסיסים.

על תכניתכם לחשב ולהציג את שטח (area) והיקף (perimeter, circumference) הטרפז.

ההדפסה של הפלט חייבת להיות בדיוק לפי הפורמט הבא:

The area of the trapezoid is 7.0

The perimeter of the trapezoid is 12.0644

הערה: מה שכתוב כאן בגופן נטוי לא צריך להיות נטוי בפלט שלכם. כאן כתבנו זאת רק כדי להדגיש את המשתנים. כמובן שהנתונים כאן הם אלו שמתאימים לאיור לעיל. לכל טרפז יהיו תוצאות אחרות. גם רמת הדיוק יכולה להיות לפי מה שתדפיס התכנית. אין צורך ליפות את ההדפסה.

הנחות:

1. אתם יכולים להניח שהמשתמש אכן הכניס שישה מספרים שלמים ולא אותיות או תווים אחרים.
2. אתם יכולים להניח שהמספרים שהוכנסו אכן מהווים קלט תקין לטרפז. שני המספרים הראשונים הם הנקודה השמאלית והמספר השלישי הוא האורך של הבסיס הראשון וכך גם לגבי הבסיס השני. אין ערבוב.
3. אתם לא יכולים להניח מי מבין הבסיסים יופיע ראשון. כלומר, יתכן והמשתמש יזין את נתוני הבסיס העליון קודם ויתכן שיזין את נתוני הבסיס התחתון קודם.
4. אפשר להשתמש בידע שרכשתם מפתרון שאלה 1.

התכנית שכתבתם צריכה להיות במחלקה בשם Trapezoid.

בממ"ן זה אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Line ו-Trapezoid), 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 Trapezoid.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: 2018 מועד אחרון להגשה: 25.11.2017

(ת)

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

שאלה 1 - 35 נקודות

המחלקה Time1 מייצגת זמן -

למחלקה Time1 יש את התכונות הפרטיות (instance variables) הבאות:

- `int _hour` – שמייצגת את השעה (בין 0 ל-23);
- `int _minute` – שמייצגת את הדקה (בין 0 ל-59).

למחלקה Time1 הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים (שעה ודקה)

```
public Time1(int h, int m)
```

אם אחד הפרמטרים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0.

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

```
public Time1 (Time1 t)
```

בנוסף הוגדרו במחלקה השיטות:

- שיטות האחזור:

○ `int getHour()` המחזירה את השעות.

○ `int getMinute()` המחזירה את הדקות.

- השיטות הקובעות:

○ `void setHour (int num)` המשנה את ערכה של השעה להיות `num`. אם `num`

הוא לא בתחום 0-23, הערך של `_hour` לא משתנה.

○ `void setMinute (int num)` המשנה את ערכה של הדקה להיות `num`. אם `num`

הוא לא בתחום 0-59, הערך של `_minute` לא משתנה.

- השיטה toString() שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים. יש להקפיד שהשיטה תחזיר מחרוזת בת 5 תווים בדיוק.

```
public String toString()
```

- int minFromMidnight () – שיטה שמחזירה כמה דקות עברו מאז חצות הלילה. למשל, אם הזמן המאוחר באובייקט הוא 07:30, יוחזר הערך 450.

- השיטה equals המקבלת כפרמטר זמן מסוים ובודקת אם הוא זהה לזמן שמיוצג על ידי האובייקט עליו מופעלת השיטה. אם כן, השיטה תחזיר true ואם לא, יוחזר false.

```
public boolean equals (Time1 other)
```

- השיטה before המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה קודם בזמן לאובייקט שמתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. למשל, 07:30 הוא לפני 13:24.

```
public boolean before (Time1 other)
```

- השיטה after המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה מאוחר בזמן לאובייקט שמתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. השיטה הזו משתמשת אך ורק בשיטה before שהוגדרה לעיל. אסור להשתמש בשום אופרטור אחר או שיטה אחרת.

```
public boolean after (Time1 other)
```

- השיטה difference המקבלת כפרמטר זמן מסוים ומחזירה את ההפרש בדקות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. שימו לב, אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.

```
public int difference(Time1 other)
```

עליכם לכתוב את המחלקה Time1 לפי ההגדרות לעיל.

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

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונות _hour ו- _minute. מותר להגדיר קבועים נוספים למחלקה.

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

שאלה 2 - 35 נקודות

המחלקה Time2 מייצגת זמן, לפי מספר הדקות שעברו מאז חצות הלילה ועד לזמן שמייצג האובייקט.

לדוגמא, אם הזמן שמיוצג על ידי האובייקט הוא: 07:30 (כלומר 7 בבוקר, 30 דקות), הוא ייוצג על-ידי הערך 450 שכן, $7 * 60 + 30 = 450$

למחלקה Time2 יש, אם כן, תכונה פרטית (instance variable) אחת, והיא מספר הדקות הזה.

`int _minFromMid`

עליכם לכתוב את המימוש של המחלקה Time2.

ה-API של שתי המחלקות Time1 ו-Time2 זהה לחלוטין! רק הייצוג הפנימי של האובייקטים (התכונות) שונה.

<i>Time2 (int h, int m)</i>	בנאים
<i>Time2 (Time2 other)</i>	אם אחד הפרמטרים שהתקבל בבנאי שמקבל 2 ערכים אינו בתחום הנכון, אז לא משתמשים בערכו כדי לחשב את הערך של התכונה הפרטית. (כלומר משנים את ערכו של כל פרמטר שאינו בתחום ל-0)
<i>int getHour()</i>	שיטות מאחזרות
<i>int getMinute()</i>	
<i>void setHour (int num)</i>	שיטות קובעות בשיטות הקובעות אם הפרמטר אינו תקין יש להשאיר את התכונה ללא שינוי
<i>void setMinute (int num)</i>	
<i>int minFromMidnight ()</i>	שיטה המחזירה כמה דקות עברו מאז חצות הלילה
<i>boolean equals(Time2 other)</i>	שיטה הבודקת האם הזמנים שווים
<i>boolean after(Time2 other)</i>	השיטה בודקת האם הזמן שעליו הופעלה השיטה מאוחר לזמן שהתקבל כפרמטר (השיטה הזו יכולה להשתמש אך ורק בשיטה <i>before</i> ולא בהשוואות נוספות)

<i>int difference(Time2 other)</i>	שיטה המחזירה את ההפרש בדקות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.
<i>String toString()</i>	שיטה המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים. יש להקפיד שהשיטה תחזיר מחרוזת בעלת 5 תווים בדיוק.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונה `_minFromMid`.

מותר להגדיר קבועים נוספים למחלקה.

במילים אחרות, חתימות השיטות של המחלקה `Time1` זהות לחלוטין לאלו של `Time2`, לבד מהמקרים בהם מתקבל זמן כפרמטר לשיטה, ואז במקום שכתוב `Time1` צריך להיות `Time2`.

שימו לב, גם השיטה `toString` של המחלקה `Time2` צריכה להיות לפי זו של המחלקה `Time1`, כלומר להדפיס את הזמן לפי שעות ודקות ולא לפי מספר הדקות מאז חצות.

אין להשתמש בשיטות ובבנאים של המחלקה `Time1` במחלקה `Time2`. מדובר במימושים חלופיים למחלקה של הזמן.

שאלה 3 - 30 נקודות

המחלקה Flight מייצגת טיסה.

למחלקה Flight התכונות הפרטיות (instance variables) הבאות:

- String _origin – שם העיר ממנה ממריאה הטיסה.
- String _destination – שם העיר בה נוחתת הטיסה.
- Time1 _departure – זמן ההמראה של הטיסה.
- int _flightDuration – משך זמן הטיסה בדקות.
- int _noOfPassengers – מספר הנוסעים בטיסה.
- boolean _isFull – האם הטיסה מלאה?
- int price – מחיר לכרטיס טיסה לאדם.

כמו כן קיים במחלקה קבוע שלם MAX_CAPACITY המציין את המספר המקסימלי של נוסעים על טיסה - 250.

למחלקה Flight יש שני בנאים:

- בנאי אחד שמקבל כפרמטרים: שם עיר ההמראה, שם עיר הנחיתה, שני מספרים שלמים המהווים את זמן ההמראה של הטיסה, מספר שלם המייצג את משך זמן הטיסה בדקות, מספר שלם המייצג את מספר הנוסעים בטיסה ומספר שלם המייצג מחיר כרטיס טיסה לאדם.

שימו לב להנחות הבאות:

- אם מספר הנוסעים גדול מהמספר המקסימלי של נוסעים האפשרי, הערך שיינתן לתכונה של מספר הנוסעים יהיה המספר המקסימלי ולא הפרמטר.
- אם מספר הנוסעים קטן מ-0, ינתן הערך 0.
- את התכונה הבוליאנית צריך לקבוע לפי מספר הנוסעים והקבוע המציין את מספר הנוסעים המקסימלי האפשרי.
- אם הפרמטר המייצג את משך זמן הטיסה בדקות קטן מ-0, הערך שיינתן לתכונה של זמן הטיסה יהיה 0.
- אם הפרמטר המייצג את מחיר כרטיס טיסה קטן מ-0, הערך שיינתן לתכונה זו יהיה 0.
- אם המספרים המהווים את זמן ההמראה של הטיסה שגויים, הטיפול בכך יתבצע במחלקה Time1.

- בנאי העתקה המקבל טיסה אחרת, ומעתיק את ערכיה.

במחלקה הוגדרו פעולות get ו-set לפי השמות המקובלים. ראו פרטים מדויקים ב-API

כמו כן הוגדרו השיטות הבאות :

- `Equals` שיטה המקבלת טיסה אחרת כפרמטר ומחזירה `true` אם הטיסה שעליה השיטה מופעלת והטיסה שהתקבלה כפרמטר זהות. הזהות תיקבע לפי שמות הערים של ההמראה והנחיתה וזמן ההמראה.
- `getArrivalTime` שיטה המחשבת את זמן הנחיתה של הטיסה ומחזירה זמן זה.
- `addPassengers` שיטה בוליאנית המקבלת מספר של נוסעים `num`, ומוסיפה אותם לטיסה, אם יש בה מקום. אם יש, היא מחזירה `true`, אם אין מקום לכולם, היא לא מוסיפה אף אחד, ומחזירה `false`. שימו לב שצריך לעדכן גם את התכונה הבוליאנית `isFull` במקרה והיא אמורה להשתנות.
- `isCheaper` שיטה המקבלת טיסה אחרת כפרמטר ומחזירה `true` אם כרטיס לטיסה עליה מופעלת השיטה הוא זול יותר מכרטיס לטיסה שהתקבלה כפרמטר, אחרת יוחזר `false`.
- `totalPrice` שיטה המחשבת את התשלום הכולל שהתקבל מכל נוסעי הטיסה, ומחזירה ערך זה.
- `landsEarlier` שיטה המקבלת טיסה אחרת כפרמטר ומחזירה `true` אם הטיסה עליה מופעלת השיטה נוחתת לפני הטיסה שהתקבלה כפרמטר, אחרת יוחזר `false`.
- `toString` שיטה המחזירה מחרוזת ובה נתוני הטיסה הבאים (בלבד), לפי הפורמט הבא :
`Flight from _origin to _destination departs at _departure. Flight is full.`
אם הטיסה לא מלאה, יודפס `Flight is not full`.

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`.

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

שימו לב ששמנו טסטרים לשלוש המחלקות באתר הקורס. חובה שטסטרים אלו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרים לא ירוצו ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

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 : Time2.java, Time1.java, Flight.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: 2018 מועד אחרון להגשה: 9.12.2017

(ת)

במטלה זו אנו משתמשים במחלקות Time1 ו-Flight שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקות Time1 ו-Flight שכתבתם או בקבצים Time1.class ו-Flight.class שיהיו באתר בספר הדיגיטלי של יחידות 5 – 6 בצמוד למטלה 13. נשים את הקבצים האלו 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%)

המחלקה Airport מייצגת את לוח הטיסות בשדה התעופה ביממה.

הייצוג נעשה על-ידי מערך ששומר את רשימת הטיסות. התכונות במחלקה הן:

- מערך של הטיסות `Flight [] _flightsSchedule`
- מספר הטיסות בלוח הטיסות (במערך) `int _noOfFlights`
- שם העיר בה נמצא שדה התעופה `String _airport`

כמו כן קיים במחלקה קבוע שלם MAX_FLIGHTS המציין את המספר המקסימלי של טיסות ביממה – 200.

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

עליכם לכתוב את המימוש ב-Java של המחלקה Airport. מימוש המחלקה כולל את הסעיפים שלהלן:

1. הגדרת הקבועים של המחלקה.
2. הגדרת התכונות של המחלקה.
3. בנאי שמקבל את שם העיר בה נמצא שדה התעופה ומאתחל את תכונות המחלקה כך שמערך הטיסות יהיה בגודל מקסימלי.
4. שיטה (addFlight) בוליאנית המוסיפה טיסה ללוח הטיסות. השיטה מקבלת את הטיסה כפרמטר. השיטה מחזירה ערך true אם ההוספה התבצעה כשורה, אם לא, השיטה תחזיר false. שימו לב שהמקור או היעד של הטיסה חייבים להיות העיר בה נמצא שדה התעופה.
5. שיטה (removeFlight) בוליאנית המוחקת טיסה מלוח הטיסות. השיטה מקבלת את הטיסה כפרמטר. השיטה מחזירה ערך true אם המחיקה התבצעה כשורה, אם לא, השיטה תחזיר false.
6. שיטה (firstFlightFromDestination) המקבלת עיר כלשהי place, מחזירה את הזמן בו ממריאה הטיסה הראשונה באותו יום מהמקום place. אם אין אף טיסה באותו יום מהמקום place יוחזר null.
7. שיטה (howManyFullFlights) המחזירה מספר האומר כמה טיסות מלאות יש באותו יום.
8. שיטה (howManyFlightsBetween) המקבלת שתי ערים, city1 ו-city2 ומחזירה מספר האומר כמה טיסות יש באותו יום הממריאות מ-city1 ונוחתות ב-city2, או ממריאות מ-city2 ונוחתות ב-city1.
9. שיטה (mostPopularDestination) המחזירה את העיר הכי פופולרית באותו יום (כלומר העיר בה נוחתות הכי הרבה טיסות).
10. שיטה (mostExpensiveTicket) המחזירה את הטיסה שהכרטיס שלה הוא היקר ביותר.
11. שיטה (longestFlight) המחזירה את הטיסה הארוכה ביותר במערך הטיסות.
12. שיטה (toString) המחזירה מחרוזת המתארת את כל הטיסות במערך הטיסות כסדרן לפי המערך (אין צורך למיין לפי זמנים או משהו אחר). כל טיסה תהיה בשורה נפרדת. ובתחילה תהיה כותרת. ראו את הדוגמא הבאה:

The flights for airport Tel-Aviv today are:

Flight from Tel-Aviv to London departs at 12:00. Flight is full

Flight from New York to Tel-Aviv departs at 10:50. Flight is full

Flight from Tel-Aviv to Paris departs at 11:35. Flight is not full

לפניכם רשימת החתימות של הבנאי ושיטות המחלקה:

- `public Airport(String city)`
- `public boolean addFlight(Flight f)`
- `public boolean removeFlight(Flight f)`
- `public Time1 firstFlightFromDestination (String place)`
- `public String toString()`
- `public int howManyFullFlights()`
- `public int howManyFlightsBetween (String city1, String city2)`
- `public String mostPopularDestination()`
- `public Flight mostExpensiveTicket()`
- `public Flight longestFlight()`

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 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 `Airport.java`, עטפו אותו בקובץ `zip` ושלחו. אין לשלוח קבצים נוספים.

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: 2018 מועד אחרון להגשה: 23.12.2017

(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 int myFunc(char x, int y)
```

אילו מהחתימות הבאות אינה מהווה העמסה חוקית של השיטה myFunc?

- א. `public int myFunc()`
- ב. `public double myFunc(char y, int x)`
- ג. `public void myFunc(int x)`
- ד. `public double myFunc(double a, int b)`

שאלה 2

נתונה המחלקה הבאה:

```
public class A {  
    public double method1(String s) { ... }  
}
```

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

דריסה (overriding) של השיטה method1?

- א. `public int method1(String y)`
- ב. `private int method1(char x)`
- ג. `public void method1()`
- ד. `public double method1(String y)`

שאלה 3

נניח שיש בפרויקט ארבע מחלקות – TV, Laptop, Computer, ElectronicDevice, 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 TV, Laptop, Computer כולן יורשות מהמחלקה ElectronicDevice
- ב. המחלקות TV ו- Computer יורשות מהמחלקה ElectronicDevice.
- ג. המחלקות TV ו- Computer יורשות מהמחלקה ElectronicDevice והמחלקה Laptop יורשת מהמחלקה Computer
- ד. המחלקה ElectronicDevice יורשת מהמחלקות TV ו- Computer, Laptop
- ה. המחלקה TV יורשת מהמחלקה ElectronicDevice והמחלקה Laptop יורשת מהמחלקה Computer

שאלה 4

נתונות 3 מחלקות A, B, ו- C כאשר מחלקה B יורשת מהמחלקה A והמחלקה C יורשת מהמחלקה B. כמו כן נתון שלמחלקה A מוגדרת התכונה `_x` האם יש גישה מתוך הקוד של המחלקה C לתכונה `_x` של המחלקה A?

- א. מתוך המחלקה C אין גישה לתכונות של המחלקה A אפילו אם הם מוגדרות כ- `protected` מכיוון שמדובר בשתי רמות של ירושה
- ב. יש גישה רק בתנאי שהתכונה `_x` במחלקה A מוגדרת כ- `protected`
- ג. יש גישה לתכונה `_x` של המחלקה A ע"י הפקודה `super.super._x`
- ד. יש גישה בכל מקרה מכיוון שהמחלקה C יורשת בעקיפין מהמחלקה A

שאלה 5

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 שיורשת 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

שאלה 6

אילו מהשיטות/בנאים הבאים לא מוגדרים במחלקה Object?

- א. `public String toString()`
- ב. `public boolean equals(Object obj)`
- ג. `public Object()`
- ד. `public void println(String s)`

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. שינוי המאפיין `name_` ל-`protected` יביא לשגיאת ריצה.
- ב. שינוי המאפיין `name_` ל-`public` יביא לשגיאת ריצה.
- ג. שינוי המאפיין `name_` ל-`default` יביא לשגיאת ריצה.
- ד. הרצת התוכנית תביא לשגיאת ריצה.
- ה. שינוי המאפיין `name_` ל-`Protected` יביא לשגיאת קומפילציה.
- ו. שינוי המאפיין `name_` ל-`public` יביא לשגיאת קומפילציה.
- ז. שינוי המאפיין `name_` ל-`default` יביא לשגיאת קומפילציה.
- ח. הרצת התוכנית תביא לשגיאת קומפילציה.

הכתוב להלן מתייחס לשאלות 8 - 11

לפניכם המחלקות C, D. כל אחת בקובץ נפרד, כמובן:

```
public class C
{
    public void foo (D d)
    {
        System.out.println("cd");
    }
}
```

```
public class D extends C
{
    public void foo(C c)
    {
        System.out.println("dc");
    }

    public void foo(D d)
    {
        System.out.println("dd");
    }
}
```

בשיטה main שנמצאת במחלקה Driver באותו פרוייקט, נכתבו הפקודות הבאות:

```
public class Driver
{
    public static void main (String [] args)
    {
        C cd = new D();
        D dd = (D)cd;

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

השאלות בלתי תלויות אחת בשניה.

שאלה 8

```
System.out.println(dd.foo(dd));
```

בעקבות הפעלת השורה

- א. תהיה שגיאת קומפילציה
- ב. תהיה שגיאת זמן ריצה
- ג. על הפלט יודפס cd
- ד. על הפלט יודפס dc
- ה. על הפלט יודפס dd

שאלה 9

```
System.out.println(dd.foo(cd));
```

בעקבות הפעלת השורה

- א. תהיה שגיאת קומפילציה
- ב. תהיה שגיאת זמן ריצה
- ג. על הפלט יודפס cd
- ד. על הפלט יודפס dc
- ה. על הפלט יודפס dd

שאלה 10

```
System.out.println(cd.foo(dd));
```

בעקבות הפעלת השורה

- א. תהיה שגיאת קומפילציה
- ב. תהיה שגיאת זמן ריצה
- ג. על הפלט יודפס cd
- ד. על הפלט יודפס dc
- ה. על הפלט יודפס dd

שאלה 11

```
System.out.println(cd.foo(cd));
```

בעקבות הפעלת השורה

- א. תהיה שגיאת קומפילציה
- ב. תהיה שגיאת זמן ריצה
- ג. על הפלט יודפס cd
- ד. על הפלט יודפס dc
- ה. על הפלט יודפס dd

הכתוב להלן מתייחס לשאלות 12 - 18

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 12

System.out.println(a3.equals(a1));

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 13

`System.out.println(a4.equals(a2)) ;`

בעקבות הפעלת השורה

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

שאלה 14

`System.out.println(a1.equals(a2)) ;`

בעקבות הפעלת השורה

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

שאלה 15

`System.out.println(a2.equals(b1)) ;`

בעקבות הפעלת השורה

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

שאלה 16

```
System.out.println(b1.equals(a1));
```

בעקבות הפעלת השורה

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

שאלה 17

```
System.out.println(b2.equals(b1));
```

בעקבות הפעלת השורה

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

שאלה 18

```
System.out.println(b1.equals(a4));
```

בעקבות הפעלת השורה

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

שאלה 19

מה נוכל להגיד בוודאות על חלק התכנית הבא, שרץ ללא שגיאה?

אין להניח שנעשה שימוש במוסכמות כלשהן לגבי השמות (naming conventions).

```
plane.rise (meter) ;
```

- א. rise חייב להיות שיטה.
- ב. meter חייב להיות משתנה פשוט (primitive).
- ג. rise חייב להיות שדה (instance variable).
- ד. plane הוא בהכרח שם של מחלקה.
- ה. plane הוא בהכרח אובייקט.
- ו. אף אחד מהתשובות אינה נכונה.

שאלה 20

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 2 ו-3 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: 2018 מועד אחרון להגשה: 13.1.2018

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)

כתבו שיטה סטטית המקבלת כפרמטרים מערך ממיון a של מספרים שלמים, ומספר שלם נוסף x . השיטה צריכה להחזיר את מספר המופעים של המספר x במערך a . חתימת השיטה היא:

```
public static int count (int[] a, int x)
```

לדוגמא,

עבור המערך

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-5	-5	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	67	67	99

והמספר $x = -5$ השיטה תחזיר את הערך 2.

עבור אותו מערך והמספר $x = 2$ השיטה תחזיר את הערך 5.

עבור אותו מערך והמספר $x = 8$ השיטה תחזיר את הערך 0.

שימו לב:

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

אל תשכחו לתעד את השיטה שכתבתם.

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

שאלה 2 - 25 נקודות (להגשה)

בהינתן סדרה כלשהי של $2n$ סיביות (ביטים – bits), שיש בה n אפסים ו- n אחדים, אפשר לחשב את המספר המינימלי של החלפות בין שתי סיביות הדרושות כדי ליצור את הסדרה המסורגת (alternating sequence) של סיביות, כלומר 01010101...01 או 10101010...10.

לדוגמא, עבור הסדרה 00011011 הפלט יהיה 2, כיון שמספיקות שתי החלפות בין סיביות כדי להגיע לסדרה 01010101: החלפה אחת היא בין הסיביות המודגשות 0001**1**011 וההחלפה השניה היא בין הסיביות המודגשות 000110**1**1 (יכולות להיות גם שתי החלפות אחרות, למשל אחת היא 0001**1**011 והשניה היא 000110**1**1)

עליכם לכתוב שיטה סטטית המקבלת מחרוזת תווים שיש בה $2n$ תווים שהם אך ורק '0' או '1', והיא מכילה n תווי '0' ו- n תווי '1' בדיוק. אפשר להניח זאת ואין צורך לבדוק זאת! השיטה צריכה להחזיר את המספר המינימלי של ההחלפות כמתואר לעיל.

חתימת השיטה היא :

```
public static int alternating (String s)
```

שימו לב, לא משנה אם ההחלפות יביאו את המחרוזת למצב "0101...01" או "1010...10". המספר המוחזר מהשיטה צריך להיות המינימלי מבין אלו שיביאו למצב אחד או למצב השני. לדוגמא, עבור המחרוזת "00101011", כדי להגיע למחרוזת "01010101" דרושות שלוש החלפות, בעוד שלהגיע למחרוזת "10101010" מספיקה החלפה אחת בלבד (התו הראשון והאחרון), ולכן השיטה צריכה להחזיר 1.

בפתרון הבעיה אסור להשתמש בשיטות מהמחלקה String, פרט לשיטות charAt ו-length , בהן מותר בשימוש.

שימו לב:

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

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם. אל תשכחו לתעד את מה שכתבתם!

שאלה 3- 25 נקודות (להגשה)

נתון מערך שמלא במספרים שלמים חיוביים ממש.

נגדיר מסלול חוקי במערך כסדרה של אינדקסים במערך, המתחילה באינדקס 0 (התא הראשון במערך) ומתקדמת במערך מספר צעדים ימינה או שמאלה לפי הערך שבתא. שימו לב שאי אפשר להתקדם מעבר לגבולות המערך. המסלול צריך להסתיים בתא האחרון במערך.

כתבו שיטה **רקורסיבית** בוליאנית המקבלת מערך מלא בנתונים, ומחזירה true אם ישנו מסלול חוקי במערך, ו- false אחרת.
חתימת השיטה היא:

```
public static boolean isWay(int[] a)
```

דוגמאות:

• עבור המערך:

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

התשובה שתוחזר תהיה true שכן ישנו מסלול שמתחיל בתא 0, הולך שני תאים ימינה לתא 2, משם תא אחד שמאלה לתא 1, ומשם ארבעה תאים ימינה לתא 5, משם שוב שני תאים ימינה לתא 7, משם שלושה תאים שמאלה לתא 4 ומשם ארבעה תאים ימינה לתא 8 שהוא האחרון במערך.

• עבור המערך:

0	1	2	3	4	5	6
1	4	3	1	2	4	3

התשובה שתוחזר תהיה false שכן אין אף מסלול שמתחיל בתא 0 ומגיע לתא 6 שהוא האחרון במערך, לפי הקפיצות ימינה או שמאלה. אם נתחיל בתא 0, נוזז תא אחד ימינה לתא אחד, משם חייבים לזוז ימינה ארבעה תאים, כי שמאלה אי אפשר בגלל גבולות המערך, וכך מגיעים לתא 5, משם חייבים לזוז שמאלה ארבעה תאים, כי ימינה אי אפשר, (שימו לב שמתא 5 אין אפשרות התקדם 4 צעדים ימינה, כיוון שתא 6 הוא כבר סוף המערך) ולכן מגיעים שוב לתא 1, כך שזהו תהליך אינסופי שלא מגיע לתא 6 לעולם.
שימו לב להימנע מרקורסיות אינסופיות כאלו.

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

מוותר להשתמש במערך עזר.

אפשר לשנות את המערך המקורי.

אפשר להשתמש בהעמסת-יתר (overloading).

אסור להשתמש במשתנים סטטיים (גלובליים)!

שאלה 4- 25 נקודות (להגשה)

נתון מערך דו-ממדי המכיל בתוכו מספרים שלמים שונים זה מזה. נגדיר כי זוג אינדקסים (i,j) מהווים **גבעה** (hill) כאשר הערך בתא $[i][j]$ גדול (ממש) מכל ארבעת שכניו הקיימים, 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.

לדוגמא, במערך הבא, התאים $[1][1]$, $[1][3]$, $[3][3]$ הם גבעות.

	0	1	2	3
0	3	8	7	1
1	5	15	2	4
2	2	1	3	2
3	4	6	7	52

כתבו שיטה סטטית **רקורסיבית** המקבלת כפרמטר מערך דו-ממדי כמתואר ומדפיסה מסלול אפשרי מהתא $[0][0]$ עד לגבעה כלשהי במערך. המסלול צריך להיות "מסלול טיפוס", כלומר, אפשר לעבור מתא אחד לשני רק אם עולים (ערך המספר בתא הנוכחי קטן ממש מערך המספר בתא אליו הולכים). כמוכך שמותר לעבור מתא אחד רק לארבעת שכניו (כמתואר לעיל). שימו לב שכיון שכל איברי המערך שונים זה מזה, בהכרח נובע כי חייבת להיות גבעה אחת לפחות במערך.

לדוגמא, במערך לעיל פלט אפשרי יהיה: $(0,0)$ $(0,1)$ $(1,1)$
או: $(0,0)$ $(1,0)$ $(2,0)$ $(3,0)$ $(3,1)$ $(3,2)$ $(3,3)$

חתימת השיטה:

```
public static void printPath (int[] [] mat)
```

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 לשנות את המערך המקורי. אסור להשתמש במערך עזר.

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: 1

מועד אחרון להגשה: 20.1.2018

סמסטר: 2018א

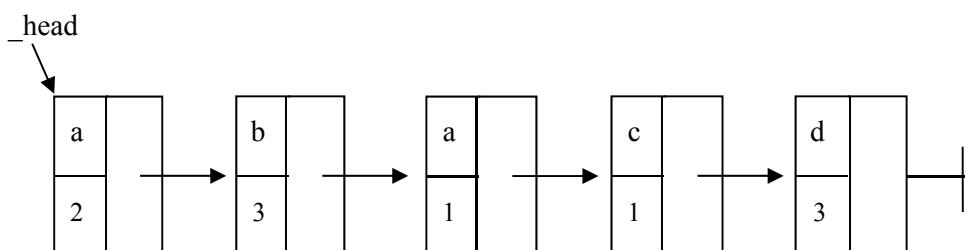
שאלה 1 - להרצה (100%)

אנו רוצים לייצג מחרוזת תווים ברשימה מקושרת.

בכל צומת ברשימה יהיו שלושה שדות:

1. האות עצמה;
2. מספר הפעמים שהיא מופיעה ברצף;
3. מצביע לצומת הבא ברשימה.

כך למשל המחרוזת "aabbbacddd" תיוצג על-ידי הרשימה הבאה:



המחלקה CharNode שלהלן מייצגת צומת ברשימה:

```
public class CharNode
{
    private char _data;
    private int _value;
    private CharNode _next;

    public CharNode(char c, int val, CharNode n) {
        _data = c;
        _value = val;
        _next = n;
    }
}
```

```

public CharNode getNext( )      { return _next; }
public void setNext(CharNode node) { _next = node; }
public int getValue()           { return _value; }
public void setValue(int v)     { _value = v; }
public char getData()           { return _data; }
public void setData(char c)     { _data = c; }
}

```

המחלקה StringList מייצגת את הרשימה כולה :

```

public class StringList
{
    private CharNode _head;

    public StringList( ) {
        _head = null;
    }

    public StringList(CharNode node) {
        _head = node;
    }
}

```

עליכם להוסיף למחלקה StringList שיטות לפי המפורט:

שימו לב, אלו שיטות מהמחלקה String ואנו רוצים לממש אותן בעזרת הייצוג של מחרוזת על-ידי רשימה כמפורט לעיל.

- `public char charAt(int i)` - המחזירה את התו במקום ה-`i` במחרוזת. הניחו שערכו של `i` נמצא בטווח הנכון.
- `public StringList concat (String str)` - המחזירה מחרוזת המורכבת מהמחרוזת עליה היא מופעלת ובסופה משורשרת המחרוזת `str`.
- `public int indexOf (int ch)` - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של התו `ch`. אם התו `ch` לא מופיע במחרוזת, יוחזר -1. **שימו לב**, הפרמטר המעביר את התו `ch` בשיטות `indexOf` הוא מסוג `int` ולא `char`. כך זה במחלקה `String`, ואם נעביר תו ולא מספר שלם, בעצם יועבר ערך הקוד שלו, וזה בסדר.
- `public int indexOf (int ch, int fromIndex)` - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של התו `ch`, כאשר החיפוש מתחיל באינדקס `fromIndex`. אם התו `ch` לא מופיע במחרוזת, יוחזר -1. אם ערכו של `fromIndex` לא נמצא בטווח, השיטה תחזיר -1.

- `public boolean equals (String str)` - המחזירה `true` אם המחרוזות עליה היא מופעלת וזהה למחרוזת `str`. אחרת יוחזר `false`. את השיטה הזו עליכם לכתוב ברקורסיה, ללא שימוש בלולאות בכלל.
- `public int compareTo (String str)` - המשווה בין המחרוזות עליה מופעלת השיטה למחרוזת `str` שבפרמטר. השיטה מחזירה את הערך 0 אם המחרוזות שוות. אם המחרוזות שבאובייקט קטנה לקסיקוגרפית מהמחרוזת `str` שבפרמטר יוחזר מספר שלילי, ואם המחרוזות שבאובייקט גדולה לקסיקוגרפית מהמחרוזת `str` יוחזר ערך חיובי.
- `public StringList substring(int i)` - המחזירה את הסיפא המתחילה במקום ה- `i` במחרוזת עליה היא מופעלת. כלומר, את התת-מחרוזת מהמקום ה- `i` עד לסוף המחרוזת. לדוגמא, אם `s = "abc"` אז `s.substring(1)` יחזיר את `"bc"`. הניחו שערכו של `i` נמצא בטווח הנכון.
- `public StringList substring(int i, int j)` - המחזירה את התת-מחרוזת המתחילה במקום ה- `i` ומסתיימת במקום ה- `j` (לא כולל) במחרוזת עליה היא מופעלת. לדוגמא, אם `s = "abcdefg"` אז `s.substring(2,5)` יחזיר את `"cde"`. הניחו שהערכים של `i` ושל `j` נמצאים בטווח הנכון.
- `public int length()` - המחזירה את אורך המחרוזת עליה היא מופעלת.

דוגמאות –

אם `s = "abcae"`, אז:

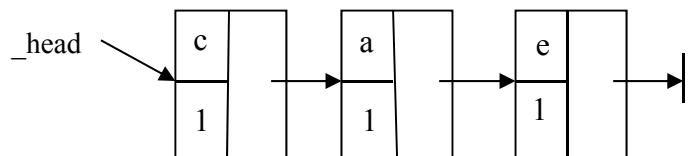
`s.length() = 5`

`s.charAt(1) = b`

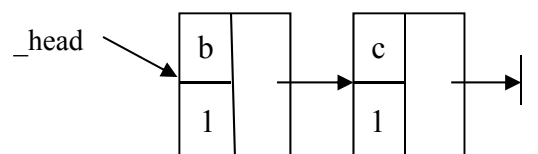
`s.indexOf('a') = 0`, `s.indexOf('a',2) = 3`, `s.indexOf('d') = -1`

`s.compareTo("abcde") = ערך חיובי`

`s.substring(2)` תחזיר את הרשימה הבאה:



`s.substring(1,3)` תחזיר את הרשימה הבאה:



שימו לב לכל מקרי השגיאה האפשריים!

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

שימו לב:

1. אסור להעביר למחרוזת ולהשתמש בפעולות מחרוזת.
2. אסור להשתמש במחלקות מוכנות כבר של Java.
3. מותר ורצוי להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים.

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 קבצי Java הבאים: StringList.java, CharNode.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: 2018 מועד אחרון להגשה: 29.1.2018

(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!

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. FIFO הוא מושג שמתאר התנהגות של תור – האחרון שנכנס הוא הראשון שיוצא.

ב. FIFO הוא מושג שמתאר התנהגות של מחסנית – האחרון שנכנס הוא הראשון שיוצא.

ג. FIFO הוא מושג שמתאר התנהגות של תור – הראשון שנכנס הוא הראשון שיוצא.

ד. FIFO הוא מושג שמתאר התנהגות של מחסנית – הראשון 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. 234561

ג. 125643

ד. 345612

ה. 654321

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 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 ובחזרה מהראש אל הזנב
- ד. מדפיסה את הערכים שבמחסנית s1 מראש המחסנית אל זנבה ובחזרה מהזנב אל הראש

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) - 4;
for (int i = 0; i < len; i++){
    x = s1.pop();
    s2.push(x);
}

len = length(s1) - 4;
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-4 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 ארבעה איברים והערך של x יהיה זהה לערך בתחתית המחסנית $s1$.
- ו. $s1$ תכלול ארבעה איברים והערך של x יהיה זהה לערך שבראש המחסנית $s2$.
- ז. אף אחת מהתשובות א-ו לעיל אינה נכונה.

```

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.	2	3	4	5	6	7
ב.	7	6	5	4	3	2
ג.	5	6	2	7	4	3
ד.	6	7	3	4	2	5

שאלה 6. לפניהם תיאור של שלושה מקרים :

- i. רשימה של הזמנת חדרים למלון. ההזמנות ממולאות על בסיס הראשון שנרשם הוא הראשון שמקבל חדר.
- ii. ניסיון של עכבר למצוא נתיב מילוט במבוך. העכבר צריך להיות מסוגל לחזור על צעדיו חזרה מהסוף לנקודת ההתחלה כשהוא מנסה לברוח.
- iii. אחסון של צעדים במשחק איקס-מיקס-דריקס. על המערכת להיות מסוגלת להציג סרט של התקדמות המשחק בסיום.

מהו מבנה הנתונים המתאים ביותר לשימוש בכל אחד מהמקרים המתוארים (נניח בפתרון בעיה על בסיס המקרים האלו)?

- | | | |
|---------------------|-----------|----------------|
| א. i מערך ממוין | ii מחסנית | iii מחסנית |
| ב. i תור | ii מחסנית | iii מערך ממוין |
| ג. i תור | ii מחסנית | iii תור |
| ד. i עץ חיפוש בינרי | ii מחסנית | iii תור |

הכתוב להלן מתייחס לשאלות 7 – 9 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())
    {
        DataNode node = q.retrieve();
        secret (q);
        q.enqueue (node);
    }
}
```

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

שאלה 7.

אם התור q2 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 q1 צריך להיות:

א. אף תור q1 לא יחזיר ערך true בקריאה לעיל.

ב. {2, 3, 6, 1, 4, 5}

ג. {5, 4, 1, 6, 3, 2}

ד. {4, 5, 6, 1, 2, 3}

ה. {4, 1, 6, 3, 2, 5}

ו. {5, 5, 5, 2, 2, 2}

שאלה 8.

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

- בודקת אם לתורים q_1 ו- q_2 יש אותו מספר איברים.
- בודקת אם התורים q_1 ו- q_2 מכילים אותו איבר בראש התור ובזנב התור.
- בודקת אם התורים q_1 ו- q_2 מכילים אותם איברים (הסדר לא חשוב)
- בודקת אם התורים q_1 ו- q_2 מכילים אותם איברים (הסדר כן חשוב)

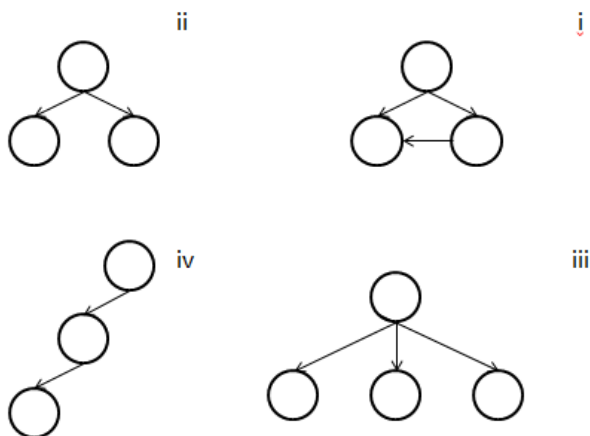
שאלה 9.

כאשר השיטה secret מקבלת כפרמטר תור q , היא :

- הופכת את הסדר בין איברי התור.
- מחזירה את התור כמו שהוא.
- מחזירה את התור כשבראשו נמצא האיבר המקורי ואחריו שאר האיברים בסדר כלשהו
- אי אפשר לדעת מה עושה השיטה באופן כללי.

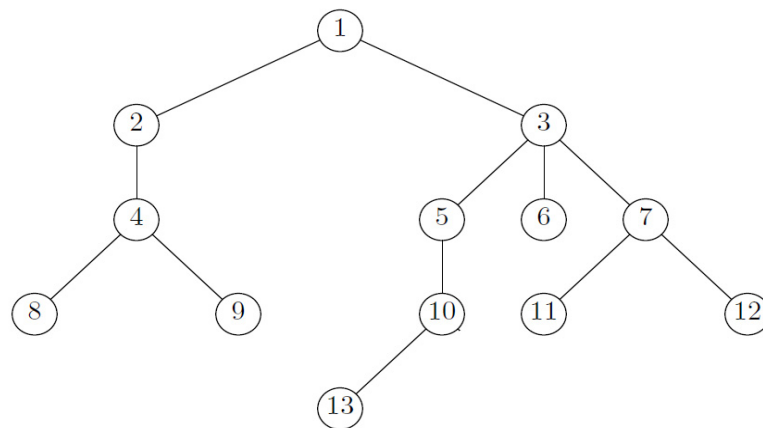
שאלה 10. לפניכם ארבעה תרשימים כאשר כל עיגול מייצג צומת מסוג Node. קבעו עבור כל אחד

מהם את התיאור המתאים ביותר עבורו :



- | | | | |
|------------|-------------|-----------|-------------|
| 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 |

שאלה 11. נתון עץ ששורשו הוא הצומת 1 :



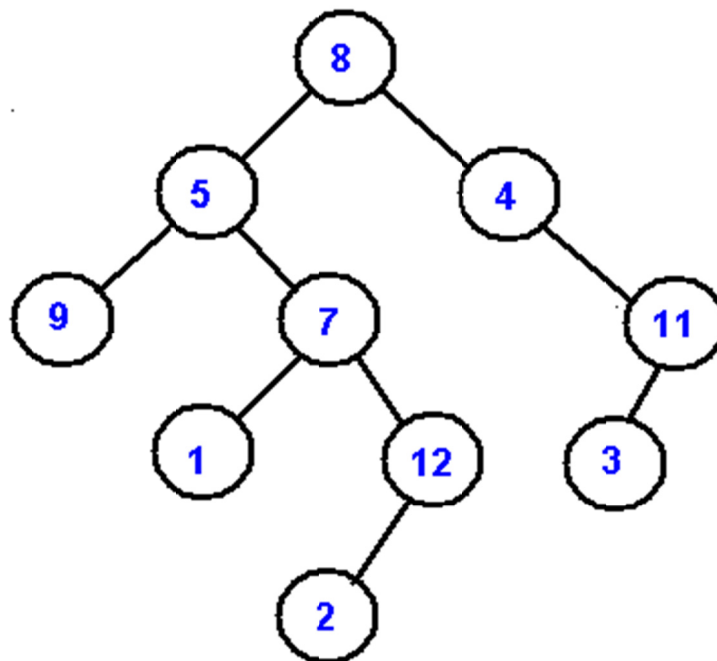
נתונות הטענות הבאות :

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. אף תשובה אינה נכונה
- ה. כל הטענות נכונות

שאלה 12. נתון העץ הבינרי הבא



נתונות הטענות הבאות :

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

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 13. לאחר הפעלת סיור InOrder ו PostOrder על עץ בינארי כלשהו התקבלו התוצאות הבאות (משמאל לימין):

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

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

אם נפעיל סיור PreOrder 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. 1 2 3 4 5 6 7 8 9 10

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

שאלה 14. לאחר הפעלת סיור PreOrder על עץ חיפוש בינארי כלשהו התקבלה התוצאה הבאה :

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

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

א. 7 8 9 4 3 1 2 5 6 10

ב. 1 4 3 2 5 7 8 9 6 10

ג. 7 8 10 9 6 4 5 2 3 1

ד. 10 9 8 7 6 5 4 3 2 1

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

קטע הקוד הבא מתיחס לשאלות 15 ו-16.

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

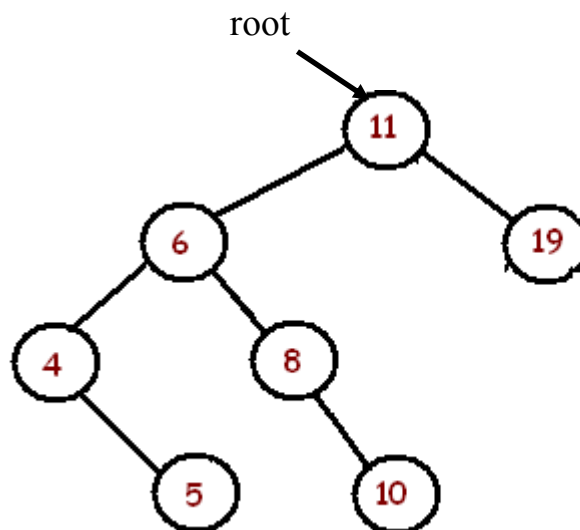
```
public int what(Node root)
{
    if(root == null)
        return 0;
    int x = what(root.getLeft());
    int y = what(root.getRight());
    return root.getValue() + f(x,y);
}

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

השיטה what מקבלת כפרמטר מצביע לשורש העץ T .

שאלה 15.

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



סמנו את התשובה הנכונה

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

שאלה 16.

סמנו את התשובה הנכונה :

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

השאלות 17 - 19 להלן מתייחסות לקטע הבא.

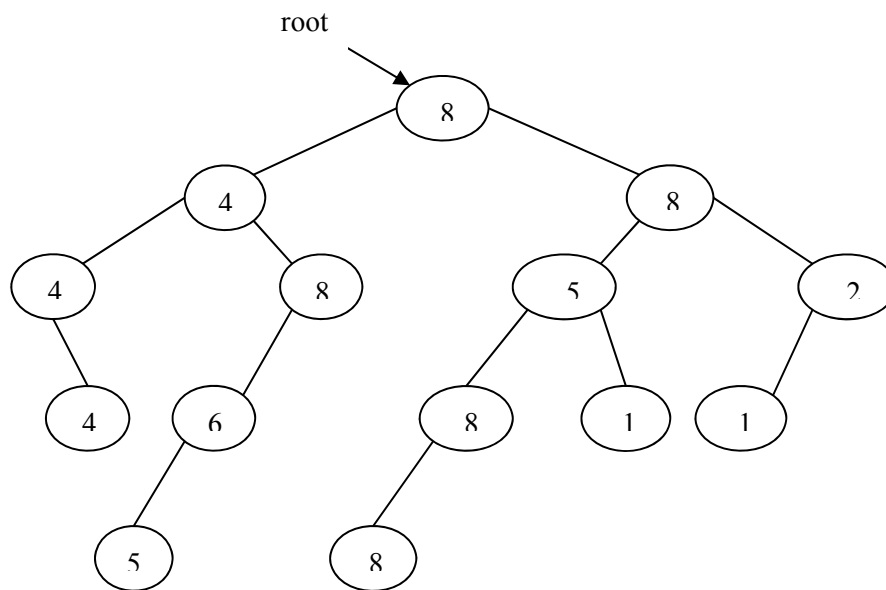
המחלקה Node מממשת צומת בעץ בינרי (לפי ההרצאה ביחידה 12).

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 17.

מה יודפס כתוצאה מהפקודה :

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

א. false

ב. true

שאלה 18.

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 שתי הטענות א וגם ב

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

שאלה 19.

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 ערכים עוקבים אחד לשני
- אף אחת מהתשובות לעיל (א – ד) אינה נכונה

שאלה 20. בהינתן עץ חיפוש בינרי, איזה מהמשפטים הבאים אינו נכון?

- שורש העץ הוא בהכרח האיבר המינימלי בעץ
- אם רוצים להוסיף איבר מסוים x לעץ יש רק מקום אחד שמתאים לו בעץ כך שעדיין העץ ימשיך להיות עץ חיפוש בינרי
- אם נתקדם בעץ כל פעם ימינה (כשאפשר) 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.
- בין המשפטים א-ד יש יותר ממשפט אחד שאינו נכון.

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