

האוניברסיטה הפתוחה

20441

**מבוא למדעי המחשב**

**ושפת Java**

חוברת הקורס – אביב 2024

כתבה: תמר וילנר

מרץ 2024 – סמסטר אביב – תשפ"ד

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנטים
ד	1. לוח זמנים ופעילויות
ו	2. תיאור המטלות
ו	2.1 מבנה המטלות
ז	2.2 חומר הלימוד הדרוש לפתרון המטלות
ז	2.3 ניקוד המטלות
י	3. התנאים לקבלת נקודות זכות בקורס
1	ממ"ן 11
7	ממ"ן 12
15	ממ"ן 13
23	ממ"ן 14



## אל הסטודנטים

אנו מקדמים את פניכם בברכה עם הצטרפותכם אל הלומדים בקורס "מבוא למדעי המחשב ושפת Java".

הקורס מבוסס על הרצאותיהם של ד"ר אמיר גורן ותמר וילנר. ההרצאות המצולמות נמצאות באתר הקורס ומלוות במצגות. חוברות השקפים שקיבלתם הן אלו המלוות את סרטי הוידאו שבאתר.

לקורס קיים אתר באינטרנט בו תמצאו את חומרי הלמידה בקורס. בנוסף, באתר הקורס תמצאו חומרים כתובים ומצולמים במהלך הקורס. גם הם חובה ללימוד הקורס. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס.

אתר הקורס הוא למעשה "ביתו" של הקורס. באתר תמצאו הסברים, הפניות והבהרות נוספות. כמו כן, באתר קיימת קבוצת דיון המאפשרת לכם לשאול שאלות את צוות הקורס וגם להתייעץ עם הסטודנטים האחרים בקורס. בקבוצת הדיון יינתן מקום לשאלת שאלות בעניין חומר הלימוד, המטלות, סביבת העבודה ועוד.

אתר הקורס בנוי בשיטת ה"רצפים" (רצפי למידה מקוונים), הרעיון הוא שהתכנים בקורס מחולקים למנות קטנות, המאורגנות באתר הקורס בצורה סדורה על פי רצף למידה המותאם ללוח הזמנים של הסמסטר. מנות התוכן מוצגות בסוגי מדיה מגוונים למשל וידאו, תרגילים אינטראקטיביים, סימולציות ועוד, המעשירות את חוויית הלמידה של הסטודנטים ומאפשרות לתרגל באופן אקטיבי את החומר הנלמד.

בסביבת הלמידה אוֹפֵּל פותח פורמט ייחודי המותאם לפדגוגית רצפי הלמידה. בפורמט הרצפים ממשק המשתמש והעיצוב הויזואלי מדגישים לסטודנטים ולסטודנטיות את רצף הלמידה המומלץ. האתר מוצג לכל סטודנט וסטודנטית במבט אישי בהתאם לקצב ההתקדמות במהלך הסמסטר.

מומלץ להיכנס לאתר לפחות פעמיים בשבוע, לעקוב אחר ההודעות בלוח ההודעות ובקבוצת הדיון. בבעיות טכניות כגון קושי בכניסה לחלקים שונים באתר וכד' אנא פנו אל המוקד בטלפון 09-7782222 או בדואר. אנא עדכנו / בדקו את פרטיכם על-מנת שתיכללו ברשימת התפוצה של דואר אלקטרוני של הקורס.

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

**אתר הקורס הוא חלק בלתי נפרד מהקורס, והסטודנטים מחויבים להיכנס אליו ולהתעדכן בכל מה שקורה בו. זהו המקום בו תמצאו את כל ההודעות החשובות לגבי לימוד הקורס, המטלות והבחינות. לא יישלחו הודעות/הוראות כתובות בדואר רגיל או אלקטרוני. כל ההודעות המתפרסמות באתר, כל השינויים וכל ההסברים מחייבים את כל הסטודנטים.**

**גם המטלות שתכתבו יישלחו אלינו דרך אתר הקורס תחת הכפתור "מערכת שליחת מטלות". הסבר מפורט על השימוש במערכת זו יישלח אליכם בנפרד.**

פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ס בכתובת:

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

התכניות בקורס "מבוא למדעי המחשב ושפת Java" נכתבות בשפת התכנות Java, ואנו נעבוד בסביבת העבודה BlueJ שפותחה במיוחד להוראת תכנות בשפת Java. BlueJ פותחה כחלק מפרויקט מחקר אוניברסיטאי על הוראת תכנות מונחה-עצמים למתחילים. המערכת פותחה ומתוחזקת על-ידי קבוצת מחקר משותפת של אוניברסיטת Deakin ממלבורן, אוסטרליה, עם מכון Maersk באוניברסיטה של דרום דנמרק ועם האוניברסיטה של Kent ב-Canterbury, אנגליה. הפרויקט נתמך על-ידי חברת Sun Microsystems.

מתוך הנאמר באתר האינטרנט של BlueJ: (כתובת האתר היא [www.bluej.org](http://www.bluej.org))

*The aim of BlueJ is to provide an easy-to-use teaching environment for the Java language that facilitates the teaching of Java to first year students. Special emphasis has been placed on visualization and interaction techniques to create a highly interactive environment that encourages experimentation and exploration.*

כפי שאמרנו, אנו נעבוד בקורס עם סביבה זו, ותמיכה טכנית תינתן לסביבת עבודה זו בלבד. יחד עם זאת, אתם יכולים לעבוד עם כל סביבה אחרת שתמצאו, אבל עליכם להשתמש אך ורק במרכיבי השפה שאנחנו לומדים אותם בקורס (דהיינו בהרצאות המצולמות של ד"ר אמיר גורן של תמר וילנר).

הורדת סביבת העבודה מהאתר של BlueJ היא חינם. הוראות התקנה ל-BlueJ ול-Java Platform נמצאות באתר שלהם, בנוסף, באתר הקורס, ביחידה 1, בתוך "מדריכי עזר וקישורים" תוכלו למצוא מדריך להתקנת Java. וסביבת העבודה BlueJ, שימו לב שמדריך זה הוא קצת ישן (ובוודאי הגרסאות המצוינות בו), ולכן עדיף לעבוד לפי ההסברים והקישורים שיש באתר של BlueJ עצמו. [www.bluej.org](http://www.bluej.org)

באותו מקום, בתוך "מדריכי עזר וקישורים" שביחידה 1 תמצאו סרט על סביבת העבודה BlueJ. בסרט זה תוכלו לראות איך עובדים עם הסביבה, ואיך כותבים ומריצים בה תכניות ב-Java.

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט [www.openu.ac.il/Library](http://www.openu.ac.il/Library).

בשאלות הקשורות ללימודי מדעי המחשב באופן כללי תוכל לפנות ליועצים האקדמיים מתחום מדעי המחשב, על-פי הרשימה המופיעה בידיעון האקדמי.

כדי לקבל ייעוץ בכל הנוגע לקורס זה, תוכלו לפנות אל תמר וילנר, מרכזת ההוראה של הקורס, בדואר אלקטרוני בכתובת [tami@openu.ac.il](mailto:tami@openu.ac.il). **בכל מכתב חובה לכתוב שם מלא ומספר ת"ז, אחרת לא אוכל לטפל בפניה!**

#### **לתשומת לב הסטודנטים הלומדים בחו"ל:**

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

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

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

**בשל המצב הבטחוני, אני כמעט שלא נמצאת במשרד, ולכן אני לא מקיימת שעות ייעוץ טלפוניות בסמסטר זה. הדרך הטובה ביותר לפנות אלי היא באמצעות הדואר האלקטרוני [tami@openu.ac.il](mailto:tami@openu.ac.il), ולא לשכוח לכתוב שם מלא ומספר ת"ז.**

אני מאחלת לכם לימוד פורה ומהנה.

בברכה,

תמר וילנר

מרכזת ההוראה בקורס

**1. לוח זמנים ופעילויות (2024 / 20441)**

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה* בקבוצות רגילות	תאריך אחרון למשלוח ממ"ן (למנחה)
1	22.03.2024-17.03.2024 (ה תענית אסתר)	יחידות 1 ו-2 – הקדמה ויסודות השפה		
2	29.03.2024-24.03.2024 (א פורים)	יחידה 2 – יסודות השפה	מפגש 1	
3	05.04.2024-31.03.2024	יחידות 3 ו-4 שימוש במחלקות נתונות, וכתיבת מחלקות		ממ"ן 11 6.4.2024
4	12.04.2024-07.04.2024	יחידה 4 – כתיבת מחלקות	מפגש 2	
5	19.04.2024-14.04.2024	יחידה 4 – כתיבת מחלקות		
6	26.04.2024-21.04.2024 (ב-ו פסח)	יחידה 4 – כתיבת מחלקות		
7	03.05.2024-28.04.2024 (א-ב פסח)	יחידה 5 – לולאות	מפגש 3	ממ"ן 12 4.5.2024
8	10.05.2024-05.05.2024 (ב יום הזכרון לשואה)	יחידה 6 – מערכים		
9	17.05.2024-12.05.2024 (ב יום הזיכרון, ג יום העצמאות)	יחידה 9 – יעילות	מפגש 4	

\* 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	24.05.2024-19.05.2024	יחידות 9 ו-10 – יעילות ורקורסיה		
11	31.05.2024-26.05.2024 (א ל"ג בעומר)	יחידה 10 – רקורסיה	מפגש 5	
12	07.06.2024-02.06.2024	יחידה 11 – רשימות	מפגש 6	ממ"ן 13 8.6.2024
13	14.06.2024-09.06.2024 (ד שבועות)	יחידות 11 - 12 רשימות ועצים בינריים	מפגש 7	
14	21.06.2024-16.06.2024	יחידה 12 – עצים בינריים	מפגש 8	ממ"ן 14 22.6.2024

מועדי בחינות הגמר יפורסמו בנפרד

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

## 2. תיאור המטלות

בקורס זה **ארבע** מטלות שעליכם לפתור ולהגיש במהלך הקורס. יש להגיש את מטלות המנחה (ממ"נים) באמצעות **מערכת המטלות המקוונת שנמצאת באתר הקורס**.  
להלן תמצאו הסבר על אופן הפתרון וכיצד לשלוח את המטלה למנחה. אם שאלה במטלה אינה ברורה לכם, אל תהססו להתקשר אל מרכזת ההוראה או אל אחד המנחים (בשעות הייעוץ הטלפוני שלהם **בלבד**) לצורך קבלת הסבר.

### 2.1 מבנה המטלות

המטלות שמוגשות למנחה הן שאלות הרצה. בשאלה הרצה יש לכתוב תכנית ולהריץ אותה במחשב.  
הקפדה על שמות מחלקות ושיטות (ציבוריות), לפי הנדרש, היא הכרחית. כל חריגה מההגדרות (אפילו החלפה בודדת של אות גדולה בקטנה, למשל) תגרום לבדיקה האוטומטית שלנו להיכשל וכתוצאה מכך לנזק בלתי הפיך בציון.

**עליכם לבדוק שהתכנית שכתבת מבצעת את הנדרש ממנה ללא טעויות, ובפרט שהיא עוברת קומפילציה. תכנית שאינה עוברת קומפילציה לא תיבדק!**

#### תיעוד

בכל תכנית צריך להוסיף תיעוד בתחילת התכנית המסביר את האלגוריתם בו השתמשם ואת מבנה התכנית. בגוף התכנית צריך להוסיף תיעוד המסביר מהו תפקידו של כל משתנה, מה מבצע כל קטע חשוב בתכנית וכל הסבר נוסף החשוב להבנת מהלך פעולתה של התכנית (את התיעוד יש לכתוב **באנגלית בלבד**). יש להקפיד על בחירת שמות משמעותיים למשתנים (באנגלית).

במשך הקורס, כאשר נגיע ליחידה הרלוונטית, נלמד איך לתעד את התכניות שלנו בעזרת תיעוד שנקרא API. מרגע זה תצטרכו להגיש את כל הממ"נים מתועדים לפי הנחיות ה-API, כפי שיילמד.

הקפידו על אופן שליחת מטלה – **קובץ דחוס מסוג ZIP בלבד** (לא RAR!), המכיל את כל הקבצים הרלוונטיים לפתרון. במידה ומדובר בפתרון שאלה "יבשה", יש לענות עליה בקובץ מסוג DOC או RTF (לא DOCX!) – מי שלא בטוח, שיפנה למנחה מבעוד מועד. במידה ומדובר בפתרון שאלה "להרצה" יש לצרף את הקבצים מסוג JAVA בלבד (לא CLASS או כל קובץ אחר). כאמור, את כל הקבצים הרלוונטיים (מסוג JAVA ו/או RTF/DOC), יש לארוז בקובץ דחוס אחד מסוג ZIP. **חריגה מהוראות אלו תגרור הורדה בציון.**

הסבר מפורט לגבי שליחת הקבצים המתאימים יינתן בכל מטלה בנפרד. המטלות בקורס זה יוגשו בעזרת מערכת שליחת המטלות שבאתר הקורס.

## 2.2 חומר הלימוד הדרוש לפתרון המטלות

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

מטלה	חומר הלימוד הנדרש לפתרון
ממ"ן 11	יחידות 1 - 2
ממ"ן 12	יחידות 3 - 4
ממ"ן 13	יחידות 9 - 10
ממ"ן 14	יחידה 11

### שימו לב!

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

## 2.3 ניקוד המטלות

ניתן לצבור עד 10 נקודות. חובה להגיש מטלות במשקל מינימלי של 6 נקודות לפחות.

להלן פירוט הניקוד לכל מטלה:

ממ"ן/ממ"ח	ניקוד
11	1
12	3
13	3
14	3

כדי לעבור את הקורס, צריך (בין היתר) להגיש מטלות במשקל של 6 נקודות לפחות. סטודנט שמגיש את כל המטלות, משקל המטלות שלו הוא 10 נקודות. בהתאם למשקל המטלות, נקבע אחוז הציון של בחינת הגמר בציון הסופי.

חישוב אחוז הציון של הבחינה – אם הוגשו מטלות במשקל 10 נקודות, הבחינה שווה 90% מהציון הסופי של הקורס. אם המטלות שהוגשו הן במשקל 6, הבחינה שווה 94%. וכך גם לכל המספרים ביניהם. לדוגמא, אם הוגשו מטלות 11, 12, 13 המשקל שלהן הוא  $7 = 1 + 3 + 3$  (ללא קשר לציון שהתקבל בכל מטלה), ואז משקל הבחינה בציון הסופי הוא 93%.

### איך מחושב ציון הקורס ?

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

#### לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:  
אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי. זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.  
**זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.**

לדוגמא, אם הציונים במטלות ובבחינה הם אלו:

מטלה	משקל	ציון
11	1	50
12	3	50
14	3	90
בחינת הגמר	93	75

משקל הבחינה אמור להיות 93%.

הציון אמור להיות:

$$(50 \times 1 + 50 \times 3 + 90 \times 3 + 75 \times 93) / 100 = 74.45 \rightarrow 74$$

שימו לב שבמטלות 11, 12 הציון נמוך מציון הבחינה. אם נוריד את שתייהן, יישארו 3 נקודות. אם נוריד רק את מטלה 11 יישארו 6 נקודות, ומשקל הבחינה יהיה 94%. אם נוריד רק את מטלה 12 יישארו 4 נקודות, וזה לא מספיק. לכן כדאי להוריד רק את מטלה 11 (כלומר להניח כאילו לא הוגשה כלל).

חישוב הציון ייעשה, אפוא, כך:

$$(50 \times 3 + 90 \times 3 + 75 \times 94) / 100 = 74.70 \rightarrow 75$$

כלומר, הציון הסופי הוא 75.

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס.

סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע

בטלפון 09-7782222 או יעדכנו בעצמם באתר שאלתא <http://www.openu.ac.il/sheilta>

**קורסים** → **ציוני מטלות ובחינות** → **הזנת ציון 0 למטלות רשות שלא הוגשו.**

יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהמוצק המשוקלל של המטלות והבחינה יהיה נמוך מ-60). לכן, אם אתם רוצים להזין ציון 0 במטלה אחת או יותר, בדקו היטב באיזו מטלה כדאי לכם להזין 0, ואל תעשו זאת ליותר מטלות מהדרוש.

**כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.**

### **הכנת המטלות חייבת להיעשות על-ידי כל סטודנט בנפרד.**

**מטלות שלא יבוצעו באופן עצמאי – ייפסלו!!!**

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

### **מועדי הגשת המטלות**

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

### **דחייה בהגשת מטלות**

- אנחנו נאשר איחורים רק כאשר יש בקשה מראש לאיחור כזה. בקשה מנומקת לאיחור של עד שבוע יש להפנות למנחה הקבוצה. במקרים חריגים ביותר של דחייה של למעלה משבוע, תצטרכו לפנות למרכזת ההוראה לקבלת אישור כזה. שוב, האישור צריך להתקבל מלכתחילה ולא בדיעבד. שימו לב שאם תאריך ההגשה של המטלה הוא במוצאי שבת, יש להגיש את הבקשה לדחיה עד יום חמישי (כולל). לא ביום שישי ולא בשבת! אלו אינם ימי עבודה!
- מי שיאחר בהגשת המטלה ללא קבלת אישור מראש, יורדו לו 3 נקודות מהציון לכל יום איחור, וגם זה רק עד שבוע. לאחר שבוע המטלה תיבדק למשוב בלבד, ולא תחשב בשקלול הציון.
- לגבי מילואימניקים – אתם יכולים לפנות אל המנחים שלכם ולסכם איתם את מועדי ההגשה למטלות המתאימים לכם ולמנחים. לכל מטלה המוגשת באיחור צריך מכתב/אישור המנמק את סיבת האיחור.

**נזכיר שוב:**

**לבחינת הגמר רשאי לגשת רק סטודנט שצבר 6 נקודות לפחות בהגשת המטלות.**

**3. התנאים לקבלת נקודות זכות בקורס**

- א) צבירת משקל של 6 נקודות לפחות במטלות.
- ב) ציון של 60 נקודות לפחות בבחינת הגמר.
- ג) ציון סופי בקורס של 60 נקודות לפחות.

# מטלת מנחה (ממ"ן) 11

הקורס: מבוא למדעי המחשב ושפת Java (20441)

נושאי המטלה: יסודות השפה

חומר הלימוד למטלה: יחידות 1-2

משקל המטלה: 1 נקודה

מספר השאלות: 2

מועד אחרון להגשה: 6.4.2024

סמסטר: 2024

(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 ההיקף והשטח של משולש.

התכנית תקרא מהקלט שהכניס המשתמש את האורך של שלוש הצלעות של המשולש, ותדפיס למסך את היקפו של המשולש ואת שטחו. אורכי הצלעות יהיו מספרים טבעיים (כלומר שלמים חיוביים).

שימו לב שייתכן שהמשתמש יכניס שלושה מספרים שלא יכולים להוות אורכי צלעות של משולש (למשל: 1, 2, 4 - למה?). אתם צריכים לבדוק זאת ולהודיע אם זה אכן קרה.

לשם חישוב שטח המשולש, ניתן להשתמש בנוסחת Heron (בויקיפדיה [כאן](#)) הקובעת כי שטח המשולש שווה לשורש הריבועי של  $p(p-a)(p-b)(p-c)$ , כאשר  $a, b$  ו- $c$  הם אורכי שלוש הצלעות של המשולש, ו- $p$  הוא מחצית היקפו. כלומר:

$$p = \frac{a + b + c}{2}$$

$$s = \sqrt{p(p-a)(p-b)(p-c)}$$

עליכם לבדוק את תקינות הקלט. אם הקלט לא תקין, התכנית צריכה להדפיס את נתוני הקלט בליווי הודעות מתאימות.

התכנית שכתבתם צריכה להיות במחלקה בשם `Triangle`.

לעזרתכם, כתבנו כאן את תחילת קוד המחלקה. עליכם להשלים את החסר (גם את התיעוד החסר).

אתם חייבים להשתמש בתחילת הקוד שלנו בדיוק כפי שהוא מופיע כאן.

```
import java.util.Scanner;
public class Triangle
{
    public static void main (String [] args)
    {
        Scanner scan = new Scanner (System.in);
        System.out.println ("This program calculates the area "
            + "and the perimeter of a given triangle. ");
        System.out.println ("Please enter the three lengths"
            + " of the triangle's sides");
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();

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



## לדוגמא,

- אם הנתונים שהמשתמש הכניס הם (משמאל לימין):

3    4    5

(כלומר, הכוונה היא ש-  $a = 3$   $b = 4$   $c = 5$ , אבל בקלט מכניסים רק את המספרים)

**ההדפסה של הפלט חייבת להיות בדיוק לפי הפורמט הבא:**

The lengths of the triangle sides are: 3, 4, 5

The perimeter of the triangle is: 12

The area of the triangle is: 6.0

שימו לב שהערכים הובאו כאן לצורך הדוגמה. אתם צריכים להדפיס לפי הערכים שהוכנסו בקלט.

**דייקו בהדפסות לפי הרווחים וסימני הפיסוק.**

לגבי הדפסת השטח – אין צורך לעגל לספרה אחת לאחר הנקודה, אלא להדפיס את המספר כפי שהתקבל.

- אם הנתונים שהמשתמש הכניס הם (משמאל לימין):

2    4    1

**ההדפסה של הפלט חייבת להיות בדיוק לפי הפורמט הבא:**

The lengths of the triangle sides are: 2, 4, 1

The given three sides don't represent a triangle

## לשם קריאה מהקלט השתמשו במחלקה Scanner.

כדי להשתמש בה צריך לכתוב בראשית התכנית את השורה

```
import java.util.Scanner;
```

אפשר למצוא את הממשק של המחלקה Scanner בתוך סעיף 2.6 Interactive Programs מהספר

Java Software Solutions שנמצא לאחר הסרטון 2.4, שם מובאות חלק מהשיטות.

הסברים על המחלקה והשימוש בה אפשר למצוא באתר הקורס בתוך "יחידה 2" ב"מדריכי עזר

וקישורים" בקובץ "מדריך לעבודה עם המחלקה Scanner לקבלת קלט מהמשתמש".

**על מנת לחשב שורש ריבועי של מספר, ניתן להשתמש בשיטה  $\text{Math.sqrt}(x)$ , שהיא שיטה של**

Java שנמצאת במחלקה  $\text{Math}$ . 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 ( $\text{int}$ ) או ממשי ( $\text{double}$ ). 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  $\text{Math}$ . 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 ( $\text{int}$ ) או ממשי ( $\text{double}$ ). השיטה מחזירה מספר ממשי (גם אם  $a$  בחזקת  $b$  הוא מספר שלם).

## שאלה 2 - להרצה (60%)

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

מתוך [ויקיפדיה](#):

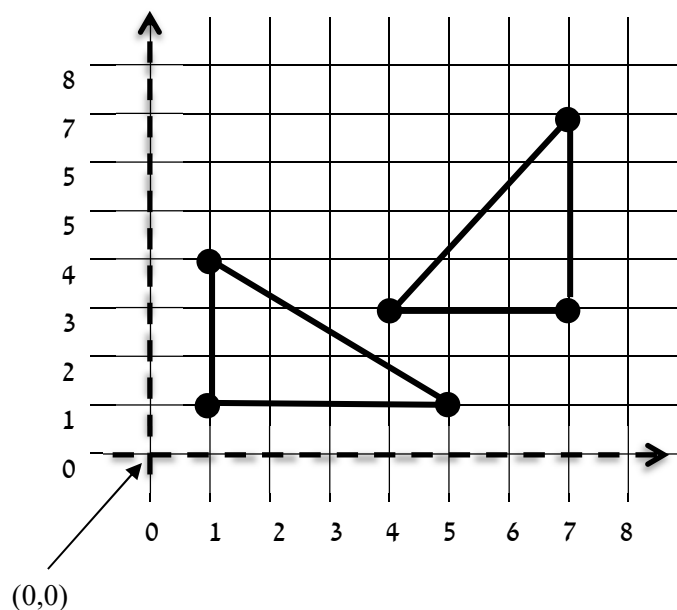
משולשים חופפים הם זוג משולשים שניתן להזיז, לסובב או לשקף אותם כך שהם יתלכדו זה עם זה, כלומר שלוש הצלעות שלהם ושלוש הזוויות שלהם שוות בהתאמה. אינטואיטיבית, שני משולשים חופפים הם בעצם שני עותקים שונים של אותו משולש.

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. התכנית צריכה להדפיס הודעות כמפורט להלן. התכנית צריכה להשתמש במשפט החפיפה "צלע-צלע-צלע".

לדוגמא: שני המשולשים שבאיור להלן חופפים. בדקו זאת.



והפלט שהתכנית תדפיס יהיה:

The first triangle is (1.0,1.0) (5.0,1.0) (1.0,4.0).

Its lengths are 4.0, 5.0, 3.0.

The second triangle is (4.0,3.0) (7.0,3.0) (7.0,7.0).

Its lengths are 3.0, 4.0, 5.0.

The triangles are congruent.

### 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.

במקרה שהמשולשים חופפים יודפס:

The triangles are congruent.

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.

**דייקו בהדפסות לפי הרווחים וסימני הפיסוק.**

להלן דוגמא לקטע קוד הקורא מהקלט את המספרים הממשיים המהווים את הקואורדינטות של שלושת הקדקודים של המשולש הראשון. הקטע אמור להיות בתוך השיטה main כמובן:

```
System.out.println ("Please enter the coordinates of the 3 " +  
                    "vertices of the first triangle:");  
System.out.println ("Enter the first vertex (2 real numbers)");  
double x11 = scan.nextDouble();  
double y11 = scan.nextDouble();  
  
System.out.println ("Enter the second vertex (2 real numbers)");  
double x12 = scan.nextDouble();  
double y12 = scan.nextDouble();  
  
System.out.println ("Enter the third vertex (2 real numbers)");  
double x13 = scan.nextDouble();  
double y13 = scan.nextDouble();
```

### הנחות:

1. אתם יכולים להניח שהמשתמש אכן הכניס שישה זוגות של מספרים ממשיים ולא אותיות או תווים אחרים.
2. אתם יכולים להניח שהמספרים שהוכנסו אכן מהווים קדקודים של שני משולשים.
3. אתם יכולים להניח שששת המספרים הראשונים הם הקואורדינטות של שלושת הקדקודים של המשולש הראשון והשישה הבאים הם הקואורדינטות של שלושת הקדקודים של המשולש השני. אין ערבוב.

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

## בשתי השאלות במטלה זו -

- הקפידו שפלט התוכנית יהיה בדיוק כפי שרשום בהנחיות! אין להוסיף הודעות על מה שנכתב מפורשות בהנחיות ויש לדייק (כולל אותיות גדולות/קטנות וסימני פיסוק) בהודעות המפורטות. אחרת יורדו לכם נקודות!
- אתם צריכים לכתוב את התכנית של כל אחת משתי השאלות במחלקה אחת (מחלקה אחת עבור כל שאלה בשמות Triangle ו-Congruent), והכל בשיטה main. **אין להוסיף שיטות נוספות!**
- **אסור להשתמש בלולאות ו/או במערכים!**

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לשאלה 1 כולל את הקובץ Triangle.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 וכתובת מחלקות

מספר השאלות: 2 משקל המטלה: 3 נקודות

סמסטר: 2024 מועד אחרון להגשה: 4.5.2024

(ת)

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

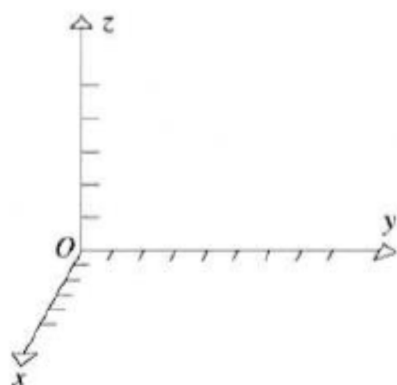
בהרצאות הקורס של ד"ר אמיר גורן, הוגדרה המחלקה Point שמייצגת נקודה במישור, לפי מערכת הצירים הקרטזית (Cartesian system).

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

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

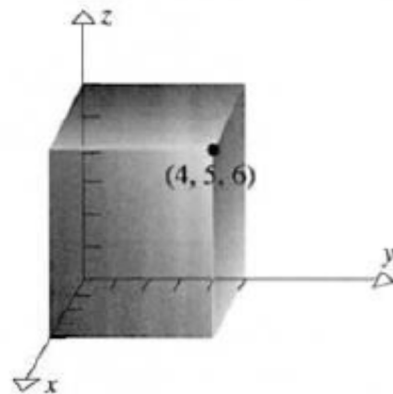
במטלה זו נייצג נקודה במרחב התלת-מימדי, שמוגדרת על ידי שלוש קואורדינטות, על ציר ה-`X`, על ציר ה-`Y` ועל ציר ה-`Z`.

זהו המרחב התלת-מימדי:



אובייקט מסוג נקודה מאופיין על-ידי שלוש הקואורדינטות  $(x, y, z)$  כשכל אחת מהן היא מספר ממשי מטיפוס `double`.

באיור הבא אפשר לראות את הנקודה  $A = (4.0, 5.0, 6.0)$



## שאלה 1 - 30 נקודות

נגדיר מחלקה הנקראת `Point3D` המייצגת נקודה במרחב התלת-מימדי.

למחלקה `Point3D` יש את התכונות הפרטיות (instance variables) הבאות:

- `_x` מסוג `double` – שמייצגת את ערך הקואורדינטה על ציר ה-`X`;
- `_y` מסוג `double` – שמייצגת את ערך הקואורדינטה על ציר ה-`Y`;
- `_z` מסוג `double` – שמייצגת את ערך הקואורדינטה על ציר ה-`Z`;

הבנאים (constructors) המוגדרים במחלקה `Point3D` הם:

- בנאי ריק, שלא מקבל פרמטרים, ומאתחל את הנקודה להיות בראשית הצירים  $(0.0, 0.0, 0.0)$ .
- בנאי המקבל כפרמטרים שלושה מספרים `x, y, z` מטיפוס `double` ומאתחל את הנקודה בהתאמה למספרים אלו  $(x, y, z)$ .
- בנאי העתקה המקבל כפרמטר אובייקט אחר מסוג `Point3D` ומעתיק אותו.

השיטות המוגדרות במחלקה `Point3D` הן:

- שיטות האחזור:
  - `getX()` `double` המחזירה את ערכה של קואורדינטת ה-`x`.
  - `getY()` `double` המחזירה את ערכה של קואורדינטת ה-`y`.
  - `getZ()` `double` המחזירה את ערכה של קואורדינטת ה-`z`.
- השיטות הקובעות:
  - `setX (double num)` `void` המשנה את ערכה של קואורדינטת ה-`x` להיות `num`.
  - `setY (double num)` `void` המשנה את ערכה של קואורדינטת ה-`y` להיות `num`.
  - `setZ (double num)` `void` המשנה את ערכה של קואורדינטת ה-`z` להיות `num`.

## וכן השיטות הבאות:

- `String toString()` - שיטה המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המתמטי המקובל -  $(x,y,z)$ . כך, המחרוזת (3.0,4.0,5.0) מייצגת את הנקודה שקואורדינטת ה-x שלה היא 3.0 קואורדינטת ה-y שלה היא 4.0, וקואורדינטת ה-z שלה היא 5.0. **שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים וללא תווים נוספים.** אין צורך להקפיד בעניין מספר הספרות העשרוניות שאחרי הנקודה.
  - `boolean equals (Point3D other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה והנקודה שהתקבלה כפרמטר זהות.
  - `boolean isAbove (Point3D other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **מעל** לנקודה שהתקבלה כפרמטר, לפי ציר ה-Z. (באיור לעיל, הנקודה A נמצאת מעל לנקודה (0,0,0) ראשית הצירים)
  - `boolean isUnder (Point3D 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, ולא בשיטה equals או בתכונות.**
  - `boolean isLeft (Point3D other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **משמאל** לנקודה שהתקבלה כפרמטר, לפי ציר ה-Y. (באיור לעיל, הנקודה (0,0,0) נמצאת משמאל לנקודה A)
  - `boolean isRight (Point3D 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, ולא בשיטה equals או בתכונות.**
  - `boolean isBehind (Point3D other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **מאחורי** הנקודה שהתקבלה כפרמטר, לפי ציר ה-X. (באיור לעיל, הנקודה (0,0,0) נמצאת מאחורי הנקודה A)
  - `boolean isInFrontOf (Point3D other)` - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **לפני** הנקודה שהתקבלה כפרמטר. **השיטה הזו משתמשת אך ורק בשיטה isBehind שהוגדרה לעיל, ולא בשיטה equals או בתכונות.**
  - `void move (double dx, double dy, double dz)` - המזיזה את הנקודה ב-dx על ציר ה-X, ב-dy על ציר ה-Y וב-dz על ציר ה-Z.
  - `double distance (Point3D p)` - שיטה שמקבלת נקודה כפרמטר ומחזירה את המרחק בין הנקודה שעליה הופעלה והנקודה שהתקבלה כפרמטר.
- חישוב המרחק בין שתי נקודות תלת-ממדיות  $(x_1,y_1,z_1)$  ו-  $(x_2,y_2,z_2)$  נעשה לפי הנוסחה שלהלן:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

בכתיבת השיטה distance ניתן להשתמש בשיטות Math.sqrt, Math.pow

**עליכם לכתוב את המחלקה Point3D לפי ההגדרות לעיל.**

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

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

## שאלה 2 - 70 נקודות

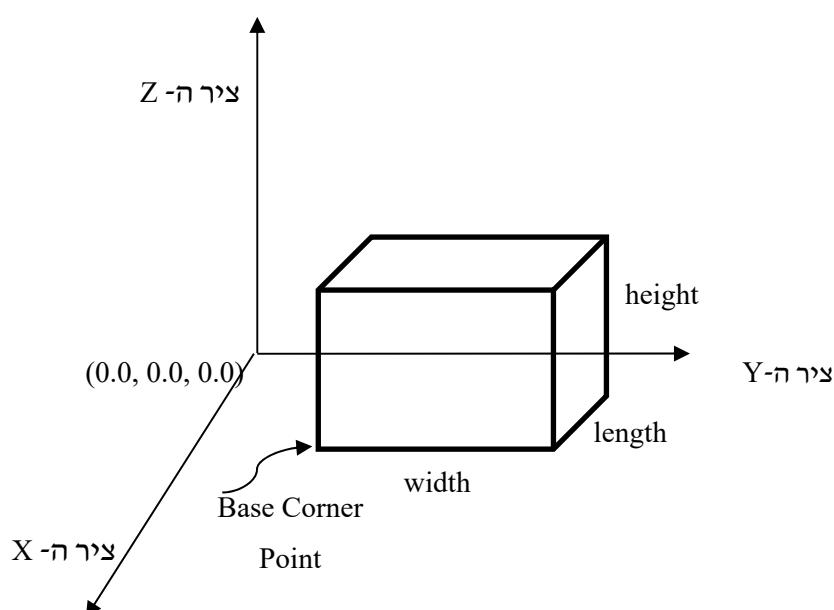
נגדיר מחלקה הנקראת Box3D המייצגת תיבה (box) בשלושה מימדים.

אובייקט מסוג תיבה מאופיין על-ידי המיקום של אחת מפינותיו (התחתונה, שמאלית, קדמית) שאנו נתייחס אליה כאל הפינה הבסיסית של התיבה, ועל-ידי שלושת המימדים שתמיד נמדדים כערכים חיוביים, כלפי מעלה, ימינה וקדימה (כלומר: גובה, רוחב ואורך).

**שימו לב שצלעות התיבה מקבילות לשלושת הצירים.**

השתמשו בטיפוס Point3D כדי לייצג את הפינה הבסיסית של התיבה, ובטיפוס int כדי לייצג את מימדי התיבה (אלו תכונות המחלקה).

זו התיבה שתתקבל:





### למחלקה Box3D יש את התכונות הפרטיות (instance variables) הבאות:

- Point3D\_base – שמייצגת את הפינה הקדמית, שמאלית, תחתונה;
- int\_length – שמייצגת את אורך התיבה (מקביל לציר ה-X);
- int\_width – שמייצגת את רוחב התיבה (מקביל לציר ה-Y);
- int\_height – שמייצגת את גובה התיבה (מקביל לציר ה-Z);

### הבנאים (constructors) המוגדרים במחלקה Point3D הם:

- בנאי ריק, שלא מקבל פרמטרים, ויוצר תיבה כאשר הפינה הבסיסית שלה נמצאת בראשית הצירים (0.0, 0.0, 0.0), והאורך, הרוחב והגובה שלה, כולם שווים ל-1.
- בנאי שמקבל נקודה מהטיפוס Point3D המהווה את הפינה הבסיסית של התיבה, ושלושה מספרים שלמים המהווים את האורך, הרוחב והגובה של התיבה, ויוצר את התיבה לפי נתונים אלו. אם אחד (או יותר) מהמספרים המהווים את האורך, הרוחב והגובה אינו חיובי ממש, הוא מקבל את הערך 1.
- בנאי העתקה המקבל כפרמטר אובייקט אחר מסוג Box3D ומעתיק אותו.

### השיטות המוגדרות במחלקה Box3D הן:

#### • שיטות האחזור:

- int getLength() המחזירה את אורך התיבה.
- int getWidth() המחזירה את רוחב התיבה.
- int getHeight() המחזירה את גובה התיבה.
- Point3D getBase() המחזירה עותק של הפינה הבסיסית של התיבה.

#### • השיטות הקובעות:

- void setLength (int num) המשנה את ערכו של אורך התיבה x להיות num. אם הערך אינו חיובי ממש, לא יתבצע כלום.
- void setWidth (int num) המשנה את ערכו של רוחב התיבה x להיות num. אם הערך אינו חיובי ממש, לא יתבצע כלום.
- void setHeight (int num) המשנה את ערכו של גובה התיבה x להיות num. אם הערך אינו חיובי ממש, לא יתבצע כלום.
- void setBase (Point3D p) המשנה את הפינה הבסיסית של התיבה להיות p. שימו לב להימנע מ-aliasing

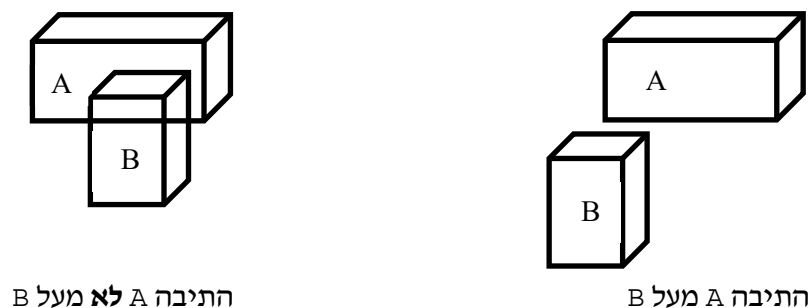
### כמו כן הוגדרו השיטות הבאות:

- `String toString()` - שיטה המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג הבא:  
The base point is XXX, length = XXX, width = XXX, height = XXX  
כאשר במקום XXX יופיעו הערכים שיש בתכונות (לא מודפסות בגופן נטוי).  
**לדוגמא,**  
The base point is (3.0,4.0,5.0), length = 10, width = 14, height = 6  
**שימו לב לדייק במחרוזת לפי הכתוב כאן.** ללא רווחים נוספים וללא תווים נוספים. אין צורך להקפיד בעניין מספר הספרות העשרוניות שאחרי הנקודה.
- `boolean equals(Box3D other)` – שיטה בוליאנית שמקבלת תיבה כפרמטר ומחזירה האם התיבה שעליה הופעלה השיטה והתיבה שהתקבלה כפרמטר זהות.
- `Box3D move(double dx, double dy, double dz)` – שיטה שמקבלת שלושה מספרים ממשיים ומחזירה **תיבה חדשה** שמוזזת לפי הפרמטרים בהתאמה. שימו לב שהתיבה שעליה הופעלה השיטה נשארת ללא שינוי.
- `Point3D getUpRightBackPoint()` המחזירה את **הנקודה** שהיא העליונה-ימנית-אחורית של התיבה.
- `Point3D getCenter()` המחזירה את **הנקודה** שהיא מרכז התיבה.
- `double distance(Box3D other)` – שיטה שמקבלת תיבה כפרמטר ומחזירה את המרחק שבין שתי הנקודות שהן **מרכזי התיבות**.
- `int getVolume()` המחזירה את **נפח** התיבה.
- `int getSurfaceArea()` המחזירה את **שטח הפנים** של התיבה.
- תזכורת מתמטית** – שטח הפנים של התיבה הוא הסכום של שטחי ששת הפאות (faces) של התיבה.
- `boolean isLargerCapacity(Box3D other)` – שיטה בוליאנית המקבלת כפרמטר תיבה אחרת, ומחזירה true אם התיבה שעליה מופעלת השיטה גדולה (מבחינת כמות, נפח) מהתיבה שהועברה כפרמטר ו- false אחרת.
- `boolean contains(Box3D other)` – שיטה בוליאנית המקבלת כפרמטר תיבה אחרת, ומחזירה true אם התיבה שעליה מופעלת השיטה יכולה להכיל **ממש** את התיבה שהועברה כפרמטר ו- false אחרת. כלומר, האם אפשר להכניס **ממש** את התיבה שהועברה כפרמטר לתוך התיבה עליה הופעלה השיטה. **התיבה צריכה להיכנס בלי לסובב אותה.** אין התייחסות לפינה הבסיסית של התיבה, אלא רק לממדי התיבה. שימו לב שהתיבה לא יכולה להכיל תיבה אחרת שאחד או יותר מהמימדים שלה זהים.

שימו לב שיש שוני מהשיטה isLargerCapacity. **לדוגמא**, התיבה A שהאורך, רוחב וגובה שלה הם בהתאמה 3, 10, 5 לא מכילה את התיבה B שהממדים שלה בהתאמה הם 4, 2, 5. אבל בכל זאת A גדולה מ-B מבחינת הנפח. אגב, גם התיבה שממדיה הם 5, 2, 4 לא מכילה **ממש** את B כי יש להן אותם ממדים.

- boolean isAbove (Box3D other) – שיטה בוליאנית המקבלת כפרמטר תיבה אחרת, ומחזירה true אם התיבה עליה מופעלת השיטה נמצאת מעל התיבה שהתקבלה כפרמטר, ו-false אחרת. שימו לב, כדי שיוחזר true, **כל** התיבה העליונה צריכה להיות **ממש מעל** כל התיבה התחתונה.

ראו את הציורים הבאים :



בשיטה זו מותר לכם להשתמש בשיטות של המחלקה Box3D, ומהמחלקה Point3D מותר לכם להשתמש אך ורק בשיטה isAbove.

עליכם לכתוב את המחלקה Box3D לפי ההגדרות לעיל. הגדרות מדויקות לפי API תמצאו באתר הקורס בתת-פרק "מטלה 12" שביחידה 4.

## שימו לב, בכל שאלות המטלה :

- אסור להוסיף תכונות למחלקות. לא פרטיות ולא ציבוריות.
- מותר להוסיף שיטות פרטיות אבל לא ציבוריות.
- בכל השיטות במטלה שמקבלות אובייקט כפרמטר אפשר להניח שמתקבל אובייקט שאותחל ואינו שווה ל-null.
- שימו לב לא לבצע aliasing במקומות המועדים.
- במטלה זו אסור להשתמש בלולאות ו/או במערכים!
- אין להשתמש בחומר שלא נלמד בקורס, כלומר בהרצאות של ד"ר אמיר גורן.

- הקפידו להשתמש בשיטות שכבר כתבתם, גם אם הן במחלקות אחרות, ואל תכתבו מחדש קוד חוזרני.
- הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.
- עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר.

שימו לב ששמנו טסטרס לשתי המחלקות באתר הקורס. חובה שטסטרס אלו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרס ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרס לא ירוצו בגלל שגיאות קומפילציה הציון במטלה יהיה אפס.

## הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת יורדו לכם הרבה נקודות!**
3. עליכם להריץ את הטסטרס שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרס 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 בשני קובצי Java הבאים : Point3D.java, Box3D.java . אין צורך להגיש את קובצי ה-API שכתבתם.
5. ארזו את שני הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

## ב ה צ ל ח ה

# מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

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

מספר השאלות: 4 משקל המטלה: 3 נקודות

סמסטר: 2024 מועד אחרון להגשה: 8.6.2024

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. אנו ממליצים מאוד לענות עליהן ללא הרצה במחשב (כפי שמקובל בבחינת הגמר) ולאחר מכן להריץ.

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex13 (בדיוק). את התשובות לשאלות על הסיבוכיות כתבו (באנגלית בלבד) כחלק מה-API של השאלה הרלוונטית.

## שאלה 1 - 25 נקודות

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

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

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

	0
c =	5

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

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

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

**חתימת השיטה היא:**

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

**שימו לב:**

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד. מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם? הסבירו תשובתכם.

**שאלה 2 - 25 נקודות**

נתון כלי לצבירת מי גשם. חתך הכלי מתואר על ידי מערך הגבהים heights.

למשל, עבור מערך הגבהים {2, 1, 1, 4, 1, 1, 2, 3}

צורת הכלי היא:

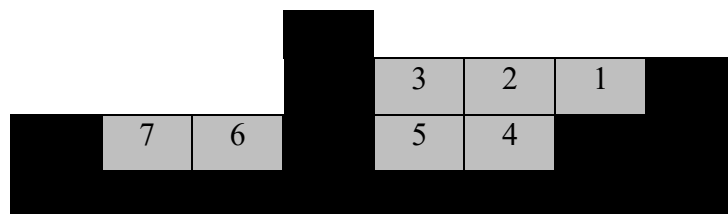


כתבו שיטה סטטית המקבלת כפרמטר מערך גבהים ומחזירה את כמות הגשם שניתן לצבור בכלי עם מערך גבהים נתון.

**ניתן להניח שכל הערכים במערך (הגבהים) הם מספרים שלמים חיוביים ממש.**

למשל, עבור הכלי לעיל, ניתן לצבור 7 יחידות גשם (האיזורים האפורים שבאיור להלן).

רמז: יש לחשב בכל מקום בכלי מה גובה העמודה המקסימלית / המינימלית מימינו ומשמאלו.



**חתימת השיטה היא:**

```
public static int waterVolume (int [] heights)
```

**שימו לב:**

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

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. הסבירו תשובתכם.

## הערות לגבי שאלות 1 ו- 2:

- ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.
- אסור לשנות את המערכים בשתי השאלות, גם אם הם חוזרים לקדמותם לאחר השיטה!
- כתבו (באנגלית בלבד) כחלק מה- API של השאלה מה סיבוכיות הזמן (Time complexity) וסיבוכיות המקום (Space complexity) של השיטה שכתבתם. הסבירו תשובתכם.
- אל תשכחו לתעד (באנגלית) את מה שכתבתם!

## שאלה 3 - 25 נקודות

נתונה המחלקה Password.

למחלקה Password יש בנאי שמקבל מספר שלם חיובי n. הבנאי מייצר סיסמא אקראית באורך n, המורכבת מאותיות הקטנות של האלפבית האנגלי (בלבד). (הבנאי שומר את הסיסמא הזו כתכונה של האובייקט).  
**חתימת הבנאי היא:**

```
public Password (int length)
```

בנוסף במחלקה קיימות השיטות:

1. שיטה ציבורית בשם getPassword המחזירה את הסיסמא הסודית.

**חתימת השיטה היא:**

```
public String getPassword ()
```

2. שיטה ציבורית בוליאנית בשם isPassword המקבלת מחרוזת תווים st, ומחזירה את הערך true אם המחרוזת st זהה בדיוק לסיסמא הסודית שיצר הבנאי, ו- false אחרת.

**חתימת השיטה היא:**

```
public boolean isPassword (String st)
```

עליכם לכתוב שיטה **סטטית רקורסיבית** בשם findPassword שמקבלת אובייקט מהמחלקה Password ואורך סיסמא length. השיטה צריכה לפצח ולהחזיר את הסיסמא.

**חתימת השיטה שלכם צריכה להיות:**

```
public static String findPassword (Password p, int length)
```

אם נכתוב שיטה main כדלהלן :

```
public static void main(String[] args)
{
    Password p = new Password(5);
    System.out.println (p.getPassword());
    System.out.println (Ex13.findPassword(p,5));
}
```

השיטה main תיצור אובייקט מהמחלקה Password ובו סיסמא סודית, ואז תדפיס את הסיסמא הסודית פעמיים. פעם אחת בעזרת השיטה getPassword ופעם שניה כפי שנמצאה לאחר החיפוש. כמובן שהשיטות יצטרכו להחזיר אותה מחרוזת.

שימו לב, המספר 5 כאן הוא רק דוגמא. אי אפשר להניח שאורך הסיסמא הוא 5. אורך הסיסמא אינו ידוע ואינו מוגבל! שימו לב, האורך של המחרוזת אינו קבוע, ובכל זאת אנו ממליצים לא לנסות להריץ עם סיסמא שאורכה גדול מ-6.

המחלקה Password נמצאת באתר הקורס בצורת class בלבד. עליכם להשתמש בה בהנחה שאתם יכולים להשתמש בבנאי ובשיטות שהוזכרו לעיל בלבד!

#### הערות חשובות:

- השיטה שתכתבו צריכה להיות רקורסיבית ללא לולאות בכלל!
- **רמז** - הדרך למצוא את הסיסמא היא על ידי בדיקת כל האפשרויות לסיסמא באורך n, ובדיקת כל אחת מהאפשרויות, אם היא הנכונה.
- אינכם יכולים לכתוב 26 קריאות רקורסיביות לשיטה. חשבו על דרך אחרת!
- בשיטה findPassword שאתם כותבים אסור לכם להשתמש בשיטה getPassword. השיטה getPassword ניתנת לכם רק כדי שתוכלו לבדוק את עצמכם.
- אם ברצונכם להשתמש בשיטות מהמחלקה String בכתיבת השיטה findPassword, מותר לכם להשתמש אך ורק בשיטות הבאות (ולא באף שיטה אחרת מהמחלקה!):  
charAt, equals, length, substring
- שימו לב שאין חובה להשתמש בשיטות אלו (בחלקן או בכלן), אבל אי אפשר להשתמש בשיטות אחרות מהמחלקה String.
- אסור להשתמש באף מחלקה אחרת!

**השיטה שתכתבו צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.**  
ראו הערה חשובה לאחר שאלה 4.



## שאלה 4 - 25 נקודות

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

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

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

0	0	0	0	1
0	1	1	1	0
0	0	1	1	0
1	0	0	0	0
1	1	0	0	0

0	0	0	0	1
0	1	1	1	0
0	0	1	1	0
1	0	0	0	0
1	1	0	0	0

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

**חתימת השיטה היא:**

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

השיטה שתכתבו צריכה להיות **רקורסיבית** ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות. אפשר להשתמש בהעמסת יתר (overloading) וכן מותר לשנות את המערך.

### הערות לגבי שאלות 3 ו-4:

- מותר להשתמש בהעמסת-יתר (Overloading)
- אסור להשתמש במשתנים סטטיים (גלובליים)!
- אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לתעד את מה שכתבתם!

בשאלה 3 מותר להשתמש בשיטות `charAt`, `equals`, `length` ו- `substring` מהמחלקה `String`.

בשאלה 4 מותר לשנות את המערך במהלך השיטה, אבל בסופה הוא צריך לחזור למצבו המקורי.

### שימו לב:

- בכל השאלות - אל תשכחו לתעד (באנגלית בלבד) את מה שכתבתם!
- שמנו טסטר למחלקה `Ex13` באתר הקורס. חובה שהטסטר ירוץ ללא שגיאות קומפילציה עם המחלקה שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה.
- אם הטסטר לא ירוץ ללא שגיאות קומפילציה הציון במטלה יהיה אפס **ללא אפשרות ערעור**.
- אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.

### הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות יהיו **בדיוק** כפי שמוגדר בממ"ן.
3. עליכם לתעד (באנגלית בלבד) את כל השיטות שאתם כותבים בתיעוד API ובתיעוד פנימי המסביר מה עשיתם בשיטה. בתיעוד זה כתבו גם מה הסיבוכיות של השיטות (בשאלות 1 ו-2).
4. את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם `Ex13` (**בדיוק**). ארוז את הקובץ בתוך קובץ `zip`. אין לשלוח קבצים נוספים.

### בהצלחה

## 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

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)

## 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) ;

# מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

נושא המטלה: רשימות מקושרות

חומר הלימוד למטלה: יחידה 11

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 22.6.2024

סמסטר: 2024ב

במטלה זו נעסוק ברשימות מקושרות של מספרים שלמים.

בכל השאלות נשתמש ברשימה דו-סטטית. לשם כך מצורפים לממ"ן זה קובץ המחלקה `IntNodeTwo.java` המייצגת חוליה ברשימה, וקובץ המחלקה `IntListTwo.java` המייצגת רשימה מקושרת כזו.

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

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

הורידו את שתי המחלקות הללו אל המחשב שלכם ועבדו על הקובץ `IntListTwo.java`. אל תמחקו ואל תשנו דבר בקבצים שהעתקתם, רק הוסיפו להם !

הערות חשובות לגבי כל השאלות בממ"ן זה:

- אסור לכם להעביר את תוכן הרשימה למערך ואז לעבוד על המערך. כך גם לא להעביר את תוכן הרשימה לרשימה אחרת ועבודה עליה. פתרון בגישה כזאת יוביל להורדה של כמעט כל הנקודות על השאלה.

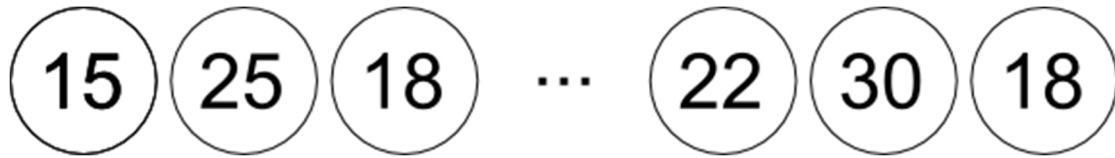
- מותר לשיטות שאתם כותבים לשנות את תוכן או את מבנה הרשימה עליה אתם עובדים. עם זאת כאשר השיטה מסיימת את עבודתה, המבנה ותוכן הרשימה חייבים להיות בדיוק כמו שהיו לפני הקריאה לשיטה.
- אסור להוסיף תכונות למחלקות `IntNodeTwo` ו-`IntListTwo`. קבועים מותר להוסיף.
- אפשר להניח שהפרמטר שמתקבל לכל אחת מהשיטות אינו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

## הערות חשובות לגבי שאלות 1 ו- 2 :

- השיטות שתכתבו צריכות להיות יעילות ככל הניתן מבחינת סיבוכיות הזמן. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות זמן גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.
- בחישוב הסיבוכיות צריך לחשב גם את הזמן של שיטות העזר, ככל שתצאו לכתוב כאלו.
- סיבוכיות המקום של השיטות שלכם צריכה להיות קבועה :  $O(1)$  . בין השאר, אין לכתוב פתרון רקורסיבי לשאלות אלו.
- כתבו (באנגלית בלבד) כחלק מה- API של השאלה מה סיבוכיות הזמן (Time complexity) של השיטות שכתבתם. הסבירו תשובתכם.

## שאלה 1 - 25 נקודות

אמיר ותמר משחקים במשחק. על לוח המשחק מונחים  $K$  מטבעות בשורה.  $K$  הוא מספר זוגי. לכל מטבע בשורה יש ערך שהוא מספר חיובי (ממש, גדול מ-0), והשחקנים יודעים בתחילת המשחק את ערכיהם של כל המטבעות.



המשחק מתנהל בתורות, לסירוגין. בכל תור, השחקן שמשחק בוחר מטבע מאחד משני קצוות השורה ולוקח אותו לקופה שלו. לאחר  $K$  תורות נגמרים המטבעות בשורה. בשלב זה סופרים את סכום ערכי המטבעות שבקופה של כל אחד מהשחקנים. השחקן שצבר סכום גדול יותר, מנצח במשחק. במקרה של שוויון בסכומים, המשחק מוכרז כתיקו.

אמיר משחק ראשון. הוא ממש לא רוצה להפסיד. לא אכפת לו אם המשחק יסתיים בתיקו או בניצחון שלו. מצאו אסטרטגיה עבור אמיר שתבטיח שהוא לא יפסיד במשחק.

כאשר תורה של תמר, היא **תמיד** תיקח את הגדול מבין שני המטבעות שבקצוות.

כמובן שאמיר יכול לחשב מראש את עץ המהלכים המלא של המשחק: איך תמר יכולה להגיב לכל מהלך שלו, ואז איך הוא יגיב לכל מהלך שלה, וכו'. הבעיה בפתרון כזה היא שהעץ הזה עצום: מספר המשחקים השונים שאמיר ותמר יכולים לשחק הוא  $2^K$  ואם  $K$  גדול, גם למחשב החזק ביותר בעולם, החישוב הזה ייקח טריליוני שנים.

לכן, אנחנו מעוניינים בפתרון **יעיל** לבעיה – כזה שדורש מאמיר לבצע מעט פעולות כדי לחשב לעצמו אסטרטגיה. כזכור,  $K$  (מספר המטבעות) הוא זוגי, כדי שלא יהיה יתרון במספר המטבעות לאחד מהצדדים.

נניח שרשימה מקושרת מסוג `IntListTwo` מכילה את ערכי המטבעות לפי סדר הופעתם בשורה שעל לוח המשחק.

הוסיפו למחלקה `IntListTwo` שיטה בשם `win` המבטיחה שאמיר יגיע אל הרווח המירבי בסוף המשחק. מותר לכם להניח שאמיר יודע מהי האסטרטגיה של תמר כפי שהוזכרה לעיל.

השיטה מדפיסה את בחירות השחקנים בכל שלב במשחק. בסוף התהליך יודפס מה היה הסכום הכולל של אמיר ומה הסכום הכולל של תמר.

השיטה מחזירה את ההפרש בין מה שהרוויח אמיר למה שהרוויחה תמר.

**חתימת השיטה היא :**

```
public int win ()
```

והיא מופיעה ריקה בקובץ `IntListTwo.java` המצורף לממ"ן זה. מלאו אותה בקוד שלכם.

**לדוגמא**, אם רשימת המטבעות שבמערך היא (הערך משמאל הוא `_head` של הרשימה, הערך מימין הוא `_tail` של הרשימה) :

`15 ↔ 19 ↔ 21 ↔ 13 ↔ 14 ↔ 30 ↔ 23 ↔ 16 ↔ null`

הפלט המצופה ייראה כך :

Amir took 16

Tamar took 23

Amir took 30

Tamar took 15

Amir took 19

Tamar took 21

Amir took 13

Tamar took 14

Final Score:

Amir total 78

Tamar total 73

והשיטה תחזיר 5, תוצאת החיסור 73 – 78.

אפשר להניח שברשימת המטבעות יש לפחות שני מטבעות, שאורך הרשימה זוגי, וכן שערכי כל המטבעות גדולים ממש מ-0.

**ושוב, קראו את ההערות הרשומות לעיל, הן בהתייחס לכל השאלות והן בהתייחס לשאלות היעילות (1 ו-2) !**



## שאלה 2 - 25 נקודות

מצאו במחלקה IntListTwo את השיטות what ו-f.

- א. מה מבצעת השיטה what על רשימה מלאה במספרים? הסבירו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת. כלומר, כתבו מה המשמעות של הערך המוחזר מהשיטה what, ומה מודפס על הפלט? (3 נק')
- ב. מה סיבוכיות הזמן והמקום הכוללת של השיטה what? (1 נק')
- ג. כתבו במחלקה IntListTwo שיטה betterWhat שמקבלת גם היא פרמטר יחיד int num, כך שתבצע בדיוק את מה שמבצעת what, אך בסיבוכיות זמן ריצה טובה יותר. השיטה מופיעה ריקה בקובץ IntListTwo.java המצורף לממ"ן זה. מלאו אותה בקוד שלכם. (20 נק')
- ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (1 נק')

את התשובות לסעיפים א ב ד כתבו ב-API של השיטה betterWhat.

אפשר להניח שברשימה עליה עובדת השיטה יש לפחות צומת אחד, וכן שהערכים שבצמתים הם רק חיוביים (גדולים מ-0).

ושוב, קראו את ההערות הרשומות לעיל, הן בהתייחס לכל השאלות והן בהתייחס לשאלות היעילות (1 ו-2) !

## הערות חשובות לגבי שאלות 3 ו-4 :

- השיטות שתכתבו צריכות להיות ללא כל שימוש בלולאות מכל סוג שהוא, כלומר עליהן להיות רקורסיביות.
- מותר לכם לכתוב שיטות עזר, אך גם הן חייבות להיות ללא כל שימוש בלולאות מכל סוג שהוא.
- כל שימוש בלולאה בתשובה לאחת מהשאלות 3 או 4 יוביל להורדה משמעותית ביותר של נקודות.
- אפשר להשתמש בהעמסת-יתר (overloading).
- אין צורך לדאוג ליעילות השיטות הרקורסיביות, ואין צורך לציין מה סיבוכיות הזמן שלהן. בקורסים מתקדמים יותר תלמדו כיצד לשלב רקורסיה ויעילות.

## שאלה 3 - 25 נקודות

נתונה רשימה של מספרים שלמים list. נגדיר ש-list1 היא רשימת-בת (sub-list) של list אם כל הערכים שנמצאים ב-list1 נמצאים גם ב-list באותו סדר בדיוק, אבל הם לא חייבים להיות רצופים.

למשל אם הרשימה list היא  $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5$ , אזי כל אחת מהרשימות הבאות היא רשימת-בת שלה :

$1$ ,  $2 \leftrightarrow 3$ ,  $1 \leftrightarrow 3 \leftrightarrow 4$ ,  $2 \leftrightarrow 5$ ,  $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5$

לעומת זאת הרשימות הבאות אינן רשימות-בת של list :

$3 \leftrightarrow 2$ ,  $1 \leftrightarrow 1 \leftrightarrow 2$ ,  $6$ ,  $5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1$ ,  $1 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 3$

נתונות שתי רשימות של מספרים שלמים, list1 ו-list2. אנו מחפשים את רשימת-הבת הגדולה ביותר המשותפת ל-list1 ול-list2.

כיתבו במחלקה IntListTwo שיטה ציבורית longestCommonSublist המקבלת כפרמטר רשימה, ומחזירה את האורך המקסימלי של רשימת-הבת המשותפת לרשימה עליה מופעלת השיטה (this) ולרשימה שהשיטה מקבלת כפרמטר.

**חתימת השיטה היא:**

```
public int longestCommonSublist(IntListTwo list2)
```

והיא מופיעה ריקה בקובץ IntListTwo.java המצורף לממ"ן זה. מלאו אותה בקוד שלכם.

לדוגמה בהינתן הרשימות הבאות :

L1 : 1↔2↔-3↔4↔5↔-6↔7↔8↔9

L2 : 2↔-3↔10↔0↔-15↔-6↔8

L3 : -15↔10↔4

L4 : -56↔156↔-12

הערך שהשיטה longestCommonSublist תחזיר על L1 עם L2 הוא 4  
(האורך של 2↔-3↔-6↔8).

הערך שהשיטה longestCommonSublist תחזיר על L2 עם L3 הוא 1 (האורך של -15 או 10, בשני המקרים רשימה עם איבר בודד).

הערך שהשיטה longestCommonSublist תחזיר על L4 עם כל אחת מהרשימות האחרות הוא 0, כיון שאין להן שום ערך במשותף.

אפשר להניח שברשימה עליה עובדת השיטה יש לפחות צומת אחד. הערכים שבצמתים יכולים להיות כלשהם : חיוביים, שליליים ואפילו 0.

## שאלה 4 - 25 נקודות

כתבו במחלקה IntListTwo שיטה ציבורית maxEqualValue המחזירה את האורך המקסימלי של תת-רשימה רציפה גדולה ביותר שהיא חלק מהרשימה עליה מופעלת השיטה (this), ואשר כל הערכים בה זהים.

לדוגמה,

אם הרשימה עליה מופעלת השיטה היא 2 ↔ 0 ↔ -3 ↔ -3 ↔ -3 ↔ -3 ↔ -1 ↔ -1 ↔ -1

השיטה תחזיר 4, כיון שבין תתי הרשימות הרציפות שכל ערכיהן זהים הארוכה ביותר היא

-3 ↔ -3 ↔ -3 ↔ -3. שאורכה 4.

אם הרשימה עליה מופעלת השיטה היא 7 ↔ -9 ↔ 7 השיטה תחזיר 1 כיון שאין בה שני ערכים זהים רצופים זה לזה.

**חתימת השיטה היא :**

```
public int maxEqualValue()
```

והיא מופיעה ריקה בקובץ IntListTwo.java המצורף לממ"ן זה. מלאו אותה בקוד שלכם.

אפשר להניח שברשימה עליה עובדת השיטה יש לפחות צומת אחד. הערכים שבצמתים יכולים להיות כלשהם : חיוביים, שליליים ואפילו 0.

**ושוב, קראו את ההערות הרשומות לעיל, הן בהתייחס לכל השאלות והן בהתייחס לשאלות הרקורסיה (3 ו-4) !**

## **הגשה:**

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת המחלקה לא תעבור קומפילציה והציון יהיה 0.**
3. עליכם להריץ את הטסטר שנמצא באתר הקורס על המחלקה שכתבתם. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקה. מאד מומלץ להוסיף לו בדיקות. שימו לב שאם הטסטר לא יעבור קומפילציה מול המחלקה שכתבתם, הציון על המטלה יהיה אפס. אם יש שיטה שאתם מעוניינים לדלג עליה, עליכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטר יעבור קומפילציה.
4. **גם במטלה זו – אם הוספתם הדפסות שלא ביקשנו בשיטות שכתבתם, כדי להיעזר בהן בפתרון השאלה, עליכם למחוק הדפסות אלו לפני ההגשה. הדפסות מיותרות כאלו יורידו בניקוד.**
5. את התשובות לכל השאלות יש להגיש בקובץ Java ששמו `IntListTwo` שאנחנו שמנו באתר. אין להוסיף אף קובץ אחר!
6. שימו לב שהקובץ שאתם שולחים חייב להיות אותו הקובץ שאנחנו שמנו באתר, רק עם התוספות שלכם.
7. שימו לב שהתכונות של ראש וזנב הרשימה בקובץ `IntListTwo` לא הוגדרו ב-private. זאת לא טעות אלא נועד להקל על בדיקת הממ"ן שלכם. כלומר, באופן יוצא דופן (ובניגוד למה שעליכם לעשות בכל הקשר אחר) בראש המחלקה שלכם צריך להופיע בדיוק כך :  
`IntNodeTwo _head, _tail;`  
ולא  
`private IntNodeTwo _head, _tail;`
8. **אין להגיש את קובץ ה-API שכתבתם, וגם לא את הקובץ `IntNodeTwo`**
9. **ארזו את הקובץ `IntListTwo.java` בקובץ zip יחיד ושלחו אותו בלבד.**

## **ב ה צ ל ח ה**