

# 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.

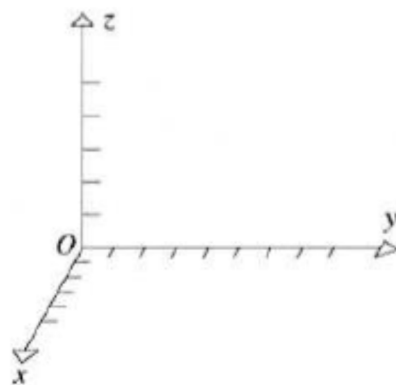
בהרצאות הקורס של ד"ר אמיר גורן, הוגדרה המחלקה Point שמייצגת נקודה במישור, לפי מערכת הצירים הקרטזית (Cartesian system).

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;

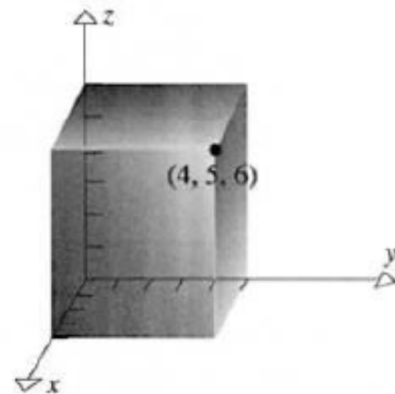
במטלה זו נייצג נקודה במרחב התלת-ממדי, שמוגדרת על ידי שלוש קואורדינטות, על ציר ה-X, על ציר ה-Y ועל ציר ה-Z.

זהו המרחב התלת-ממדי:



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-ממדי.

למחלקה `Point3D` יש 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

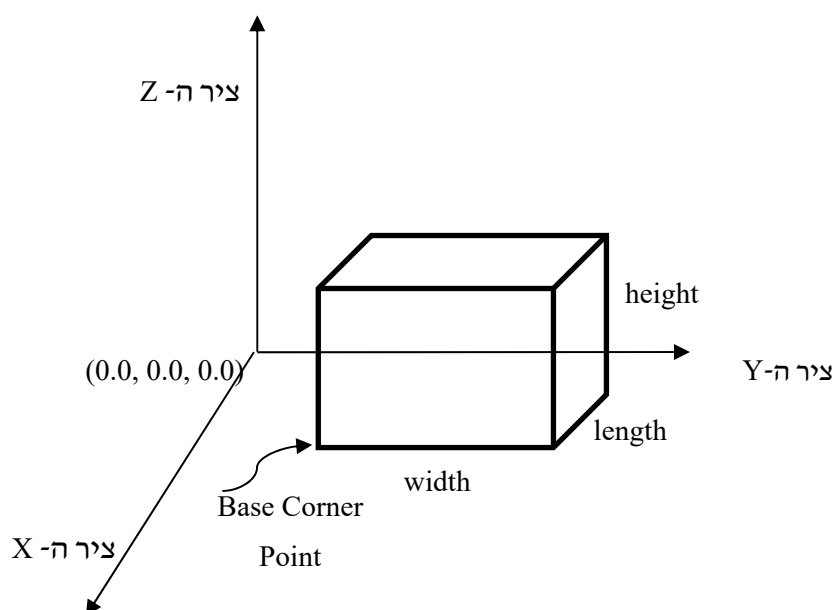
נגדיר מחלקה הנקראת Box3D המייצגת תיבה (box) בשלושה ממדים.

אובייקט מסוג תיבה מאופיין על-ידי המיקום של אחת מפינותיו (התחתונה, שמאלית, קדמית) שאנו נתייחס אליה כאל הפינה הבסיסית של התיבה, ועל-ידי שלושת הממדים: גובה, אורך, רוחב, שתמיד נמדדים כערכים חיוביים. הגובה נמדד מהפינה הבסיסית כלפי מעלה בכיוון החיובי של ציר ה-z, הרוחב נמדד מהפינה הבסיסית ימינה בכיוון החיובי של ציר ה-y, והאורך נמדד מהפינה הבסיסית אחורה בכיוון השלילי של ציר ה-x.

שימו לב שצלעות התיבה מקבילות לשלושת הצירים.

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

זו התיבה שתתקבל:



### למחלקה Box3D יש 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.

**לדוגמא: אם התיבה היא**

**The base point is (3.0,4.0,5.0), length = 2, width = 4, height = 1**

**פעולה זו תחזיר את הנקודה: (1.0,8.0,6.0)**

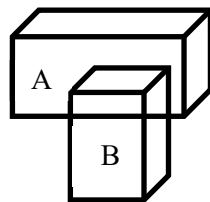
- `Point3D getCenter()` המחזירה את **הנקודה** שהיא מרכז התיבה.
- `double distance(Box3D other)` – שיטה שמקבלת תיבה כפרמטר ומחזירה את המרחק שבין שתי הנקודות שהן **מרכזי התיבות**.
- `int getVolume()` המחזירה את **נפח** התיבה.
- `int getSurfaceArea()` המחזירה את **שטח הפנים** של התיבה.
- **תזכורת מתמטית** – 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 מהממדים שלה זהים.

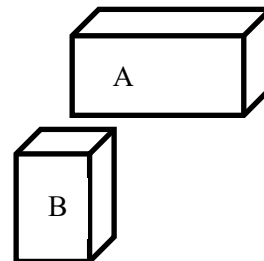
שימו לב שיש שוני מהשיטה isLargerCapacity. לדוגמא, התיבה A שהאורך, רוחב וגובה שלה הם בהתאמה 3, 10, 5 לא מכילה את התיבה B שהממדים שלה בהתאמה הם 4, 2, 5. אבל בכל זאת A גדולה מ-B מבחינת הנפח. אגב, גם התיבה שממדיה הם 4, 2, 5 לא מכילה ממש את B כי יש להן אותם ממדים.

- 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 :



התיבה A לא מעל B



התיבה A מעל B

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- בחומר שאינו כלול 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. ארוז את שני הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

## ב ה צ ל ח ה