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: 2017ב **מועד אחרון להגשה:** 29.4.2017

(ת)

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

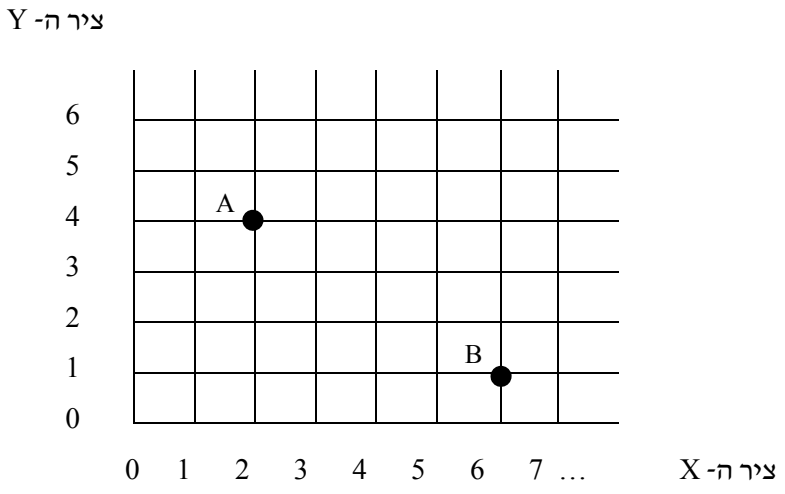
שאלה 1 - 20 נקודות

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

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

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

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



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

המחלקה Point תייצג נקודה במישור ברביע הראשון בלבד .

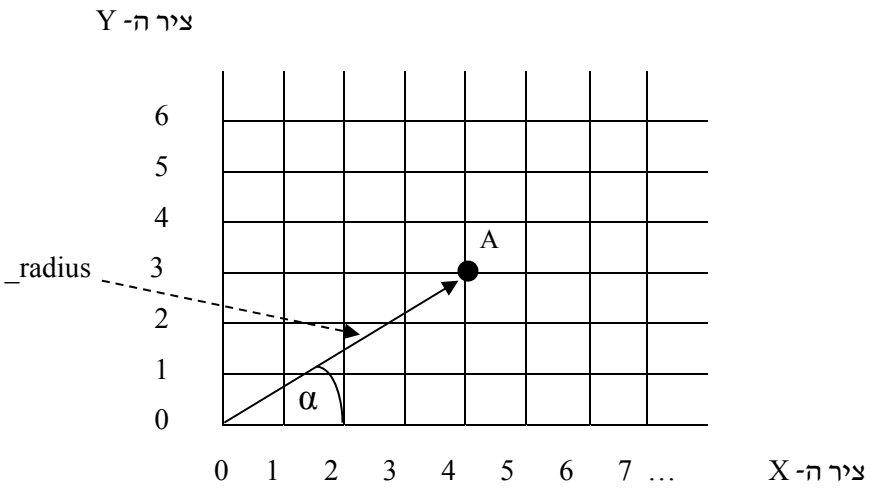
למחלקה Point יש את התכונות הפרטיות (instance variables) הבאות:

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

לדוגמא, הנה מסומנת הנקודה A (שנמצאת בקואורדינטות (4.0,3.0) במרחב:

כאן אורך הוקטור שמחבר את הנקודה (0.0,0.0) עם (4.0,3.0) הוא 5.0

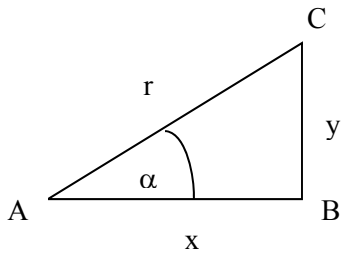
וערכה של הזוית α (alpha) הוא 36.87 מעלות (וברדיאנים - 0.64)



תזכורת מתמטית קצרה

לדוגמא, נתון המשולש ישר הזווית הבא:

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



$$\sin(\propto) = {}^{y}/_{r}$$

$$\cos(\propto) = {}^{x}/_{r}$$

$$\tan(\propto) = {}^{y}/_{x}$$

- $sin\,\alpha = {}^{y}/_{r}$
- $cos\,\alpha = {}^{x}/_{r}$
- $tan\,\alpha = {}^{y}/_{x}$
- מכאן, כאשר נתונים אורכי הצלעות x ו- y, אפשר לחשב את הזווית α על-ידי שימוש בנוסחה ההפוכה arctan . כלומר, $\propto = arctan\left(\frac{y}{x}\right)$ **ערך הזווית בראדיאנים.**

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

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

כזכור, על מנת לחשב שורש ריבועי של מספר, ניתן להשתמש בשיטה Math.sqrt(x), שהיא שיטה של Java שנמצאת במחלקה Math. כדי להשתמש בה אין צורך לייבא אף מחלקה, אלא לקרוא לה בשמה המלא Math.sqrt(x) כאשר במקום הפרמטר x כותבים את הביטוי שממנו רוצים להוציא שורש ריבועי.

הפרמטר x של השיטה הזו יכול להיות מטיפוס שלם (int) או ממשי (double). השיטה מחזירה מספר ממשי (גם אם השורש הריבועי של x הוא מספר שלם).

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

כדי להימנע מאי דיוקים קטנים בחישובים של ממשיים , לדוגמא קבלת תוצאה 3.9999999994 במקום 4.0 השתמש בנוסחה: Math.round(d*10000)/(double)10000 כאשר d מכיל את המספר הממשי שמעוניינים לעגל.

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

אפשר למצוא את ה- API של המחלקה Math בכתובת

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

שימו לב:

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

להזכירכם:

Π =3.14159… = Math.PI = 180°

Math.PI / 2 = 90° וכן הלאה...

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

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

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

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

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

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

- האחד - בנאי המקבל שני פרמטרים המהווים את ערכי התכונות שיהיו לנקודה.

public Point(double x, double y)

אם אחד הפרמטרים שהתקבל הוא שלילי, הוא צריך להיות מאותחל ל- 0.

- השני - בנאי העתקה המקבל נקודה אחרת, ומעתיק את ערכיה.

public Point (Point other)

בנוסף, הוגדרו במחלקה השיטות:

- שיטות **האחזור**:
  - double getX() המחזירה את ערכה של קואורדינטת ה- x.
  - double getY() המחזירה את ערכה של קואורדינטת ה- y.
- השיטות **הקובעות**:
  - void setX (double num) המשנה את ערכה של קואורדינטת ה- x להיות num.
    אם num הוא מספר שלילי, הערך של x לא משתנה.
  - void setY (double num) המשנה את ערכה של קואורדינטת ה- y להיות num.
    אם num הוא מספר שלילי, הערך של y לא משתנה.
- השיטה toString שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המתמטי המקובל - (x,y). **כלומר, להדפיס את הנקודה לפי מערכת הצירים הקרטזית.** כך, המחרוזת (3.0,4.0) מייצגת את הנקודה שקואורדינטת ה- x שלה היא 3.0 וקואורדינטת ה- y שלה היא 4.0. **שימו לב לדייק במחרוזת לפי הכתוב כאן**. ללא רווחים וללא תווים נוספים.
- boolean equals (Point other) – שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה והנקודה שהתקבלה כפרמטר זהות.
- boolean isAbove (Point other) - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **מעל** לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת מעל לנקודה B)
- boolean isUnder (Point other) - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **מתחת** לנקודה שהתקבלה כפרמטר. **השיטה הזו משתמשת אך ורק בשיטה isAbove שהוגדרה לעיל.**
- boolean isLeft (Point other) - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **משמאל** לנקודה שהתקבלה כפרמטר. (באיור למעלה, הנקודה A נמצאת משמאל לנקודה B)

- boolean isRight (Point other) - שיטה שמקבלת נקודה כפרמטר ומחזירה האם הנקודה שעליה הופעלה השיטה נמצאת **מימין** לנקודה שהתקבלה כפרמטר. **השיטה הזו משתמשת אך ורק בשיטה isLeft שהוגדרה לעיל.**
- double distance (Point p)– שיטה שמקבלת נקודה כפרמטר ומחזירה את המרחק בין הנקודה שעליה הופעלה והנקודה שהתקבלה כפרמטר. לעזרתכם, הנוסחה לחישוב מרחק בין הנקודה (x1,y1) , (x2,y2) היא $\sqrt[2]{(y2-y1)^2+(x2-x1)^2}$
- void move (double dx, double dy) – המזיזה את הנקודה ב- dx על ציר ה- X וב- dy על ציר ה- Y. אם התזוזה מזיזה את הנקודה מחוץ לרביע הראשון של מערכת הצירים, הנקודה תישאר במקומה ולא תזוז.

עליכם לכתוב את המחלקה Point לפי ההגדרות לעיל.

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

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

מותר להשתמש אך ורק בשיטות הבאות מהמחלקה *Math*:

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

שאלה 2 - 40 נקודות

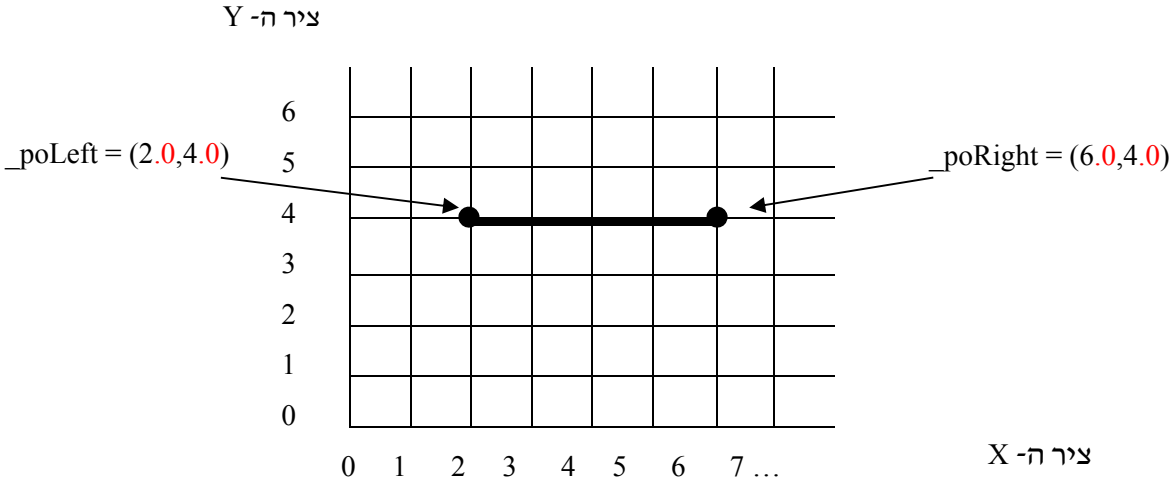
המחלקה Segment1 מייצגת מקטע במישור המקביל לציר ה- x .

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

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

לדוגמא, הנה מסומן המקטע המחבר את שתי הנקודות

_poLeft = (2.0,4.0) ו- _poRight = (6.0,4.0) במרחב:



שימו לב, כל המקטעים מקבילים לציר ה- X.

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

אי אפשר להוסיף תכונות פרטיות למחלקה זו.

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

- בנאי המקבל שני פרמטרים המהווים את הנקודות השמאלית והימנית של המקטע.

```
public Segment1 (Point left, Point right)
```

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

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

לדוגמא, אם הנקודה left היא (3.0, 4.0) והנקודה right היא (5.0, 6.0) אזי הבנאי ייצור מקטע שהנקודה _poLeft שלו תהיה (3.0, 4.0) והנקודה _poRight שלו תהיה (5.0, 4.0).

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

  גם כאן:

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

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

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

- בנאי העתקה המקבל מקטע אחר, ומעתיק את ערכיו.

```
public Segment1 (Segment1 other)
```

בנוסף הוגדרו במחלקה השיטות:

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

(3.0,4.0)---(5.0,4.0)

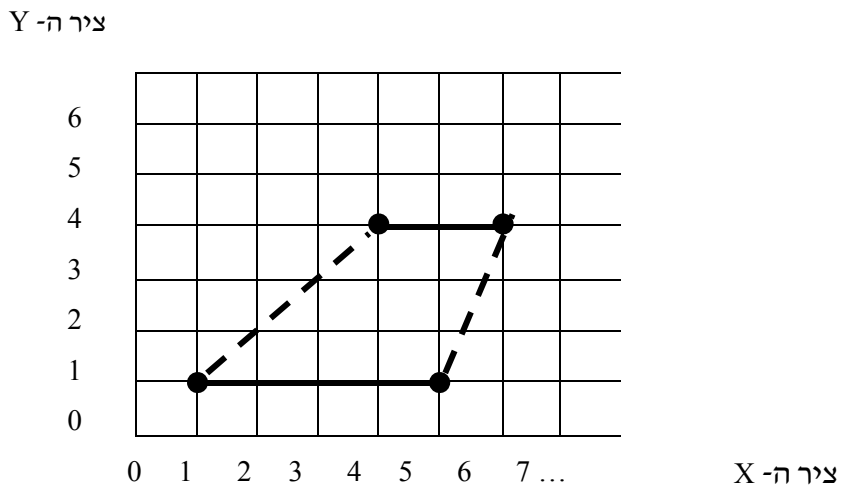
שימו לב לדייק במחרוזת לפי הכתוב כאן. ללא רווחים וללא תווים נוספים.

- boolean equals (Segment1 other) – שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה והמקטע שהתקבל כפרמטר זהים.
- boolean isAbove (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא **מעל** למקטע שהתקבל כפרמטר.
- boolean isUnder (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא **מתחת** למקטע שהתקבל כפרמטר. **השיטה הזו משתמשת אך ורק בשיטה isAbove שהוגדרה לעיל.**
- boolean isLeft (Segment1 other) - שיטה שמקבלת מקטע כפרמטר ומחזירה האם המקטע שעליו הופעלה השיטה נמצא **כולו משמאל** למקטע שהתקבל כפרמטר.

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

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

לדוגמא, באיור הבא,



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

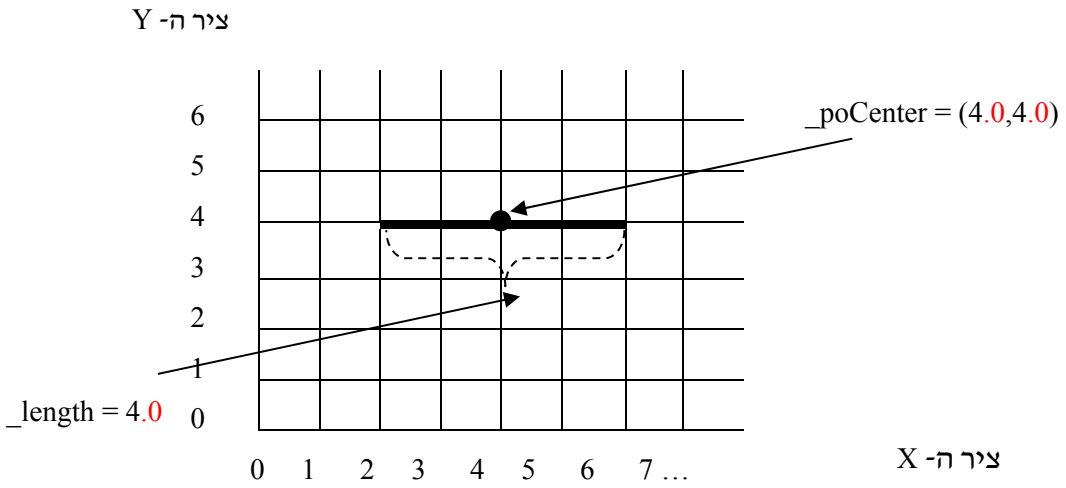
שאלה 3 - 40 נקודות

המחלקה Segment2 מייצגת מקטע במישור המקביל לציר ה- x .

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

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

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



עליכם לכתוב מימוש למחלקה Segment2, כך שהיא תבצע בדיוק את אותן שיטות שמבצעת המחלקה Segment1. למרות שהייצוג הפנימי של האובייקטים (התכונות) שונה.

על השיטות הכתובות עבור מחלקות Segment1 ו- Segment2 להיות זהות מבחינת שם ופונקציונליות. עם זאת, שימו לב ששיטות מקבילות בשתי המחלקות אינן מקבלות בהכרח את אותם הפרמטרים (ראו את ה- API המדויק באתר).

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

שימו לב, אסור להוסיף תכונות פרטיות.

מותר להוסיף שיטות פרטיות.

אין להשתמש במספרים בקוד. יש להוסיף קבועים (final) עבור כל מספר קבוע ולהשתמש בקבוע בקוד.

בכל השיטות במטלה שמקבלות אובייקט כפרמטר אפשר להניח שמתקבל אובייקט שאותחל ואינו שווה ל- null.

שימו לב ששמנו טסטרים 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.

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

שימו לב לא לבצע aliasing במקומות המועדים.

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

הגשה

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

ב ה צ ל ח ה