

מטלת מנחה (ממ"ן) 15

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 11 נושא המטלה: רשימה מקושרת

מספר השאלות: 2 משקל המטלה: 5 נקודות

סמסטר: 2022 מועד אחרון להגשה: 22.1.2022

(ת)

כזכור, במטלה 12 הגדרנו מחלקה Point המייצגת נקודה במישור (על ידי קואורדינטות x ו-y) ומחלקה RectangleA המייצגת מלבן המקביל לצירים (על ידי הנקודה הדרום-מערבית של המלבן וכן הרוחב והגובה של המלבן).

במטלה זו נשתמש בשתי מחלקות אלו, וניצור רשימה מקושרת של מלבנים.

כדי לעשות זאת, עליכם להגדיר שתי מחלקות RectNode ו-RectList.

חובה עליכם להשתמש במחלקות של Point ו-RectangleA ששמנו באתר במטלה 15. שימו לב שהקבצים שיש באתר הם קובצי class ולא הקוד ב-Java. אי אפשר לפתוח אותם אלא להשתמש בהם בלבד. לצידם יש קובצי html שהם ה-API של המחלקות Point ו-RectangleA.

שאלה 1 - להרצה (10%)

המחלקה RectNode תייצג צומת ברשימה מקושרת שיש בו מלבן.

לכל אובייקט במחלקה יש שני שדות:

1. RectangleA _rect // המלבן
2. RectNode _next // מצביע לאיבר הבא

למחלקה זו עליכם להגדיר שלושה בנאים:

1. public RectNode (RectangleA r)
בנאי המקבל מלבן, שדה ה-next יאותחל ל-null.
2. public RectNode (RectangleA r, RectNode n)
בנאי המקבל מלבן ואיבר נוסף מטיפוס RectNode, ומאתחל את התכונות לפי הפרמטרים.
3. public RectNode (RectNode r)
בנאי העתקה.

השיטות במחלקה RectNode הן:

- public RectangleA getRect() - שיטה המחזירה עותק של המלבן שבאיבר.
- public RectNode getNext() - שיטה המחזירה מצביע לאיבר הבא.

- `public void setRect(RectangleA r)` - שיטה המקבלת מלבן ומעדכנת את תכונת המלבן שבאיבר.
- `public void setNext(RectNode next)` - שיטה המקבלת מצביע ומעדכנת את תכונת המצביע לאיבר הבא.

שימו לב: הבנאים והשיטות במחלקה זו מקבלים ומחזירים אובייקטים (מצביעים). לפיכך יש לפעול בזהירות ולקבוע מתי `aliasing` הוא נכון ומתי הוא טעות. כאשר עליכם להעתיק או להחזיר אובייקט מסוג מלבן יש להיזהר מ-`aliasing`. כאשר נרצה להחזיר או להעתיק מצביע לאיבר הבא ברשימה `aliasing` אינו טעות.

שאלה 2 - להרצה (90%)

המחלקה `RectList` תייצג רשימה מקושרת של מלבנים. הייצוג נעשה על-ידי רשימה ששומרת את רשימת המלבנים.

במחלקה זו מותר להגדיר אך ורק תכונה פרטית אחת, ראש הרשימה `_head`, שיצביע לתחילת הרשימה לאובייקט מסוג `RectNode`. אין להוסיף תכונות מעבר לתכונה זו.

עליכם לממש ב-Java את המחלקה `RectList` לפי הסעיפים להלן:

1. הגדרת התכונה של המחלקה.
2. בנאי שיוצר רשימה ריקה - מאתחל את ראש הרשימה להיות `null`.
3. שיטה `addRect` המקבלת כפרמטר מלבן ומוסיפה אותו **לסוף הרשימה**. אם המלבן הזה כבר נמצא ברשימה, השיטה לא תעשה כלום.
4. שיטה `howManyWithPoint` המקבלת כפרמטר נקודה `p` (אובייקט מהמחלקה `Point`) ומחזירה את מספר המלבנים ברשימה שהנקודה הדרום-מערבית שלהם היא `p`. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) או שאין מלבנים ברשימה שהנקודה `p` היא הנקודה הדרום-מערבית שלהם, השיטה תחזיר 0.
5. שיטה `longestDiagonal` המחזירה את אורך האלכסון הארוך ביותר בין כל המלבנים. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר 0.
6. שיטה `mostLeftRect` המחזירה העתק של הנקודה הדרום-מערבית של המלבן הכי שמאלי ברשימה. מלבן אחד יותר שמאלי ממלבן אחר אם הנקודה הדרום-מערבית שלו נמצאת משמאל לנקודה הדרום-מערבית של המלבן האחר. אם יש יותר ממלבן אחד שהם השמאליים ביותר, השיטה מחזירה **העתק של הנקודה הדרום-מערבית של המלבן הראשון מביניהם ברשימה**. **בד-תקלה**. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר `null`.

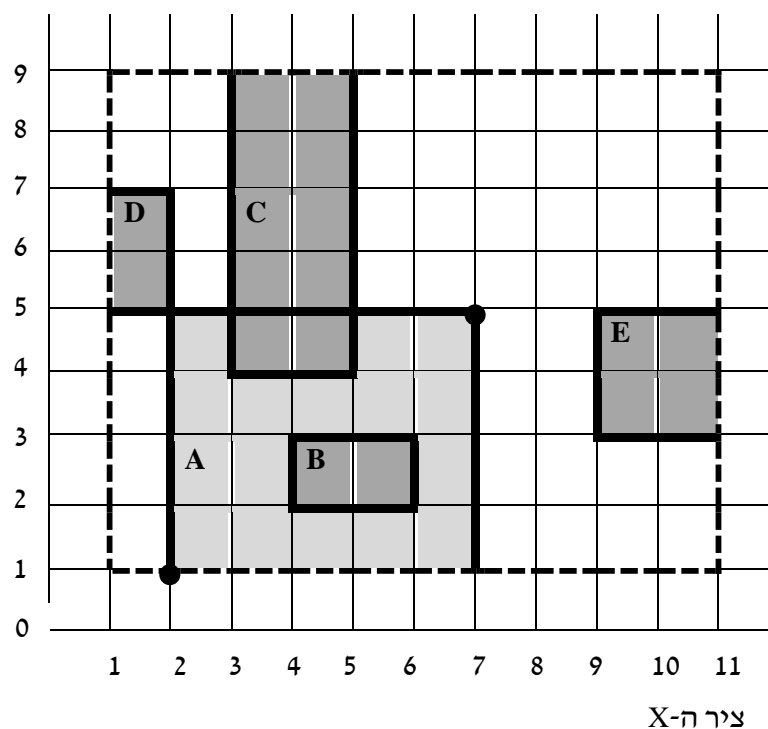
7. שיטה highestRect המחזירה העתק של הנקודה הצפון-מזרחית של המלבן הכי גבוה ברשימה. מלבן אחד יותר גבוה ממלבן אחר אם הנקודה הצפון-מזרחית שלו גבוהה יותר מזו של האחר. אם יש יותר ממלבן אחד בגובה הגבוה ביותר, השיטה מחזירה העתק של הנקודה הצפון-מזרחית של המלבן הראשון מביניהם ברשימה. **בדוגמה.**

אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר null.

8. שיטה minimalContainer המחזירה **מלבן** חדש (אובייקט מהמחלקה RectangleA) שהוא המלבן בעל השטח המינימלי המכיל את כל המלבנים שברשימה. אם אין מלבנים ברשימה (כלומר הרשימה ריקה) השיטה תחזיר null.

לדוגמא, נסתכל על האיור שלהלן:

ציר ה-Y



ציר ה-X

המלבנים המצוירים באיור הם :

- A שהנקודה הדרום-מערבית שלו היא (2,1) והנקודה הצפון-מזרחית שלו היא (4,3)
- B שהנקודה הדרום-מערבית שלו היא (4,2) והנקודה הצפון-מזרחית שלו היא (6,3)
- C שהנקודה הדרום-מערבית שלו היא (3,4) והנקודה הצפון-מזרחית שלו היא (5,9)
- D שהנקודה הדרום-מערבית שלו היא (1,5) והנקודה הצפון-מזרחית שלו היא (2,7)
- E שהנקודה הדרום-מערבית שלו היא (9,3) והנקודה הצפון-מזרחית שלו היא (11,5)

המלבן הקטן ביותר שמכיל את כל חמשת המלבנים הוא זה שהנקודה הדרום-מערבית שלו היא (1,1) והנקודה הצפון-מזרחית שלו היא (11,9). הגבולות החיצוניים שלו מסומנים בקו מקווקו באיור.

9. השיטה toString המחזירה מחרוזת תווים המייצגת את רשימת המלבנים. המחרוזת צריכה להיות בפורמט הבא: שימו לב שאין רווחים במחרוזת שמייצגת את הנקודה.

The list has 3 rectangles.

1. Width=5 Height=4 PointSW=(2,1)
2. Width=4 Height=7 PointSW=(2,1)
3. Width=5 Height=4 PointSW=(3,1)

אם אין מלבנים ברשימה, השיטה תחזיר מחרוזת בפורמט הבא:

The list has 0 rectangles.

שימו לב לא לבצע aliasing במקומות המועדים (מלבד המקומות בהם במפורש נדרש אחרת). מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.

אתם צריכים כמובן לכתוב API לשתי המחלקות.
שימו לב לכל מקרי השגיאה האפשריים!

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

שימו לב:

1. אסור להשתמש במחלקות מוכנות כבר של Java.
2. מותר ורצוי להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים, לדוגמא Math.sqrt.

שימו לב ששמנו טסטר למחלקה RectList באתר הקורס. חובה לוודא שטסטר זה ירוץ ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה לשיטה ובתוך גוף השיטה החזירו ערך סתמי כדי שהטסטר ירוץ עם המחלקות ללא שגיאות קומפילציה. מי שיגיש מטלה שלא עוברת קומפילציה הציון במטלה שלו יהיה אפס!

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות והמחלקות יהיו בדיוק לפי הוראות הממ"ן.
3. את התשובות לשאלות יש להגיש בשני קובצי Java הבאים: RectNode.java, ו- RectList.java ארוזים יחד בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.