Supervisor Task (MMAN)

Course: 20441 - Introduction to Computer Science and the Java Language

Task subject: A linked list

Study material for the assignment: Unit 11

Task weight: 5 points

Number of questions: 2

Deadline for submission: 2022.1.22

Semester: 2022 a

) ÿ (

Remember, in Task 12 we defined a Point class **that represents a point on a plane** (by coordinates x and y)

And a **class RectangleA representing a rectangle parallel to the axes)** by the southwestern point of the rectangle

And the width and height of the rectangle.

In this task we will use these two classes, and create a linked list of rectangles.

To do this, you must set up two RectNode and RectList classes

You must use the Point and RectangleA classes that we put on the site in Task 15. Note that the files on the site are

class files **and not the code in Java . You can not open them but use**

In them only. Next to them are **html files which are the API** of the **Point and RectangleA classes .**

Question 1 - to run)% 10)

The **RectNode class will represent a node in a linked list that has a rectangle.**

Each object in the class has two fields:

1. RectangleA _rect // The Rectangle

next _RectNode. 2 Pointer to the next limb //

You must define three builders for this class:

1. public RectNode (RectangleA r)

The next _ field will be initialized to null. A builder who receives a rectangle,

2. public RectNode (RectangleA r, RectNode n)

A constructor that receives a rectangle and another member of the RectNode type, and initializes the properties according to the parameters.

3. public RectNode (RectNode r)

Copy builder.

The methods in the RectNode class are:

- () getRect RectangleA public - A method that returns a copy of the rectangle in the organ.
- () getNext RectNode public - A method that returns a pointer to the next member.

You and update a rectangle that receives a method - public void setRect (RectangleA r) •

The feature of the rectangle in the limb.

You update a pointer that accepts a method - public void setNext (RectNode next) •

The pointer feature to the next limb.

Note: The builders and methods in this class accept and return objects (pointers).

Care must be taken to determine when aliasing is true and when it is a mistake. When you need to copy

Or return a rectangle object should be careful about aliasing. When we want to return or copy

Pointer to the next member in the aliasing list is not a mistake.

Question 2 - to run) 90%)

The **RectList class will represent a linked list of rectangles.**

The representation is made by a list that maintains the list of rectangles.

In this class it is allowed to define only one private attribute, the head of the list, which will point to the beginning

The list for a RectNode object. Do not add attributes beyond this attribute.

You must implement the **RectList class** in **Java according to the sections below:**

1. Defining the attribute of the class.

2. Builder that creates an empty list - initializes the list head to null .

3. AddRect method that accepts as a rectangle parameter and adds it **to the end of the list.** If this rectangle

Already on the list, the method will do nothing.

4. HowManyWithPoint method that receives as point parameter p (object from class Point)

And returns the number of rectangles in the list whose southwestern point is p .

Is their southwest point, the method will return 0.

5. The longestDiagonal method that returns the longest diagonal length between all the rectangles.

If there are no rectangles in the list) that is, the list is empty (the method will return 0.

6. MostLeftRect method returns a copy of the southwest **point** of the leftmost rectangle in the list. One rectangle

is more left-handed than another rectangle if the southwestern point

Its is to the left of the southwestern point of the other rectangle. If there is more than one rectangle they

The method returns a copy of the southwestern point of , are the leftmost

The first rectangle among them on the list. She~~came across~~ it. If there are no rectangles in the list) that is

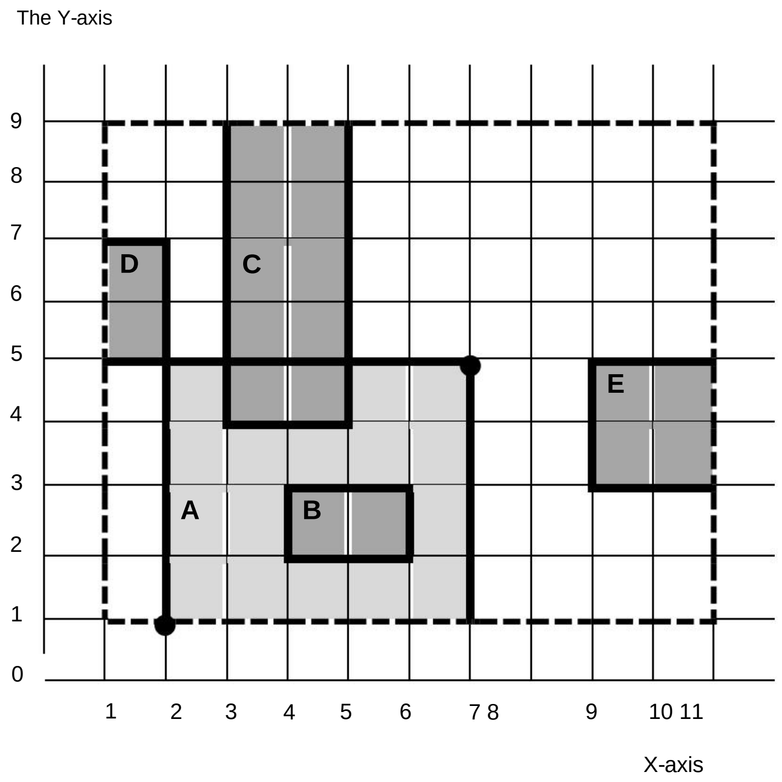
The list is empty (the method will return null.

7. The highestRect method returns a copy of the northeast **point** of the highest rectangle in the list. One rectangle is higher than another rectangle if its northeastern point is higher than that of the other. If there is more than one rectangle at the highest height, the method returns a copy of the northeast point ~~of the first re~~ctangle on the list. She came across it. If there are no rectangles in the list) that is, the list is empty (the method will return null.

8. MinimalContainer method that returns a new **rectangle** (object from the RectangleA class)

Which is the rectangle with the minimum area that contains all the rectangles in the list. If there are no rectangles in the list) that is, the list is empty (the method will return null.

For example, let's look at the illustration below:



The rectangles drawn in the figure are:

- A whose southwestern point is (1,2) and whose northeastern point is (5,7 (
- B whose southwestern point is (2,4) and whose northeastern point is (3,6 (
- C whose southwestern point is (4,3) and whose northeastern point is (9,5 (
- D whose southwestern point is (5,1) and its northeastern point is (7,2 (
- E whose southwestern point is (3,9) and whose northeastern point is (5,11 (

The smallest rectangle containing all five rectangles is the one whose southwestern point is (1,1) and its northeastern point is (9,11 .) Its outer boundaries

Are marked with a dashed line in the figure.

9. The toString method returns a string of characters representing the list of rectangles. The string Should be in the following format: Note that there are no spaces in the string representing the dot.

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)

If there are no rectangles in the list, the method will return a string in the following format:

The list has 0 rectangles.

Be careful not to perform aliasing in the time locations (except where explicitly required otherwise).

It is allowed to add additional methods (private), as you see fit.

You should of course write an API for both departments.

Pay attention to all possible errors!

Write as a comment in the API what the time and place complications of each method you have written are.

Take care of the effectiveness of the methods you have written!

pay attention:

·1. Java-ready classes should not be used

2. It is permissible and desirable to use departments that were given in the lecture and are in the slide booklet, for example

.Math.sqrt

Please note that we have placed a tester for the RectList department on the course website. This tester must be verified

Will run without compilation errors with your departments. If there is a method that you did not write, write a signature for the method and within the body of the method return a random value so that the tester will run with the classes without compilation errors. Whoever submits an assignment that does not pass the grade compilation

In his task will be zero!

submission

1. The submission of the financial statement is done only electronically, through the assignment submission system.

2. Make sure that the names of the methods and departments are exactly according to the provisions of the MMAN.

3. The answers to the questions must be submitted in the following two Java files : java.RectNode , and java.RectList packed together in a single zip file . Do not send additional files.