Assignment (MMAN) 12

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

Study material for the assignment: units 3-4 - **Assignments:** Using classes and writing classrooms

Task weight: 4 points

Number of questions: 3

Deadline for submission: 2021.11.27

Semester: 2022 a

(A)

The purpose of this task is to teach you the basics of self-indulgence.

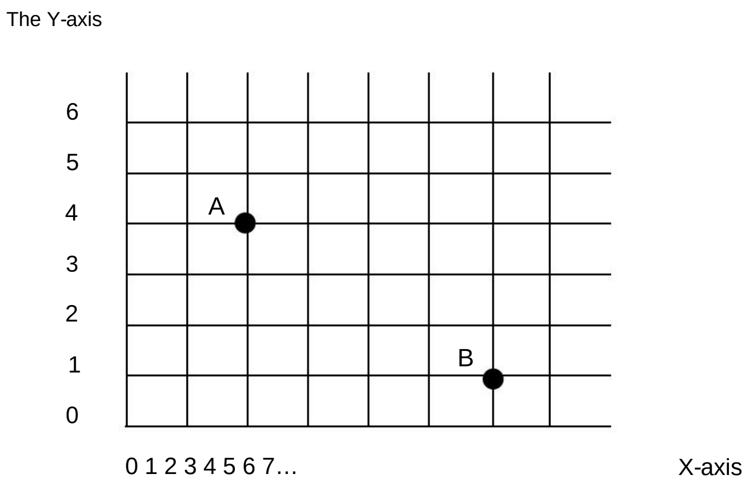
Codesÿ 20 - 1 question

The **Point class represents a code in the plane, according to the Cartesian system of axes .**

The Point class contains the following variables (variables instance):

- x _int - which represents the position on the x-axis .
- y _int - which represents the position on the axis of the Y -axis.

For example, the two codes (4,2 = (A and 1,6 = (B) are denoted in space:



The Point class has been defined as constructors :**(**

- The one - on the island that receives two parameters that constitute the values of the properties that will be coded.

public Point (int x, int y)

- The value - in a copy island that receives another code, and copies its values.

public Point (Point other)

In addition, the methods were defined in the department:

- **Retrieval** methods :

o () getX int which returns the value of the x-coordinate.

o () getY int which returns the value of the y-coordinate.

- The **determining** methods :

o) num int (setX void Increase the value of the x-coordinate to be num.

o) num int (setY void Increase the value of the y-coordinate to be num.

- The toString () method that returns the contents of the object as a string of characters according to the mathematical representation

The usual - (y, x. (Thus, the string (4,3) represents the code whose x-coordinate is 3

And its y-coordinate is 4. **Pay attention to be precise in the string as written here.** No spaces

And without ÿadditions characters.

As a other parameter in the name and a codaÿ couchÿ that accepts a method - boolean equals (Point other) •

And returns the mother code on which the method was applied and the other code obtained as an identity parameter.

That is, returns true if the code values on which the method is run are equal to the other code values.

•) other Point (isAbove boolean - a method that receives a code as a parameter and returns whether the code on which the

method was run is **above** the code received as a parameter.

Code A is located above Code B (

•) other Point (isUnder boolean - a method that receives a code as a parameter and returns the mother

The code on which the method was applied is **below the** code obtained as a parameter. **This method**

Use only the isAbove method defined above and do not use additional operations. Do not access the contents

of the codes.

•) other Point (isLeft boolean - a method that receives a code as a parameter and returns whether the code on which the

method was applied is located to the **left** of the code received as a parameter.

Code A is located to the left of Code B (

•) other Point (isRight boolean - a method that receives a code as a parameter and returns whether the code on which the

method was run is located to the **right** of the code received as a parameter. **This method uses**

Only in the isLeft method defined above and do not use ÿAddition operations. Do not access the contents of

the codes.

-And X -the axis on deltaX -b encodes the moving - void move (int deltaX, int deltaY) •

deltaY on the Y axis.

•) p Point (distance double– a method that receives a ÿÿÿ code as a parameter and returns the distance between

The code on which the method was applied and the code obtained as a parameter.

Mathematical reminder:

To calculate the distance between two ÿÿÿÿpoints - (y2, x2), (y1, x1 - (use the following formula:

$$d\ ÿ\ \sqrt{(x\ ÿ\ x_{1})_{2}^{2}\ ÿ\ (\ ÿ\ ÿ\ y\ )_{1\ 2}^{\quad 2}}$$

If you want to calculate a square root of a number, you can use the (x (sqrt.Math) method, which is a Java method

You're in the Math class. To use it you do not need to import any class, but call it by its name

The full (x (sqrt.Math) where instead of the parameter x write the expression from which you want to extract a root

Squares.

The x parameter of this method can be of type int (or real (double) (the method returns

A real number (even if the square root of x is an integer).

You must write the Point class **according to the definitions above.**

Codesÿ 40 - 2 question

The RectangleA class represents a rectangle whose sides are parallel to the axes.

The RectangleA class has the following variables (variables instance):

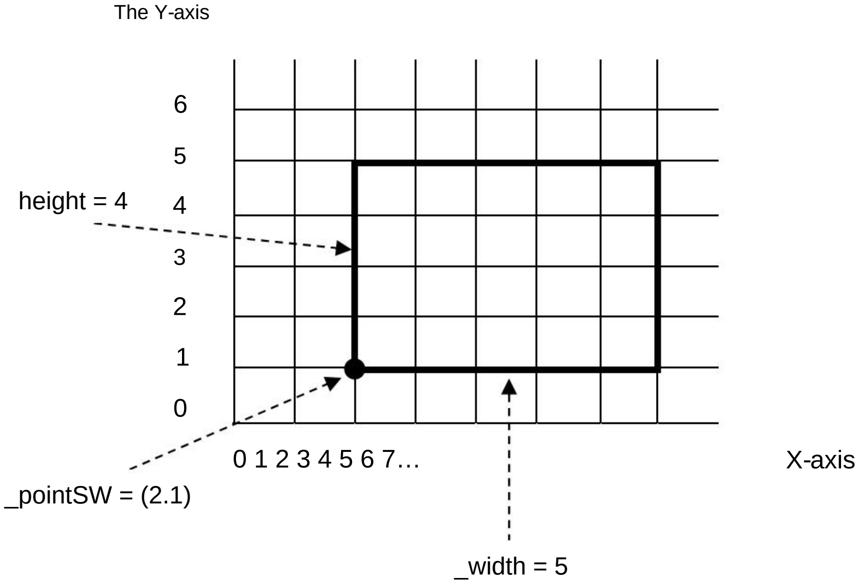
Rectangle Width - int _width •

The rectangle height - int _height •

• pointSW _Point - Location of the southwestern point of the rectangle.

Here is an example of a rectangle:

Its southwestern point is in position (1, 2), height 4 and width 5.



The RectangleA department has four defined:

- One - the recipient of integers as parameters, the first is the width of the rectangle and the integer Is the height. The southwestern point will be at the beginning of the axes (0, 0). The length and width must Be really positive wholes. If any of them are like this, it will be initialized to be 1.

public RectangleA (int w, int h)

- The value - which receives parameters with values for the appropriate properties: the southwestern code. Will be accepted as a parameter (as a code, which is null, and the width and height will be accepted as parameters. The length And the width must be really positive integers. If any of them are like that, the plan will be initialized To be 1.

public RectangleA (Point p, int w, int h)

- The third - the recipient of two ÿÿÿÿpoints as parameters. The first code is the southwestern code (sw (and the ÿÿÿkoda shÿya is the northeast koda) (ne. (You can prove that the ÿkoda is indeed sw ÿ is located **just** southwest of code ne. In particular, both codes can not be Same value x or same value y. No need to check this. It is also possible to assume that both codes Transmitted as parameters are null.

public RectangleA (Point sw, Point ne)

- Fourth - In non-copying, which receives an object from the RectangleA class and copies its values.

public RectangleA (RectangleA r)

Also, the retrieval methods (get) and the determining methods (set) were defined according to the accepted names for all intents and purposes. Of the class, and the toString method that returns the rectangular characters as a string of characters.

- public int getWidth ()
- public int getHeight ()
- public Point getPointSW ()
- public void setWidth (int w)
- public void setHeight (int h)
- public void setPointSW (Point p)
- public String toString ()

Remarks:

- In the setWidth and setHeight methods, if the parameter is really positive, nothing will be done, and the plan Stay where she was.

- In the toString method the string representing the rectangle in the figure above will look exactly like this (without Gains ÿAdditions and no ÿAdditions) -

Width = 5 Height = 4 PointSW = (2,1)

To the RectangleA class ÿAdd the following methods as well:

- A method that returns the perimeter of the rectangle

public int getPerimeter ()

- A method that returns the area of the rectangle

public int getArea ()

- A method that moves the rectangle to another location, the method gets whole deltaX and deltaY Teachers how much to move the rectangle on the X-axis and the Y-axis respectively.

public void move (int deltaX, int deltaY)

- A Boolean method that accepts a rectangle as a parameter and returns whether the rectangle on which the method was applied is the same To a rectangle obtained as a parameter (identical to the mesh of all intents).

public boolean equals (RectangleA other)

- A method that returns the length of the diagonal in a rectangle.

public double getDiagonalLength ()

- A Boolean method that accepts a rectangle as a parameter and returns whether the area of the rectangle on which it is applied The method is larger than the area of the rectangle obtained as a parameter.

public boolean isLarger (RectangleA other)

- A method that returns the northeastern point of the rectangle.

public Point getPointNE ()

- A method that extends the sides of the rectangle so that what was width becomes height, and height becomes Be width. The southwestern Koda is a hotbed.

public void changeSides ()

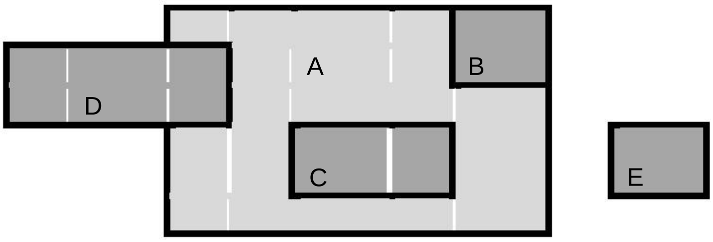
- A Boolean method that accepts a rectangle ÿAdds r and returns true if the rectangle on which the method ÿÿÿ is found

Whole within the rectangle r). Also common ribs ÿthink "inside").

public boolean isIn (RectangleA r)

For example,

The rectangle A shown below, contains rectangles B and C , but does not contain rectangles D and E.



- Boolean method that accepts a rectangle ÿadds r and returns true if there is an overlap between the rectangles and false

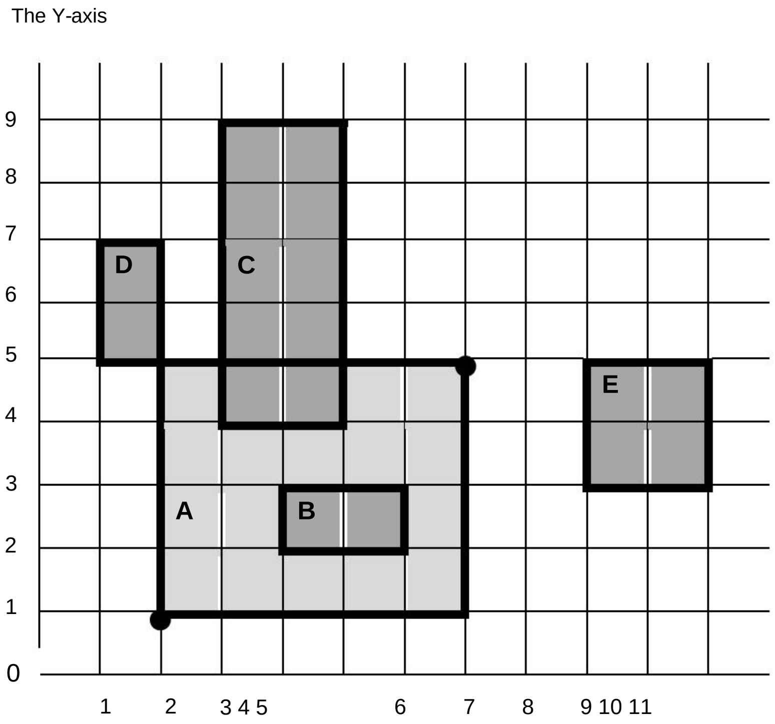
Other.

Reminder -

ÿÿ Said that there is an overlap between the two hearts if there are ÿÿÿÿcodes in the plane that are found in the two hearts.

public boolean overlap (RectangleA r)

For example, ÿLook at the illustration below:



The rectangle A pictured above, its southwestern point is (1,2) and its northeastern point

His is (5,7.) The codes are plotted in the graph in prominent circles.

All of the following rectangles overlap with rectangle A:

- 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 whose northeastern point is (7,2 (

Note that the rectangle A and D overlap in only one point (5,2).

While the rectangle

- E whose southwestern point is (3,9) and whose northeastern point is (5,11 (

·Io overlaps with no ÿÿÿcode to rectangle A

You must write the RectangleA class according to the definitions above (and according to what is written in the API On the site).

Be careful not to perform aliasing in the deadlines.

Codesÿ 40 - 3 question

The **RectangleB class represents a rectangle whose sides are parallel to the axes.**

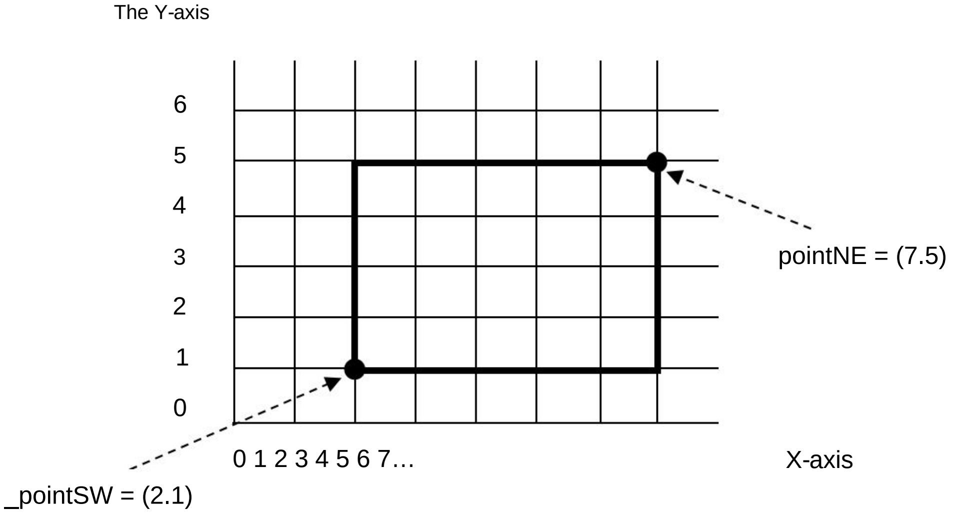
The RectangleB department has the following variables (variables instance):

- pointSW _Point - Location of the southwestern point of the rectangle.
- pointNE _Point - The location of the northeastern point of the rectangle.

Here is an example of a rectangle presented in question 2:

Its southwestern point is in position (1, 2 , (

Its northeastern point is in position (5, 7)



The RectangleB department has four defined:

- One - the recipient of integers as parameters, the first is the width of the rectangle and the integer

Is the height. The southwestern point will be at the beginning of the axes (0, 0). The length and width must

Be really positive wholes. If any of them are like this, it will be initialized to be 1.

public RectangleB (int w, int h)

- The value - which receives parameters with values for the appropriate properties: the southwestern code.

Will be accepted as a parameter (as a code, which is null, and the width and height will be accepted as parameters. The length

And the width must be really positive integers. If any of them are like that, the plan will be initialized

To be 1.

public RectangleB (Point p, int w, int h)

- The third - the recipient of two ÿÿÿÿpoints as parameters. The first code is the southwestern code

(sw (and the ÿÿÿkoda shÿya is the northeast koda) (ne. (You can prove that the ÿkoda is indeed

sw ÿ is located **just** southwest of Koda ne. There is no need to check this. It is also possible to express

That the two codes passed as parameters are null.

public RectangleB (Point sw, Point ne)

- The fourth - in non-copying, which receives an object from the RectangleB class and copies its values.

public RectangleB (RectangleB r)

The methods of the RectangleB class **are exactly the same as those of the RectangleA .**

Except in cases where a rectangle is obtained as a parameter to the method, then where written RectangleA should be

.RectangleB

In the **RectangleB class Do not use the RectangleA** class **or the methods written in it, and vice versa! goddess**

Two separate and completely independent departments!

Note that the toString method of the **RectangleB class should also be according to that of the class**

RectangleA , that is, print the southwest codec, latitude and longitude.

Precise API definitions can be found on the course website in the " **API Files"** subdirectory .

The animals are important:

1. In all the methods in the task that accept an object as a parameter, it can be assumed that an initialized object is obtained

And is equal to **null.**

2. You must refrain from **aliasing** methods and proponents.

3. If while writing the requested methods you want to use additional auxiliary methods, they must

Be **private .**

4. To avoid duplication of code in the RectangleA and RectangleB classes, use as needed

In existing methods in Point departments.

, RectangleA and RectangleB you wrote using the 5. You must well document the Point departments

API, as demonstrated in the lectures. To create the html file that contains the API, you need to go through

For documentation or interface mode (depending on which version of BlueJ you have installed) on the button

Top right (click on the arrow), on the class screen. When you switch to the documentation mode will be created at the same time

A file containing the department's API documentation, named html.RectangleA, html.RectangleB, and

html.Point. This file is located in the doc subfolder you found inside the folder containing the.

Your project.

You must document all the departments you will write in the API as well as in the chemical documentation.

You can of course use the API comments found on the site. **Note that you do not have to ship**

The html files, but you definitely need to prepare them.

6. Note that testers have been named for the three departments on the course website. Must make sure

These testers will run without compilation errors with your departments. if there is

A method that you did not write, wrote a signature for the method and within the body of the method returned a random value

So that the testers will run with the classes without compilation errors. Who will submit an assignment

That does not pass compilation The score in his assignment will be zero!

7. Clarification: The behavior of the departments in questions 2 and 3 in all methods should be the same.

Therefore, with the setPointSW method in the RectangleB class, the width values must be taken care of

And the altitude will not change, i.e. update the northeastern code as well.

Exact definitions for attendees and API requirements can be found on the course website.

submission

1. The submission of the financial statement was done only electronically, through the task sending system.

2. Remember that adherence to the names of departments and methods (public), as required, is necessary. Any deviation from the settings (even a single replacement of a capital letter in kata, for example) will cause its automatic check to fail resulting in irreversible damage to the score. Therefore, make sure that the names of the departments and methods are exactly as defined in the MMAN. **Otherwise you will be dropped Lots of ÿcodes!**

3. You must run the testers found on the course website on the departments you have written. Note that the testers do not cover all the options, and in particular not the end cases. They only check the names of the methods in the departments. It is highly recommended to add tests to them.

As long as RectangleB.java and RectangleA.java, the Point.java files are submitted to you . 4 **Submission of additional files will be deducted from the score. It is your responsibility to submit the affected files.**

5. Wrap the three files in a single zip file and send. **Do not send files and extensions.**

Successfully