AP CSA

Unit 2 Homework

Balchunas

This homework allows you to demonstrate your knowledge of object oriented programming. These classes and methods are useful in graphics software used to make games and animations. You will create two classes that work together to represent rectangles in the (x, y) planes. You will write methods related to Axis-Aligned Bounding Box (AABB) collisions useful for graphics software. The submission will have three `.java` files: `Point.java`, `Rectangle.java`, and `Runner.java`.

Point

Your goal here is to create a class that represents a single point in the (x, y) plane. x and y can represent any real number. The class specification is as follows. Create a class `Point` that has the following.

Attributes

x	stores the x-coordinate of the point
y	stores the y-coordinate of the point

Constructors

Point()	creates a Point object at (0, 0)
Point(x, y)	Creates a Point object at (x, y)

Methods

toString()	returns the representation (x, y)
equals(other)	True if this (x, y) pair is equal to other (x, y)
getX()	returns the x coordinate of this point
getY()	returns the y coordinate of this point
setX(x)	sets the x coordinate of this point to x
setY(y)	sets the x coordinate of this point to y
slopeTo(other)	returns the slope between this and other. Return Double.POSITIVE_INFINITY for vertical lines

Rectangle

Your goal here is to create a class that represents a rectangle. A rectangle consists of four points: (topLeft, topRight, bottomLeft, bottomRight). The class specification is as follows. Create a class `Rectangle` that has the following.

Attributes

topLeft	Point at the top left of the rectangle
topRight	Point at the top right of the rectangle
bottomLeft	Point at the bottom left of the rectangle
bottomRight	Point at the bottom right of the rectangle

Constructor

Rectangle(topLeft, topRight, bottomLeft, bottomRight)	create a Rectangle with these four coordinates

Methods

toString()	returns string of (topLeft, topRight, bottomLeft, bottomRight)
equals(other)	True if this rectangle has the same four defining points as another
isOverlapping(other)	True if this rectangle overlaps other in the x-y plane
getArea()	returns the area of the rectangle
isSquare()	returns True if the rectangle is a square
isValid()	returns True if the four points make a rectangle, False otherwise

The `isValid()` method would return false if the topRight point was below the bottomRight point for example. It would also return false if any of the four line segments making the rectangle are sloped in any way.

Medical Record

Your goal here is to create a class that represents a medical record. Each medical record has a patient name, patient ID number, and whether or not the person needs a translator.

Attributes

name	the name of the patient
patientID	the ID of the patient (as a string)
needsTranslator	True if this patient needs a translator

Constructors

MedicalRecord(String name)	creates a medial record of a patient with name and no need for translator
MedicalRecord(String name, boolean	creates a medial record of a patient with name and no need

needsTranslator)	for translator

Both constructors should assign a 6 digit random ID between `00000` and `99999`. Note that the ID must be 5 digits long, e.g. `00001` instead of `1`. Duplicate IDs are okay for this particular program.

Methods

toString()	returns the representation ({ID}, {name}, translator required) if translator is needed and ({ID}, {name}) if no translator is needed
equals(other)	True if this has all the same attributes as other
getName()	returns the x coordinate of this point
toggleTranslator()	toggles whether or not the patient requires a translator
getPaitentID()	returns the patient ID
getNumRecords()	returns the number of records total. This should be a static method.
clearRecord()	reset all the attributes to their default values to clear the record

Runner

Inside of runner you will be writing code that demonstrates the functionality of the previous classes.

Use a Scanner object to test **ONE** of the classes. Demonstrate that you know how to read from `System.in` with the Scanner.

Create a few points, compare them, and find slopes between them. Make sure you're demonstrating all the functionality, e.g. vertical slopes, horizontal slopes, positive slopes, and negative slopes. Create a few rectangles to test the rectangle class. You want to create some rectangles that overlap, some that do not, as well as squares. Include examples of rectangles that are *not* rectangles and show they are not valid using the `isValid` method. Create a few medical records and show you know how to call the static method.