RAZR

RAZR DEVELOPER CODING TEST

Introduction

In the following exercise, you will be asked to implement a small Java project. Feel free to add functionality that is not explicitly asked for if you feel the need for it.

Please read all the questions before you start coding. Come to the interview prepared to discuss your code including design decisions you made.

The general approach to the interview is that we sit down together and walk through your solution with you guiding the conversation. There are no tricks here, we're just looking to see how you like to approach problems and think through a solution.

Please share your solution with us some time before the interview however you see fit.

Section 1 - Shapes

The Circle Object

Create a Circle object with the following behaviour:

1. Instances can be created with any desired radius
2. Instances have a 'getArea' method that returns the area of the circle
3. Instances have a 'toString' method that returns the following:

$$Circle:\ Radius = X,\ Area\ =\ Y$$

(where X and Y are the radius and the area of the circle)

The Square Object

Create a Square object with the following behaviour:

1. Instances can be created with any length of sides
2. Instances have a 'getArea' method that returns the area of the square
3. Instances have a 'toString' method that returns the following:

$$Square:\ Size = X,\ Area\ =\ Y$$

(where X and Y are the side length and area of the square)

RAZR

Shape Sorting

Create the capability to input an array of Circles and Squares, and return the array sorted by area, in **descending** order.

Shape Generation

Create the capability to generate an array of 50 squares of varying **side length** between 1 and 100, and 50 circles of varying **diameter** between 1 and 100. The sizes of each shape should be randomly generated.

Searching

Create the capability to return a shape object(s) from an unsorted array of Circles and Squares with the area closest to the mean (rounded up).

Section 2 - REST API

1. Using the functionality created above, expose RESTful resource endpoints for Circles and Squares that support GET, POST and DELETE. POSTed objects should persist for the purposes of the GET and DELETE.

2. Create a resource that returns in response to a GET request an array of 50 squares of varying **side length** between 1 and 100, and 50 circles of varying **diameter** between 1 and 100. The sizes of each shape should be randomly generated. All the randomly generated objects should persist.

3. Create a resource that sorts the persisted Shape objects by area and returns them as an array.

4. Create a resource that returns a shape or shapes with the area closest to the mean from the persisted Shape objects.

Notes:

- Ensure you're using a version of Java >= 7
- You can use any Java frameworks you think are appropriate to the exercise. Bring all the tools you want to bear on this and don't feel limited or that you need to solve this any particular way.
- Any references to data persistence in the above requirements only extends to a single instance of the application and does not need to persist when restarting the application. In memory data storage is fine, this is just a demo application.