The following are 3 separate coding tests that should be developed using Java. Submit the code to your GitHub account and send a link to your account to the sender of this document. If you do not have a GitHub account, please create one first.

Given the following Shape interface:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public interface Shape {

public double getArea();

public double getPerimeter();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

Implement the classes: Triangle, Circle, and Rectangle

Write test classes for each implementation

Extra bonus fun: Implement ellipse and/or square \*/

/\* Triangle code \*/

public class Triangle extends Shape {

private final double a, b, c; // sides

public Triangle() {

this(1,1,1);

}

public Triangle(double a, double b, double c) {

this.a = a;

this.b = b;

this.c = c;

}

@Override

public double area() {

// Heron's formula:

// A = SquareRoot(s \* (s - a) \* (s - b) \* (s - c))

// where s = (a + b + c) / 2, or 1/2 of the perimeter of the triangle

double s = (a + b + c) / 2;

return Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

@Override

public double perimeter() {

// P = a + b + c

return a + b + c;

}

}

/\* Circle code \*/

public class Circle extends Shape {

private final double radius;

final double pi = Math.PI;

public Circle() {

this(1);

}

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

// A = π r^2

return pi \* Math.pow(radius, 2);

}

public double perimeter() {

// P = 2πr

return 2 \* pi \* radius;

}

}

/\* Rectangle code \*/

public class Rectangle extends Shape {

private final double width, length; //sides

public Rectangle() {

this(1,1);

}

public Rectangle(double width, double length) {

this.width = width;

this.length = length;

}

@Override

public double area() {

// A = w \* l

return width \* length;

}

@Override

public double perimeter() {

// P = 2(w + l)

return 2 \* (width + length);

}

}

/\* Testing Shapes code \*/

public class TestShape {

public static void main(String[] args) {

// Rectangle test

double width = 5, length = 7;

Shape rectangle = new Rectangle(width, length);

System.out.println("Rectangle width: " + width + " and length: " + length

+ "\nResulting area: " + rectangle.area()

+ "\nResulting perimeter: " + rectangle.perimeter() + "\n");

// Circle test

double radius = 5;

Shape circle = new Circle(radius);

System.out.println("Circle radius: " + radius

+ "\nResulting Area: " + circle.area()

+ "\nResulting Perimeter: " + circle.perimeter() + "\n");

// Triangle test

double a = 5, b = 3, c = 4;

Shape triangle = new Triangle(a,b,c);

System.out.println("Triangle sides lengths: " + a + ", " + b + ", " + c

+ "\nResulting Area: " + triangle.area()

+ "\nResulting Perimeter: " + triangle.perimeter() + "\n");

}

}

Given the following Java collection questions, implement the following using the Java Collection Framework.

Define the following 2 collections

* one contains integers 1 thru 10 (random order)
* one contains 6 thru 15 (random order)
* Print both collections
* Print the size of both collections

Combine the above two collections into a third collection

* Collection cannot have any duplicates
* Collection must be sorted
* Remove the middle entry
* Print the collection in reverse order
* Print the size of the collection

This is for both condition above.

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

public class Listings {

public static void main(String... args) {

{

int[] array1={1, 9, 3, 5, 2, 6, 8, 7, 10, 4};

for (int number : array1)

{ System.out.print(number + " "); }

System.out.println();

System.out.println(array1.length + " elements in the collection." );

System.out.println();

int[] array2={6, 8, 7, 10, 15, 11, 14, 13, 12 };

for (int number2 : array2)

{ System.out.print(number2); }

System.out.println();

System.out.println(array2.length + " elements in the collection.");

System.out.println();

int[] numbers3={};

int x = 0;

for (int number : array1)

{

numbers3[x] = number;

System.out.print(number);

}

System.out.println();

System.out.println(numbers3.length + " elements in the collection.");

System.out.println();

System.out.println();

for (int number : array2)

{

//if (!(numbers3.contains(number)))

//Arrays.asList(numbers3).contains(number);

//if(!(ArrayUtils.contains(numbers3,number)))

if(!(Arrays.asList(numbers3).contains(number)))

{

numbers3[x] = number;

System.out.print(number);

}

}

System.out.println();

System.out.println(numbers3.length + " elements in the collection.");

System.out.println();

for (int number : numbers3)

{

System.out.print(number);

}

System.out.println();

System.out.println(numbers3.length + " elements in the collection.");

System.out.println();

int temp = 1;

for (int j = 0; j < numbers3.length - 1 && temp > 0; j++)

{

temp = 0;

for (int i = 0; i < numbers3.length - 1; i++)

{

if (numbers3[i] > numbers3[i + 1])

{

temp = numbers3[i];

numbers3[i] = numbers3[i + 1];

numbers3[i + 1] = temp;

}

}

}

for (int number : numbers3)

{

System.out.print(number);

}

System.out.println();

/\* nothing works \*/

//numbers3.remove(numbers3.length / 2);

//Arrays.asList(numbers3).removes(numbers3.length / 2);

ArrayUtils.removeElement(numbers3, (numbers3.length / 2) );

//ArrayUtils.remove(numbers3, (numbers3.length / 2));

//remove(numbers3,numbers3.length / 2);

/\* still nothing \*/

for (int number : numbers3)

{

System.out.print(number);

}

System.out.println();

for (int i = (numbers3.length - 1); i > -1; i--)

{

System.out.print(numbers3[i]);

}

System.out.println();

System.out.println(numbers3.length + " elements in the collection.");

}

}

}

Define a collection that contains a key/value pair

* Add 5 key/value pairs to the collection
* Add a duplicate key/value pair (one of the five that was added)
  + - Check for duplicate key and print key/value if found (do not add it to the collection)
* Print the key/value pairs
* Print the size of the collection when done
* Comment the code and explain why the specific collection type was chosen.

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

public class CrunchifyHashmapToArrayList {

public static void main(String... args) {

HashMap<String, Integer> companyDetails = new HashMap<String, Integer>();

// create hashmap with keys and values (CompanyName, #Employees)

companyDetails.put("eBay", 4444);

companyDetails.put("Paypal", 5555);

companyDetails.put("IBM", 6666);

companyDetails.put("Google", 7777);

companyDetails.put("Yahoo", 8888);

System.out.println("==> Size of companyDetails Map: " + companyDetails.size());

Iterator it = companyDetails.entrySet().iterator();

while (it.hasNext()) {

Map.Entry pairs = (Map.Entry) it.next();

System.out.println(pairs.getKey() + " = " + pairs.getValue());

}

// Converting HashMap keys into ArrayList

List<String> keyList = new ArrayList<String>(companyDetails.keySet());

System.out.println("\n==> Size of Key list: " + keyList.size());

}

}

Given the following lines of code:

for (int i = 1; i <= 5; i++) {

for (int j = 1; j <= (5 - i); j++) {

System.out.print(".");

}

for (int k = 1; k <= i; k++) {

System.out.print(i);

}

System.out.println();

}

Output:

....1

...22

..333

.4444

55555

Rewrite the code to only use 1 for-loop (You can use any Java built-in library you want).

How would you rewrite this for the nth number instead of ending at 5?

int Nth = 9;

for (int i = 1; i <= Nth; i++)

{

int j = 1;

while (j <= (Nth - i))

{

System.out.print(".");

j++;

}

int k = 1;

while ( k <= i)

{

System.out.print(i);

k++;

}

System.out.println();

}

}

FAQ:

Q: What if I have questions about this test?

A: Contact us with any questions via email to the sender of this document.

Q: What programming languages can I use?

A: Java only.

Q: How should I submit my test?

A: Submit the code to your GitHub account and send a link to your account to the sender of this document. If you do not have a GitHub account, please create one first.