

Status	Finished
Started	Friday, 18 October 2024, 9:30 AM
Completed	Friday, 18 October 2024, 10.15 AM
Duration	45 mins 13 secs

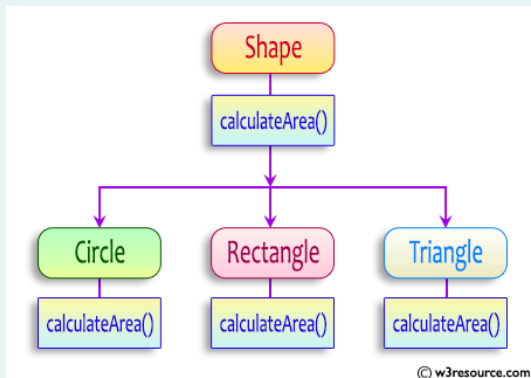
Question 1

Correct

Marked out of 5.00

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:



```

abstract class Shape {
    public abstract double calculateArea() ;
}

```

```
System.out.printf("Area of a Triangle :%.2f\n",((0.5)*base*height)); // use this statement
```

sample Input :

```

4 // radius of the circle to calculate area PI*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle

```

OUTPUT:

Area of a circle :50.27

Area of a Rectangle :30.00

Area of a Triangle :6.00

For example:

Test	Input	Result
1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00
2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 // Abstract class Shape
4 abstract class Shape {
5     public abstract double calculateArea();
6 }
7
8 // Circle class
9 class Circle extends Shape {
10     private double radius;
11
12     public Circle(double radius) {

```

```

13         this.radius = radius;
14     }
15
16     @Override
17     public double calculateArea() {
18         return Math.PI * radius * radius; // Area of circle:  $\pi r^2$ 
19     }
20 }
21
22 // Rectangle class
23 class Rectangle extends Shape {
24     private double length;
25     private double breadth;
26
27     public Rectangle(double length, double breadth) {
28         this.length = length;
29         this.breadth = breadth;
30     }
31
32     @Override
33     public double calculateArea() {
34         return length * breadth; // Area of rectangle: length * breadth
35     }
36 }
37
38 // Triangle class
39 class Triangle extends Shape {
40     private double base;
41     private double height;
42
43     public Triangle(double base, double height) {
44         this.base = base;
45         this.height = height;
46     }
47
48     @Override
49     public double calculateArea() {
50         return 0.5 * base * height; // Area of triangle:  $0.5 * \text{base} * \text{height}$ 
51     }
52 }

```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

1. Final Variable:

- Once a variable is declared **final**, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

```
final int MAX_SPEED = 120; // Constant value, cannot be changed
```

2. Final Method:

- A method declared **final** cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

```
public final void display() {  
    System.out.println("This is a final method.");  
}
```

3. Final Class:

- A class declared as **final** cannot be subclassed (i.e., no other class can inherit from it).
- It is used to prevent a class from being extended and modified.
- ```
public final class Vehicle {
 // class code
}
```

**Given a Java Program that contains the bug in it, your task is to clear the bug to the output.**

**you should delete any piece of code.**

**For example:**

| Test | Result                                                                |
|------|-----------------------------------------------------------------------|
| 1    | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. |

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 // Final class definition
2 final class FinalExample {
3 // Final variable
4 final int MAX_SPEED = 120; // Constant value
5
6 // Final method
7 public final void display() {
8 System.out.println("The maximum speed is: " + MAX_SPEED + " km/h");
9 }
10 }
11
12 // Main class to test the final class
13 public class Test {
14 public static void main(String[] args) {
15 // Create an instance of FinalExample
16 FinalExample example = new FinalExample();
17 example.display();
18
19 // Uncommenting the following line will result in a compile-time error
20 // because FinalExample is a final class and cannot be subclassed.
21 // class SubclassExample extends FinalExample { }
22
23 System.out.println("This is a subclass of FinalExample.");
24 }
25 }
26
27
28
```

|   | Test | Expected                                                              | Got                                                                   |   |
|---|------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|---|
| ✓ | 1    | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. | ✓ |

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

**For example:**

| Input                  | Result           |
|------------------------|------------------|
| 3<br>oreo sirish apple | oreoapple        |
| 2<br>Mango banana      | no matches found |
| 3<br>Ate Ace Girl      | ateace           |

**Answer:** (penalty regime: 0 %)

```
1
2 import java.util.Scanner;
3
4 public class VowelStringExtractor {
5
6 // Method to extract strings with vowels as first and last characters
7 public static String extractVowelStrings(String[] stringArray) {
8 StringBuilder result = new StringBuilder();
9 String vowels = "aeiouAEIOU"; // String containing all vowels
10
11 // Iterate through the array of strings
12 for (String s : stringArray) {
13 // Check if the string is not empty and if both the first and last characters are vowels
14 if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 && vowels.indexOf(s.charAt(s.length() - 1)) != -1) {
15 result.append(s); // Append matching string to the result
16 }
17 }
18
19 // Return the concatenated string in lowercase or "no matches found"
```

```

20 // return the concatenated string in lower case if no matches found
21 return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";
22 }
23
24 public static void main(String[] args) {
25 Scanner scanner = new Scanner(System.in);
26
27 // Input for the number of strings
28
29 int n = scanner.nextInt();
30 scanner.nextLine(); // Consume the newline character
31
32 // Input for the strings in one line
33
34 String input = scanner.nextLine();
35 String[] strings = input.split(" "); // Split input into an array
36
37 // Process and output the result
38 String result = extractVowelStrings(strings);
39 System.out.println(result);
40
41 scanner.close(); // Close the scanner
42 }

```

|   | Input                  | Expected         | Got              |   |
|---|------------------------|------------------|------------------|---|
| ✓ | 3<br>oreo sirish apple | oreoapple        | oreoapple        | ✓ |
| ✓ | 2<br>Mango banana      | no matches found | no matches found | ✓ |
| ✓ | 3<br>Ate Ace Girl      | ateace           | ateace           | ✓ |

Passed all tests! ✓

◀ Lab-08-MCQ

Jump to...

FindStringCode ▶