

CS23333-Object Oriented Programming Using Java-2023

[Dashboard](#) / [My courses](#) / [CS23333-OOPJ-2023](#) / [Lab-08 - Polymorphism, Abstract Classes, final Keyword](#) / [Lab-08-Logic Building](#)

Quiz navigation

1

2

3

[Show one page at a time](#)

[Finish review](#)

Status	Finished
Started	Monday, 7 October 2024, 5:39 PM
Completed	Monday, 7 October 2024, 5:53 PM
Duration	14 mins 9 secs

Question 1

Correct

Marked out of 5.00

[Flag question](#)

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 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 public class VowelStrings {
4
5     public static boolean isVowel(char ch) {
6         ch = Character.toLowerCase(ch);
7         return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
8     }
9
10    public static String findVowelStrings(String[] words) {
11        StringBuilder result = new StringBuilder();
12
13        for (String word : words) {
14            if (word.length() > 0 && isVowel(word.charAt(0)) && isVowel(word.charAt(word.length() - 1))) {
15                result.append(word);
16            }
17        }
18
19        return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";
20    }
21
22    public static void main(String[] args) {
23        Scanner scanner = new Scanner(System.in);
24
25        int n = scanner.nextInt();
26        scanner.nextLine();
27
28        Ate Ace Girl
29
30    }
31 }
```

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

[Flag question](#)

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:

```
classDiagram
    class Shape {
        +calculateArea()
    }
    class Circle {
        +calculateArea()
    }
    class Rectangle {
        +calculateArea()
    }
    class Triangle {
        +calculateArea()
    }
    Shape <|-- Circle
    Shape <|-- Rectangle
    Shape <|-- Triangle
```

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:

```
classDiagram
    class Shape {
        +calculateArea()
    }
    class Circle {
        +calculateArea()
    }
    class Rectangle {
        +calculateArea()
    }
    class Triangle {
        +calculateArea()
    }
    Shape <|-- Circle
    Shape <|-- Rectangle
    Shape <|-- Triangle
```

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

```
1 class FinalExample {
2
3     final int maxSpeed = 120;
4     public final void displayMaxSpeed() {
5         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
6     }
7 }
8
9 class SubClass extends FinalExample {
10     public void showDetails() {
11         System.out.println("This is a subclass of FinalExample.");
12     }
13 }
14
15 public class prog {
16     public static void main(String[] args) {
17         FinalExample obj = new FinalExample();
18         obj.displayMaxSpeed();
19
20         SubClass subObj = new SubClass();
21         subObj.showDetails();
22     }
23 }
24
```

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

Passed all tests! ✓

Finish review

→ Lab-08-MCQ

Jump to...

9

FindStringCode →

```
1 class Circle extends Shape {
2     double radius;
3
4     Circle(double radius) {
5         this.radius = radius;
6     }
7
8     @Override
9     public double calculateArea() {
10         return Math.PI * radius * radius;
11     }
12 }
13
14 class Rectangle extends Shape {
15     double length, breadth;
16
17     Rectangle(double length, double breadth) {
18         this.length = length;
19         this.breadth = breadth;
20     }
21
22     @Override
23     public double calculateArea() {
24         return length * breadth;
25     }
26 }
27
28 class Triangle extends Shape {
29     double base, height;
30
31     Triangle(double base, double height) {
32         this.base = base;
33         this.height = height;
34     }
35
36     @Override
37     public double calculateArea() {
38         return 0.5 * base * height;
39     }
40 }
41
42 public class Main {
43     public static void main(String[] args) {
44         Scanner scanner = new Scanner(System.in);
45         double radius = scanner.nextDouble();
46     }
47 }
```

```
1 abstract class Shape {
2     public abstract double calculateArea();
3 }
4
5 class Circle extends Shape {
6     double radius;
7
8     Circle(double radius) {
9         this.radius = radius;
10    }
11
12    @Override
13    public double calculateArea() {
14        return Math.PI * radius * radius;
15    }
16 }
17
18 class Rectangle extends Shape {
19     double length, breadth;
20
21    Rectangle(double length, double breadth) {
22        this.length = length;
23        this.breadth = breadth;
24    }
25
26    @Override
27    public double calculateArea() {
28        return length * breadth;
29    }
30 }
```

```
32 }
33
34 class Triangle extends Shape {
    // 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 This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

Reset answer

```
1 class FinalExample {
2
3     final int maxSpeed = 120;
4     public final void displayMaxSpeed() {
5         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
6     }
7 }
8
9 class SubClass extends FinalExample {
10     public void showDetails() {
11         System.out.println("This is a subclass of FinalExample.");
12     }
13 }
14
15 public class prog {
16     public static void main(String[] args) {
17         FinalExample obj = new FinalExample();
18         obj.displayMaxSpeed();
19
20         SubClass subObj = new SubClass();
21         subObj.showDetails();
22     }
23 }
24
```