

OOPS

Week 8

1)

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

```
1 import java.util.Scanner;
2 public class Main{
3     public static void main(String[] args){
4         Scanner sc=new Scanner(System.in);
5         int a=sc.nextInt(),c=0;
6         sc.nextLine();
7         String []arr=sc.nextLine().split(" ");
8         for(int i=0;i<a;i++){
9             String w=arr[i].toLowerCase();
10            char s1=w.charAt(0);
11            char s2=w.charAt(arr[i].length()-1);
12            int f1=0,f2=0;
13            if(s1=='a' || s1=='e' || s1=='i' || s1=='o' || s1=='u') f1=1;
14            if(s2=='a' || s2=='e' || s2=='i' || s2=='o' || s2=='u') f2=1;
15            if(f1==1 && f2==1)System.out.print(w);
16            else c++;
17        }
18        if(c==a)System.out.println("no matches found");
19    }
20 }
```

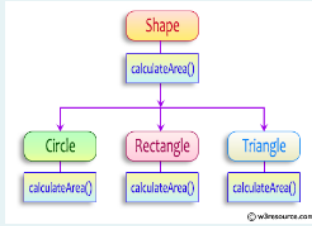
	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

2)

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
```

```
import java.util.*;
```

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

```
class Circle extends Shape {
    private double radius;

    Circle(double r) {
        radius = r;
    }

    double calculateArea() {
        return Math.PI * radius * radius;
    }
}
```

```
class Rectangle extends Shape {
    private double length;
    private double breadth;

    Rectangle(double l, double b) {
        length = l;
        breadth = b;
    }
}
```

```
}  
  
double calculateArea() {  
    return length * breadth;  
}  
}
```

```
class Triangle extends Shape {  
    private double base;  
    private double height;  
    Triangle(double b, double h) {  
        base = b;  
        height = h;  
    }  
    double calculateArea() {  
        return 0.5 * base * height;  
    }  
}
```

```
public class Prog {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        double r = sc.nextDouble();  
        Shape circle = new Circle(r);  
        System.out.println("Area of a circle: "+String.format("%.2f",circle.calculateArea()));  
        double length = sc.nextDouble();  
        double breadth = sc.nextDouble();  
        Shape rectangle = new Rectangle(length, breadth);  
        System.out.println("Area of a Rectangle: " + String.format("%.2f",rectangle.calculateArea()));  
        double base = sc.nextDouble();  
        double height = sc.nextDouble();  
        Shape triangle = new Triangle(base, height);
```

```

        System.out.println("Area of a Triangle: " + String.format("%.2f",triangle.calculateArea()));
    }
}

```

	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! ✓

3)

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
}
```

```
1 class FinalExample {
2
3 // Final variable
4 final int maxSpeed = 120;
5
6 // Final method
7 public final void displayMaxSpeed() {
8 System.out.println("The maximum speed is: " + maxSpeed + " km/h");
9 }
10 }
11 class SubClass extends FinalExample {
12 /*public void displayMaxSpeed() {
13 System.out.println("Cannot override a final method");
14 }*/
15 // You can create new methods here
16 public void showDetails() {
17 System.out.println("This is a subclass of FinalExample.");
18 }
19 }
20 class prog {
21 public static void main(String[] args) {
22 FinalExample obj = new FinalExample();
23 obj.displayMaxSpeed();
24
25 SubClass subObj = new SubClass();
26 subObj.showDetails();
27 }
28 }
29
```

|   | 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! ✓

DHIVYA SHREE K 230