| | |
|---|---|
| **Status** | Finished |
| **Started** | Wednesday, 16 October 2024, 12:56 AM |
| **Completed** | Wednesday, 16 October 2024, 1:40 AM |
| **Duration** | 43 mins 36 secs |

Question **1**

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 %)

```
import java.util.*;
class prog{
    public static void main(String arh[]){
        Scanner sc= new Scanner(System.in);
        int n = sc.nextInt();
        String arr[]= new String[n];
        sc.nextLine();
        String str= sc.nextLine();
        String temp="";
        int j=0;
        int l=str.length();
        for(int i=0;i<l;i++){
            if(str.charAt(i)==' '){
                arr[j]=temp;
                temp="";
                j++;
            }
            else{
                temp+=str.charAt(i);
```

```
                temp= Str.charAt(i);
            }
        }
        arr[j]=temp;
        String s="";
        char []cha={'a','A','e','E','i','I','o','O','u','U'};
        for(int i=0;i<n;i++){
            int c=0;
            char [] ar = arr[i].toCharArray();
            char ch1=ar[0];
            char ch2 = ar[ar.length-1];
            for(char k: cha){
                if(k==ch1){
                    c++;
                }
                if(k==ch2){
                    c++;//s+=arr[i];
                }
            }
            if(c==2){
                s+=arr[i];
            }
    }
    if(s==""){
        System.out.print("no matches found");

    }
    else{
        System.out.print(s.toLowerCase());
    }
}
}
```

| | 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! ✓

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

```
class FinalExample {

    // Final variable
           final     int maxSpeed = 120;

    // Final method
    public           void displayMaxSpeed() {
     System.out.println         ("The maximum speed is: " + maxSpeed + " km/h");
    }
}

class SubClass extends FinalExample {
//@Override
    public void displayMaxSpeed() {
        System.out.println("Cannot override a final method");
    }

    // You can create new methods here
    public void showDetails() {
        System.out.println("This is a subclass of FinalExample.");
    }
}

class prog {
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();

        SubClass subObj = new SubClass();
        subObj.showDetails();
    }
```

```
32 }
33
```

| | 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! ✓

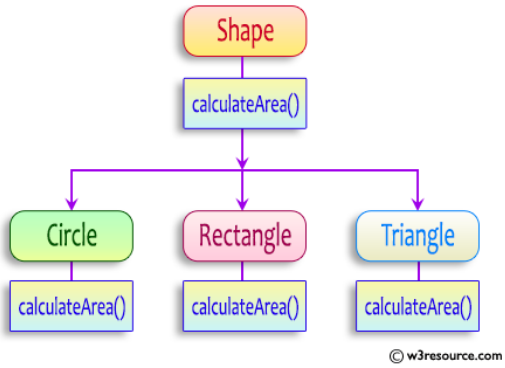
Question **3**

Correct

Marked out of 5.00

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

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

```
import java.util.*;
abstract class Shape{
    abstract void calculatearea();
}
class Circle extends Shape{
    float rad;
    Circle(float rad){
        this.rad=rad;
    }
    void calculatearea(){
        System.out.format("Area of a circle: %.2f\n",3.14159*rad*rad);
    }
```

```
}
class Rectangle extends Shape{
    float l;
    float b;
    Rectangle(float l, float b){
        this.l=l;
        this.b=b;
    }
    void calculatearea(){
        System.out.format("Area of a Rectangle: %.2f\n",(l*b));
    }
}
class Triangle extends Shape{
    float ba;
    float h;
    Triangle(float ba, float h){
        this.ba=ba;
        this.h=h;
    }
    void calculatearea(){
        System.out.format("Area of a Triangle: %.2f\n",(0.5*ba*h));
    }
}
class prog{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        float rad=sc.nextFloat();
        float l= sc.nextFloat();
        float b= sc.nextFloat();
        float ba= sc.nextFloat();
        float h = sc.nextFloat();
        Circle c= new Circle(rad);
        Rectangle r = new Rectangle(l,b);
        Triangle t = new Triangle(ba,h);
        c.calculatearea();
        r.calculatearea();
        t.calculatearea();

    }
}
```

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

Passed all tests! ✓

◄ Lab-08-MCQ

Jump to...

FindStringCode ►