Documentation for Custom Annotation Program

This Java program demonstrates the use of custom annotations. It covers marker interfaces, single-value annotations, and multi-value annotations. In this example, we create a custom annotation named 'Smartphone' with a single value 'os', and apply it to a class 'NokiaService'. The program then uses reflection to retrieve the annotation values.

1. Marker Interface

Marker interfaces are those which do not contain any values or methods. In the provided code, the commented-out interface '@interface Smartphone' is an example of a marker interface.

2. Single Value Annotation

Single-value annotations are those that contain only one value. In this program, '@interface Smartphone' is used as a single-value annotation. It takes a single parameter 'os' which indicates the operating system of the smartphone.

3. Meta Annotations

'@Target' and '@Retention' are examples of meta-annotations in this program:

- '@Target(ElementType.TYPE)': Specifies that the annotation can be applied to classes, interfaces, or enums.
- '@Retention(RetentionPolicy.RUNTIME)': Indicates that the annotation is available at runtime.

4. NokiaService Class

The 'NokiaService' class represents a Nokia smartphone with attributes such as model, size, and storage. This class is annotated with '@Smartphone(os = "Android")', indicating that the operating system is Android.

5. Reflection API

The program uses Java's Reflection API to access the annotation information at runtime. The 'getClass()' method is used to get the class of the 'NokiaService' object, and 'getAnnotation()' retrieves the annotation. Typecasting is then performed to get the annotation's value ('os').

6. Code

```
// Java code for Custom Annotation Example


//Marker Interface

// @interface Smartphone {}


@Target(ElementType.TYPE)

@Retention(RetentionPolicy.RUNTIME)

public @interface Smartphone {

    String os();

}


@Smartphone(os = "Android")

class NokiaService {

    String model;

    double size;

    int storage;


    public NokiaService(String model, double size, int storage) {

        this.model = model;

        this.size = size;

        this.storage = storage;

    }

}
```

```
public class CustomAnnotation {

    public static void main(String[] args) {

        NokiaService objectNokiaService = new NokiaService("Nokia mobile", 5.8, 6);

        System.out.println("Model is: " + objectNokiaService.model);

        System.out.println("Size is: " + objectNokiaService.size);

        System.out.println("Storage is: " + objectNokiaService.storage);


        Class c = objectNokiaService.getClass();

        Annotation an = c.getAnnotation(Smartphone.class);

        Smartphone s = (Smartphone) an;

        System.out.println("Smart phone OS is : " + s.os());

    }

}
```