Doubts

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface Description {
    String author();
    String description();
    int version() default 0;
}
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface Descriptions {
    Description[] value();
}
```

For this if we have not repeatable defined…

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

// Define Description annotation
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface Description {
    String author();
    String description();
    int version() default 0;
}

// Define Descriptions annotation
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface Descriptions {
    Description[] value();
}

// Example usage
public class MyClass {

    @Description(author = "John", description = "This is a sample method",
version = 1)
    public void sampleMethod() {
        // Method implementation
    }
    @Descriptions({
        @Description(author = "Alice", description = "First method", version = 1),
        @Description(author = "Bob", description = "Second method", version = 2)
    })
    public void multipleDescriptions() {
        // Method implementation
    }

    public static void main(String[] args) {
        // Retrieve annotations and process them
        MyClass obj = new MyClass();
        try {
            // Get Description annotation from sampleMethod
            Description description =
obj.getClass().getMethod("sampleMethod").getAnnotation(Description.class);
            if (description != null) {
                System.out.println("Author: " + description.author());
                System.out.println("Description: " + description.description());
                System.out.println("Version: " + description.version());
            }

            // Get Descriptions annotation from multipleDescriptions
            Descriptions descriptions =
obj.getClass().getMethod("multipleDescriptions").getAnnotation(Descriptions.c
lass);
            if (descriptions != null) {
                for (Description desc : descriptions.value()) {
                    System.out.println("Author: " + desc.author());
                    System.out.println("Description: " + desc.description());
                    System.out.println("Version: " + desc.version());
                }
            }
        } catch (NoSuchMethodException e) {
            e.printStackTrace();
        }
    }
}
```

Conclusion

Here you learned how to define custom annotation in Java:

- define annotation retention (SOURCE, CLASS or RUNTIME) or use default one,
- specify annotation target type or use default one,
- create a container class if you want it to be @Repeatable,
- decide if the annotation is @Documented or @Inherited,
- write processing class using java reflection.