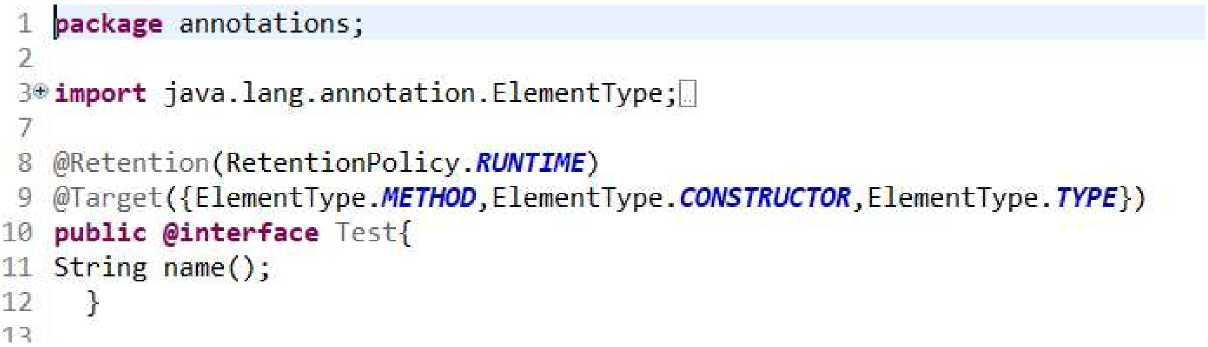
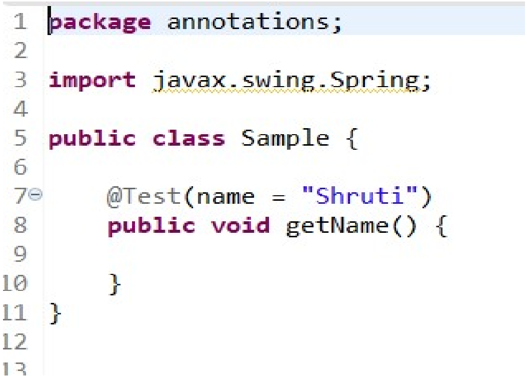
ANNOTATION ASSIGNMENT

Name:Shruti Bhosale

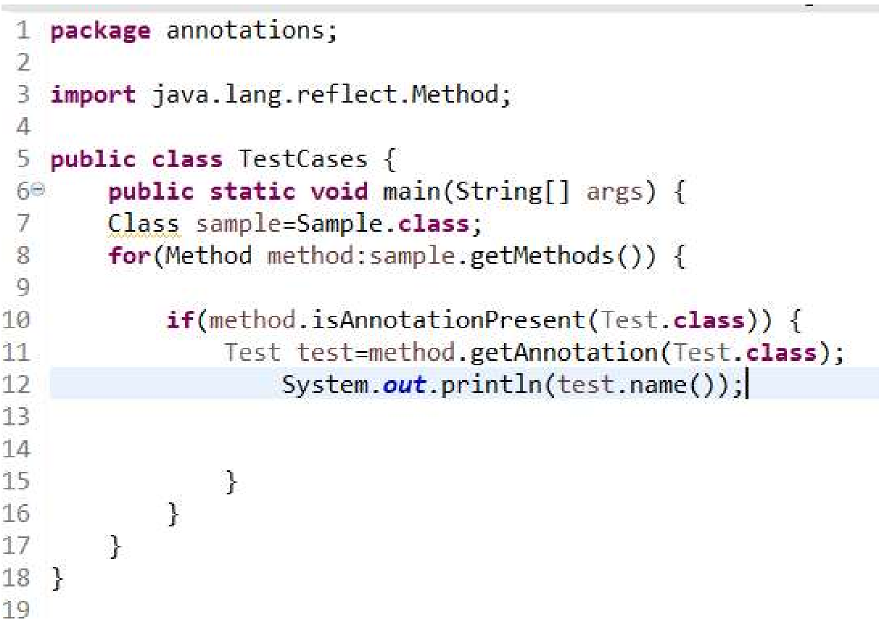
1)Custom Annotation @Type:



```java
package annotations;

import java.lang.annotation.ElementType;

@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.METHOD,ElementType.CONSTRUCTOR,ElementType.TYPE})
public @interface Test{
String name();
  }
```

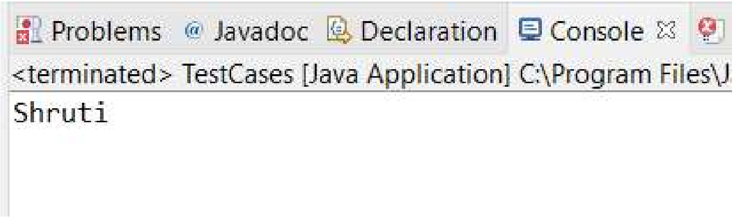


```java
package annotations;

import javax.swing.Spring;

public class Sample {

    @Test(name = "Shruti")
    public void getName() {

    }
}
```



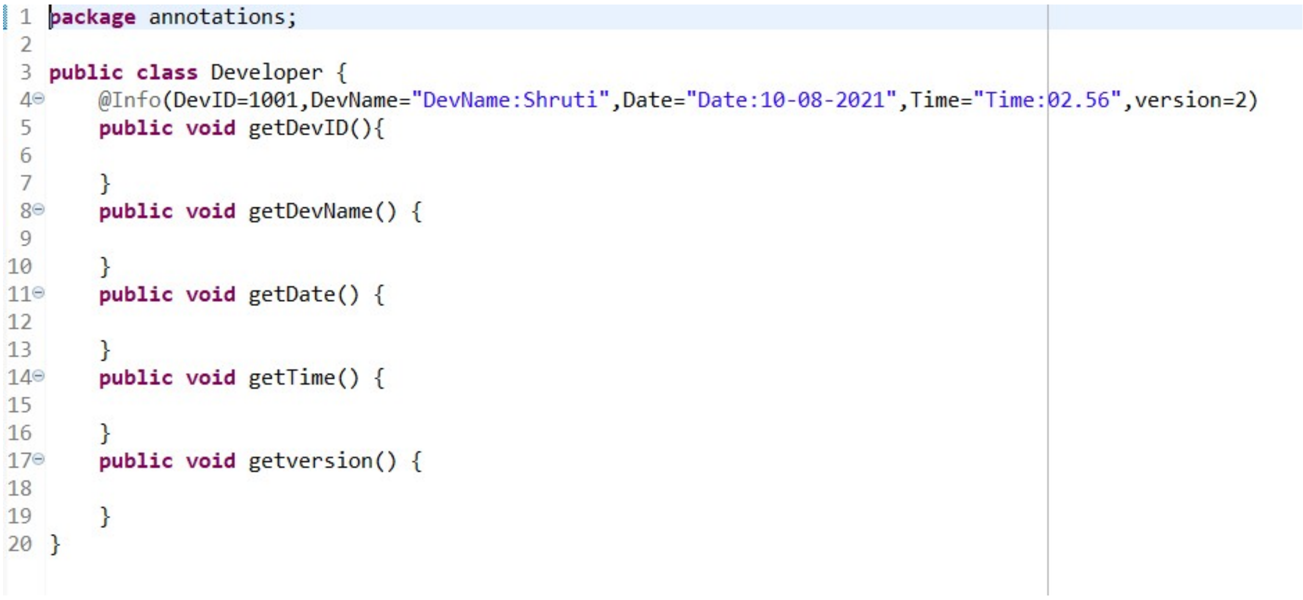
```java
package annotations;

import java.lang.reflect.Method;

public class TestCases {
    public static void main(String[] args) {
    Class sample=Sample.class;
    for(Method method:sample.getMethods()) {

        if(method.isAnnotationPresent(Test.class)) {
            Test test=method.getAnnotation(Test.class);
                System.out.println(test.name());


            }
        }
    }
}
```

Output:

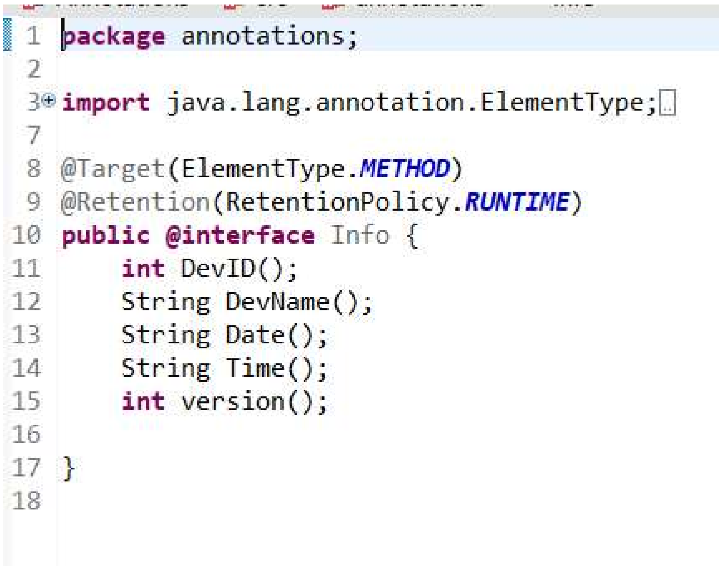


```
<terminated> TestCases [Java Application] C:\Program Files\J
Shruti
```

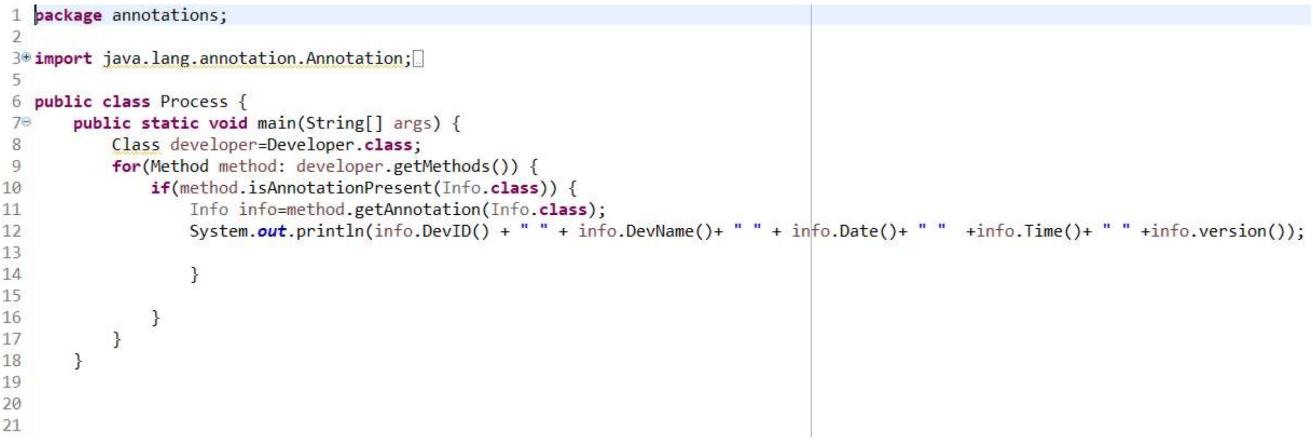
2)Developer:



```java
package annotations;

public class Developer {
    @Info(DevID=1001,DevName="DevName:Shruti",Date="Date:10-08-2021",Time="Time:02.56",version=2)
    public void getDevID(){

    }
    public void getDevName() {

    }
    public void getDate() {

    }
    public void getTime() {

    }
    public void getversion() {

    }
}
```

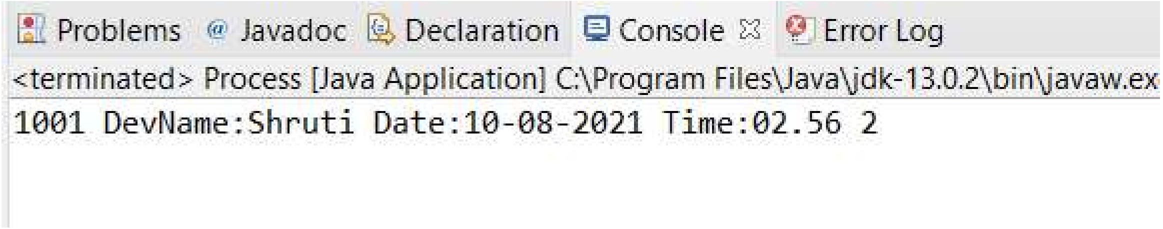


```java
package annotations;

import java.lang.annotation.ElementType;

@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface Info {
    int DevID();
    String DevName();
    String Date();
    String Time();
    int version();

}

```



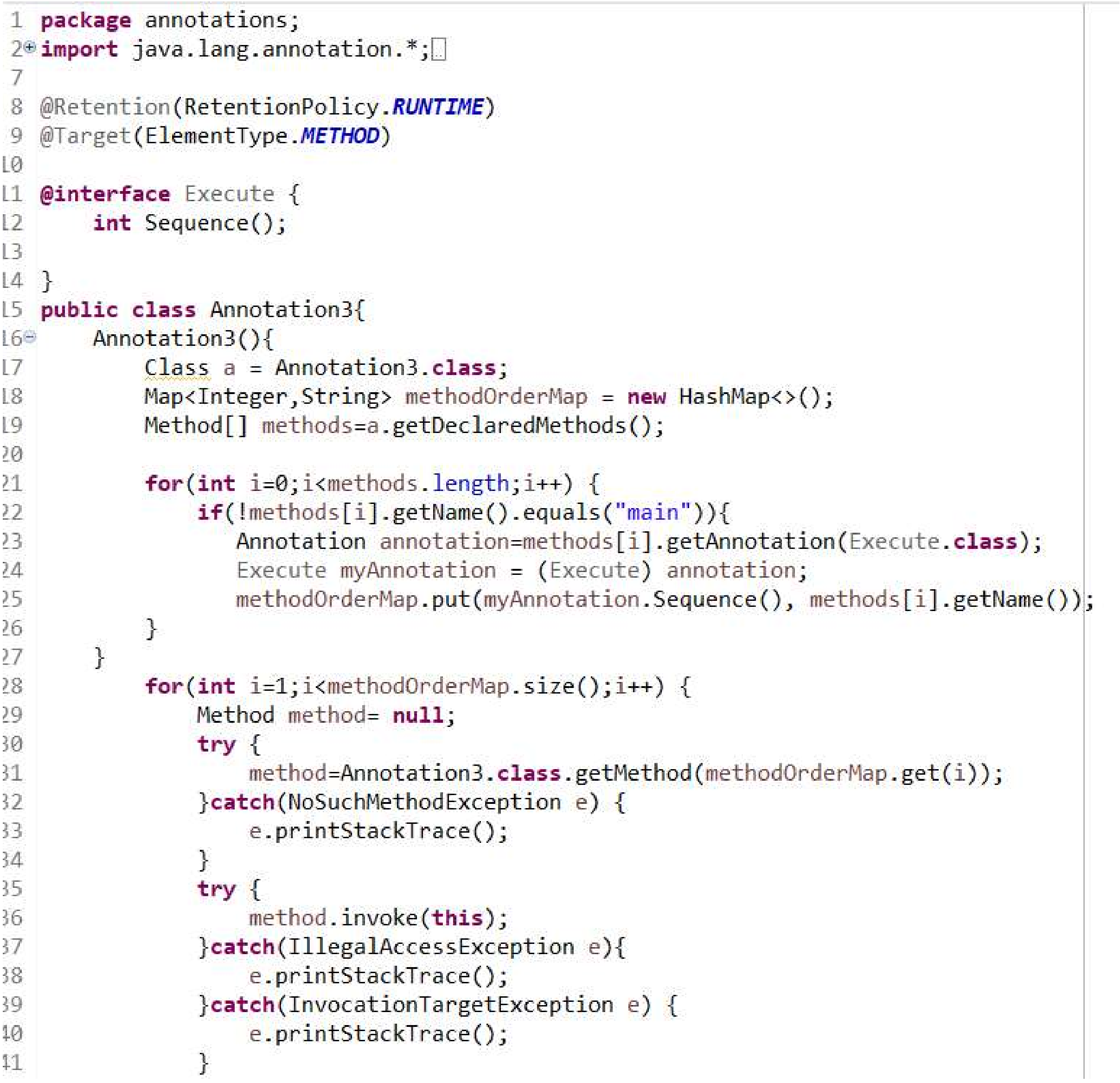
```
package annotations;

import java.lang.annotation.Annotation;

public class Process {
    public static void main(String[] args) {
        Class developer=Developer.class;
        for(Method method: developer.getMethods()) {
            if(method.isAnnotationPresent(Info.class)) {
                Info info=method.getAnnotation(Info.class);
                System.out.println(info.DevID() + " " + info.DevName()+ " " + info.Date()+ " "  +info.Time()+ " " +info.version());

                }

            }
        }
    }



```

Output:

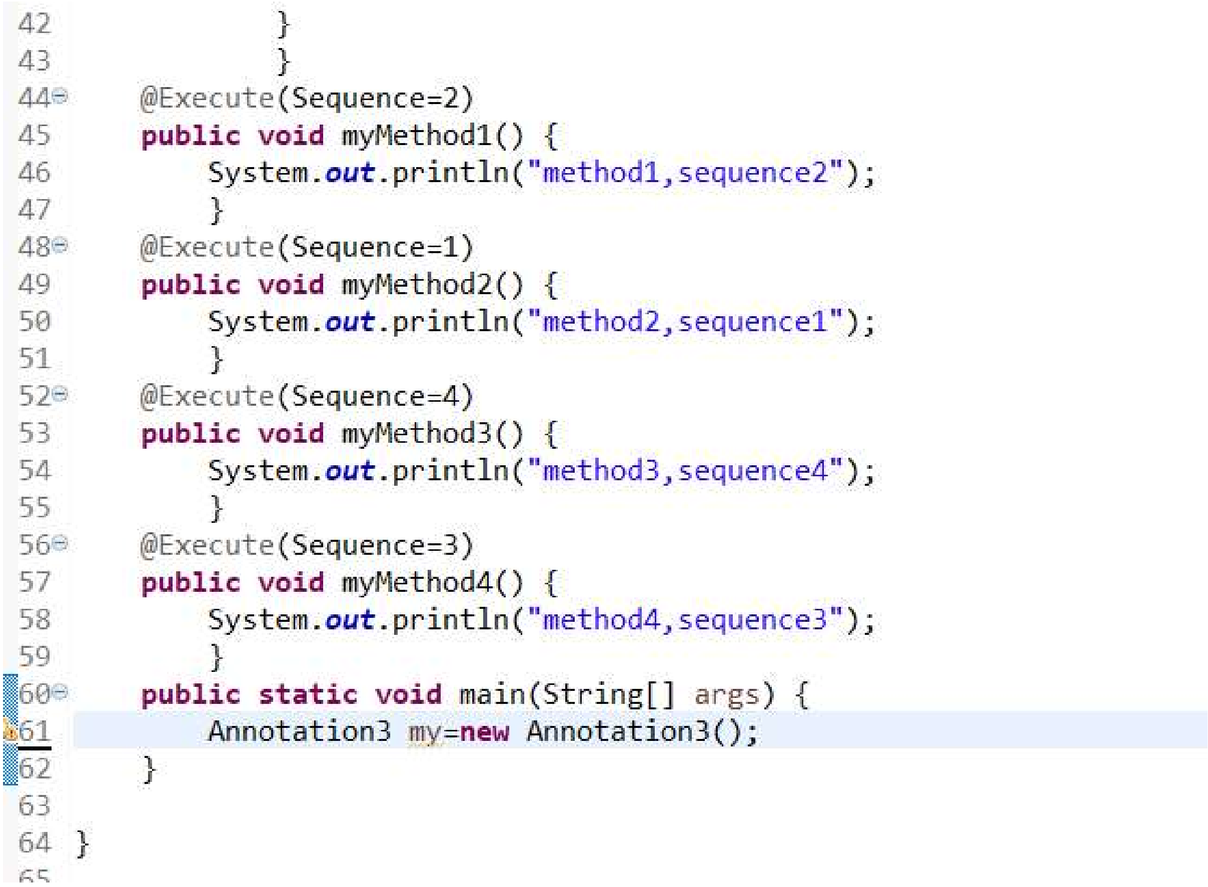


```
<terminated> Process [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.ex
1001 DevName:Shruti Date:10-08-2021 Time:02.56 2
```

3)Reflect:

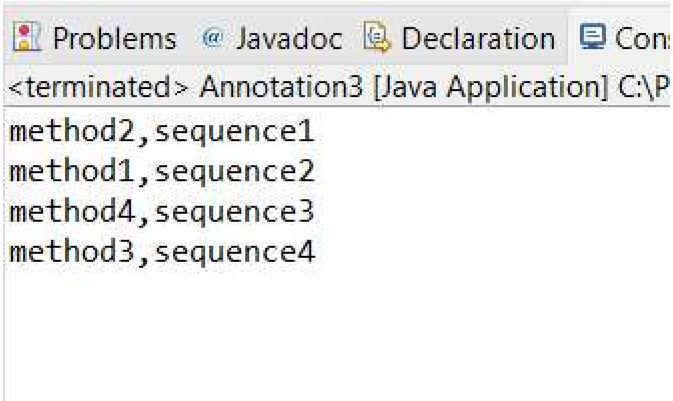


```java
package annotations;
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)

@interface Execute {
    int Sequence();

}
public class Annotation3{
    Annotation3(){
        Class a = Annotation3.class;
        Map<Integer,String> methodOrderMap = new HashMap<>();
        Method[] methods=a.getDeclaredMethods();

        for(int i=0;i<methods.length;i++) {
            if(!methods[i].getName().equals("main")){
               Annotation annotation=methods[i].getAnnotation(Execute.class);
               Execute myAnnotation = (Execute) annotation;
               methodOrderMap.put(myAnnotation.Sequence(), methods[i].getName());
        }
    }
        for(int i=1;i<methodOrderMap.size();i++) {
            Method method= null;
            try {
                method=Annotation3.class.getMethod(methodOrderMap.get(i));
            }catch(NoSuchMethodException e) {
                e.printStackTrace();
            }
            try {
                method.invoke(this);
            }catch(IllegalAccessException e){
                e.printStackTrace();
            }catch(InvocationTargetException e) {
                e.printStackTrace();
            }
```



```java
            }
            }
    @Execute(Sequence=2)
    public void myMethod1() {
        System.out.println("method1,sequence2");
        }
    @Execute(Sequence=1)
    public void myMethod2() {
        System.out.println("method2,sequence1");
        }
    @Execute(Sequence=4)
    public void myMethod3() {
        System.out.println("method3,sequence4");
        }
    @Execute(Sequence=3)
    public void myMethod4() {
        System.out.println("method4,sequence3");
        }
    public static void main(String[] args) {
        Annotation3 my=new Annotation3();
    }

}
```

Output:



```
<terminated> Annotation3 [Java Application] C:\P
method2,sequence1
method1,sequence2
method4,sequence3
method3,sequence4
```