1. **package** custom;

**import** java.lang.annotation.ElementType;

**import** java.lang.annotation.Retention;

**import** java.lang.annotation.RetentionPolicy;

**import** java.lang.annotation.Target;

@Retention(RetentionPolicy.***RUNTIME***)

@Target({ElementType.***METHOD***, ElementType.***CONSTRUCTOR***, ElementType.***PARAMETER***, ElementType.***TYPE***})

**@interface** Test {

}

@Test

**public** **class** Annot1 {

**public** **static** **void** main(String[] args)

{

@Test

**public** **void** T1() {

System.***out***.println("This method is a Test-case");

}

@Test

**public** **static** **void** T2() {

}

}

@Test

**public** **void** T3() {

}

}

1. package custom;

import java.lang.annotation.Annotation;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

import java.lang.reflect.Constructor;

@Retention(RetentionPolicy.RUNTIME)

@Target({ElementType.TYPE, ElementType.METHOD, ElementType.CONSTRUCTOR, ElementType.PARAMETER, ElementType.FIELD})

@interface Info {

int AuthorID();

String Author() default "Lakshman";

String Supervisor() default "Ram";

String Date();

String Time();

int Version();

String Description() default "This method has the information of the developer";

}

@Info(AuthorID = 11, Date = "11-11-2011", Time = "11:00", Version = 11)

class Details{

@Info(AuthorID = 12, Date = "12-12-2012", Time = "12:00", Version = 12)

public Details() {}

}

public class Annot2 {

@SuppressWarnings({ "unchecked", "rawtypes" })

public static void main(String[] args) throws NoSuchMethodException{

Details d = new Details();

Class c = d.getClass();

Annotation anc = c.getAnnotation(Info.class);

Info i = (Info)anc;

System.out.println("\n\nDevelopers ID: "+i.AuthorID()+" ,\nDevelopers name: "+ i.Author()+ " ,\nDevelopers Supervisor: "+i.Supervisor()+ " ,\nDate: "+i.Date()+" ,\nTime: "+i.Time()+" ,\nVersion: "+i.Version()+" ,\nDescription of the Class: "+i.Description()+" ,\nHashcode: "+i.hashCode());

try {

Constructor[] m = Details.class.getConstructors();

Annotation anm = m[0].getAnnotation(Info.class);

Info a = (Info)anm;

System.out.println("\n\nDevelopers ID: "+a.AuthorID()+" ,\nDevelopers name: "+ a.Author()+ " ,\nDevelopers Supervisor: "+a.Supervisor()+ " ,\nDate: "+a.Date()+" ,\nTime: "+a.Time()+" ,\nVersion: "+a.Version()+" ,\nDescription of the Class: "+a.Description()+" ,\nHashcode: "+a.hashCode());

}

catch(NullPointerException e) {

System.out.println("NullPointerException");

}

}

}

1. **package** custom;

**import** java.lang.annotation.ElementType;

**import** java.lang.annotation.Retention;

**import** java.lang.annotation.RetentionPolicy;

**import** java.lang.annotation.Target;

**import** java.lang.reflect.Method;

@Retention(RetentionPolicy.***RUNTIME***)

@Target(ElementType.***METHOD***)

**@interface** execute{

**int** sequence();

}

**class** Myclass {

@execute(sequence = 2)

**public** **void** T1() {

System.***out***.println("This method prints first");

}

@execute(sequence = 1)

**public** **void** T2() {

System.***out***.println("This method prints third");

}

@execute(sequence = 3)

**public** **void** T3() {

System.***out***.println("This method prints second");

}

}

**public** **class** Annot3{

**public** **static** **void** main(String[] args) {

Myclass mc = **new** Myclass();

Method[] c = mc.getClass().getMethods();

**for**(Method m: c) {

execute exe = m.getAnnotation(execute.**class**);

**if**(exe != **null**) {

**try** {

m.invoke(mc);

}**catch**(Exception e) {

e.printStackTrace();

}

}

}

}

}