Assignments 7 -tanaya jadhav

import java.lang.annotation.\*;  
  
//Marker annotations  
//single value annotation  
//multivalued  
//metaannotation  
@Target(ElementType.*TYPE*)  
@Retention(RetentionPolicy.*RUNTIME*)  
@interface Test{  
 String os() default "Symbain";  
 int version() default 1;  
 }  
  
  
@Test(os="andorid",version=6)  
class NokiaAseries {  
 String model;  
 int size;  
  
 public NokiaAseries(String model, int size) {  
 this.model = model;  
 this.size = size;  
 }  
}  
  
public class Annotationtest {  
 public static void main(String[] args)  
 {  
  
 NokiaAseries obj = new NokiaAseries("fire",5);  
 Class c = obj.getClass();  
 Annotation on = c.getAnnotation(Test.class);  
 Test s = (Test) on;  
 System.*out*.println(s.os());  
  
 }  
}

output:-

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\lib\idea\_rt.jar=58940:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Tanaya\IdeaProjects\Corejava\out\production\Corejava Annotationtest

andorid

Process finished with exit code 0

2)

import java.lang.annotation.\*;  
  
//Marker annotations  
//single value annotation  
//multivalued  
//metaannotation  
@Target(ElementType.*TYPE*)  
@Retention(RetentionPolicy.*RUNTIME*)  
@interface Info{  
 String name() default "Tanaya";  
 String Date() default "09/08/2021";  
  
  
 }  
@Info(name = "Amit",Date = "88/99/00")  
class Developer {  
 int DeveloperID;  
 String Date;  
 String Time;  
 int version;  
  
 public Developer(int DeveloperID,String Date,String Time, int version) {  
 this.DeveloperID = DeveloperID;  
 this.Date = Date;  
 this.Time = Time;  
 this.version = version;  
 }  
}  
  
public class Annotationtest {  
 public static void main(String[] args)  
 {  
  
 Developer obj = new Developer(101,"12/2000","12:00",1000);  
 Class c = obj.getClass();  
 Annotation on = c.getAnnotation(Info.class);  
 Info s = (Info) on;  
 System.*out*.println(s.name());  
  
 }  
}

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\lib\idea\_rt.jar=60320:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Tanaya\IdeaProjects\Corejava\out\production\Corejava Annotationtest

Amit

Process finished with exit code 0

import java.lang.annotation.\*;  
  
//Marker annotations  
//single value annotation  
//multivalued  
//metaannotation  
@Target(ElementType.*TYPE*)  
@Retention(RetentionPolicy.*RUNTIME*)  
@interface Info{  
 String name() default "Tanaya";  
 String Date() default "09/08/2021";  
  
  
 }  
@Info(name = "Amit",Date = "88/99/00")  
class Developer {  
 int DeveloperID;  
 String Date;  
 String Time;  
 int version;  
  
 public Developer(int DeveloperID,String Date,String Time, int version) {  
 this.DeveloperID = DeveloperID;  
 this.Date = Date;  
 this.Time = Time;  
 this.version = version;  
 }  
}  
  
public class Annotationtest {  
 public static void main(String[] args)  
 {  
  
 Developer obj = new Developer(101,"12/2000","12:00",1000);  
 Class c = obj.getClass();  
 Annotation on = c.getAnnotation(Info.class);  
 Info s = (Info) on;  
 System.*out*.println(s.Date());  
  
 }  
}

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\lib\idea\_rt.jar=60323:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Tanaya\IdeaProjects\Corejava\out\production\Corejava Annotationtest

88/99/00

Process finished with exit code 0

import java.lang.annotation.\*;  
  
//Marker annotations  
//single value annotation  
//multivalued  
//metaannotation  
@Target(ElementType.*TYPE*)  
@Retention(RetentionPolicy.*RUNTIME*)  
@interface Info{  
 String name() default "Tanaya";  
 String Date() default "09/08/2021";  
  
  
 }  
@Info(name = "Amit",Date = "88/99/00")  
class Developer {  
 int DeveloperID;  
 String Date;  
 String Time;  
 int version;  
  
 public Developer(int DeveloperID,String Date,String Time, int version) {  
 this.DeveloperID = DeveloperID;  
 this.Date = Date;  
 this.Time = Time;  
 this.version = version;  
 }  
}  
  
public class Annotationtest {  
 public static void main(String[] args)  
 {  
  
 Developer obj = new Developer(101,"12/2000","12:00",1000);  
 Class c = obj.getClass();  
 Annotation on = c.getAnnotation(Info.class);  
 Info s = (Info) on;  
 obj.getClass();  
 System.*out*.println(obj);  
  
 }  
}

output:-

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\lib\idea\_rt.jar=56222:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Tanaya\IdeaProjects\Corejava\out\production\Corejava Annotationtest

Developer@50040f0c

Process finished with exit code 0

3)

import java.lang.annotation.\*;  
import java.lang.reflect.InvocationTargetException;  
import java.lang.reflect.Method;  
import java.util.HashMap;  
import java.util.Map;  
  
  
@Retention(RetentionPolicy.*RUNTIME*)  
@Target(ElementType.*METHOD*)  
@interface Execute{  
  
 int sequence();  
}  
public class MyClass {  
 MyClass(){  
 Class aClass = MyClass.class;  
 Map<Integer,String> methodOrderMap = new HashMap<>();  
 Method[] methods = aClass.getDeclaredMethods();//allows all methods  
 //hash map madhe element add kele  
 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;//mapping  
 methodOrderMap.put(myAnnotation.sequence(),methods[i].getName());  
 }  
 }  
 //method sathi  
 for(int i=1;i<=methodOrderMap.size();i++){  
 Method method = null;  
 try {  
 method = MyClass.class.getMethod(methodOrderMap.get(i));  
 } catch (NoSuchMethodException e) {  
 e.printStackTrace();  
 }  
  
 try {  
 method.invoke(this);  
 } catch (IllegalAccessException e) {  
 e.printStackTrace();  
 } catch (InvocationTargetException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 }  
  
 @Execute(sequence=2)  
 public void myMethod1(){  
 System.*out*.println("this is method 1 but sequence 2");  
 }  
  
 @Execute(sequence = 1)  
 public void myMethod2(){  
 System.*out*.println("this is method 2 but is sequence 1" );  
 }  
  
 @Execute(sequence = 4)  
 public void myMethod3(){  
 System.*out*.println("this is method 3 and sequence 4" );  
 }  
  
 @Execute(sequence = 3)  
 public void myMethod4(){  
 System.*out*.println("this is method 4 but sequence 3");  
 }  
  
 public static void main(String args[]){  
 MyClass my = new MyClass();  
 }  
}

output:-

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\lib\idea\_rt.jar=64355:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Tanaya\IdeaProjects\Corejava\out\production\Corejava MyClass

this is method 2 but is sequence 1

this is method 1 but sequence 2

this is method 4 but sequence 3

this is method 3 and sequence 4

Process finished with exit code 0