TASK 01

Tags Annotation:

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Tags;

import org.junit.jupiter.api.Assertions.assertEquals;

class TestCase02 {

    @Test

    @Tags("firstPriority")

    void testMethod01() {

    }

    @Test

    @Tag("firstPriority")

    void runTestcase02() {

    }

    @Tags("fastTag")

    void testMethod03() {

    }

    @Test

    @Tag("slowTag")

    void runTestcase04() {

    }

}

Pom.xml:

<project xmlns="[http://maven.apache.org/POM/4.0.0"](http://maven.apache.org/POM/4.0.0%22)

      xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance%22)

      xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> [https://maven.apache.org/xsd/maven-4.0.0.xsd">](https://maven.apache.org/xsd/maven-4.0.0.xsd%22%3E)

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>  
<artifactId>junit5demo</artifactId>  
<version>0.0.1-SNAPSHOT</version>

  <dependencies>  
<dependency>  
<groupId>org.junit.jupiter</groupId>  
<artifactId>junit-jupiter</artifactId>  
<version>5.10.0</version>  
<scope>test</scope>  
</dependency>  
</dependencies>  
<build>  
<plugins>  
<plugin>  
<groupId>org.apache.maven.plugins</groupId>  
<artifactId>maven-surefire-plugin</artifactId>  
<version>3.1.2</version>  
<configuration>  
<includeTags>firstPriority,fastTag</includeTags>  
<excludeTags>slowTag</excludeTags>  
</configuration>  
</plugin>  
</plugins>  
</build>

 </project>

Code:

package Aug2;  
  
import org.junit.jupiter.api.Tag;  
import org.junit.jupiter.api.Test;  
  
  
class TestCase02 {  
 @Test  
 @Tag("firstPriority")  
 void testAddition() {  
  
 }  
  
 @Test  
 @Tag("firstPriority")  
 void runTestcase02() {  
  
 }  
 @Tag("fastTag")  
 void testMethod03() {  
  
 }  
  
 @Test  
 @Tag("slowTag")  
 void runTestcase04() {  
  
 }  
}

output:

Picked up JAVA\_TOOL\_OPTIONS: -Dlog4j2.formatMsgNoLookups=true

Process finished with exit code 0

**Factory Method Design Pattern**

Product :->       Pizza  interface/abstract class

        preparation();

        baking();

        cutting();

        boxing();

ConcreteProduct --> PepperoniPizza

        //implement all the features of Pizza.

Creator --->          PizzaFactory interface/abstract class

        Pizza createPizza(); - type is Pizza

Concretecreator ---> PepperoniPizzaFactory

        @Override

        Pizza createPizza() {

            ....

            retrun new PepperoniPizza();

        }

//driver class

        class Sams {

            psvm() {

                PizzaFactory pfobj = new PepperoniPizzaFactory();

                Pizza pobj = pfobj.createPizza();

            }

    }

Code:

package Aug2;  
  
public interface Pizza {  
 void preparation();  
 void baking();  
 void cutting();  
 void boxing();  
}  
public class PepperoniPizza implements Pizza {  
  
 @Override  
 public void preparation() {  
 System.*out*.println("Preparing Pepperoni Pizza...");  
 }  
  
 @Override  
 public void baking() {  
 System.*out*.println("Baking Pepperoni Pizza...");  
 }  
  
 @Override  
 public void cutting() {  
 System.*out*.println("Cutting Pepperoni Pizza...");  
 }  
  
 @Override  
 public void boxing() {  
 System.*out*.println("Boxing Pepperoni Pizza...");  
 }  
}  
public abstract class PizzaFactory {  
 public abstract Pizza createPizza();  
}  
public class PepperoniPizzaFactory extends PizzaFactory {  
  
 @Override  
 public Pizza createPizza() {  
 return new PepperoniPizza();  
 }  
}  
public class Sams {  
 public static void main(String[] args) {  
 PizzaFactory factory = new PepperoniPizzaFactory();  
 Pizza pizza = factory.createPizza();  
  
 pizza.preparation();  
 pizza.baking();  
 pizza.cutting();  
 pizza.boxing();  
 }  
}

output:

Preparing Pepperoni Pizza...

Baking Pepperoni Pizza...

Cutting Pepperoni Pizza...

Boxing Pepperoni Pizza...

Process finished with exit code 0