第八章：了解数据驱动的测试 测试题（12） 单选题：

1. Jessica开发了学生管理应用程序，其中包含网页，类和数据库。开发后，jessica希望对应用程序执行数据驱动的测试。为此。他创建了数据文件并向其中添加了一下代码：（ ）

Jessica has developed a student management application that includes web pages, classes, and databases. After development, jessica wanted to perform a data-driven test on the application. to this end. He created the data file and added the code to it

<dataset>

<studentinfo studentID="001" studentName="John" studentMarks="85"/>

<studentinfo studentID="002" studentName="Sarah" studentMarks="95"/>

<studentinfo studentID="003" studentName="Roger" studentMarks="65"/>

</dataset>

A: xml

B: xlsx

C: csv

D: txt

2，以下哪个格式是指通过用户定义的标记以逻辑格式存储的数据？（ ）

Which of the following formats refers to data stored in a logical format through user-defined tags?

A：csv文件

B: 数据库

C: XML文件

D: Excel文件

1. Lolly开发了一个使用mysql数据库的图书管理系统，开发后。她希望使用DbUnit测试此应用程序。她希望在测试用例开始时删除现有值并将输入文件中指定的新值插入到数据库，然而，不希望在测试用例结束时执行任何操作。为此她编写了一下代码段：

Lolly developed a database using the mysql database management system, after development. She wants to use DbUnit to test this application. She wants to delete the existing value at the beginning of the test case and insert the new value specified in the input file into the database, however, do not want to do anything at the end of the test case. For which she wrote a code snippet

public class CustomerDaoTest extends DatabaseTestCase{

public CustomerDaoTest() {

}

@Override

protected IDatabaseConnection getConnection() throws Exception {

Class.forName("com.mysql.jdbc.Driver");

String url="jdbc:mysql://localhost:3306/lolly";

String user="root";

String password="root";

Connection conn=DriverManager.getConnection(url, user, password);

return new DatabaseConnection(conn);

}

@Override

protected IDataSet getDataSet() throws Exception {

InputStream inputStream=new FileInputStream("/input.xml");

IDataSet dataset=new FlatXmlDataSetBuilder().build(inputStream);

return dataset;

}

@Test

public void testSave() throws Exception{

CustomerDao dao=new CustomerDao();

int i=dao.save("zhangsan", "99999");

IDataSet dataSet=getConnection().createDataSet();

ITable table1=dataSet.getTable("customs\_info");

IDataSet data=new FlatXmlDataSetBuilder().build(new FileInputStream("save.xml"));

ITable t=data.getTable("customs\_info");

Assertion.assertEquals(t, table1);

}

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.REFRESH;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.DELETE\_ALL;

}

}

但是，上述代码段没有按预期执行。她应该使用以下哪个代码段实现以上需求？

However, the above code snippet is not executed as expected. Which of the following code snippets should I use to achieve the above requirements

A:

public class CustomerDaoTest extends DatabaseTestCase{

…………………………………..

…………………….

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.REFRESH;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.NONE;

}

}

**B:**

public class CustomerDaoTest extends DatabaseTestCase{

…………………………………..

…………………….

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.REFRESH;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.NONE;

}

}

**C:**

public class CustomerDaoTest extends DatabaseTestCase{

…………………………………..

…………………….

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.CLEAN\_INSERT;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.DELETE\_ALL;

}

}

**D:**

public class CustomerDaoTest extends DatabaseTestCase{

…………………………………..

…………………….

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.CLEAN\_INSERT;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.NONE;

}

}

1. 你开发了应用程序以计算数字的阶乘。现在，您被分配的任务是创建一个测试用例以通过传递不同类

的值（例如一个正的一位数、一个正的两位数，以及一个负数）测试以上类。为此，您创建了一下参数化方法：

You have developed an application to calculate the factorial of the numbers. Now, the task you are assigned is to create a test case to test the class by passing different classes of values (for example, a positive one digit, a positive two digits, and a negative number). To do this, you created a parameterization method

@Parameters

public static Collection<Integer[]> getParemeter(){

Integer[][] faceArrs={

{200,218},{500,558},{100,100}

};

return Arrays.asList(faceArrs);

}

您将使用以下哪个代码段来封装以上@ Parameters 方法?

Which of the following code snippets will you use to encapsulate the above @ Parameters method

**A:**

@RunWith(Parameterized.class)

public class BuyBizTest {

private int a;

private long expected;

public BuyBizTest(long expected , int a ) {

this. expected = expected;

this. a = a;

}

@Parameters

public static Collection<Integer[]> getParemeter(){

Integer[][] paremeters={

{200,218},{500,558},{100,100}

};

return Arrays.asList(paremeters);

}

…………………………………….

**B:**

@RunWith(Parameterized.class)

public class BuyBizTest {

private int a;

private long expected;

public BuyBizTest(int a , long expected ) {

this. a = a;

this. expected = expected;

}

@Parameters

public static Collection<Integer[]> getParemeter(){

Integer[][] paremeters={

{200,218},{500,558},{100,100}

};

return Arrays.asList(paremeters);

}

…………………………………….

**C:**

@RunWith(Parameters.class)

public class BuyBizTest {

private int a;

private long expected;

public BuyBizTest(int a , long expected ) {

this. a = a;

this. expected = expected;

}

@Parameters

public static Collection<Integer[]> getParemeter(){

Integer[][] paremeters={

{200,218},{500,558},{100,100}

};

return Arrays.asList(paremeters);

}

…………………………………….

**D:**

@RunWith(Parameters.class)

public class BuyBizTest {

private int a;

private long expected;

public BuyBizTest((long expected , int a) {

this. expected = expected;

this. a = a;

}

@Parameters

public static Collection<Integer[]> getParemeter(){

Integer[][] paremeters={

{200,218},{500,558},{100,100}

};

return Arrays.asList(paremeters);

}

…………………………………….

5, 事务的哪个特性确保事务完全通过或者完全失败，因为事务不能部分执行。因此数据库中的事务需要测试其完整性？

Which attribute of the transaction ensures that the transaction passes completely or fails completely, because the transaction can not be partially implemented. So the transaction in the database needs to test its integrity

A:原子性

B:一致性

C:隔离性

D:持久性

6,请考虑一下语句：

语句A：持久性表示事务对数据所做的更改必须永久保存在系统中。

语句B：隔离性表示并发事务所做的任何数据修改必须与其他并发事务所做的修改隔离。

Consider the following statement:

   Statement A: Persistence Indicates that the transaction's changes to the data must be permanently saved in the system.

   Statement B: Isolated Indicates that any data modifications made by concurrent transactions must be quarantined with modifications made by other concurrent transactions. A: 语句A和语句B均为False

B: 语句A为False,B为true

C: 语句A和语句B均为true

D: 语句A为true,B为False

7，您为一所学院开发了应用程序以管理其教师数据。开发以后，您需要使用众多输入值测试值测试该应用程序并验证期望值与实际结果。因此，您计划为输入和期望值创建不同的数据文件，而不是为每组值创建多个测试用例。您通过执行以上任务可实现数据驱动测试的哪个优点？

You have developed an application for a college to manage its teacher data. After development, you need to test the application with a number of input value test values and verify the expected and actual results. Therefore, you plan to create different data files for input and expectations, rather than creating multiple test cases for each group of values. What are the advantages of data-driven testing that you can achieve by performing the above tasks?

A:可管理的记录

B:需要使用测试计划中的变更更新数据文件

C:降低了测试用例中的冗余

D:独立测试脚本

8,Emma开发了一个使用MySQL数据库的图书管理应用程序。开发后，她希望使用DbUnit测试此应用程序。她希望在测试用例开始时更新现有值并将输入文件中指定的新值插入到数据库。

此外，她希望在测试用例结束时在数据库表中删除数据文件中提供的所有行。她应该使用以下哪个代码段实现以上需求？

Emma has developed a library management application that uses the MySQL database. After development, she wants to use DbUnit to test this application. She wants to update the existing values at the beginning of the test case and insert the new values specified in the input file into the database. In addition, she wants to delete all the rows provided in the data file in the database table at the end of the test case. Which of the following code snippets should I use to achieve the above requirements

**A:**

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.CLEAN\_INSERT;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.DELETE

}

**B:**

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.CLEAN\_INSERT;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.NONE;

}

**C:**

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.REFRESH;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.DELETE\_ALL;

}

**D:**

@Override

protected DatabaseOperation getSetUpOperation() throws Exception {

return DatabaseOperation.CLEAN\_INSERT;

}

@Override

protected DatabaseOperation getTearDownOperation() throws Exception {

return DatabaseOperation.TRUNCATE\_TABLE;

}

9，以下哪个格式是指存储在逗号分隔文件中的数据？

Which of the following formats refers to the data stored in a comma-delimited file

A: CSV

B: DataBase

C: XML

D: Excel

10,以下哪个数据库操作不对数据库表执行任何操作

Which of the following database operations does not perform any operations on the database tables

A: DatabaseOperation.NONE

B: DatabaseOperation.TRUNCATE\_TABLE

C: DatabaseOperation.DELETE\_ALL

D: DatabaseOperation.REFRESH

11，DatabaseTestCase类是一个抽象类，需要您重写以下两个方法的实现：

问哪两个方法？

The DatabaseTestCase class is an abstract class that requires you to override the implementation of the following two methods:

Ask which two methods

A: protected DatabaseOperation getSetUpOperation() throws Exception

protected DatabaseOperation getTearDownOperation() throws Exception

B: protected abstract IDatabaseConnection getConnection() throws Exception

Protected abstract IDataSet getDataSet() throws Exception

C: protected DatabaseOperation getTearDownOperation() throws Exception

protected IDatabaseConnection getConnection() throws Exception

D: protected abstract DatabaseOperation getSetUpOperation() throws Exception

protected abstract IDatabaseConnection getConnection() throws Exception

12,测试数据库的\_\_\_\_\_\_\_\_\_\_\_\_是指验证一个表中的更改是否如期望那样反应在其他表中。它确保在数据库

中维护了数据的完全性和准确性，以及数据的一致性。

A:原子性

B:一致性

C:数据完整性

D:隔离性

E:持久性

The \_\_\_\_\_\_\_\_\_\_\_\_ of the test database refers to verifying that the changes in a table are responsive to other tables as expected. It ensures that in the database

    Maintaining the completeness and accuracy of the data, as well as the consistency of the data.

  A: Atomicity

  B: Consistency

  C: Data integrity

  D: Isolation

  E: Persistence

在执行数据驱动测试时：

思考studentinfo表的成精，该表存储学生ID,名字和分数。当学生们结束了特定的班级的学习时，需要从现有的班级数据中除去其所有现有数据，并且需要插入新学生的数据。可通过CLEAN\_INSERT操作实现此任务。您已创建了数据文件input.xml以保存要插入的一组新值。接着您需要创建期望的数据文件expecteddata.xml，如以下代码：

哪个是正确的格式？（）

When performing a data-driven test: Think about the results of the studentinfo table, which stores the student ID, name, and score. When students finish a particular class study, they need to remove all of their existing data from the existing class data and need to insert new student data. This can be done with the CLEAN\_INSERT operation. You have created the data file input.xml to save a new set of values to insert. Then you need to create the expected data file expecteddata.xml, as in the following code: Which is the correct format?

A:<dataset>

<studentinfo studentID=001 studentName="John" studentMarks=85/>

<studentinfo studentID=002 studentName="Sarah" studentMarks=95/>

<studentinfo studentID=003 studentName="Roger" studentMarks=65/>

</dataset>

B:<dataset>

<studentinfo studentID=’001’ studentName=’John’ studentMarks=’85’/>

<studentinfo studentID=’002’ studentName=’Sarah’ studentMarks=’95’/>

<studentinfo studentID=’003’ studentName=’Roger’ studentMarks=’65’/>

</dataset>

C:<dataset>

<studentinfo studentID="001" studentName="John" studentMarks=85/>

<studentinfo studentID="002" studentName="Sarah" studentMarks=95/>

<studentinfo studentID="003" studentName="Roger" studentMarks=65/>

</dataset>

D:<dataset>

<studentinfo studentID="001" studentName="John" studentMarks="85"/>

<studentinfo studentID="002" studentName="Sarah" studentMarks="95"/>

<studentinfo studentID="003" studentName="Roger" studentMarks="65"/>

</dataset>