Package hw3TestCases

```
package hw3TestCases
```

Classes

Class	Description
CreateAnswerQuestionTestPage	**Title:** CreateAnswerQuestionTestPage
DeleteAnswerQuestionTestPage	**Title:** DeleteAnswerQuestionTestPage
QuestionAnswerResolvedTestPage	**Title:** QuestionAnswerResolvedTestPage
ReadAnswerQuestionTestPage	**Title:** ReadAnswerQuestionTestPage
UpdateAnswerQuestionTestPage	**Title:** UpdateAnswerQuestionTestPage

Package hw3TestCases

Class CreateAnswerQuestionTestPage

java.lang.Object
    hw3TestCases.CreateAnswerQuestionTestPage

```
public class CreateAnswerQuestionTestPage
extends Object
```

Title: CreateAnswerQuestionTestPage

Description: This JUnit test class verifies that the application correctly allows creation of users, questions, and answers in the system database.

The following scenarios are tested:

- Adding a new question for a specific user
- Adding an answer to a previously inserted question

Each test ensures the underlying database operations complete successfully without throwing exceptions.

Constructor Summary

Constructors

Constructor	Description
`CreateAnswerQuestionTestPage()`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`connect()`	Establishes a database connection once before any test cases are executed.
`void`	`testAddAnswer()`	Tests whether an answer can be successfully added for a given question and user.
`void`	`testAddQuestion()`	Tests whether a question can be successfully added for a given user.

Methods inherited from class java.lang.Object

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

Constructor Details

CreateAnswerQuestionTestPage

```
public CreateAnswerQuestionTestPage()
```

Method Details

connect

```
public static void connect()
                    throws SQLException
```

Establishes a database connection once before any test cases are executed.

Throws:

`SQLException` - if a connection to the database cannot be established

testAddQuestion

```
public void testAddQuestion()
```

Tests whether a question can be successfully added for a given user.

The test simulates a student user creating a new question. It does not validate the returned data, only that the action does not throw an exception.

testAddAnswer

```
public void testAddAnswer()
                   throws SQLException
```

Tests whether an answer can be successfully added for a given question and user.

The test performs a complete flow:

- Registers the user if not already in the system
- Adds a question for that user
- Retrieves the inserted question from the database
- Adds an answer linked to that question

Throws:

`SQLException` - if any database query fails

Package hw3TestCases

Class DeleteAnswerQuestionTestPage

java.lang.Object
    hw3TestCases.DeleteAnswerQuestionTestPage

```
public class DeleteAnswerQuestionTestPage
extends Object
```

Title: DeleteAnswerQuestionTestPage

Description: This JUnit test class verifies the full flow of creating and deleting a `Question` and its associated `Answer` for a newly registered user.

The test ensures that both insertion and deletion operations can be performed without throwing exceptions, confirming data integrity and cleanup logic.

Part of HW3 test suite.

Constructor Summary

Constructors

Constructor	Description
`DeleteAnswerQuestionTestPage()`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`setup()`	Establishes a connection to the database before executing any test cases.
`void`	`testCreateAndDeleteAnswerAndQues`	Performs a complete lifecycle test: Creates a test user Creates a question and an answer Retrieves the inserted data for confirmation Deletes the answer and the question

Methods inherited from class java.lang.Object

`clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait`

Constructor Details

DeleteAnswerQuestionTestPage

```
public DeleteAnswerQuestionTestPage()
```

Method Details

setup

```
public static void setup()
                  throws SQLException
```

Establishes a connection to the database before executing any test cases.

Throws:

`SQLException` - if the database connection fails

testCreateAndDeleteAnswerAndQuestion

```
public void testCreateAndDeleteAnswerAndQuestion()
                                          throws SQLException
```

Performs a complete lifecycle test:

- Creates a test user
- Creates a question and an answer
- Retrieves the inserted data for confirmation
- Deletes the answer and the question

Each step is validated to ensure no exceptions are thrown, indicating that database insertions and deletions are functioning as expected.

Throws:

`SQLException` - if any database operation fails

Package hw3TestCases

Class QuestionAnswerResolvedTestPage

java.lang.Object
    hw3TestCases.QuestionAnswerResolvedTestPage

```
public class QuestionAnswerResolvedTestPage
extends Object
```

Title: QuestionAnswerResolvedTestPage

Description: This JUnit test class verifies whether questions in the system can be correctly marked as resolved or remain unresolved upon creation.

The test cases cover:

- Adding and resolving a question
- Adding a question that remains unresolved by default

Part of HW3 test suite focused on resolution status tracking of questions.

Constructor Summary

Constructors

Constructor	Description
`QuestionAnswerResolvedTestPage()`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`setup()`	Establishes the database connection before any test method runs.
`void`	`testAddResolvedQuestion()`	Tests the process of adding a question and marking it as resolved.
`void`	`testAddUnresolvedQuestion()`	Tests that a newly added question is unresolved by default.

Methods inherited from class java.lang.Object

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

Constructor Details

QuestionAnswerResolvedTestPage

```
public QuestionAnswerResolvedTestPage()
```

Method Details

setup

```
public static void setup()
                  throws SQLException
```

Establishes the database connection before any test method runs.

Throws:

`SQLException` - if database connection fails

testAddResolvedQuestion

```
public void testAddResolvedQuestion()
                             throws SQLException
```

Tests the process of adding a question and marking it as resolved.

This test:

- Creates a unique test user
- Adds a question on behalf of the user
- Marks the question as resolved
- Validates that the resolution flag is set correctly

Throws:

`SQLException` - if any database operation fails

testAddUnresolvedQuestion

```
public void testAddUnresolvedQuestion()
                               throws SQLException
```

Tests that a newly added question is unresolved by default.

This test:

- Creates a unique test user
- Adds a question without modifying its resolution status
- Checks that the resolved flag remains false

Throws:

`SQLException` - if any database operation fails

Package hw3TestCases

Class ReadAnswerQuestionTestPage

java.lang.Object
    hw3TestCases.ReadAnswerQuestionTestPage

```
public class ReadAnswerQuestionTestPage
extends Object
```

Title: ReadAnswerQuestionTestPage

Description: This JUnit test class verifies that the application correctly supports retrieving `Question` and `Answer` records from the database by their IDs.

Test scenarios include:

- Creating and retrieving an answer using its unique ID
- Creating and retrieving a question using its unique ID

These tests confirm the integrity of read operations after data insertion.

Constructor Summary

Constructors

Constructor	Description
`ReadAnswerQuestionTestPage()`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`setup()`	Establishes a connection to the database before any test is executed.
`void`	`testGetAnswerById()`	Tests whether an `Answer` can be correctly retrieved by its unique database ID.
`void`	`testGetQuestionById()`	Tests whether a `Question` can be correctly retrieved by its unique database ID.

Methods inherited from class java.lang.Object

clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Details

ReadAnswerQuestionTestPage

```
public ReadAnswerQuestionTestPage()
```

Method Details

setup

```
public static void setup()
                  throws SQLException
```

Establishes a connection to the database before any test is executed.

Throws:

SQLException - if a database connection cannot be established

testGetAnswerById

```
public void testGetAnswerById()
                       throws SQLException
```

Tests whether an `Answer` can be correctly retrieved by its unique database ID.

This test performs the following steps:

- Creates a user and adds a question
- Adds an answer linked to that question
- Retrieves the answer using its ID
- Verifies that the fetched answer is not null and content matches

Throws:

SQLException - if any database operation fails

testGetQuestionById

```
public void testGetQuestionById()
                         throws SQLException
```

Tests whether a `Question` can be correctly retrieved by its unique database ID.

This test:

- Registers a user and adds a new question
- Retrieves the question using its assigned ID
- Verifies that the retrieved question content matches the original

Throws:

`SQLException` - if any database operation fails

Package hw3TestCases

Class UpdateAnswerQuestionTestPage

java.lang.Object
    hw3TestCases.UpdateAnswerQuestionTestPage

```
public class UpdateAnswerQuestionTestPage
extends Object
```

Title: UpdateAnswerQuestionTestPage

Description: This JUnit test class verifies that a question's content can be successfully updated in the database using the question ID.

The test flow includes:

- Registering a new user
- Adding a question authored by that user
- Updating the content of the question
- Retrieving and verifying that the update was applied

This ensures the system's edit functionality works as expected.

Constructor Summary

Constructors

Constructor	Description
`UpdateAnswerQuestionTestPage()`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`setup()`	Connects to the database once before all test cases are executed.
`void`	`testEditQuestion()`	Tests whether an existing `Question` can be updated with new content.

Methods inherited from class java.lang.Object

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

Constructor Details

UpdateAnswerQuestionTestPage

```
public UpdateAnswerQuestionTestPage()
```

Method Details

setup

```
public static void setup()
                  throws SQLException
```

Connects to the database once before all test cases are executed.

Throws:

`SQLException` - if the database connection fails

testEditQuestion

```
public void testEditQuestion()
                      throws SQLException
```

Tests whether an existing `Question` can be updated with new content.

The test performs the following operations:

- Creates a user and inserts a question
- Modifies the question text using `updateQuestion`
- Retrieves the updated record and verifies its content

Throws:

`SQLException` - if any step of the database operation fails

Hierarchy For Package hw3TestCases

Class Hierarchy

- java.lang.**Object**
  - hw3TestCases.**CreateAnswerQuestionTestPage**
  - hw3TestCases.**DeleteAnswerQuestionTestPage**
  - hw3TestCases.**QuestionAnswerResolvedTestPage**
  - hw3TestCases.**ReadAnswerQuestionTestPage**
  - hw3TestCases.**UpdateAnswerQuestionTestPage**

Index

C D H Q R S T U
All Classes and Interfaces | All Packages

C

connect() - Static method in class hw3TestCases.CreateAnswerQuestionTestPage
  Establishes a database connection once before any test cases are executed.

CreateAnswerQuestionTestPage - Class in hw3TestCases
  Title: CreateAnswerQuestionTestPage

CreateAnswerQuestionTestPage() - Constructor for class hw3TestCases.CreateAnswerQuestionTestPage

D

DeleteAnswerQuestionTestPage - Class in hw3TestCases
  Title: DeleteAnswerQuestionTestPage

DeleteAnswerQuestionTestPage() - Constructor for class hw3TestCases.DeleteAnswerQuestionTestPage

H

hw3TestCases - package hw3TestCases

Q

QuestionAnswerResolvedTestPage - Class in hw3TestCases
  Title: QuestionAnswerResolvedTestPage

QuestionAnswerResolvedTestPage() - Constructor for class hw3TestCases.QuestionAnswerResolvedTestPage

R

ReadAnswerQuestionTestPage - Class in hw3TestCases
  Title: ReadAnswerQuestionTestPage

ReadAnswerQuestionTestPage() - Constructor for class hw3TestCases.ReadAnswerQuestionTestPage

S

setup() - Static method in class hw3TestCases.DeleteAnswerQuestionTestPage
  Establishes a connection to the database before executing any test cases.

setup() - Static method in class hw3TestCases.QuestionAnswerResolvedTestPage
  Establishes the database connection before any test method runs.

setup() - Static method in class hw3TestCases.ReadAnswerQuestionTestPage
  Establishes a connection to the database before any test is executed.

setup() - Static method in class hw3TestCases.UpdateAnswerQuestionTestPage
  Connects to the database once before all test cases are executed.

T

C D H Q R S T U
All Classes and Interfaces | All Packages