

The screenshot shows a Java package summary in the NetBeans IDE. The title bar indicates the file is located at `file:///Users/apple/NetBeansProjects/FacadDesign/dist/javadoc/package-summary.html`. The menu bar includes options like PACKAGE, CLASS, USE, TREE, INDEX, and HELP. The PACKAGE tab is selected. The search bar contains the text "Search". The main content area displays the "Unnamed Package" documentation, which includes sections for All Classes and Interfaces, Interfaces, and Classes. The Classes section lists the following entries:

Class	Description
Circle	
FacadePatternDemo	
Rectangle	
Shape	
ShapeMaker	The ShapeMaker class represents a utility for drawing various shapes.
Square	

## Unnamed Package

All Classes and Interfaces	Interfaces	Classes
Class		Description
Circle		
FacadePatternDemo		
Rectangle		
Shape		
ShapeMaker		The ShapeMaker class represents a utility for drawing various shapes.
Square		

FacadDesign - Apache NetBeans IDE 19

Projects Services Files

RoverTest.java ShapeTester.java

Source History

363.2/432.0MB

Search (⌘+F)

FacadDesign

Source Packages

<default package>

- Circle.java
- FacadePatternDemo.java
- Rectangle.java
- Shape.java
- ShapeMaker.java
- Square.java

Test Packages

<default package>

- ShapeTester.java

java Libraries Test Libraries

Factory Pattern

Source Packages

<default package>

- Circle.java
- FactoryPatternDemo.java
- Rectangle.java
- Shape.java
- ShapeFactory.java
- Square.java

Test Packages Libraries

Notifications

Priority Message Date Created

Test Results

ShapeTester

All 3 tests passed. (0.34 s)

Tests passed: 100.00 %

Test Results Output

1:1 INS

```
1 import org.junit.Test;
2 import static org.junit.Assert.*;
3
4 /**
5 * @author apple
6 * @version 2.0
7 *
8 * The ShapeTester class is responsible for testing the functionality of the ShapeMaker class,
9 * which is designed to create and draw various shapes such as Circle, Rectangle, and Square.
10 * The class contains individual test methods for each shape to ensure their drawing methods
11 * produce the expected output.
12 */
13
14 public class ShapeTester {
15
16     /**
17      * Constructs a new instance of ShapeTester.
18      * The constructor is empty as there is no need for initialization.
19      */
20     public ShapeTester() {
21
22         // Create an instance of ShapeMaker to use its methods for testing
23         ShapeMaker shapeMaker = new ShapeMaker();
24     }
25 }
```

FacadDesign - Apache NetBeans IDE 19

Projects Services Files

FacadePatternDemo.java

Source History

```
1  /**
2  * @author apple
3  * @version 2.0
4  *
5  * The FacadePatternDemo class serves as a demonstration of the Facade design pattern.
6  * It showcases the use of a simplified interface (facade) provided by the ShapeMaker class
7  * to draw various shapes without exposing the complexities of the underlying shape implementations.
8  *
9  */
10 public class FacadePatternDemo {
11
12     /**
13      * The main method demonstrates the Facade design pattern by creating a ShapeMaker object
14      * and using its simplified interface to draw a circle, rectangle, and square.
15      *
16      * @param args The command-line arguments passed to the program. These are not used in this demo.
17      */
18     public static void main(String[] args) {
19
20         // Create a ShapeMaker object
21         ShapeMaker shapeMaker = new ShapeMaker();
22
23     }
}
```

Output - FacadDesign (run)

```
run:
Circle::draw()
Rectangle::draw()
BUILD SUCCESSFUL (total time: 0 seconds)
```

Test Results Output Notifications

The screenshot shows a Java Javadoc package summary page titled "package-summary.html". The browser window has a standard OS X-style title bar with icons for close, minimize, and maximize. The address bar displays the URL: "file:///Users/apple/NetBeansProjects/Factory%20Pattern/dist/javadoc/package-summary.html". Below the address bar is a toolbar with various icons for file operations like save, print, and search. The main content area has a dark blue header with tabs: "PACKAGE" (highlighted), "CLASS", "USE", "TREE", "INDEX", and "HELP". Below the header, there's a search bar with the placeholder "Search" and a clear button. The main content area is titled "PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES".

## Unnamed Package

All Classes and Interfaces	Interfaces	Classes
		Class
		Description
		Circle
FactoryPatternDemo		
		Rectangle
		Shape
ShapeFactory		
		Square

Factory Pattern - Apache NetBeans IDE 19

Projects Services Files

RoverTest.java ShapeTester.java ShapeFactory.java FactoryTest.java

Source History

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit4TestClass.java to edit this template
4   */
5
6  import org.junit.Test;
7  import static org.junit.Assert.*;
8
9 /**
10  *
11  * @author apple
12  * @version 1.0
13  *
14  */
15 public class FactoryTest {
16
17     public FactoryTest() {
18
19         // Create an instance of ShapeFactory
20         ShapeFactory sh = new ShapeFactory();
21
22
23         /**
24          * Test of getShape method, of class ShapeFactory.
25     }
```

Notifications

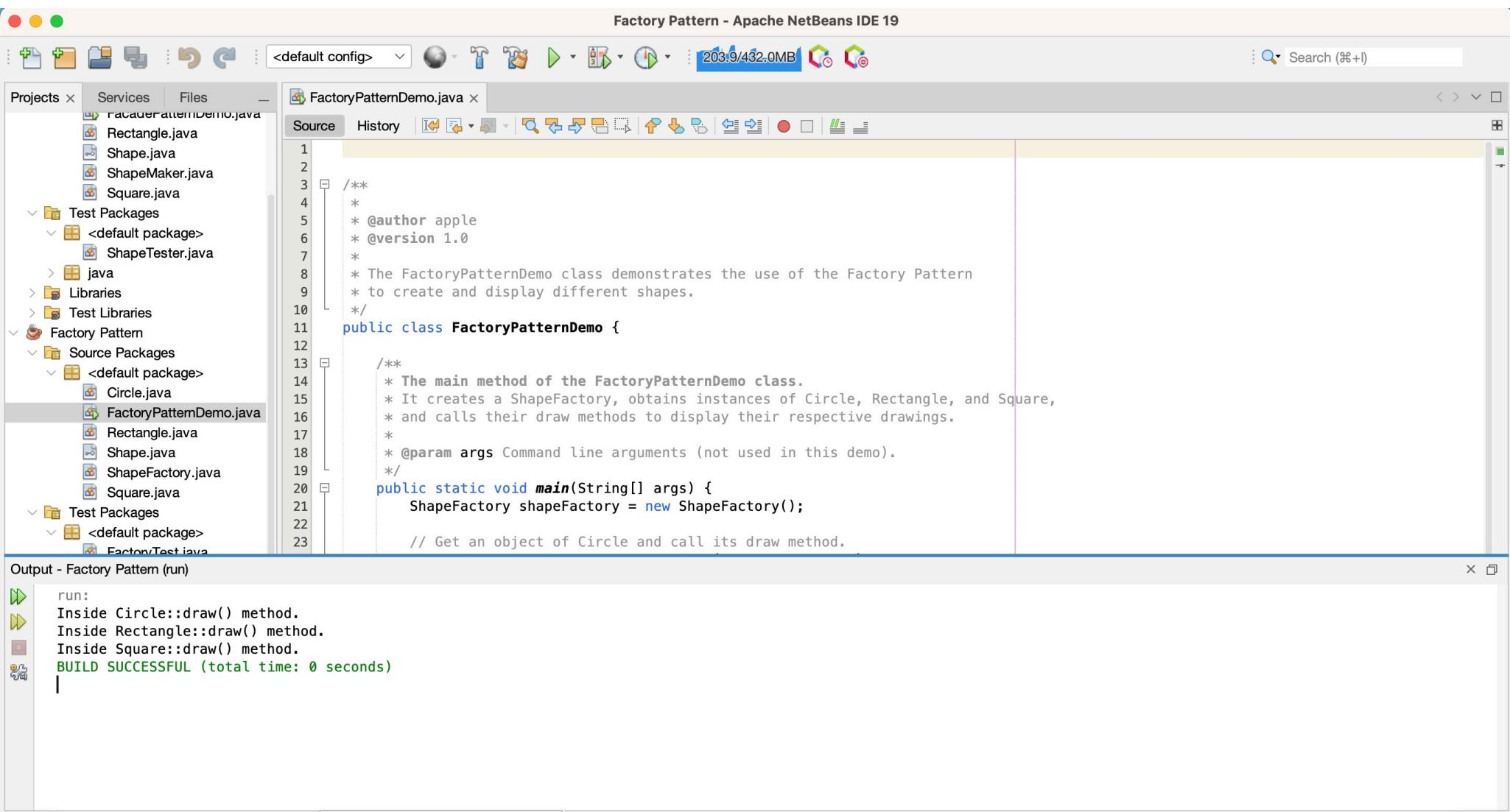
Test Results

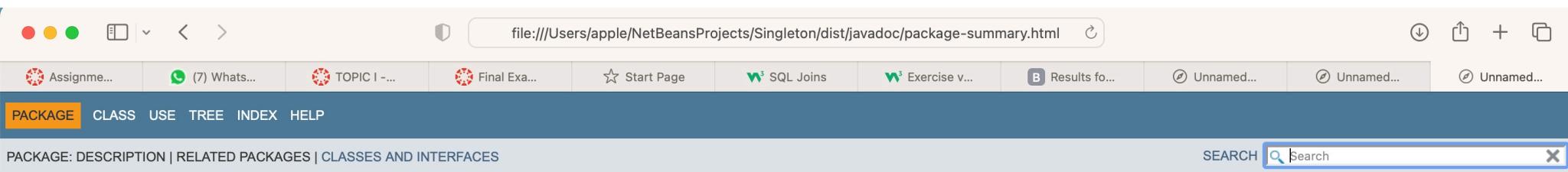
ShapeTester × FactoryTest × SingleObjectTest × SingletonTest ×

All 3 tests passed. (0.13 s)

Tests passed: 100.00 %

Test Results Output





## Unnamed Package

### Classes

Class	Description
SingleObject	
SingletonDemo	

Singleton - Apache NetBeans IDE 19

Projects x Services Files

Rover.java x AtRest.java x SingleObject.java x SingletonDemo.java x

Source History

```
1  /**
2  * 
3  * @author apple
4  * @version 3.0
5  *
6  * The SingletonDemo class demonstrates the use of the Singleton Pattern.
7  * It showcases the instantiation of a singleton object of the SingleObject class
8  * and calls its showMessage method.
9  */
10 */
11
12 public class SingletonDemo {
13
14     /**
15      * The main method of the SingletonPatternDemo class.
16      * It showcases the instantiation of a singleton object and calls its showMessage method.
17      *
18      * @param args Command line arguments (not used in this demo).
19     */

```

Test Results

SingletonTest x

All 3 tests passed. (0.358 s)

Tests passed: 100.00 %

Test Results Output Notifications

32:1 INS

Singleton - Apache NetBeans IDE 19

Projects Services Files

SingletonDemo.java

Source History

```
1  /**
2  * 
3  * @author apple
4  * @version 3.0
5  *
6  * The SingletonDemo class demonstrates the use of the Singleton Pattern.
7  * It showcases the instantiation of a singleton object of the SingleObject class
8  * and calls its showMessage method.
9  */
10 public class SingletonDemo {
11 
12     /**
13      * The main method of the SingletonPatternDemo class.
14      * It showcases the instantiation of a singleton object and calls its showMessage method.
15      *
16      * @param args Command line arguments (not used in this demo).
17      */
18     public static void main(String[] args) {
19         // Illegal construct
20         // Compile Time Error: The constructor SingleObject() is not visible
21         // SingleObject object = new SingleObject();
22     }
23 }
```

Output - Singleton (run)

```
run:
Hello World!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Test Results Output Notifications Finished building Singleton (run).

The screenshot shows a Java Javadoc package summary page titled "Unnamed Package". The page is displayed in a web browser window with a blue header bar containing navigation links like PACKAGE, CLASS, USE, TREE, INDEX, and HELP. The main content area displays a table of classes with their descriptions. The "Classes" tab is selected.

Class	Description
AtRest	The AtRest class represents a state where an object is at rest.
MoveBackward	The MoveBackward class represents a state where an object is moving backward.
MoveForward	The MoveForward class represents a state where an object is moving forward.
Rover	
State	The State class represents an abstract state that can be extended to create specific states for a rover.

StatePattern - Apache NetBeans IDE 19

Projects x Services Files

ShapeMaker.java  
Square.java

Test Packages  
Libraries  
Test Libraries

Factory Pattern  
Filter  
Interface  
JavaJDBC  
javafx  
LAB4CLASS  
Moneycoin  
Multithreading  
Multithreading Bonus  
MVC  
reg  
registration

Singleton

Source Packages

StateTest.java x

Source History

354.3/664.0MB

Search (⌘+I)

1 import org.junit.Test;  
2 import static org.junit.Assert.\*;  
3  
4 /\*\*  
5 \*  
6 \* @author apple  
7 \* @version 3.0  
8 \*/  
9  
10 public class StateTest {  
11  
12     /\*\*  
13      \* Default constructor for the RoverTest class.  
14     \*/  
15     public StateTest() {  
16     }  
17 }  
18  
19

Test Results

StateTest x

All 3 tests passed. (0.346 s)

Tests passed: 100.00 %

Test Results Output Notifications

1:1 INS

StatePattern - Apache NetBeans IDE 19

Projects Services Files

SingletonDemo.java x Rover.java x

Source History

```
1 /**
2 *
3 * @author apple
4 * @version 3.0
5 *
6 * The Rover class represents a rover with various states, including At Rest, Move Forward, and Move Backward.
7 * It provides methods to simulate the actions of pressing the right or left pedal, either a certain number of times
8 * or for a specific duration, as well as printing the current state and subState of the rover.
9 *
10 */
```

Output - StatePattern (run)

```
Current State: At Rest
Current SubState: None
Transitioning from At Rest State to Move Backward State...
Current State: Move Backward
Current SubState: Accelerate
Transitioning from Accelerate State to Constant Speed State...
Current State: Move Backward
Current SubState: Constant Speed
Transitioning from Constant Speed State to Decelerate State...
Current State: Move Backward
Current SubState: Decelerate
Transitioning from Decelerate State to Accelerate State...
Current State: Move Backward
Current SubState: Accelerate
Error: Can only press Left Pedal thrice when in Decelerate State.
Unable to move.
Current State: Move Backward
Current SubState: Accelerate
Transitioning from Accelerate State to Decelerate State...
Current State: Move Backward
Current SubState: Decelerate
Transitioning from Decelerate State to At Rest State...
Current State: At Rest
Current SubState: None
BUILD SUCCESSFUL (total time: 0 seconds)
```

Test Results Output Notifications