Object-Oriented Software Analysis and Design

School of Computer Science
University of Windsor

Test-Driven Development

Test-Driven Development

- An excellent practice promoted by the iterative and agile XP method, and applicable to the UP, is **test-driven development (TDD)**. It is also known as **test-first development**.

- TDD covers more than just unit testing (testing individual components), but this introduction will focus on its application to unit testing individual classes.

Test-Driven Development (contd.)

- In OO unit testing TDD-style, test code is written before the class to be tested, and the developer writes unit testing code for nearly all production code.
- The basic rhythm is to write a little test code, then write a little production code, make it pass the test, then write some more test code, and so forth.
- **Key Point:** The test is written first, imagining the code to be tested is written.

Test-Driven Development: Advantages

- The unit tests actually get written
- Programmer satisfaction leading to more consistent test writing
- Clarification of detailed interface and behavior
- Provable, repeatable, automated verification
- The confidence to change things

Software Testing

- As Edsger Dijkstra, an early contributor to the development of software engineering, eloquently stated (Dijkstra et al., 1972):

  "Testing can only show the presence of errors, not their absence"

- Testing is part of a broader process of software **verification** and **validation** (V & V). Verification and validation are not the same thing, although they are often confused.

  Validation: Are we building the right product?

  Verification: Are we building the product right?

Software Testing

- Typically, a commercial software system has to go through three stages of testing:
- **Development testing**, where the system is tested **during development to discover bugs and defects**. System designers and programmers are likely to be involved in the testing process.
- **Release testing**, where a separate testing team **tests a complete version of the system** before it is released to users. The aim of release testing is to check that the system meets the requirements of the system stakeholders.

Software Testing

- **User testing**, where users or potential users of a system **test the system in their own environment**. For software products, the “user” may be an internal marketing group that decides if the software can be marketed, released and sold.
    - **Acceptance testing** is one type of user testing where the customer formally **tests a system to decide if it should be accepted from the system supplier or if further development is required**.

Development Testing

There are three stages of development testing:

- **Unit testing**, where **individual program units or object classes are tested**. Unit testing should **focus on testing the functionality of objects or methods**.
- **Component testing**, where several individual units are integrated to create composite components. Component testing should **focus on testing the component interfaces** that provide access to the component functions.
- **System testing**, where some or all of the components in a system are integrated and the system is **tested as a whole**. System testing should **focus on testing component interactions**.

Unit testing

- Unit testing is the **process of testing program components**, such as methods or object classes.
- Individual functions or methods are the simplest type of component. Your tests should be calls to these routines with different input parameters.
- When you are testing object classes, you should design your tests to provide coverage of all of the features of the object.

Unit testing (contd.)

- Whenever possible, you should automate unit testing.
- In automated unit testing, you **make use of a test automation framework** (such as JUnit) to write and run your program tests.
- Unit testing frameworks provide generic test classes that you extend to create specific test cases.
- They can then run all of the tests that you have implemented and report, on the success or failure of the tests.
- An entire test suite can often be run in a few seconds so it is possible to execute all the tests every time you make a change to the program.

The Test Pyramid[1]

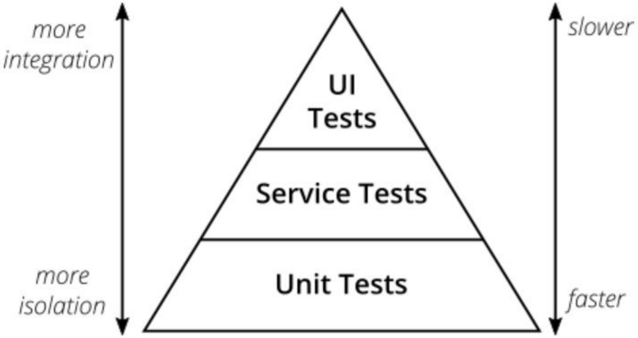


Figure 2: The Test Pyramid

[1] from Martin Fowler's article

Choosing Unit Test Cases

- Testing is expensive and time consuming, so it is important that you choose effective unit test cases. Effectiveness, in this case, means two things:
    - The test cases should show that, when used as expected, the component that you are testing does what it is supposed to do.
    - If there are defects in the component, these should be revealed by test cases.

JUnit Testing

JUnit

- JUnit is an open source unit testing framework for the Java programming language.

Some JUnit Annotations

Annotation	Description
@Test	Denotes that a method is a test method.
@DisplayName	Declares a custom display name for the test class or test method.
@BeforeEach	Denotes that the annotated method should be executed before each test method in the test class.
@AfterEach	Denotes that the annotated method should be executed after each test method in the test class.
@BeforeAll	Denotes that the annotated method should be executed before all test methods in the test class.
@AfterAll	Denotes that the annotated method should be executed after all test methods in the test class.
@Tag	Used to declare tags for filtering tests, either at the class or method level
@Disabled	Used to disable a test class or test method.

Writing Tests: Assertions

- Use the various *assertXXX()* methods to test different conditions.
- `junit.framework.TestCase`, the base class for all test cases, extends from `junit.framework.Assert`, which defines numerous overloaded *assertXXX()* methods. Your tests function by calling these methods.

Writing Tests: Assertions

Method	Description
assertEquals( )	Compares two values for equality. The test passes if the values are equal.
assertFalse( )	Evaluates a boolean expression. The test passes if the expression is false.
assertNotNull( )	Compares an object reference to null. The test passes if the reference is not null.
assertNotSame( )	Compares the memory address of two object references using the == operator. The test passes if both refer to different objects.
assertNull( )	Compares an object reference to null. The test passes if the reference is null.
assertSame( )	Compares the memory address of two object references using the == operator. The test passes if both refer to the same object.
assertTrue( )	Evaluates a boolean expression. The test passes if the expression is true.

It's Quiz Time

1. JUnit provides Assertions for testing expected results. (True or False)

Writing Tests: Example-2

- CalcJUnit

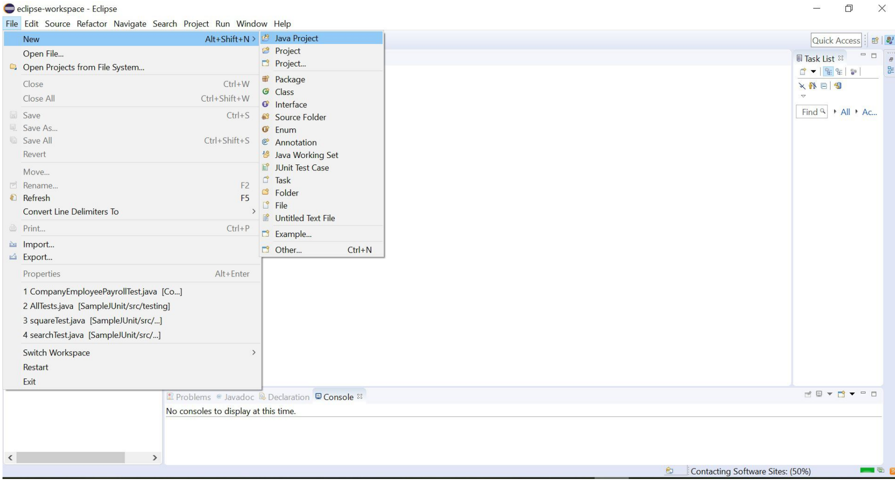
Writing Tests: Example-3

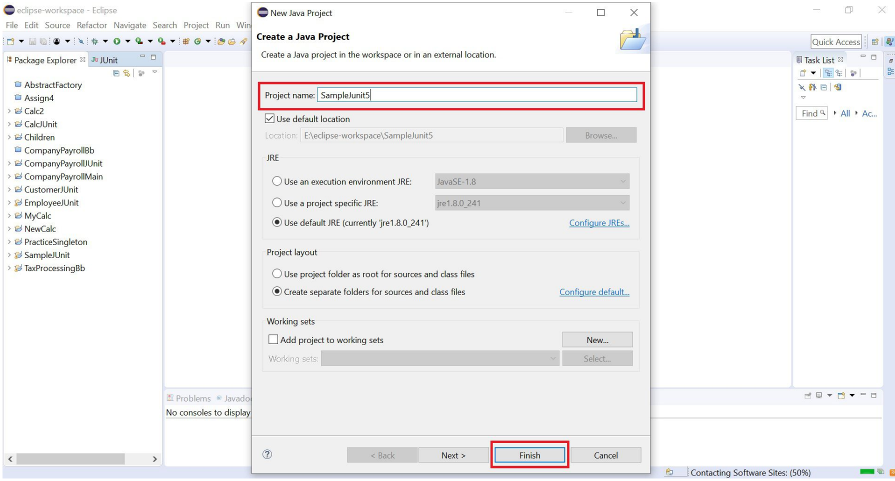
- CustomerJUnit

Writing Tests: Example-4

- EmployeeJUnit

Step-by-step instructions to create a Java project in Eclipse


eclipse-workspace - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import...
Export...
Properties Alt+Enter
1 CompanyEmployeePayrollTest.java [Co...]
2 AllTests.java [SampleJUnit/src/testing]
3 squareTest.java [SampleJUnit/src/...]
4 searchTest.java [SampleJUnit/src/...]
Switch Workspace
Restart
Exit
Java Project
Project
Project...
Package
Class
Interface
Source Folder
Enum
Annotation
Java Working Set
JUnit Test Case
Task
Folder
File
Untitled Text File
Example...
Other... Ctrl+N
Quick Access
Task List
Find All Ac...
Problems Javadoc Declaration Console
No consoles to display at this time.
Contacting Software Sites: (50%)



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: SampleJunit5

[✓] Use default location

Location: E:\eclipse-workspace\SampleJunit5 Browse...

JRE

( ) Use an execution environment JRE: JavaSE-1.8

( ) Use a project specific JRE: jre1.8.0_241

(●) Use default JRE (currently 'jre1.8.0_241') Configure JREs...

Project layout

( ) Use project folder as root for sources and class files

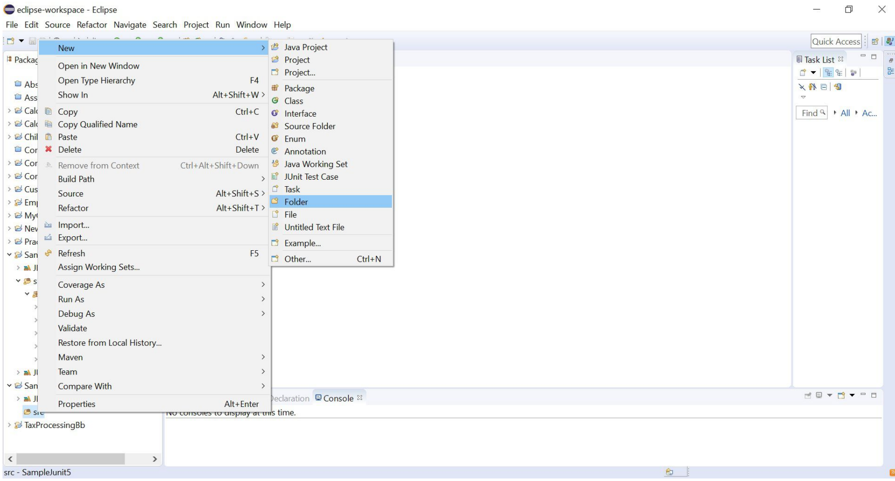
(●) Create separate folders for sources and class files Configure default...

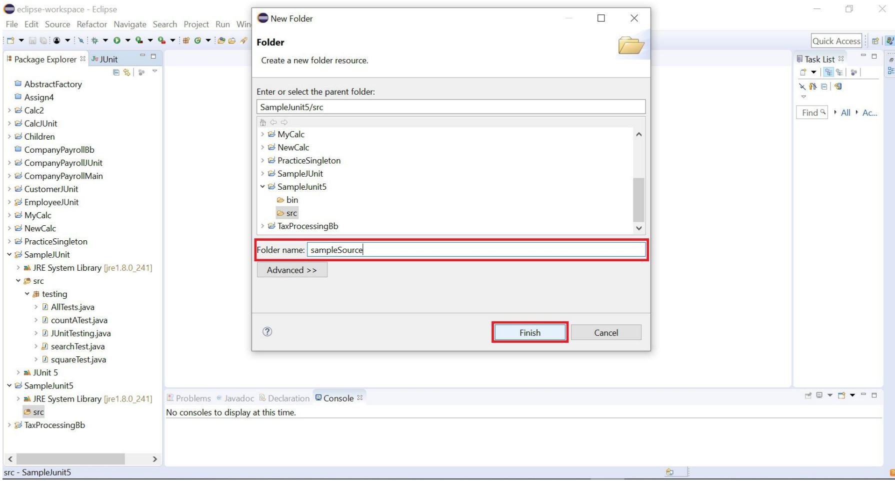
Working sets

[ ] Add project to working sets New...

Working sets: Select...

< Back | Next > | Finish | Cancel


eclipse-workspace - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Coverage As
Run As
Debug As
Validate
Restore from Local History...
Maven
Team
Compare With
Properties Alt+Enter
Java Project
Project
Project...
Package
Class
Interface
Source Folder
Enum
Annotation
Java Working Set
JUnit Test Case
Task
Folder
File
Untitled Text File
Example...
Other... Ctrl+N
Quick Access
Task List
Find
All
Ac...
Console
No consoles to display at this time.
TaxProcessingBb
src - SampleJunit5



New Folder

Folder

Create a new folder resource.

Enter or select the parent folder:

SampleJunit5/src

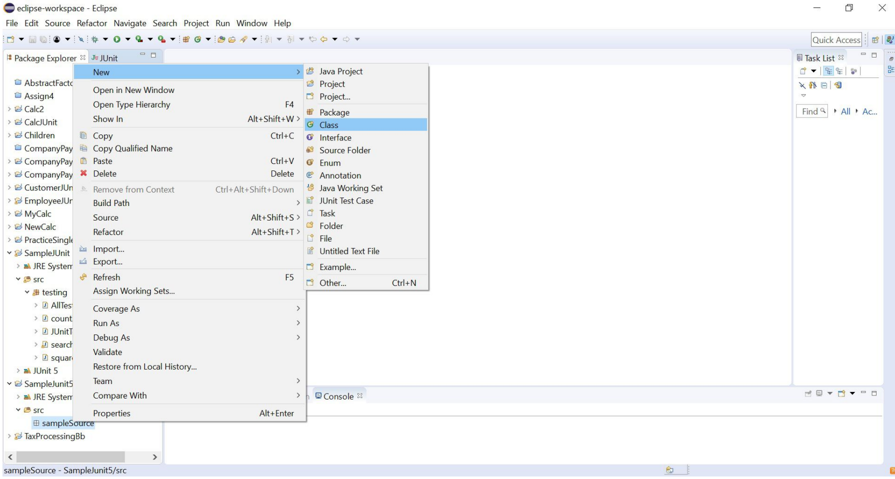
- MyCalc
- NewCalc
- PracticeSingleton
- SampleJUnit
- SampleJunit5
  - bin
  - src
- TaxProcessingBb

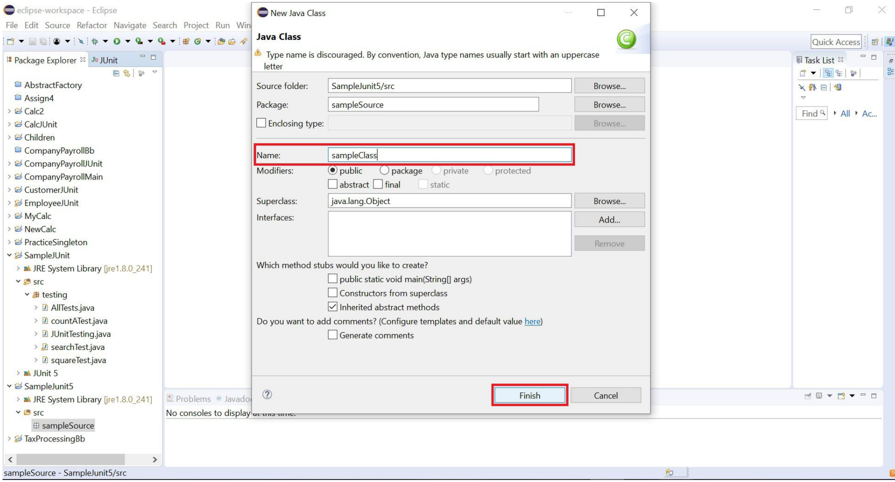
Folder name: sampleSource

Advanced >>

Finish Cancel

No consoles to display at this time.


eclipse-workspace - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
JUnit
Task List
Find
All
Ac...
AbstractFacto
Assign4
Calc2
CalcJUnit
Children
CompanyPay
CompanyPay
CompanyPay
CustomerJUn
EmployeeJUn
MyCalc
NewCalc
PracticeSingle
SampleJUnit
JRE System
src
testing
AllTes
count
JUnitT
search
squar
JUnit 5
SampleJunit5
JRE System
src
sampleSource
TaxProcessingBb
New
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Coverage As
Run As
Debug As
Validate
Restore from Local History...
Team
Compare With
Properties Alt+Enter
Java Project
Project
Project...
Package
Class
Interface
Source Folder
Enum
Annotation
Java Working Set
JUnit Test Case
Task
Folder
File
Untitled Text File
Example...
Other... Ctrl+N
Console
sampleSource - SampleJunit5/src


eclipse-workspace - Eclipse
File Edit Source Refactor Navigate Search Project Run Win
Package Explorer
JUnit
Quick Access
Task List
Find
All
Ac...
Problems
Javado

AbstractFactory
Assign4
Calc2
CalcJUnit
Children
CompanyPayrollBb
CompanyPayrollJUnit
CompanyPayrollMain
CustomerJUnit
EmployeeJUnit
MyCalc
NewCalc
PracticeSingleton
SampleJUnit
JRE System Library [jre1.8.0_241]
src
testing
AllTests.java
countATest.java
JUnitTesting.java
searchTest.java
squareTest.java
JUnit 5
SampleJunit5
JRE System Library [jre1.8.0_241]
src
sampleSource
TaxProcessingBb

No consoles to display at this time.

sampleSource - SampleJunit5/src

New Java Class

Java Class

Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder:	SampleJunit5/src	Browse...
Package:	sampleSource	Browse...
☐ Enclosing type:		Browse...
Name:	sampleClass	
Modifiers:	● public ○ package ○ private ○ protected	
	☐ abstract ☐ final ☐ static	
Superclass:	java.lang.Object	Browse...
Interfaces:		Add... / Remove

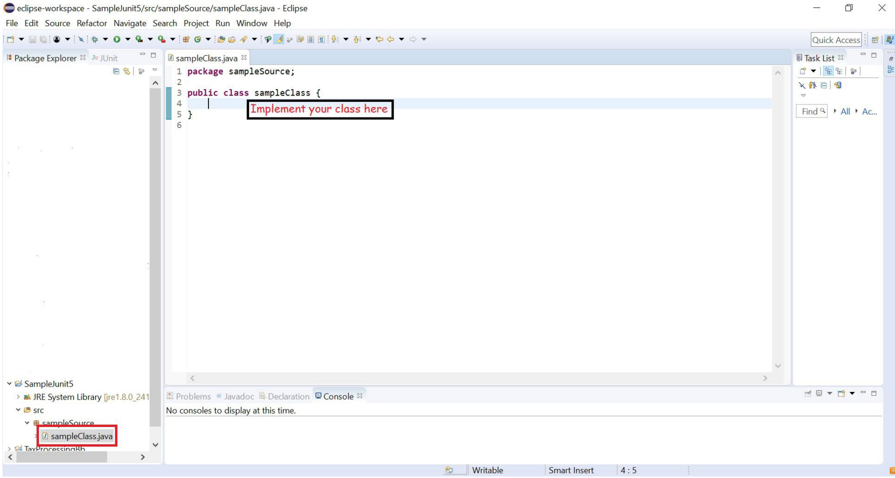
Which method stubs would you like to create?

- ☐ public static void main(String[] args)
- ☐ Constructors from superclass
- ☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

- ☐ Generate comments

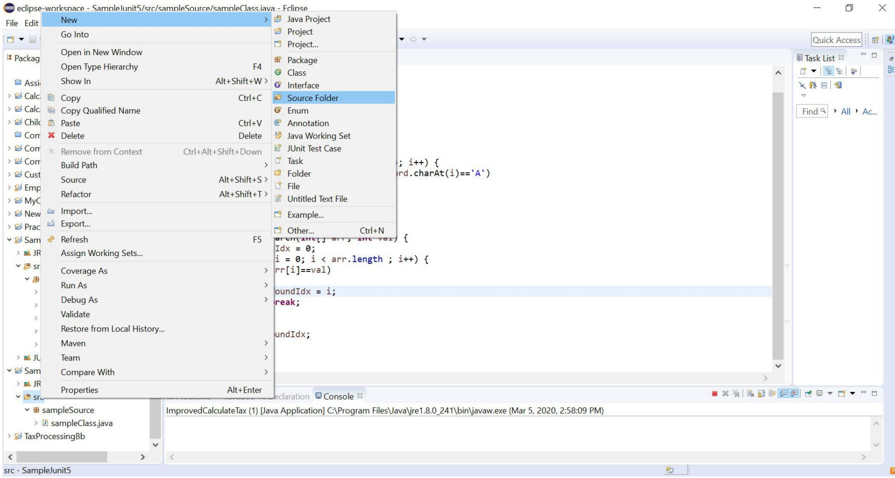
Finish Cancel



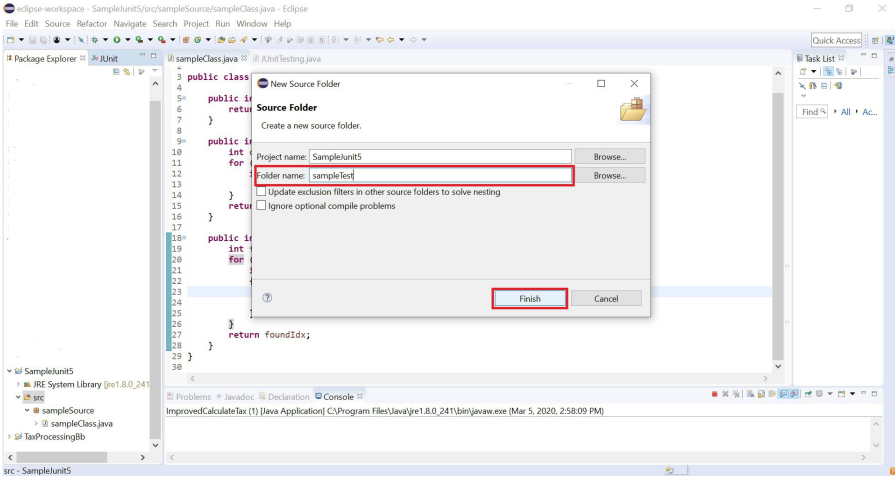
```java
package sampleSource;

public class sampleClass {
	
}
```

Implement your class here

No consoles to display at this time.



ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)


eclipse-workspace - SampleJunit5/src/sampleSource/sampleClass.java - Eclipse

```
public class
    public in
        retur
    }

    public in
        int
        for

        }
        retur
    }

    public in
        int
        for

        }
        return foundIdx;
    }
}
```

New Source Folder

Source Folder

Create a new source folder.

Project name: SampleJunit5

Folder name: sampleTest

☐ Update exclusion filters in other source folders to solve nesting

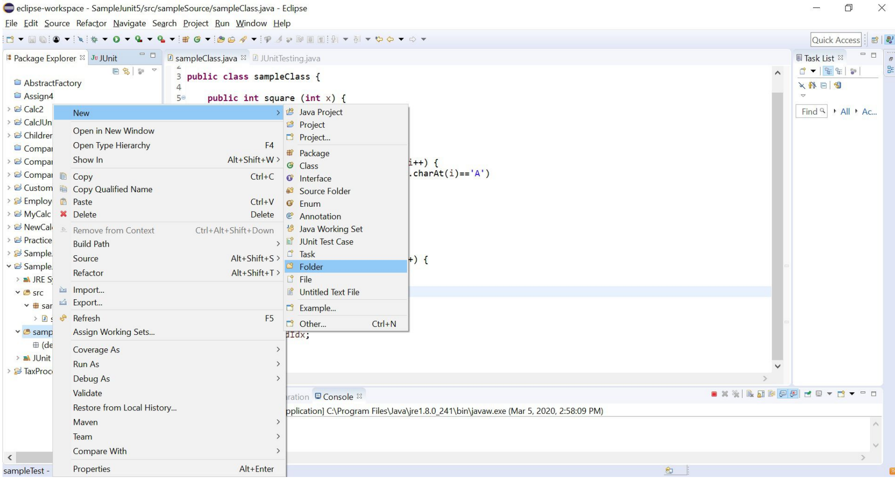
☐ Ignore optional compile problems

SampleJunit5
- JRE System Library [jre1.8.0_241
- src
  - sampleSource
    - sampleClass.java

TaxProcessingBb

```
ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)
```

src - SampleJunit5



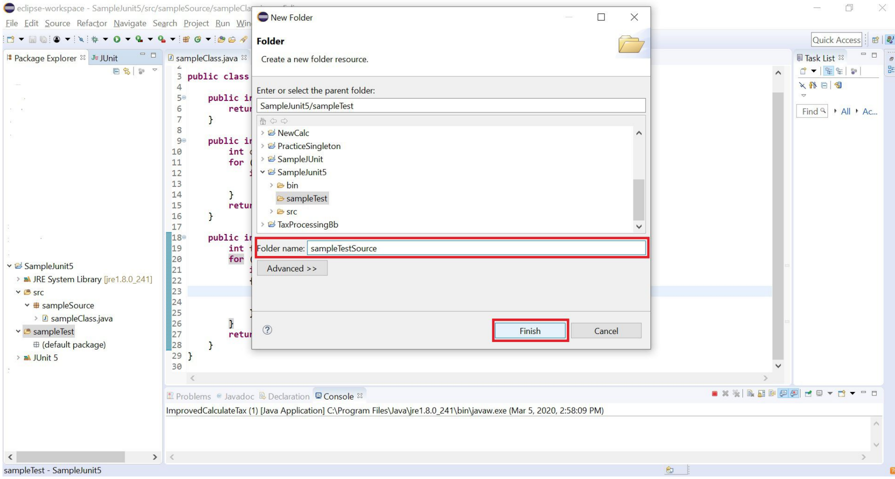
```
public class sampleClass {

    public int square (int x) {
```

```
i++) {
.charAt(i)=='A')
```

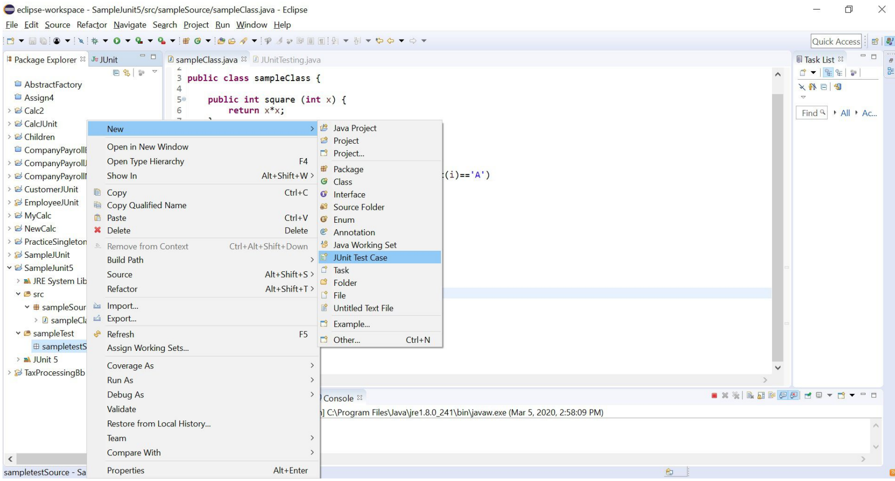
```
+) {
```

```
dIdx;
```

```
pplication] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)
```



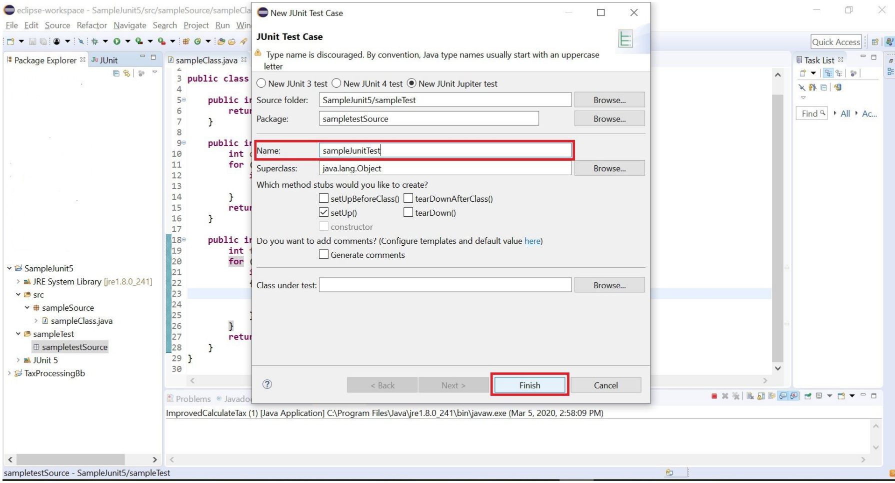
ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)



```
public class sampleClass {

    public int square (int x) {
        return x*x;
```

```
(i)=='A')
```

```
] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)
```


eclipse-workspace - SampleJunit5/src/sampleSource/sampleCla
File Edit Source Refactor Navigate Search Project Run Win
Package Explorer
JUnit
sampleClass.java
Quick Access
Task List
Find
All
Ac...
Problems
Javadoc

New JUnit Test Case

JUnit Test Case

Type name is discouraged. By convention, Java type names usually start with an uppercase letter

New JUnit 3 test [✓ New JUnit Jupiter test] New JUnit 4 test

Source folder:	SampleJunit5/sampleTest	Browse...
Package:	sampletestSource	Browse...
Name:	sampleJunitTest	
Superclass:	java.lang.Object	Browse...

Which method stubs would you like to create?

- ☐ setUpBeforeClass()
- ☐ tearDownAfterClass()
- ☑ setUp()
- ☐ tearDown()
- ☐ constructor

Do you want to add comments? (Configure templates and default value here)

- ☐ Generate comments

Class under test:

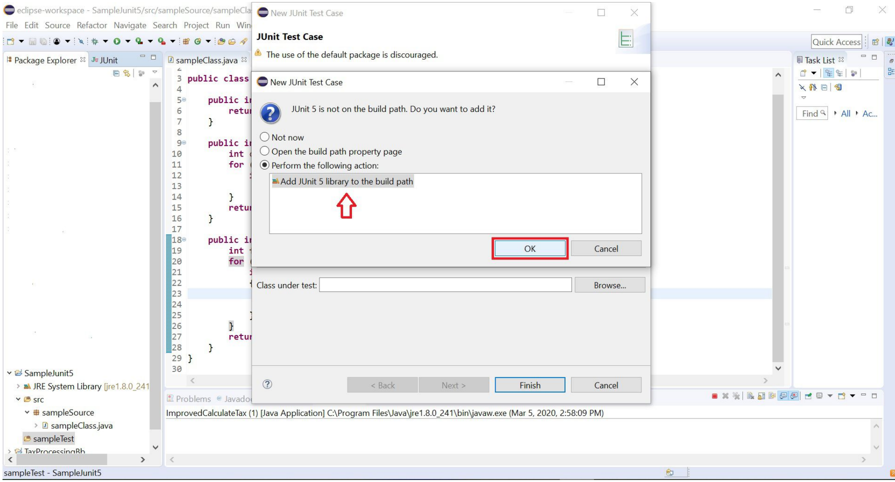
< Back Next > Finish Cancel

```
public class
    public in
        retur
    }
    public in
        int
        for
        }
        retur
    }
    public in
        int
        for
        }
        retur
    }
}
```

- SampleJunit5
  - JRE System Library [jre1.8.0_241]
  - src
    - sampleSource
      - sampleClass.java
  - sampleTest
    - sampletestSource
  - JUnit 5
- TaxProcessingBb

ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)

sampletestSource - SampleJunit5/sampleTest


eclipse-workspace - SampleJunit5/src/sampleSource/sampleCla
File Edit Source Refactor Navigate Search Project Run Win
Package Explorer
JUnit
sampleClass.java
Quick Access
Task List
Find
All
Ac...
SampleJunit5
JRE System Library [jre1.8.0_241
src
sampleSource
sampleClass.java
sampleTest
TaxProcessingBb
Problems
Javado
sampleTest - SampleJunit5

```
public class
    public i
        retu
    }
    public i
        int
        for
        }
        retu
    }
    public i
        int
        for
        }
        retu
    }
}
```

New JUnit Test Case

JUnit Test Case

The use of the default package is discouraged.

New JUnit Test Case

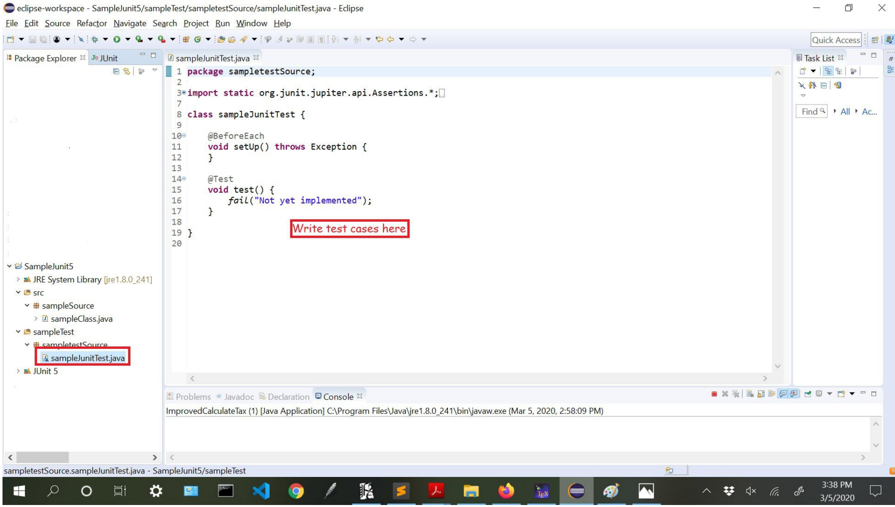
JUnit 5 is not on the build path. Do you want to add it?

- ( ) Not now
- ( ) Open the build path property page
- [✓ Perform the following action:]

Add JUnit 5 library to the build path

Class under test:

ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)


eclipse-workspace - SampleJunit5/sampleTest/sampletestSource/sampleJunitTest.java - Eclipse

```java
package sampletestSource;

import static org.junit.jupiter.api.Assertions.*;

class sampleJunitTest {

    @BeforeEach
    void setUp() throws Exception {
    }

    @Test
    void test() {
        fail("Not yet implemented");
    }

}
```

Write test cases here

- SampleJunit5
  - JRE System Library [jre1.8.0_241]
  - src
    - sampleSource
      - sampleClass.java
  - sampleTest
    - sampletestSource
      - sampleJunitTest.java
  - JUnit 5

ImprovedCalculateTax (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 2:58:09 PM)

sampletestSource.sampleJunitTest.java - SampleJunit5/sampleTest