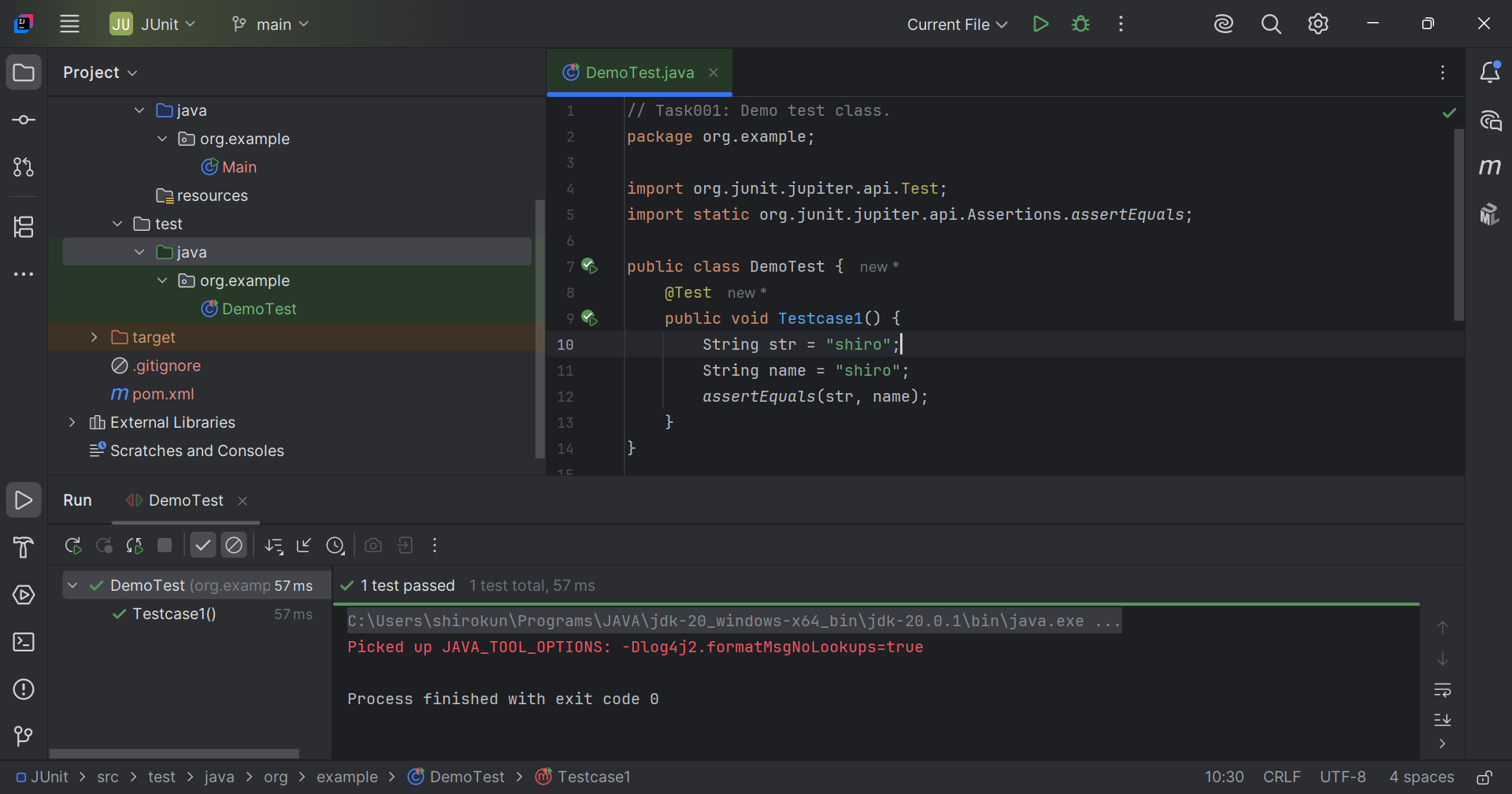
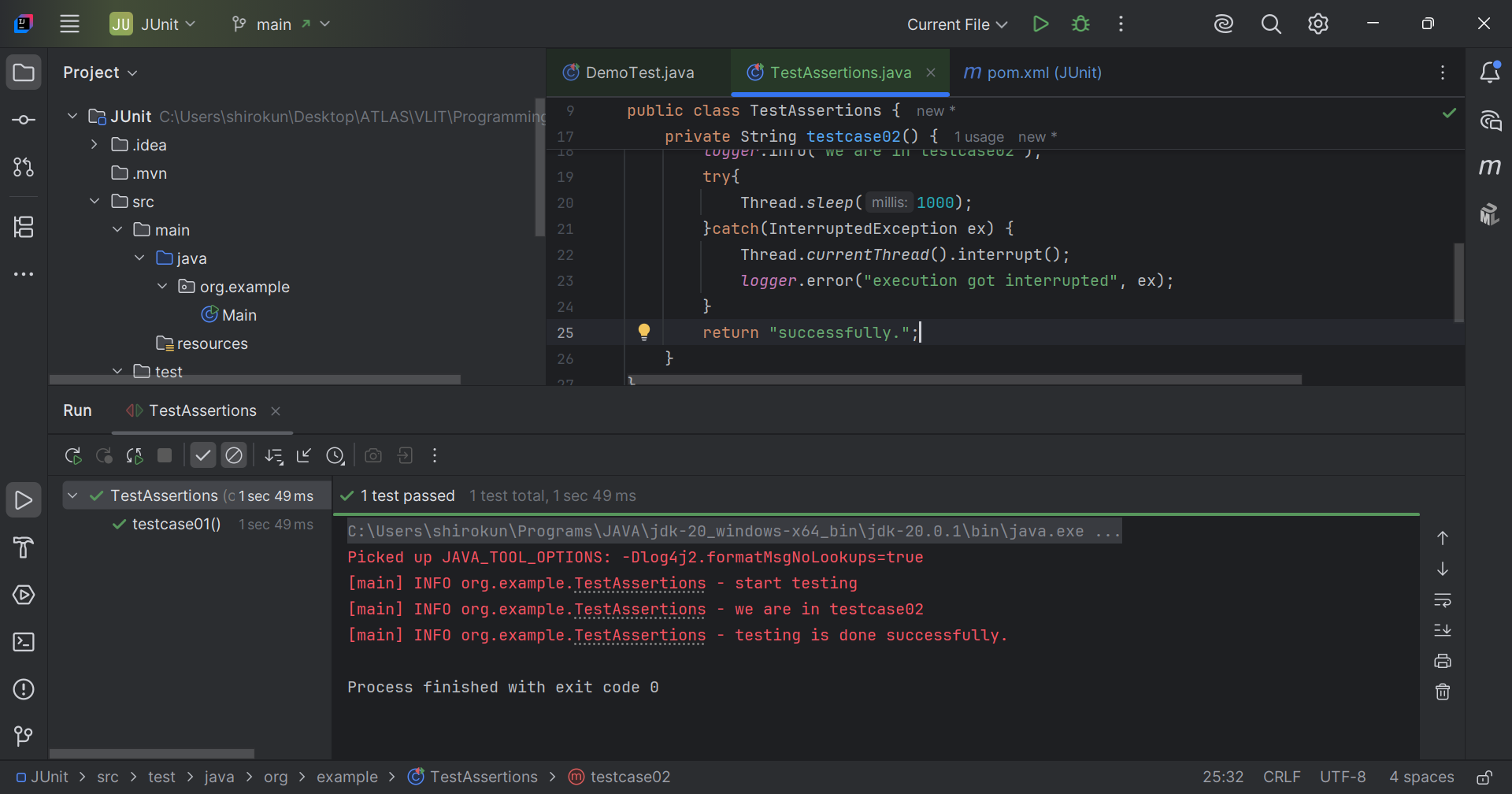
Day 36 – 26/09/2025

// Task001: Demo test class.  
package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class DemoTest {  
 @Test  
 public void Testcase1() {  
 String str = "shiro";  
 String name = "shiro";  
 *assertEquals*(str, name);  
 }  
}

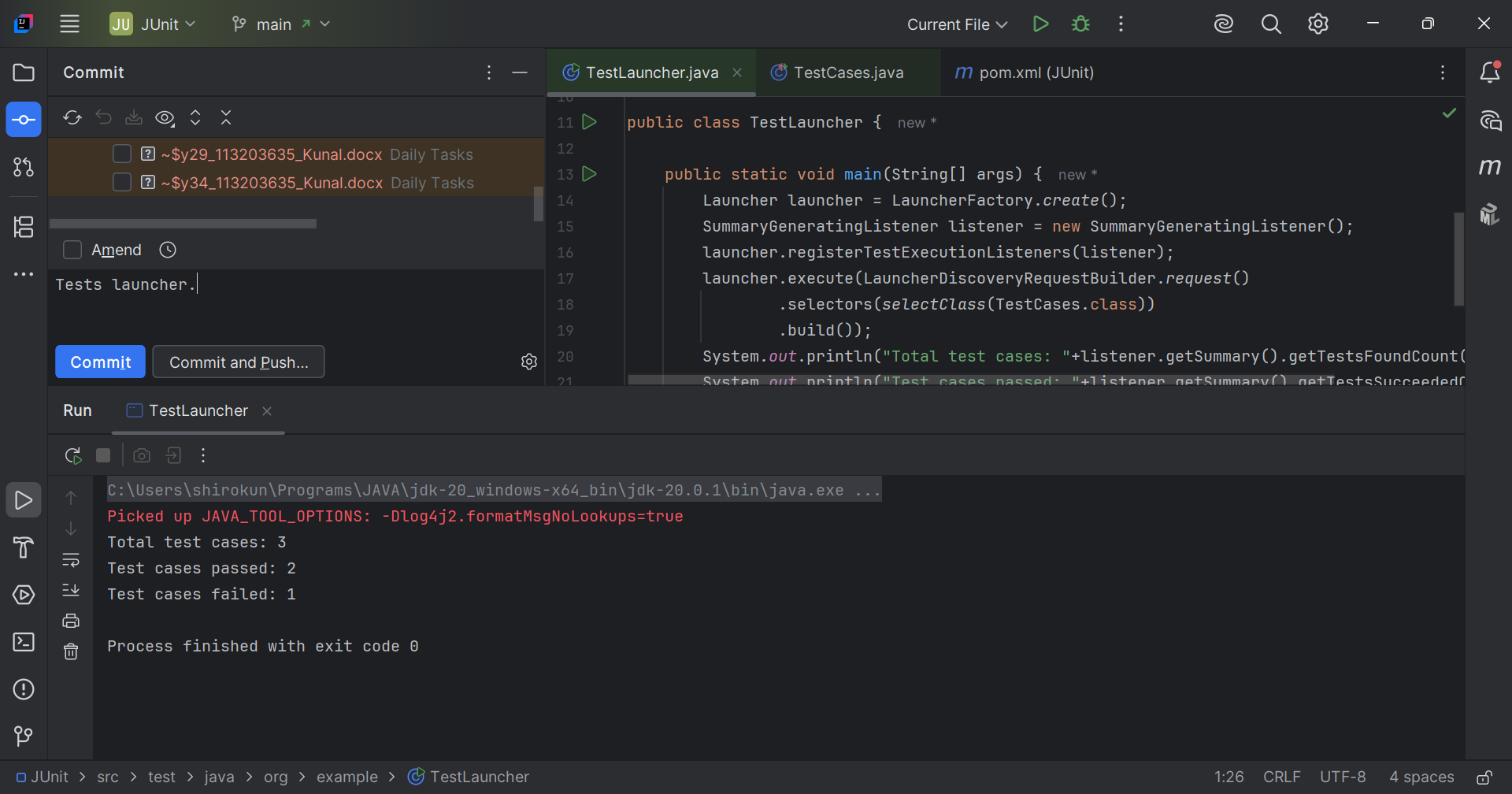


// Task002: Assertions.  
  
package org.example;  
  
import org.junit.jupiter.api.Test;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class TestAssertions {  
 private static final Logger *logger* = LoggerFactory.*getLogger*(TestAssertions.class);  
 @Test  
 void testcase01() {  
 *logger*.info("start testing");  
 String res = testcase02();  
 *logger*.info("testing is done {}", res);  
 }  
 private String testcase02() {  
 *logger*.info("we are in testcase02");  
 try{  
 Thread.*sleep*(1000);  
 }catch(InterruptedException ex) {  
 Thread.*currentThread*().interrupt();  
 *logger*.error("execution got interrupted", ex);  
 }  
 return "successfully.";  
 }  
}

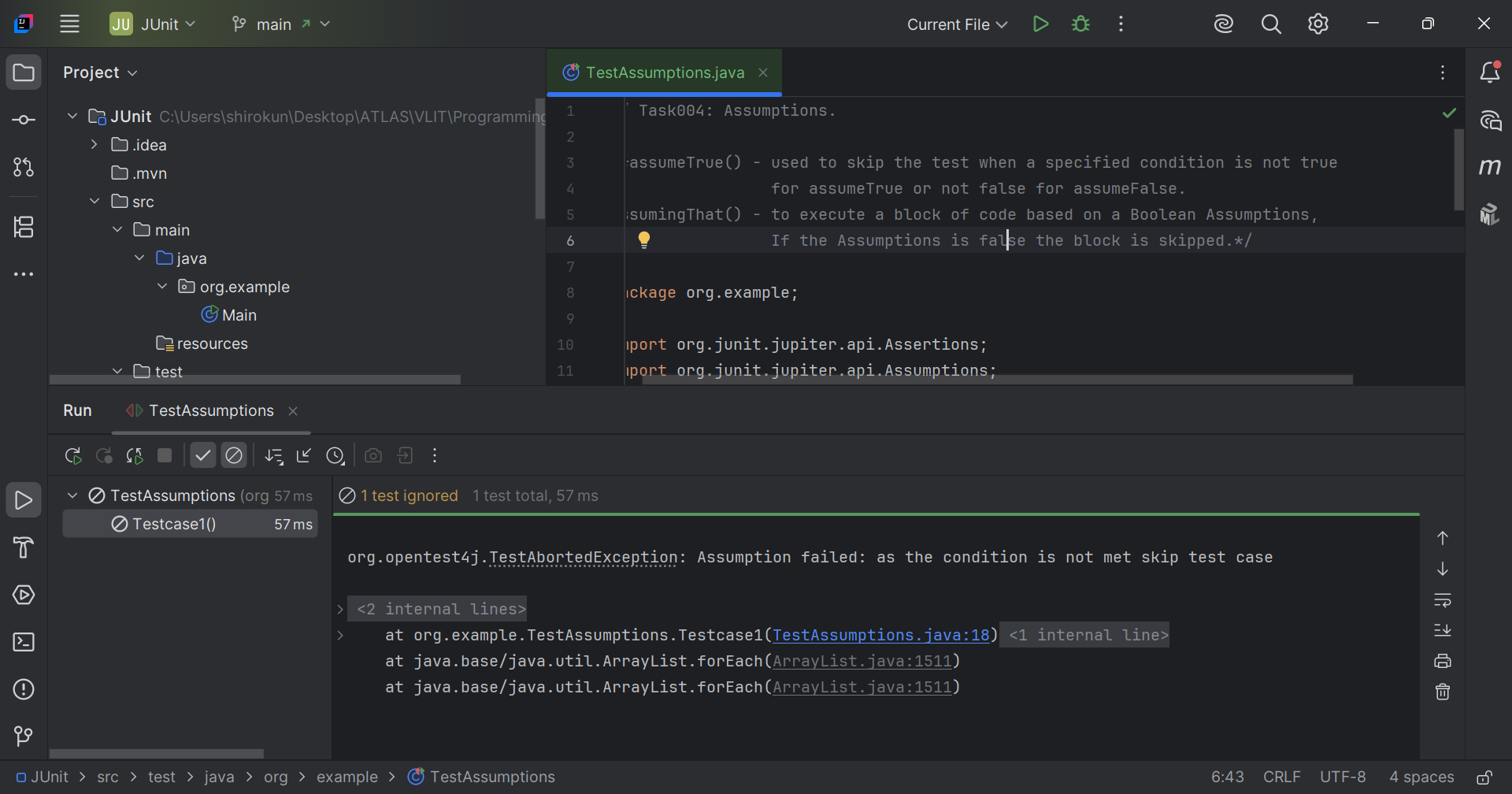


// Task003: Test launcher  
  
package org.example;  
  
import org.junit.platform.launcher.Launcher;  
import org.junit.platform.launcher.core.LauncherDiscoveryRequestBuilder;  
import org.junit.platform.launcher.core.LauncherFactory;  
import org.junit.platform.launcher.listeners.SummaryGeneratingListener;  
import static org.junit.platform.engine.discovery.DiscoverySelectors.*selectClass*;  
  
public class TestLauncher {  
  
 public static void main(String[] args) {  
 Launcher launcher = LauncherFactory.*create*();  
 SummaryGeneratingListener listener = new SummaryGeneratingListener();  
 launcher.registerTestExecutionListeners(listener);  
 launcher.execute(LauncherDiscoveryRequestBuilder.*request*()  
 .selectors(*selectClass*(TestCases.class))  
 .build());  
 System.*out*.println("Total test cases: "+listener.getSummary().getTestsFoundCount());  
 System.*out*.println("Test cases passed: "+listener.getSummary().getTestsSucceededCount());  
 System.*out*.println("Test cases failed: "+listener.getSummary().getTestsFailedCount());  
 }  
}

// Task003: Test cases for launcher.  
  
package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertTrue*;  
import static org.junit.jupiter.api.Assertions.*fail*;  
  
public class TestCases {  
  
 @Test  
 void firstTest() {  
 *assertTrue*(true);  
 }  
  
 @Test  
 void secondTest() {  
 *assertTrue*(true, "This test should pass.");  
 }  
  
 @Test  
 void thirdTest(){  
 *fail*("false");  
 }  
}



// Task004: Assumptions.  
  
/\*assumeTrue() - used to skip the test when a specified condition is not true  
 for assumeTrue or not false for assumeFalse.  
assumingThat() - to execute a block of code based on a Boolean Assumptions,  
 If the Assumptions is false the block is skipped.\*/  
  
package org.example;  
  
import org.junit.jupiter.api.Assertions;  
import org.junit.jupiter.api.Assumptions;  
import org.junit.jupiter.api.Test;  
  
public class TestAssumptions {  
 @Test  
 void Testcase1() {  
 boolean condition = "true".equalsIgnoreCase(System.*getProperty*("true"));  
 int result = testcase2();  
 Assertions.*assertEquals*(20, result, "value need to be 10");  
 Assumptions.*assumeTrue*(condition, "as the condition is not met skip test case");  
 }  
  
 private int testcase2() {  
 return 10;  
 }  
  
}



// Task005: Parameterized tests.  
  
package org.example;  
  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class TestParameterized {  
 @ParameterizedTest  
 @ValueSource(ints = {100, 25, 30, 70, 40})  
 void testSquare(int value) {  
 int result = square(value);  
 *assertEquals*(value \* value, result, " if wrong");  
 }  
  
 private int square(int number) {  
 return number \* number;  
 }  
  
}

