|  |
| --- |
| Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and |
| Teardown Methods in JUnit |

# 1. Arrange-Act-Assert Pattern Example:

import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class CalculatorTest {  
  
 @Test  
 public void testAddition() {  
 // Arrange  
 Calculator calc = new Calculator();  
  
 // Act  
 int result = calc.add(10, 20);  
  
 // Assert  
 assertEquals(30, result);  
 }  
}  
  
class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

# 2. Test Fixtures and Setup/Teardown Methods Example:

import org.junit.jupiter.api.\*;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class UserServiceTest {  
  
 private UserService userService;  
  
 @BeforeEach  
 public void setUp() {  
 userService = new UserService();  
 userService.addUser(new User("john", "pass123"));  
 }  
  
 @AfterEach  
 public void tearDown() {  
 userService.clearUsers();  
 }  
  
 @Test  
 public void testFindUser() {  
 // Arrange is handled in setUp()  
  
 // Act  
 User result = userService.findUser("john");  
  
 // Assert  
 assertNotNull(result);  
 assertEquals("john", result.getUsername());  
 }  
}  
  
class UserService {  
 private Map<String, User> users = new HashMap<>();  
  
 public void addUser(User user) {  
 users.put(user.getUsername(), user);  
 }  
  
 public User findUser(String username) {  
 return users.get(username);  
 }  
  
 public void clearUsers() {  
 users.clear();  
 }  
}  
  
class User {  
 private String username;  
 private String password;  
  
 public User(String username, String password) {  
 this.username = username;  
 this.password = password;  
 }  
  
 public String getUsername() {  
 return username;  
 }  
}

# Expected Output:

All tests should pass.  
  
Output (using a JUnit runner, like in IntelliJ or Maven):  
--------------------------------------------------------  
CalculatorTest:  
✔ testAddition() PASSED  
  
UserServiceTest:  
✔ testFindUser() PASSED