Week 2 Mandatory Hands-on Exercise Answers

# TDD using JUnit5 and Mockito - Exercise 1: Mocking and Stubbing

**CODE:**

import java.util.\*;

// Interface like a repository

interface UserRepository {

String getUserEmailById(int userId);

}

// Simulated service that depends on UserRepository

class UserService {

private UserRepository userRepository;

public UserService(UserRepository repo) {

this.userRepository = repo;

}

public String fetchUserEmail(int id) {

return userRepository.getUserEmailById(id);

}

}

// Manual "mock" class to simulate behavior

class FakeUserRepository implements UserRepository {

@Override

public String getUserEmailById(int userId) {

// Stubbed response

if (userId == 1) {

return "test@example.com";

} else {

return "unknown@example.com";

}

}

}

public class Main {

public static void main(String[] args) {

// Arrange: use fake repository instead of real database

UserRepository mockRepo = new FakeUserRepository();

UserService userService = new UserService(mockRepo);

// Act

String email = userService.fetchUserEmail(1);

// Assert (simulated)

if ("test@example.com".equals(email)) {

System.out.println("testestFetchUserEmail PASSED");

} else {

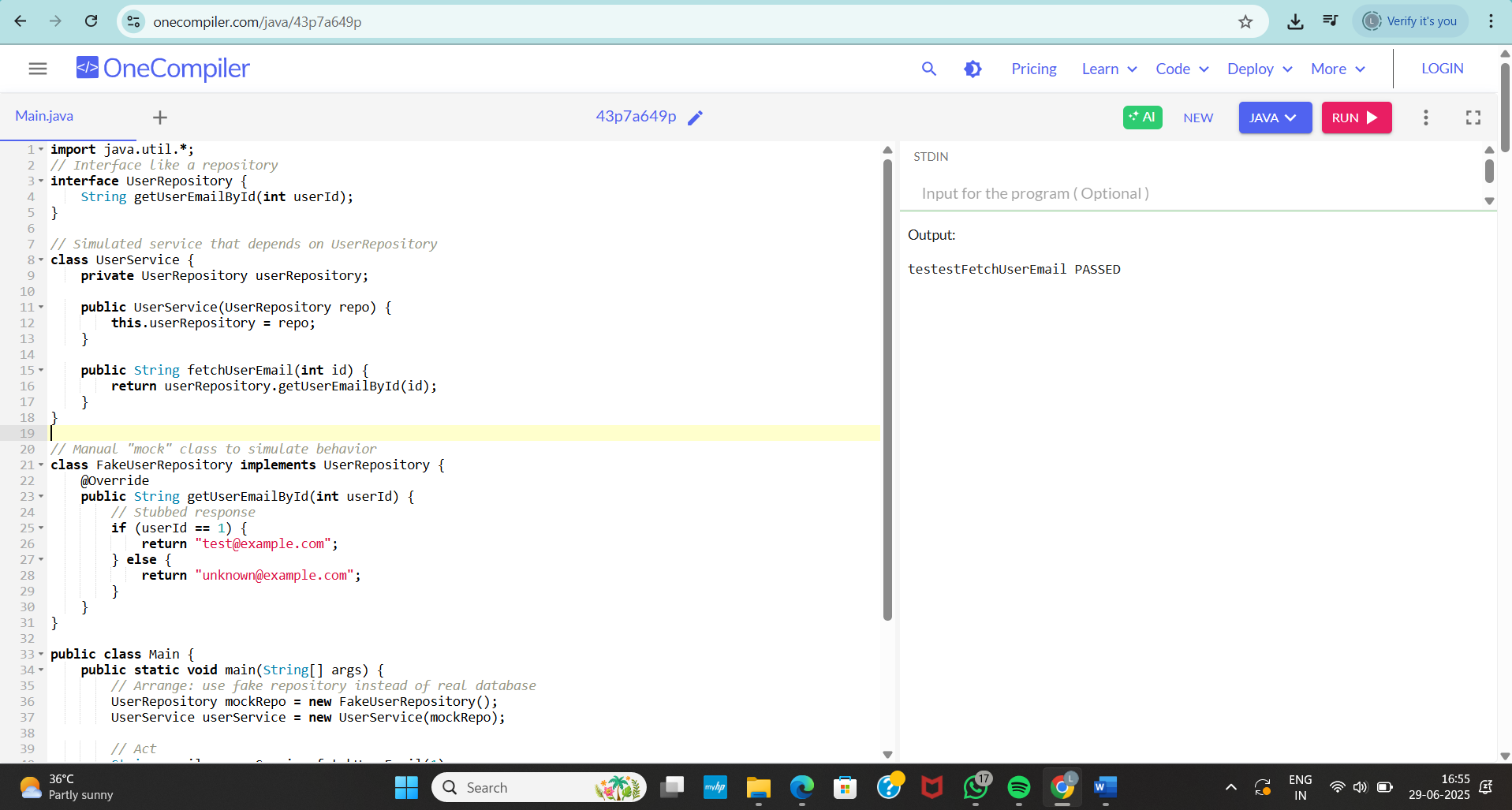
System.out.println("testFetchUserEmail FAILED: Expected test@example.com but got " + email);

}

}

}

OUTPUT:



# TDD using JUnit5 and Mockito - Exercise 2: Verifying Interactions

CODE:

interface EmailSender {

void sendEmail(String to, String message);

}

// Fake class with flag to verify interaction

class FakeEmailSender implements EmailSender {

boolean wasCalled = false;

String lastEmailSent = "";

String lastMessageSent = "";

public void sendEmail(String to, String message) {

wasCalled = true;

lastEmailSent = to;

lastMessageSent = message;

System.out.println("Email sent to " + to + ": " + message);

}

}

class NotificationService {

private EmailSender emailSender;

public NotificationService(EmailSender sender) {

this.emailSender = sender;

}

public void notifyUser(String userEmail) {

emailSender.sendEmail(userEmail, "Welcome!");

}

}

public class Main {

public static void main(String[] args) {

// Arrange

FakeEmailSender mockSender = new FakeEmailSender();

NotificationService service = new NotificationService(mockSender);

// Act

service.notifyUser("user@example.com");

// Assert - manual verification

if (mockSender.wasCalled &&

"user@example.com".equals(mockSender.lastEmailSent) &&

"Welcome!".equals(mockSender.lastMessageSent)) {

System.out.println("testNotifyUser PASSED");

} else {

System.out.println("testNotifyUser FAILED");

}

}

}

OUTPUT:

