**Exercise-1**

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

import org.junit.jupiter.api.Test;

public class CalculatorServiceTest {

@Test

public void testAdd() {

CalculatorService calculator = new CalculatorService();

int result = calculator.add(2, 3);

assertEquals(5, result);

}

}

**Exercise-2**

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

import static org.mockito.Mockito.\*;

import java.util.Optional;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

@ExtendWith(MockitoExtension.class)

public class UserServiceTest {

@Mock

private UserRepository;

@InjectMocks

private UserService userService;

@Test

public void testGetUserById() {

User mockUser = new User();

mockUser.setId(1L);

mockUser.setName("John Doe");

when(userRepository.findById(1L)).thenReturn(Optional.of(mockUser));

User result = userService.getUserById(1L);

assertEquals("John Doe", result.getName());

}

@Test

public void testGetUserByIdNotFound() {

when(userRepository.findById(1L)).thenReturn(Optional.empty());

User result = userService.getUserById(1L);

assertNull(result);

}

}

**Exercise-3**

import static org.mockito.Mockito.\*;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;

import org.springframework.boot.test.mock.mockito.MockBean;

import org.springframework.test.web.servlet.MockMvc;

@WebMvcTest(UserController.class)

public class UserControllerTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserService userService;

@Test

public void testGetUser() throws Exception {

User mockUser = new User();

mockUser.setId(1L);

mockUser.setName("John Doe");

when(userService.getUserById(1L)).thenReturn(mockUser);

mockMvc.perform(get("/users/1"))

.andExpect(status().isOk())

.andExpect(jsonPath("$.name").value("John Doe"));

}

}

**Exercise-4**

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

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.web.client.TestRestTemplate;

import org.springframework.boot.test.web.server.LocalServerPort;

import org.springframework.http.ResponseEntity;

@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM\_PORT)

public class UserIntegrationTest {

@LocalServerPort

private int port;

@Autowired

private TestRestTemplate restTemplate;

@Test

public void testGetUser() {

ResponseEntity<User> response = restTemplate.getForEntity(

"http://localhost:" + port + "/users/1", User.class);

assertEquals(200, response.getStatusCodeValue());

assertNotNull(response.getBody());

}

}

**Exercise-5**

import static org.mockito.ArgumentMatchers.\*;

import static org.mockito.Mockito.\*;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;

import org.springframework.boot.test.mock.mockito.MockBean;

import org.springframework.http.MediaType;

import org.springframework.test.web.servlet.MockMvc;

import com.fasterxml.jackson.databind.ObjectMapper;

@WebMvcTest(UserController.class)

public class UserControllerPostTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserService userService;

@Autowired

private ObjectMapper objectMapper;

@Test

public void testCreateUser() throws Exception {

User newUser = new User();

newUser.setName("Jane Doe");

User savedUser = new User();

savedUser.setId(1L);

savedUser.setName("Jane Doe");

when(userService.saveUser(any(User.class))).thenReturn(savedUser);

mockMvc.perform(post("/users")

.contentType(MediaType.APPLICATION\_JSON)

.content(objectMapper.writeValueAsString(newUser)))

.andExpect(status().isOk())

.andExpect(jsonPath("$.id").value(1L))

.andExpect(jsonPath("$.name").value("Jane Doe"));

}

}

**Exercise-6**

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

import static org.mockito.Mockito.\*;

import java.util.NoSuchElementException;

import java.util.Optional;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

@ExtendWith(MockitoExtension.class)

public class UserServiceExceptionTest {

@Mock

private UserRepository userRepository;

@InjectMocks

private UserService userService;

@Test

public void testGetUserByIdThrowsException() {

when(userRepository.findById(1L)).thenReturn(Optional.empty());

assertThrows(NoSuchElementException.class, () -> {

userService.getUserById(1L);

});

}

}

**Exercise-7**

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

import java.util.Arrays;

import java.util.List;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTest;

@DataJpaTest

public class UserRepositoryTest {

@Autowired

private UserRepository userRepository;

@Test

public void testFindByName() {

User user1 = new User();

user1.setName("John");

userRepository.save(user1);

User user2 = new User();

user2.setName("John");

userRepository.save(user2);

List<User> users = userRepository.findByName("John");

assertEquals(2, users.size());

}

}

**Exercise-8**

import static org.mockito.Mockito.\*;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;

import org.springframework.boot.test.mock.mockito.MockBean;

import org.springframework.test.web.servlet.MockMvc;

@WebMvcTest(UserController.class)

public class UserControllerExceptionTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserService userService;

@Test

public void testGetUserNotFound() throws Exception {

when(userService.getUserById(1L)).thenThrow(new NoSuchElementException());

mockMvc.perform(get("/users/1"))

.andExpect(status().isNotFound())

.andExpect(content().string("User not found"));

}

}

**Exercise-9**

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

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.CsvSource;

public class CalculatorParameterizedTest {

private CalculatorService calculator = new CalculatorService();

@ParameterizedTest

@CsvSource({

"1, 2, 3",

"5, 5, 10",

"10, -5, 5",

"0, 0, 0"

})

public void testAdd(int a, int b, int expected) {

assertEquals(expected, calculator.add(a, b));

}

}