|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №2**

по дисциплине «Тестирование и верификация программного обеспечения»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-01-19 | Субботин Е.В.  Кузин Д.О.  Залетин Н.А.  Леутина Т.А. |
| **Проверил:** | ассистент Трушин С. М. |

2021 г.

**Содержание**

[Состав команды 3](#_Toc84325357)

[1 Описание ПО 4](#_Toc84325358)

[1.1 Документация своего проекта 4](#_Toc84325359)

[1.2 Документация проекта партнёра (Кузин Данил) 4](#_Toc84325360)

[1.3 Выбор ПО для тестирования 4](#_Toc84325361)

[2 Модульное тестирование 5](#_Toc84325362)

[2.1 Выбор классов для тестирования 5](#_Toc84325364)

[2.2 Тестирование класса CartService 5](#_Toc84325365)

[2.3 Тестирование класса OrderService 6](#_Toc84325366)

[2.4 Тестирование класса UserService 7](#_Toc84325367)

[2.5 Тестирование класса ProductService 9](#_Toc84325368)

[3 Оценка качества чужого кода, соответствия документации и его надежности 11](#_Toc84325369)

Состав команды

* Субботин Евгений
* Кузин Данил
* Залетин Никита
* Леутина Таисия

# Описание ПО

## Документация своего проекта

Ссылка на ЯндексДиск с документацией:

https://drive.google.com/drive/folders/1cSNr86fbitNDk2k0\_Slt9Ro3RHEeQq6c?usp=sharing

Ссылка на гит-репозиторий с проектом:

https://github.com/Evgeniy638/catlog

## Документация проекта партнёра (Кузин Данил)

Ссылка на гит-репозиторий с проектом:

https://github.com/Shambonik/java\_kr

Ссылка на облако с документацией:

https://disk.yandex.ru/d/g73OkwhQzzEuVA

## Выбор ПО для тестирования

В качестве ПО для тестирования был выбран JUNIT 5, для создания «объектов-заглушек» использовался Mockito, также применялись библиотеки проверки утверждений, такие как AssertJ. Для генерации документации использовался Javadoc.

# Модульное тестирование



## Выбор классов для тестирования

Для тестирования были выбраны классы CartService, OrderService, ProductService и UserService, представленные на Рисунке 2.1.

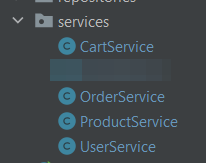


Рисунок 2.1 – Классы для тестирования

## Тестирование класса CartService

Исходный код класса CartServiceTest.java (Листинг 2.1)

Листинг 2.1 - Файл CartServiceTest.java

@ExtendWith(MockitoExtension.class)  
public class CartServiceTest {  
  
 @Mock  
 private ProductService productService;  
  
 @Test  
 public void getProducts() {  
 String meatCart = "2.1.4.3";  
 Mockito.*when*(productService.getProductById(*anyLong*())).thenReturn(new Product());  
 CartService cartService = new CartService(productService);  
 Assertions.*assertEquals*(4, cartService.getProducts(meatCart).getProductCounts().size());  
 }  
}

Был проведен один тест для метода getProducts. Тест пройден успешно и его результат приведен на Рисунке 2.2.

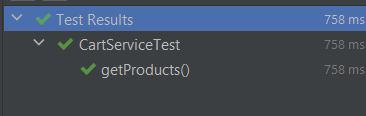


Рисунок 2.2 – Тестирование класса CartService

## Тестирование класса OrderService

Исходный код класса OrderServiceTest.java (Листинг 2.2)

Листинг 2.2 - Файл OrderServiceTest.java

@ExtendWith(MockitoExtension.class)  
public class OrderServiceTest {  
  
 @Mock  
 private ProductService productService;  
 @Mock  
 private UserService userService;  
 @Mock  
 private CartService cartService;  
 @Mock  
 private OrderRepo orderRepo;  
 @Mock  
 private ProductCountRepo productCountRepo;  
 @Mock  
 private EmailService emailService;  
  
  
 @Captor  
 ArgumentCaptor<Order> captor;  
  
 @Test  
 public void cancelOrder(){  
 Order order = new Order();  
 order.setId(1L);  
 order.setStatus(Collections.*singleton*(Order\_Status.*PROCESSING*));  
 Mockito.*when*(orderRepo.findOrderById(1L)).thenReturn(order);  
 OrderService orderService = new OrderService(productService,  
 userService, cartService, orderRepo, productCountRepo, emailService);  
 orderService.cancelOrder(order.getId());  
 Mockito.*verify*(orderRepo).save(captor.capture());  
 Order captured = captor.getValue();  
 Assertions.*assertEquals*(Collections.*singleton*(Order\_Status.*CANCELED*),  
 captured.getStatus());  
 }  
  
}

Был проведен 1 тест для метода cancelOrder. Тест пройден успешно. Его результат приведен на Рисунке 2.3.

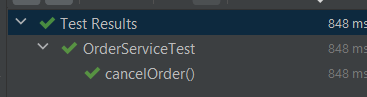


Рисунок 2.3 – Тестирование класса OrderService

## Тестирование класса UserService

Исходный код класса UserServiceTest.java (Листинг 2.3)

Листинг 2.3 - Файл UserServiceTest.java

@ExtendWith(MockitoExtension.class)  
public class UserServiceTest {  
  
 @Mock  
 private UserRepo userRepo;  
  
 @Mock  
 private PasswordEncoder passwordEncoder;  
  
 @Mock  
 private SessionRegistry sessionRegistry;  
  
 @Captor  
 ArgumentCaptor<User> captor;  
  
 @Test  
 public void getUsers() {  
 User user1 = new User();  
 user1.setName("Name1");  
 User user2 = new User();  
 user2.setName("Name2");  
 Mockito.*when*(userRepo.findAll()).thenReturn(List.*of*(user1, user2));  
 UserService userService = new UserService(userRepo, passwordEncoder, sessionRegistry);  
 Assertions.*assertEquals*(2, userService.getUsers().size());  
 Assertions.*assertEquals*("Name1", userService.getUsers().get(0).getName());  
 }  
  
 @Test  
 public void addUser() {  
 User user1 = new User();  
 user1.setUsername("Name1");  
 Mockito.*when*(userRepo.findByUsername(user1.getUsername())).thenReturn(null);  
 Mockito.*when*(passwordEncoder.encode(user1.getPassword())).thenReturn("1");  
 UserService userService = new UserService(userRepo, passwordEncoder, sessionRegistry);  
 userService.addUser(user1);  
 Mockito.*verify*(userRepo).save(captor.capture());  
 User captured = captor.getValue();  
 Assertions.*assertEquals*("Name1", captured.getUsername());  
 }  
  
 @Test  
 public void changeUserRole(){  
 User user1 = new User();  
 user1.setRoles(Set.*of*(Role.*USER*));  
 User currentUser = new User();  
 currentUser.setId(1);  
 ChangeRole changeRole = new ChangeRole();  
 changeRole.setRole(Role.*ADMIN*);  
 Mockito.*when*(userRepo.findById(2)).thenReturn(user1);  
 UserService userService = new UserService(userRepo, passwordEncoder, sessionRegistry);  
 userService.changeUserRole(currentUser, 2, changeRole);  
 Mockito.*verify*(userRepo).save(captor.capture());  
 User captured = captor.getValue();  
 Assertions.*assertEquals*(Set.*of*(Role.*ADMIN*), captured.getRoles());  
 }  
}

Было проведено 3 теста для методов getUsers, addUser, changeUserRole. Тесты пройдены успешно. Их результаты приведены на Рисунке 2.4

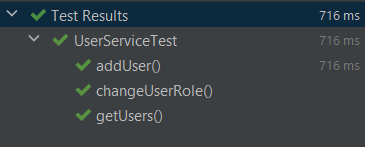


Рисунок 2.4 – Тестирование класса UserService

## Тестирование класса ProductService

Исходный код класса ProductServiceTest.java (Листинг 2.4)

Листинг 2.4 - Файл ProductServiceTest.java

@ExtendWith(MockitoExtension.class)  
public class ProductServiceTest {  
  
 @Mock  
 private ProductRepo productRepo;  
  
 @Captor  
 ArgumentCaptor<Product> captor;  
  
 @Test  
 public void getCategoryFilters(){  
 ProductService productService = new ProductService(productRepo);  
 Assertions.*assertEquals*(Arrays.*stream*(Product\_Category.*getCategories*())  
 .findFirst()  
 .get().name(),  
 productService.getCategoryFilters().getList().get(0).getName());  
 }  
  
 @Test  
 public void changeProductActive(){  
 Product product = new Product();  
 product.setId(1L);  
 product.setActive(true);  
 Mockito.*when*(productRepo.findProductById(1L)).thenReturn(product);  
 ProductService productService = new ProductService(productRepo);  
 productService.changeProductActive(product.getId());  
 Mockito.*verify*(productRepo).save(captor.capture());  
 Product captured = captor.getValue();  
 Assertions.*assertFalse*(captured.isActive());  
 }  
  
 @Test  
 public void reduceCount(){  
 Product product = new Product();  
 product.setCount(10);  
 ProductCount productCount = new ProductCount();  
 productCount.setProduct(product);  
 productCount.setCount(3);  
 ProductService productService = new ProductService(productRepo);  
 productService.reduceCount(productCount);  
 Mockito.*verify*(productRepo).save(captor.capture());  
 Product captured = captor.getValue();  
 Assertions.*assertEquals*(7, captured.getCount());  
 }  
}

Было проведено 3 теста для методов getCategoryFilters, changeProductActive, reduceCount. Тесты пройдены успешно. Их результаты приведены на Рисунке 2.5

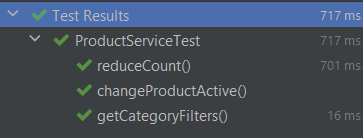


Рисунок 2.5 – Тестирование класса ProductService

# Оценка качества чужого кода, соответствия документации и его надежности

Код хорошо читается, в ходе тестирования не были выявлены несоответствия кода документации, а также код прошёл все необходимые тесты. Оценка: 5 баллов.