Conteúdo do curso imersivo

TESTES UNITÁRIOS COM JUNIT

Atualizado em 31/03/2023

Conteúdo programático

1. Introdução
1.1. Introdução ao treinamento
1.2. Como usar o suporte
1.3. Pirâmide de testes
1.4. A importância dos testes unitários
1.5. O princípio FIRST
1.6. Introdução ao Maven
1.7. Instalando a IntelliJ no Windows
1.8. Instalando a IntelliJ no Linux
1.9. Instalando o JDK do Java no Windows
1.10. Instalando o JDK do Java no Linux
1.11. Instalando o Maven no Windows
1.12. Instalando o Maven no Linux
1.13. Baixando o projeto inicial do Github
1.14. Comandos básicos do Maven
1.15. Apresentação do projeto do curso

2. Explorando o JUnit
2.1. Introdução ao JUnit
2.2. Escrevendo o seu primeiro teste unitário
2.3. Explorando as asserções
2.4. Asserções de Exceptions
2.5. Asserções em listas
2.6. Asserções de Timeout
2.7. Asserções agrupadas com AssertAll
2.8. Executando do teste com Debug
2.9. Desabilitando testes unitários
2.10. Ignorando execução dos testes condicionalmente com Assumptions
2.11. Executando testes via Maven
2.12. Desafio - Escrevendo testes faltantes do SaudacaoUtil
2.13. Refatorando classe SaudacaoUtil
2.14. Desafio - Implementando conta bancária com testes unitários

3. Organizando testes unitários
3.1. Organizando testes com o padrão Triple A
3.2. Alterando nome de exibição dos testes com @DisplayName
3.3. Formatando nomes de testes com @DisplayNameGeneration
3.4. Aplicando a nomenclatura do BDD para nomear métodos de teste
3.5. Organizando classe de testes com sub-classes e @Nested

3.6. Preparando o cenário de testes com @BeforeEach e @BeforeAll
3.7. Um teste deve ter uma única asserção?
3.8. Combinando @Nested e @BeforeEach com a nomenclatura do BDD
3.9. Desafio - Implemente a lógica e testes de um carrinho de compras

4. Stub, Mock e Spy
4.1. Implementações falsas com Stub
4.2. Introdução ao Mock
4.3. Simulando classes com Mockito
4.4. Mock com annotations
4.5. Alterando estado dos parâmetros passados no mock
4.6. Parâmetros dinâmicos
4.7. Verificando chamada de métodos com mock usando Mockito verify
4.8. Forçando uma Exception com mock
4.9. Capturando parâmetros enviados aos mocks com Argument Captor
4.10. Espionando um objeto real com Mockito
4.11. Alterando retorno de um mock após chamadas consecutivas
4.12. Verificando ordem de chamada de métodos
4.13. Usando mock em métodos estáticos
4.14. Entendendo problema de mocks não utilizados
4.15. Implementando testes no CadastroEditor no método de edição
4.16. Desafio - Criar testes do CadastroPost

5. Trabalhando com dados fictícios
5.1. Eliminando repetições de código com o Design Pattern Object Mother
5.2. Combinando Design Pattern Object Mother com Builder
5.3. Desafio - Refatorar testes antigos

6. Testes parametrizados
6.1. Testes parametrizados
6.2. Carregados dados de um CSV em um método de teste
6.3. Carregando valores de um Enum

7. Análise de cobertura de testes
7.1. Introdução a análise de cobertura de testes
7.2. Descobrindo cenário de testes não cobertos com ajuda da IntelliJ
7.3. Lacunas da análise de cobertura automática
7.4. Implementando o JaCoCo - Java Code Coverage Library

8. Asserções fluentes com AssertJ
8.1. Introdução ao AssertJ?
8.2. Asserções básicas
8.3. Asserções de Exceptions

8.4. Asserções com mensagens descritivas
8.5. Asserções customizadas
8.6. Múltiplas asserções em listas