

Testes White-box: Cobertura de Decisões

A **cobertura de decisões** é uma das **técnicas de teste white-box** mais populares. Esta técnica garante que cada decisão associada a uma **estrutura condicional** (ex: *if*, *switch*, *while*, *for*) é coberta por pelo menos um teste, para todo o espaço de resultados (*verdadeiro/falso*).

Se criarmos um **cenário de teste** com o valor **a=2** para o método do exemplo seguinte, conseguimos uma **taxa de cobertura de 50%**. Para um outro cenário onde o **a=6**, a taxa de cobertura é igual. **Combinando ambos os cenários**, conseguimos atingir **100% de cobertura**.

```
void demo(int a) {  
    if(a > 5) a = a * 3;  
  
    System.out.println(a);  
}
```

Exercício

Considere a **implementação do padrão Decorator**.

- Crie os testes para validar a condição do *if* do método *auth* da classe *Auth*.

```
public void auth(String username, String password) throws AuthException {  
    if(! username.equals("admin") || !password.equals("admin"))  
        throw new AuthException();  
}
```

- Crie os testes para validar a condição implícita do tratamento da exceção do método *auth* na classe *CommonWordsValidator*.

```
public void auth(String username, String password) throws AuthException, IOException {  
    try {  
        String dicWord = getHttpRequest(password);  
    } catch (IOException e) {  
        throw new AuthException();  
    }  
  
    super.auth(username, password);  
}
```

Para analisar o output da aplicação é necessário redirecioná-lo para uma variável.

```
@BeforeAll  
public static void setUpStreams() {  
    System.setOut(new PrintStream(outContent));  
}
```

Assim, nos testes é possível analisar o que está a ser enviado para o ecrã.

```
assertTrue(outContent.toString().contains("Hello World"));
```

No final é necessário repor a configuração.

```
@AfterAll
public static void tearDownAfterClass() {
    System.setOut(System.out);
    End e = new End();
}
```

Algumas questões para **refletir**:

- Todos os ramos de execução foram testados exaustivamente?

- De que forma as falhas na cobertura das decisões podem estar relacionadas com as limitações da aplicação desta técnica em testes white-box?

Submeta a classe com os testes **TestDecorator** para **validação**.

Specification

[JavaDocs](#)

Deadline

2020-03-30 23:55

Submit for Testing

Submit for Evaluation

Submission Done With Success !!!

```

      |
      |└─ JUnit Jupiter ✓
      |  └─ TestDecorator ✓
      |     |└─ auth_InvalidCommonWord_ThrowsException() ✓
      |     |  |└─ auth_UsernameRightAndPasswordRight_NoException() ✓
      |     |  |└─ auth_UsernameWrongAndPasswordWrong_ThrowsException() ✓
      |     |  |  |└─ auth_ValidCommonWord_NoException() ✓
      |     |  |  |└─ auth_UsernameRightAndPasswordWrong_ThrowsException() ✓
      |     |  |  |└─ auth_UsernameWrongAndPasswordRight_ThrowsException() ✓
      |     |  └─ JUnit Vintage ✓
```

Test run finished after 1508 ms

```
[ 3 containers found ]
[ 0 containers skipped ]
[ 3 containers started ]
[ 0 containers aborted ]
[ 3 containers successful ]
[ 0 containers failed ]
[ 6 tests found ]
[ 0 tests skipped ]
[ 6 tests started ]
[ 0 tests aborted ]
[ 6 tests successful ]
[ 0 tests failed ]
```