UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN

Instituto Metrópole Digital

IMD0040 - Linguagem de Programação 2

Aula Revisão

1. Analisar o projeto Leilão.
    a. Quantais as classes, os atributos e métodos. Analisando estas classes você consegue abstrair suas características e comportamentos? Comente no código os métodos.
2. Primeira funcionalidade a implementar: dado um leilão precisamos saber qual o maior e o menor lance que foi dado, para isto temos criar uma classe chamada avaliador:
    i. Esta classe contém um método chamado avalia que identifica qual o maior e o menor lance de um determinado leilão. segue o método.

```
10
11    public void avalia(Leilao leilao){
12
13        for (Lance lance : leilao.getLances()){
14            if(lance.getValor() > maiorDeTodos){
15                maiorDeTodos = lance.getValor();
16            }else if (lance.getValor() < menorDeTodos){
17                menorDeTodos = lance.getValor();
18            }
19        }
20    }
```

    ii. Analisando esse método podemos perceber que a classe Avaliador tem dois atributos, você consegue identificar quais são?
    iii. Agora crie os dois atributos privados e do tipo double. Inicialize os atributos de forma que eles garantam sempre o menor ou o maior valor, consulte a classe Double .
3. Vamos criar uma classe para testar as funcionalidades do avaliador (Ainda não com JUnit):
    i. Vamos criar 3 Usuários (Usando o construtor),
    ii. Crie um leilão (Usando o construtor)
    iii. Proponha lances nesse leilão, os lances devem seguir essa ordem: 300, 400 e 250.
    iv. Crie o avaliador e avalie o leilão, imprimindo o maior e o menor valor.
    v. Execute e veja se o valor está correto.
4. Será que nosso programa está sem erros?? para verificar isso vamos alterar as ordens dos lances para: 250, 300 e 400. Encontramos um erro, OK? isso ocorreu por que não testamos nosso software corretamente.
5. Vamos criar a classe teste:
    i. O método de teste deve ser public void e não receber argumentos, bem como ser anotado por @Test, importamos o @Teste de import org.junit.Test.
    ii. Vamos utilizar o método assertEquals da classe Assert (Assert.assertEquals), que recebe dois parâmetros: o primeiro o valor

esperado e o segundo é o valor calculado, neste nosso caso o método de retorno do Maior ou Menor Lance.

iii. Qual o resultado do teste? Qual a falha?

iv. Agora é hore de corrigir o problema.

v. Agora crie um outro teste para a situação que temos apenas um lance, o maior e o menor valor devem ser os mesmos.

6. Agora o nosso programa será alterado, precisamos saber quais sãos os 3 maiores lances. Altere a implementação da Classe Avaliador pelo conteúdo do arquivo novo_avaliador.txt. Estude essa nova classe e tente compreender as alterações.
7. Agora crie um teste para a nova funcionalidade 3 maiores, segue o teste:

```
@Test
public void testTresMaioresLances(){

    Usuario joao = new Usuario("João");
    Usuario maria = new Usuario("Maria");

    Leilao leilao = new Leilao("Playstatio 3 Novo");

    leilao.propoe(new Lance(joao, 100.0));
    leilao.propoe(new Lance(maria, 200.0));
    leilao.propoe(new Lance(joao, 300.0));
    leilao.propoe(new Lance(maria, 400.0));

    Avaliador leiloeiro = new Avaliador();

    leiloeiro.avalia(leilao);
    List<Lance> maiores =  leiloeiro.getTresMaiores();
    assertEquals(3, maiores.size());
    assertEquals(400.0, maiores.get(0).getValor(),0.00001);
    assertEquals(300.0, maiores.get(1).getValor(),0.00001);
    assertEquals(200.0, maiores.get(2).getValor(),0.00001);

}
```

8. Após executar os testes com o JUnit o que ocorreu? o testTresMaioresLances passou? e os testes anteriores?
9. Agora é hore de corrigir o problema.