



Atualize sua classe `Avaliador` com o seguinte código:

```
public class Avaliador {

    private double maiorDeTodos = Double.NEGATIVE_INFINITY;
    private double menorDeTodos = Double.POSITIVE_INFINITY;
    private List<Lance> maiores;

    public void avalia(Leilao leilao) {
        for(Lance lance : leilao.getLances()) {
            if(lance.getValor() > maiorDeTodos) maiorDeTodos = lance.getValor();
            if (lance.getValor() < menorDeTodos) menorDeTodos = lance.getValor();
        }

        pegaOsMaioresNo(leilao);
    }

    private void pegaOsMaioresNo(Leilao leilao) {
        maiores = new ArrayList<Lance>(leilao.getLances());
        Collections.sort(maiores, new Comparator<Lance>() {
            public int compare(Lance o1, Lance o2) {
                if(o1.getValor() < o2.getValor()) return 1;
                if(o1.getValor() > o2.getValor()) return -1;
                return 0;
            }
        });
        maiores = maiores.subList(0, maiores.size() > 3 ? 3 : maiores.size());
    }

    public List<Lance> getTresMaiores() {
        return this.maiores;
    }

    public double getMaiorLance() {
        return maiorDeTodos;
    }

    public double getMenorLance() {
        return menorDeTodos;
    }
}
```

Veja que agora temos o método que devolve os três maiores lances daquele leilão. Escreva os seguintes testes para esse método:

- Um leilão com 5 lances, deve encontrar os três maiores
- Um leilão com 2 lances, deve devolver apenas os dois lances que encontrou
- Um leilão sem nenhum lance, devolve lista vazia

## Opinião do instrutor

Os testes ficaram:

```
@Test
public void deveEncontrarOsTresMaioresLances() {
    Usuario joao = new Usuario("João");
    Usuario maria = new Usuario("Maria");
    Leilao leilao = new Leilao("Playstation 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, maiores.get(0).getValor(), 0.00001);
    assertEquals(300, maiores.get(1).getValor(), 0.00001);
    assertEquals(200, maiores.get(2).getValor(), 0.00001);
}

@Test
public void deveDevolverTodosLancesCasoNaoHajaNoMinimo3() {
    Usuario joao = new Usuario("João");
    Usuario maria = new Usuario("Maria");
    Leilao leilao = new Leilao("Playstation 3 Novo");

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

    Avaliador leiloeiro = new Avaliador();
    leiloeiro.avalia(leilao);

    List<Lance> maiores = leiloeiro.getTresMaiores();

    assertEquals(2, maiores.size());
    assertEquals(200, maiores.get(0).getValor(), 0.00001);
    assertEquals(100, maiores.get(1).getValor(), 0.00001);
}

@Test
public void deveDevolverListaVaziaCasoNaoHajaLances() {
    Leilao leilao = new Leilao("Playstation 3 Novo");

    Avaliador leiloeiro = new Avaliador();
    leiloeiro.avalia(leilao);

    List<Lance> maiores = leiloeiro.getTresMaiores();

    assertEquals(0, maiores.size());
}
```

Já na classe `Avaliador`, fizemos a seguinte mudança para que os testes passem:

```
maiores = maiores.subList(0, maiores.size()
> 3 ? 3 : maiores.size());
```

