

INF335 – AMBIENTES PARA CONCEPÇÃO DE SOFTWARE

TRABALHO AULA 02 – TESTES UNITÁRIOS COM JUnit

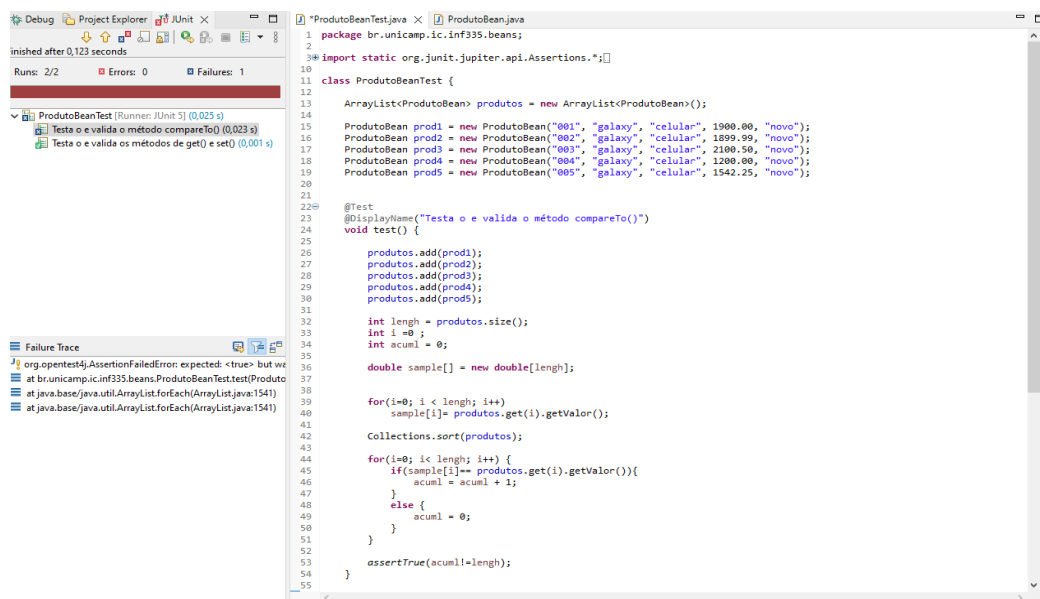
Acesso ao código corrigido: https://github.com/pedrolivr/inf_335_tarefa2

1 TESTE PARA A CLASSE ProdutoBean

A Classe ProdutoBean teve todos seus métodos testados, inclusive os métodos de `get()` e `set()`. A primeira parte do teste consiste em validar o método `compareTo()`, para isso, duas listas estão sendo comparadas, uma lista com os valores não ordenados dos produtos por ordem de criação, e outra lista ordenada depois que o método `Collections.sort()` é executado. Para auxiliar, foi criado um “contador” acumulativo que cresce na medida em que encontra os mesmos valores nas mesmas posições entre as listas.

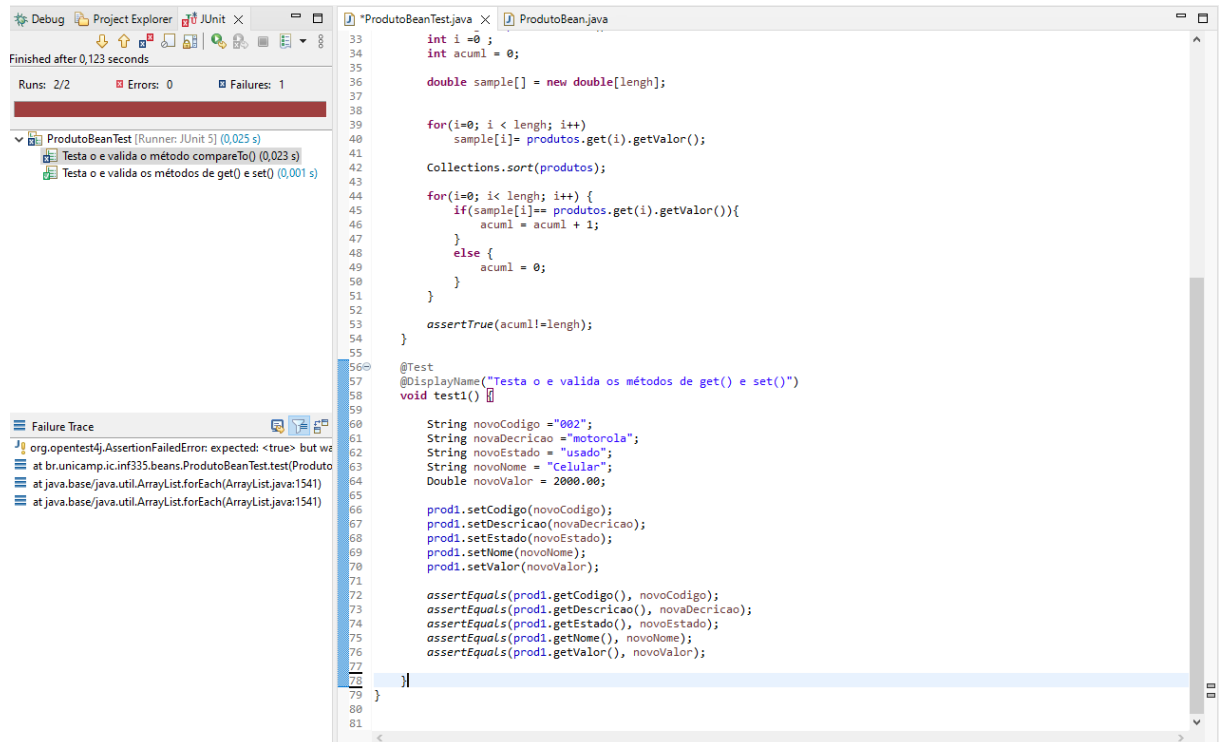
Com isso, se o contador retornar um valor exatamente do tamanho das listas, significa que o método `compareTo()` não está funcionando, uma vez que a lista dispões dos elementos nos mesmo lugares antes e depois da aplicação deste método. O teste quando executado, retorna uma falha

Figura 1



A segunda parte dos testes consiste em checar se os métodos de `get()` e `set()` funcionam de maneira correta. O teste validou os métodos.

Figura 2



1.1 Correções para a Classe ProdutoBean

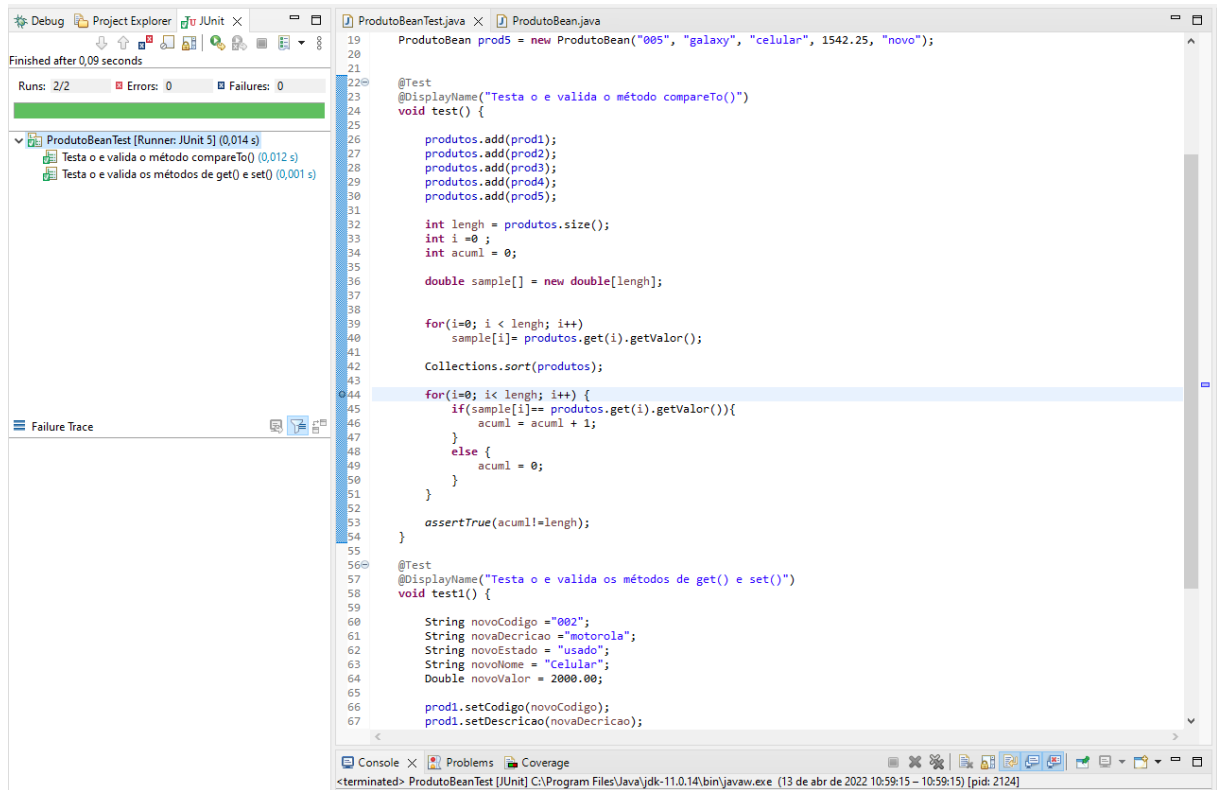
Com a mudança na comparação, é esperado que a lista após o método `Collections.sort()` apresente uma ordenação diferente da primeira lista.

Figura 3

```
@Override
public int compareTo(ProdutoBean p) {
    if (valor < p.getValor()) {
        return 1;
    } else if (valor > p.getValor()) {
        return -1;
    } else
        return 0;
}
```

Após a alteração, ambo os testes tem validações positivas após as correções.

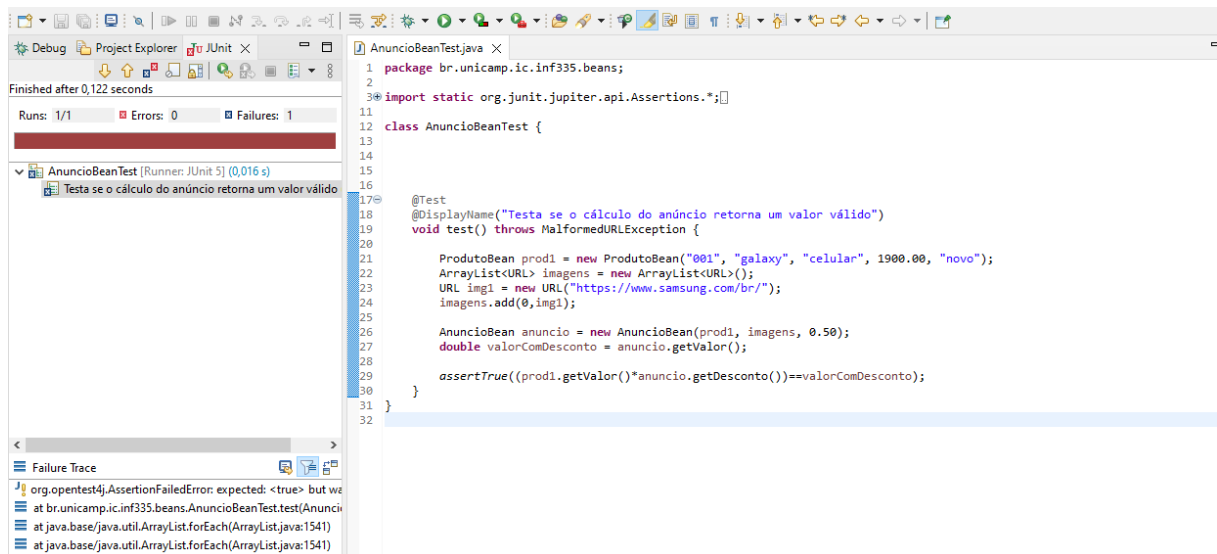
Figura 4



2 TESTE PARA A CLASSE AnuncioBean

A classe AnuncioBeanTest testa o valor do produto com o desconto aplicado, no caso abaixo o teste retorna uma falha, uma vez que o valor do produto com desconto já aplicado, não está correta.

Figura 5



2.1 A alteração sugerida na classe AnuncioBean

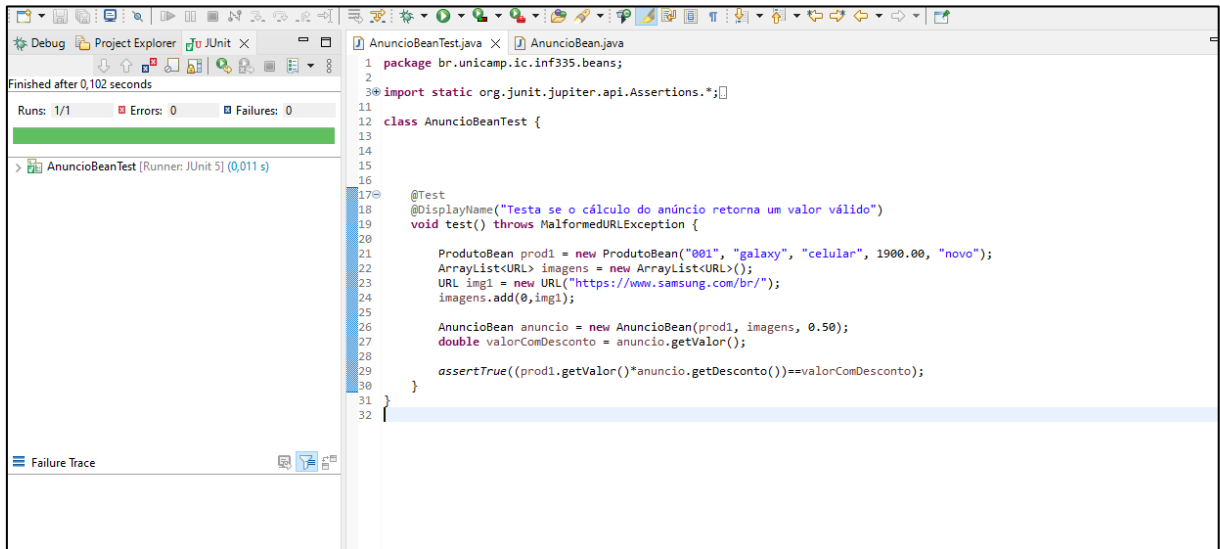
A alteração consiste na modificação da função dentro do método `getValor()`, trocando o sinal de `"/` por `"*"`.

Figura 6

```
public Double getValor() {  
    return produto.getValor()-(produto.getValor()/desconto);  
}
```

Após a correção, o teste valida os métodos.

Figura 7



3 TESTE PARA A CLASSE AnuncianteBean

A classe AnuncianteBean não retornou nenhum erro, a checagem foi feita por meio de debugger e teste unitário. Porém, a classe apenas não apresenta nenhum erro porque os itens apresentados anteriormente sofreram alterações para o correto funcionamento das outras classes. Caso as alterações não tivessem sido feitas, o teste retornaria uma falha.

Figura 8

