

Assertivas:

Usamos as assertivas para verificar o resultado do teste, importamos **Assert** e podemos ter acesso a seus métodos assertivos, se o método passar pela assertiva ele é um teste bem sucedido.

AssertTrue: Recebe um booleano e verifica se a expressão é verdadeira, usamos quando queremos que o resultado do nosso teste seja verdadeiro de acordo com o que passamos como argumento.

AssertFalse: Recebe um booleano mas verifica se a expressão retornada é falsa, claro que ao usar isso esperamos que o resultado do nosso teste seja falso.

AssertEquals: Ele existe para cada tipo de variável e o funcionamento é simples, passamos um valor esperado e verificamos se ele é igual ao valor atual. Quando comparamos valores do tipo float ou double, precisamos passar um terceiro argumento de "delta", ele é uma margem de erro para o comparativo.

O Assert não possui o conceito de unboxing e autoboxing do java, então quando formos comparar um tipo primitivo com um tipo Objeto. Então ou passamos o valor primitivo para objeto ou o objeto para tipo primitivo.

Strings também funcionam no AssertEquals, mas elas tem que ser exatamente iguais incluindo as cases. Por isso talvez valha a pena usar o AssertTrue usando métodos específicos da classe de String.

Assim como as String, a igualdade dos objetos será verificada pelo equals() do próprio objeto. Afinal, quem é melhor para comparar se um objeto é igual ao outro se não o próprio objeto em questão. Implementando o equals() e hashCode() podemos decidir o que é considerado para que um objeto seja igual ao outro.

AssertSame: Podemos usar para checar se um objeto é igual ao outro a nível de instância.

AssertNull: Meio óbvio pelo nome, mas é isso mesmo. Assertiva para se algum objeto é nulo.

Vale ressaltar que todos os asserts também tem seus modos negativos, por exemplo AssertNotNull, AssertNotTrue.