Alexandre Rodrigues 54472

Característica 1: O número é igual a zero

Domínio: n = 0

Regiões: True or False

Teste 1.1:

```
//Teste com input = 0 (excepção deve ser levantada)
@Test(expected = IlegalArgumentException.class)
public void testInverteDobre1() {
   inverteDobre(0d);
}
```

Característica 2: O número é negativo

Domínio: n<0

Regiões: True or False

```
//Teste com input = -2
@Test
public void testInverteDobre3() {
   assertEquals(-0.25d, inverteDobre(-2d), 0.001);
}
```

Característica 3: O número é positivo

Domínio: n>0

Regiões: True or False

```
//Teste com input = 2
@Test
public void testInverteDobre2() {
   assertEquals(0.25d, inverteDobre(2d), 0.001);

}
```

```
/**
* Devolve o inverso do dobro do valor fornecido em parâmetro.
* @param n um valor de tipo double.
* @requires n !=0
*/
public static double inverteDobre(double n)
```

Característica 1: Lista está vazia

Domínio: lista = []

Regiões: True or False

Teste 1.1:

```
@Test(expected = IlegalArgumentException.class)
```

```
public void contaMaioresQue1() {
   contaMaioresQue(new ArrayList<Integer>(),15);
}
```

Característica 2: Não existem números superiores ou iguais

Domínio: lista = [0,1,2,3,4], x = 5

Regiões: True or False

```
//Teste com input = 2
@Test
public void contaMaioresQue2() {
    assertEquals(0, inverteDobre(new ArrayList<Integer>(Arrays.asList(0,1,2,3,4)),5));
}
```

Característica 3: Existem números superiores ou iguais

Domínio: lista = [0,1,2,3,4], x = 1

Regiões: True or False

```
//Teste com input = -2
@Test
public void contaMaioresQue3() {
    assertEquals(4, inverteDobre(new ArrayList<Integer>(Arrays.asList(0,1,2,3,4)),1));
}
```

Característica 4: Número negativo

Domínio: lista = [1,2,3,4], x = -2

Regiões: True or False

```
//Teste com input = 2
@Test
public void contaMaioresQue4() {
    assertEquals(4, inverteDobre(new ArrayList<Integer>(Arrays.asList(1,2,3,4)),-2));
}
```

```
/**
* Retorna o número de valores superiores ou iguais a x na lista.
* @param lista - a lista contendo os elementos alvo de pesquisa.
* @param x o elemento a comparar.
* @requires lista != null && !lista.isEmpty()
*/
public static int contaMaioresQue(ArrayList<Integer> lista, int x)
```