Estruturas de Dados
Ficha Laboratorial Nº 4
Engenharia Informática – Instituto Superior de Engenharia de Coimbra

1 – Construa uma classe *DezReais* que permite ao utilizador armazenar 10 números (objectos da classe *Double*). Os números podem ser acrescentados através do método add(*Double*). Não é possível remover números. Caso a utilizador tente inserir números para além do limite, deve gerar uma excepção *Runtime* adequada.

2 – Construa um iterador adequado para a classe *DezReais*, modificando-a de forma a que passe a implementar o interface *Iterable<Double>*. Ignore, por agora a possibilidade de ocorrerem excepções para além de *UnsupportedOperationException*.

3 – Acrescente ao iterador construído na alínea anterior o suporte para excepções do tipo *NoSuchElementException.*

4 – Crie uma classe *DezReaisMutável* análoga a *DezReais* que permite a remoção elementos. Crie um iterador adequado, garantindo a lançamento correcto das excepções *IllegalStateException* e *NoSuchElementException*.

5 – Acrescente o suporte necessário para que o iterador de *DezReaisMutável* gere a excepção *ConcurrentModificationException* caso o contentor de dados seja alterado externamente.

6 – Crie um iterador alternativo que percorre apenas os elementos positivos. Este recebe uma posição e devolve o índice do próximo elemento positivo, que se encontra a seguir a essa posição, ou -1 se não existir mais nenhum positivo.

7 – Construa um algoritmo que procura o maior número num conjunto iterável de *Doubles*. Verifique se o seu algoritmo funciona de forma adequada com a classe *DezReais*, *DezReaisMutável* e com a classe *ArrayList<Double>*.

8 – Generalize o algoritmo anterior de forma a que funcione para qualquer colecção com elementos comparáveis