

POO (MiEI/LCC)

2015/2016

Ficha Prática #08

Auto-avaliação

Conteúdo

1	Objetivos	3
2	Exercícios	3

1 Objectivos

Com esta ficha pretende-se que faça uma auto-avaliação dos conhecimentos que adquiriu ao longo do semestre.

A ficha é consituída por 6 exercícios que deverá resolver e submeter na plataforma de *e-learning* até à data do teste.

2 Exercícios

1. Escreva um programa Java que leia uma quantidade não determinada de números inteiros positivos (o número que encerrar a leitura será o zero). Calcule e imprima a quantidade de números pares e ímpares assim como a média (numero real) dos valores pares.
2. Escreva um programa Java que dado um número M e um número N, inteiros positivos, lê N idades mostrando todas as idades maiores que M. No final deve mostrar a média (numero real) das idades lidas.
3. Escreva um programa Java que, dado um número inteiro N maior que 0, leia um vector (array unidimensionnal) de N elementos inteiros e imprima quantos valores pares existem no vector.
4. Escreva um programa Java que determine quantas vezes uma sub-cadeia de caracteres (fornecida pelo utilizador) aparece no interior de uma outra cadeia (também fornecida inicialmente pelo utilizador).
Considere que a cadeia 'BABABABA' contém três subcadeias 'BAB'.
5. Considere que estamos a desenvolver um software para fazer a gestão de percursos turísticos. Os utilizadores escolhem as localidades que querem visitar e criam um percurso que depois irão efectuar. Considere que é necessário desenvolver a classe *Percurso*, que permite registar as localidades que o utilizador seleccionou e permite saber em que ponto do percurso é que se está. Um dos programadores da empresa que está a desenvolver este software criou a classe que a seguir se apresenta.

```
1 public class Percurso {
2     private String nomePercurso;
3     // localidades do percurso já visitadas
4     private Stack<String> locsVisitadas;
5     // localidades ainda por visitar
6     private Stack<String> locsPorVisitar;
7     ...
8 }
```

Implemente os métodos seguintes, assumindo somente os métodos de Stack conhecidos: pop, push, top e empty. Pode ainda utilizar os métodos forEach e stream para utilizar *streams* activas. Tenha em atenção que o código deve ser robusto, sendo necessário verificar se os métodos necessitam de lançar exceções.

Tenha em atenção que todos os métodos de consulta pedidos não devem, em caso algum, destruir a informação do percurso.

- (a) construtor inicial que cria um percurso com as localidades que vão ser visitadas

```
public Percurso(Collection<String> localidades)
```

- (b) dá a localidade acessível para visitar, isto é, a localidade que podemos visitar no momento

```
public String locAVisitar()
```

- (c) avança para a próxima localidade a visitar

```
public void proxLoc()
```

- (d) num percurso, potencialmente parcialmente visitado, pretende-se remover uma localidade do mesmo, sendo que a remoção só é válida se for de uma localidade ainda não visitada

```
public void removeLocalidade(String loc)
```

6. Considere que se pretende modelar uma biblioteca de entradas multimédia. Neste momento a biblioteca contém apenas dois tipos de entradas: Albus e Filmes. Um Album pode ser caracterizado como tendo:

- um identificador único atribuído pelo dono;
- um título;
- o nome do artista ou da banda;
- número de músicas;
- tempo total das músicas do disco;

- uma flag indicando que se possui o album;
- um comentário.

Para um Filmes guarda-se a seguinte informação:

- um identificador único atribuído pelo dono;
- o título;
- o nome do realizador;
- o tempo total do filme;
- uma lista com os nomes dos actores;
- uma flag indicando que se possui o filme;
- um comentário.

Para resolver este exercício cumpra as seguintes etapas:

- (a) Construa as classes Album e Filme tendo em atenção que deve utilizar o mecanismo de herança.
- (b) Implemente a classe que modela a biblioteca sabendo que deverá satisfazer os seguintes requisitos:
 - permitir acrescentar um novo Album ou Filme às entradas já existentes.
 - permitir procurar um Album/Filme pelo seu nome.
 - permitir procurar todos os Filmes em que entra um actor (dado o seu nome).
 - permitir gravar e ler os dados de ficheiro.

Use um único Map para guardar quer Albums quer Filmes. Utilize quer iteradores internos quer externos.