



Parte 2 - 12.5 valores

Considere que se pretende criar uma aplicação, a **PodCastUM**, que faça a gestão de podcasts. A aplicação guarda os podcasts indexando-os pelo nome do mesmo. Cada podcast possui uma coleção de episódios que são ouvidos pelos utilizadores.

Cada **Episodio** tem a duração (em minutos), o título, a data em que foi lançado e um array de bytes que é o seu conteúdo (que será reproduzido depois numa aplicação cliente).

Cada **Utilizador** possui um número de utilizador, que deve ser *sequencial e atribuído automaticamente*, o seu nome, a identificação dos podcasts que segue e ainda a informação de que ouviu, em determinada data, um episódio de um podcast.

Essa informação, instância da classe **Registo**, guarda o episódio que foi escutado, a data em que foi escutado e o tempo (número de minutos) que o utilizador dedicou a esse episódio.

Considere os seguintes excertos de código:

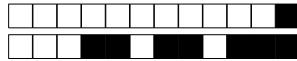
```
public class PodCastUM implements Serializable {
    private Map<String, PodCast> podcasts; // Nome PodCast -> PodCast
    // outras variáveis que considere necessárias
    ...
}
```

```
public class Registo implements Comparable<Registo>, Serializable {
    // variáveis de instância
    ...
}
```

```
public class PodCast implements Serializable {
    private String nomePodCast;
    private List<Episodio> episodios; // Ordenada por ordem de disponibilização do episódio
    ...
}
```

Considere que a estratégia de associação entre **PodCastUM** e os seus podcasts e utilizadores é de **composição**, mas tal já não é necessário na relação entre o **Registo** e o **Episodio** a que se refere, e também entre o **Utilizador** e o **PodCast** que segue.

Assuma, para as perguntas seguintes, que os métodos usuais (get, set, equals, clone, hashCode, ...) estão disponíveis, a menos que sejam solicitados, e responda às questões:



Questão 6

- 1) Efectue a declaração das variáveis de instância de PodCastUM, Utilizador, Episodio e Registo;
- 2) Codifique o método construtor `public Utilizador(String nomeUtilizador, List<String> nomesPodcasts)`, da classe `Utilizador`, que cria uma instância de utilizador e subscrive os podcasts com os nomes passados como parâmetro.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*



Questão 7

Considerando que já existe o método `public void adicionaPodCast(PodCast p)`, na classe `Utilizador`, que adiciona o podcast passado como parâmetro aos podcasts que um utilizador subscreve, **codifique o método** `public void subscrevePodCast(int numUtilizador, String nomePodcast) throws ...`, da classe `PodCastUM`, que efectua a subscrição do podcast indicado pelo utilizador cujo número é passado por parâmetro. Identifique todas as situações necessárias para que o estado dos objectos fique coerente.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*



Questão 8

Codifique os seguintes métodos:

- 1) `public void ouvirEpisodio(PodCast pod, int numEpisodio, int numMinutosEscutados)`, da classe `Utilizador`, que indica que o episódio com o número de sequência `numEpisodio` foi ouvido durante o número de minutos indicados;
- 2) O construtor da classe `Registo` que utilizar.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*



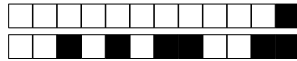
Questão 9

Codifique o método `public String podCastMaisEscutado()`, da classe `PodCastUM`, que devolve o nome do podcast que tem mais minutos escutados considerando todos os seus episódios.

Caso exista mais do que um podcast com o mesmo número de minutos escutados deverá ser escolhido aquele que tiver menos episódios e em caso de novo empate o que tiver o nome alfabeticamente maior.

Se utilizar métodos auxiliares codifique-os.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*



Questão 10

Considere que se pretende ter um novo tipo de utilizador, o `UtilizadorPremium`, que além da funcionalidade associada aos utilizadores já existentes também poderá criar uma cópia local (fazer *download*), com vinte (20) posições disponíveis, dos episódios dos podcasts nos quais tem interesse. Faça a declaração da classe `UtilizadorPremium` e codifique o método `public void guardaLocalmente(Episodio e) throws NaoExisteEspacoException` da mesma classe.

☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 Reservado aos docentes