

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

## PARTE II - 12.5 VALORES

Considere que se pretende ter um sistema que implemente uma serviço de disponibilização de podcasts. Um podcast possui um identificador (um nome) e tem associada uma lista de episódios que foram disponibilizados.

A entidade episódio de um podcast foi definida da seguinte forma:

```
public class Episodio {
    private String nome;
    private double duracao;
    private int classificacao; //dada pelos seus ouvintes (valor de 0..100)
    private List<String> conteudo; //corresponde ao texto que e' dito
                                //quando se reproduz o episodio
    private int numeroVezezTocada; //qts vezes e' que o podcast foi ouvido
    private LocalDateTime ultimaVez; //registra quando foi reproduzido
                                //ultima vez

    ...
    ...
}
```

Considere também que o sistema completo a desenvolver **SpotifyP00** guarda, além dos podcasts existentes e dos episódios destes, informação relativa aos utilizadores do sistema. Para cada utilizador guarda-se o seu identificador (que neste sistema é a String do seu email), o seu nome e a informação dos podcasts que tem subscritos.

Resolva os seguintes exercícios:

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

6. Efectue a declaração das classes `Podcast`, `Utilizador` e `SpotifyP00`, identificando apenas as variáveis existentes e codificando o método `public List<Episodio> getEpisodios(String nomePodcast)`, da classe `SpotifyP00`, que dado um identificador de podcast devolve, numa lógica de composição, uma lista com os episódios disponíveis para esse podcast.

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

7. Desenhe o Diagrama de Classes da solução **SpotifyP00**. Considere que não necessita de colocar os métodos **get** e **set**.

**Resposta:**

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

8. Codifique o método `public void remove(String nomeP) throws...`, da classe `SpotifyP00`, que remove do sistema o podcast identificado. Esta remoção não poderá ser possível se o podcast não existir registado no sistema ou se o mesmo podcast tiver utilizadores que actualmente o estejam a subscrever. Indique na assinatura do método as excepções de que necessitar (não necessita de as codificar).

**Resposta:**

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

9. Codifique o método `public Episodio getEpisodioMaisLongo(String u)`, da classe `SpotifyP00`, que para o utilizador passado por parâmetro, devolve o episódio mais longo de entre os podcasts que esse utilizador tem subscritos.

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

10. Desenvolva o método `public Map<Integer,List<Episodio>> episodiosPorClassf()`, da classe `SpotifyP00`, que associa a cada valor de classificação a lista dos episódios, de todos os podcasts, com essa mesma classificação.

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

11. Considere agora que a classe `Episodio` deverá implementar a interface `Playable`, definida como

```
public interface Playable {  
    public void play();  
}
```

Tendo em consideração que existirá um objecto chamado `System.media`, que tem o mesmo comportamento do `System.out` e que transforma em som o conteúdo em texto do episódio, altere a classe `Episodio` de modo a que implemente `Playable`.

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

12. Considere agora que se criaram novos tipos de conteúdo que passam pela disponibilização de episódios com som e vídeo. Pretende criar-se o `EpisodioVideo`, que para além do audio também possui uma lista de `Byte` que representa o conteúdo visual. Codifique a classe `EpisodioVideo`, apresentando a sua declaração e variáveis, o construtor parametrizado e a codificação do método `play`. Por simplificação assumo que, para reproduzir estes conteúdos, pode primeiro tratar do vídeo e depois do som, e que o `System.media` também sabe reproduzir vídeo.



Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

13. Considere que é possível efectuar a reprodução de um podcast por parte de um Utilizador, através do método `public void playEpisodio(String idPodCast, String nomeEpisodio) throws AlreadyPlayingException` da classe `Utilizador`. A excepção é lançada quando esse utilizador já está no momento a reproduzir um episódio. Considere que se pretende criar agora a noção de `UtilizadorPremium`, que é um utilizador que, enquanto reproduz um episódio, possui a capacidade de colocar os outros episódos que pretende reproduzir numa lista de espera.

Codifique a classe `UtilizadorPremium` com as suas variáveis de instância e a implementação do método `playEpisodio`.

Nome:\_\_\_\_\_Nº:\_\_\_\_\_Curso:\_\_\_\_\_ (C)

---

14. Codifique o método `public void gravaInfoEpisodiosParaTocarMaisTarde(String fich)`, que grava em ficheiro de texto os episódios dos `UtilizadorPremium` que estão na fila de espera para serem reproduzidos. A informação deve ficar guardada com o formato

```
Nome Utilizador
Id do Episodio - duracao
Id do Episodio - duracao
...
...
Nome Utilizador
Id do Episodio - duracao
...
```

Tenha em atenção as possíveis exceções resultantes do uso de ficheiros.