



Guilherme Bedin <guilherme0bedin@gmail.com>

#7DaysOfCode - Java 7/7: Ordenando elementos1 mensagem

Paulo Silveira <paulo@7daysofcode.io>

13 de junho de 2022 08:03

Responder a: reply-caelum57945.activehosted.1321.8172.46734@d32a.emsend3.com

Para: Guilherme Bedin <guilherme0bedin@gmail.com>



Olá, Guilherme Bedin!

Esse é o seu último dia de **#7DaysOfCode** com Java! Você avançou muito no seu projeto e, para fechar com chave de ouro, você irá voltar a mexer com o seu modelo.

E você pode me perguntar: mas por quê? Ao usar a API do IMDB, os filmes já vêm com uma ordem predefinida, descendente, baseada na nota dos filmes (rating). Tudo bem até aí, mas **e se você quiser ordenar os filmes com base em algum outro critério**, talvez pela ordem alfabética do nome ou pelo ano?

Bora praticar então?

No mundo Java, é possível ordenar uma coleção usando o método `sort()` da classe `Collections`, cuja base é algo como:

```
Collections.sort(contentList); //mas qual é o critério de ordenação?
```

Esse método sabe ordenar listas, desde que você defina uma regra - é aí que vai entrar o seu modelo. A partir dele, você poderá definir essa regra, que também é chamada de “ordem natural”. Ou seja, para ter essa regra no modelo, você deverá **implementar um método bem definido pela interface Comparable**. Dessa forma, os objetos da classe se tornarão comparáveis.

Bora lá então? Vou te dar novamente um pequeno passo-a-passo para ajudar:

1. Implemente a interface `Comparable<? extends Content>` na classe (ou record) `Movie` (e também na classe `Series`, se você a tiver criado). Para começar, você pode implementar o método usando a nota (`rating`) como parâmetro de comparação. Por exemplo:

```
public int compareTo(Content outro) {  
    return this.rating().compareTo(outro.  
rating());  
}
```

No método `main`, para ordenar a lista e gerar o HTML, use:

```
Collections.sort(contentList);
```

2. O método `sort()` está sobrecarregado, então você pode passar um `Comparator` como segundo parâmetro para inverter a lista:

```
Collections.sort(contentList,  
Comparator.reverseOrder());
```

EXERCÍCIO EXTRAS

Não pare por aí! Vamos usar este último dia para praticar mais e

mais de Java!

1. Implemente outras formas de comparação usando um `Comparator`. Você pode usar o método `Comparator.comparing(a, b)`, que recebe a lista e o método que será utilizado na comparação.
2. Se você implementou a API da Marvel, junte as listas das séries e dos filmes em uma só. Para tal, use a API de `Streams` para fazer um `join`, algo como:

```
List<? extends Content> mixedList =  
Stream.of(seriesList, moviesList)  
    .flatMap(Collection::stream)  
    .collect(Collectors.toList());
```

3. Ao misturar as duas listas, pode fazer sentido indicar na sua página HTML se o elemento da lista é uma série ou filme. Para tal, crie um novo método `type()` na interface `Content`. Implemente esse método no modelo (em `Series` e em `Movie`) e mostre o resultado no seu HTML.

Monty Python's Flying Circus (1969)

Movie



Nota: 8.8 - Ano: 1969

Repare acima na palavra “*Movie*” abaixo do título e acima do pôster, indicando que se trata de um filme (e não de uma série ou história em quadrinhos).

DICA

O Comparador também serve para definir novas regras de ordenação usando “*method references*”. Por exemplo, você pode ordenar pelo ano:

```
Collections.sort(contentList,  
Comparator.comparing(Content::year));
```

Personalize o seu código da maneira que achar mais legal e

não se esqueça de compartilhá-lo no seu GitHub ou nas suas redes sociais com a hashtag **#7DaysOfCode**, e também com **#feedback7DoC** caso você precise de alguma ajuda.

EXTRA

Para saber mais sobre ordenação de coleções com `Comparable` e `Comparator`, você pode ler este artigo do blog da Alura.

Ufa, você chegou no final da jornada!

Espero muito que você tenha gostado do nosso projeto, porque eu me diverti muito enquanto o criava. Fique à vontade para me dar um feedback lá no meu Twitter (@paulo_caelum), vou ficar feliz em ouvir sobre os seus sucessos!

Amanhã eu vou te enviar um e-mail com a minha resolução do projeto completo e com um formulário para saber o que você achou desses desafios, se tem algum comentário ou alguma sugestão.

Bons estudos daqui pra frente, obrigado pela companhia e bom trabalho!

Paulo Silveira

CEO e fundador da Alura

Enviado para: guilherme0bedin@gmail.com

[Cancelar a inscrição](#)

Alura, Rua Vergueiro 3185 , São Paulo - SP, 04101-300, Brasil