

Aula Prática – 23/10/2017

Exercício 1.1: Faça o download da classe *Book.java* que está no AVA. Ele contém a estrutura básica para uma classe *Book*, que define dois campos e um construtor para inicializar os campos. Neste e nos próximos exercícios, você adicionará mais recursos à estrutura da classe.

Adicione à classe dois métodos de acesso - *getAuthor* e *getTitle* - que retornem os campos *author* e *title* como seus respectivos resultados. Teste sua classe criando algumas instâncias e chamando os métodos.

Exercício 1.2: Adicione dois métodos, *printAuthor* e *printTitle*, à estrutura básica da classe *Book*. Imprima os campos de autor e de título, respectivamente, na janela do terminal.

Exercício 1.3: Adicione mais um campo, *pages*, à classe *Book* para armazenar o número de páginas. Esse deve ser do tipo *int* e seu valor inicial deve ser passado para o único construtor, junto com as strings *author* e *title*. Inclua um método de acesso *getPages* apropriado para este campo.

Exercício 1.4: Adicione um método, *printDetails*, à classe *Book*. Isso deve imprimir detalhes do autor, título e páginas na janela de terminal. Você escolhe como os detalhes são formatados. Por exemplo, todos os três itens poderiam ser impressos em uma única linha ou cada um poderia ser impresso em uma linha separada. Você também pode escolher incluir algum texto explicativo para ajudar um usuário a determinar qual é o autor e qual é o título, por exemplo :

```
Title : Robinson Crusoe, Author : Daniel Defoe, Pages : 232
```

Exercício 1.5: Adicione mais campo, *refNumber*, para a classe *Book*. Esse campo pode armazenar um número de referência para uma biblioteca, por exemplo. Ele deve ser de tipo *String* e ser inicializado com a string de comprimento zero ("") no construtor, pois seu valor inicial não é passado em um parâmetro para o construtor. Em vez disso, defina um método modificador para ele com a seguinte assinatura :

```
public void set RefNumber(string ref)
```

O corpo desse método deve atribuir O valor do parâmetro ao campo *refNumber*. Adicione um correspondente de acesso *getRefNumber* para ajudá-lo a verificar se o método modificador funciona corretamente.

Exercício 1.6: Modifique seu método *printDetails* para Incluir impressão do número de referência. Entretanto, O método deve imprimir o número de referência somente se tiver sido configurado, isto é, se a string *refNumber* tiver um comprimento diferente de zero. Se ela não tiver sido configurada, então imprima a string "ZZZ" no lugar. Dica : Utilize uma instrução condicional cujo teste chama o método *length* na string *refNumber*.

Exercício 1.7: Modifique seu mutator de *setRefNumber* de modo que configure o campo *refNumber* somente se o parâmetro for uma string de pelo menos três caracteres. Se for inferior a três imprima uma mensagem de erro e deixe o campo inalterado.

Exercício 1.8: Adicione um campo de inteiro, *borrowed*, à classe *Book*. Isso mantém uma contagem do número de vezes que um livro foi emprestado. Adicione um método modificador, *borrow*, à classe. Isso deve incrementar o campo por 1 toda vez que ele for chamado. Inclua um de acesso *getBorrowed*, que retorna o valor desse novo campo como seu resultado. Modifique *printDetails* de modo que inclua o valor desse campo com um trecho de texto explicativo.

Exercício 1.9: Exercício de desafio.

Crie um classe chamada *Library*, que seja capaz de armazenar uma lista de objetos da classe *Book*. Esta lista pode ser implementada utilizando a mesma estratégia da classe *Lista* que utilizamos em aulas anteriores (a inserção aumenta o tamanho do vetor quando necessário). A classe *Library* deve conter os seguintes métodos:

- **Construtor:** recebe como parâmetros um inteiro que define o tamanho inicial da lista (vetor);
- **void add(Book book)** : insere um livro na lista (book); caso não haja mais espaço no vetor, realocar os elementos em um novo vetor, com o dobro do tamanho;
- **void print()** : imprime toda a lista de livros no console;
- **Book remove(int index)** : remove do vetor o livro da posição indicada pelo parâmetro *index* e retorna o conteúdo;
- **int findTitle(String title)** : dado um título de livro, realiza a busca na lista e retorna o índice onde o livro está armazenado. Caso não encontre o título, retorna -1.

Exercício 1.10: Exercício de desafio.

Acrescente um novo método de busca na classe anterior, *Book[] findTitles(String title)*, que recebe um parâmetro que pode conter o título parcial de um livro e retorna um vetor com todos os livros que contém a string passada como parâmetro. Caso nenhum livro seja encontrado, o método deve retornar null.