

Lista de exercícios extra

1. Este exercício tem como objetivo estudar e treinar a criação de classes com atributos e métodos.

Lembre-se de utilizar as convenções para declaração de nomes de métodos, atributos e classes.

a) Criar uma classe chamada `Livro`, contendo os seguintes atributos:

- ➡ `nome`
- ➡ `ISBN`
- ➡ `autor`
- ➡ `data de edição`
- ➡ `editora`
- ➡ `número de páginas`

b) A classe também deverá conter os seguintes métodos:

- ➡ `imprime()`: Este método deverá imprimir todos os dados do livro
- ➡ `inicializa()`: Este método deverá receber como parâmetros os valores de inicialização de todos os atributos e fazer a inicialização de cada um deles

c) Crie uma classe chamada `TesteLivro`, que dentro do método `main` realiza as seguintes tarefas:

- ➡ Declare e construa um objeto do tipo `Livro` chamado `livro1`
- ➡ Faça uma chamada ao método `imprime()` do objeto
- ➡ Inicialize o `livro1` com valores através do método `inicializa()`
- ➡ Faça uma chamada ao método `imprime()` do objeto
- ➡ Altere os valores dos atributos do livro acessando diretamente o atributo
- ➡ Faça uma chamada ao método `imprime()` do objeto

d) Crie o diagrama de classes para representar a classe `Livro`

2) Encapsule (crie os métodos de acesso, getters e setters e utilize o modificador `private` nos atributos) os atributos da classe `Livro` adequadamente e faça as modificações necessárias na classe `TesteLivro`.

3) Sobrecarregue (faça overload) do método `inicializa()` recebendo todos os parâmetros menos o número de páginas.

Na classe `TesteLivro` construa mais um objeto `livro` e utilize o método `inicializa()` que você acabou de criar e o método `imprime()` para visualizar os resultados.

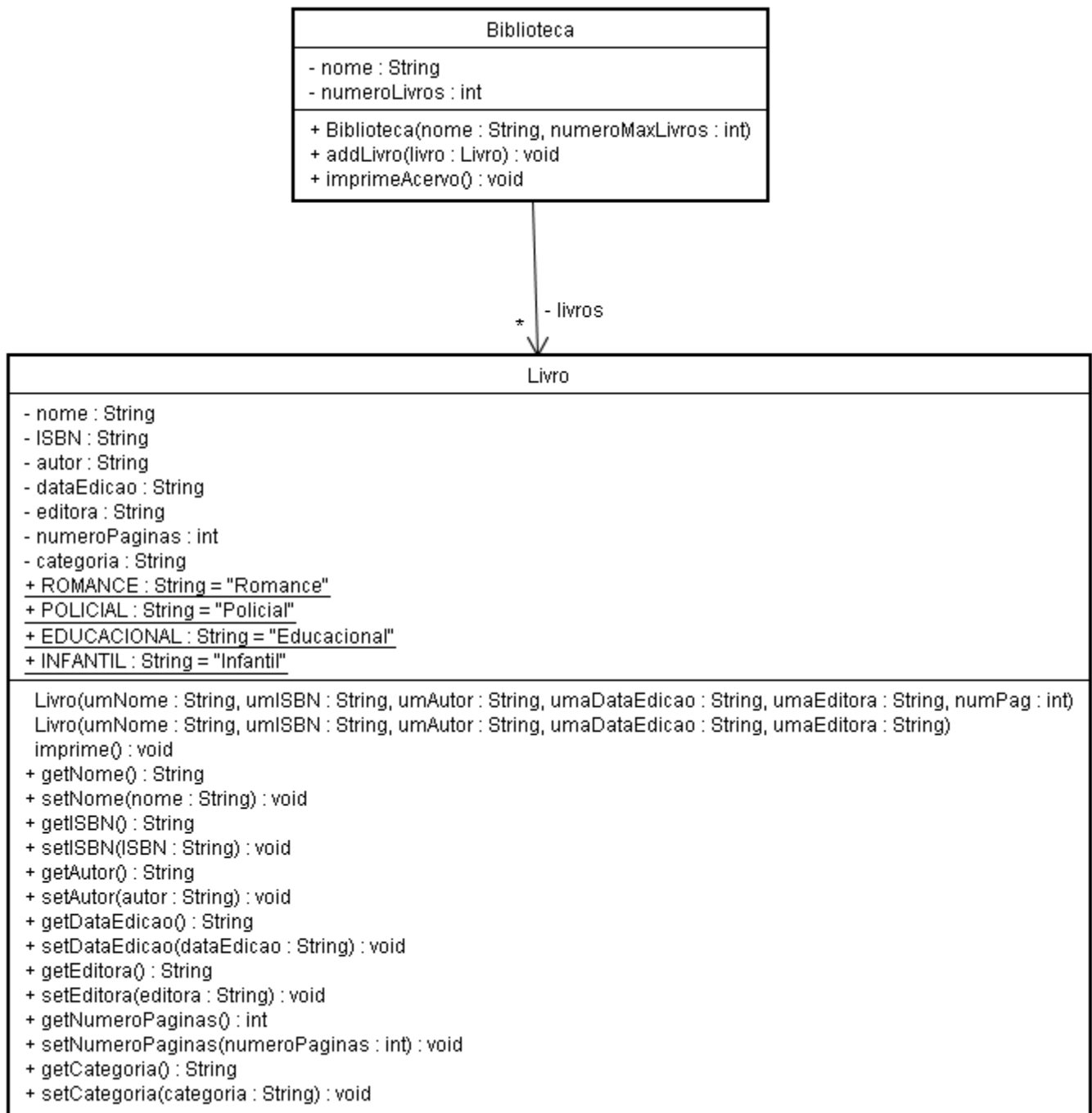
4) Defina dois construtores que julgar necessários na classe `Livro` e remova os métodos `inicializa()` da classe. Faça as alterações necessárias na classe `TestaLivro`.

5) Crie 4 possíveis categorias para um `Livro` (`Romance`, `Policia`, `Educacional` e `Infantil`), para isto, crie 4 constantes estáticas do tipo `String` na classe `Livro`. Adicione um atributo `categoria` do tipo `String` na classe `Livro` e o encapsule.

Altere o método `imprime()` da classe `Livro` fazendo com que ele imprima também o valor do atributo `categoria`.

Teste o método e as constantes na classe `TesteLivro`.

6) Implemente o seguinte diagrama de classes utilizando a classe `Livro` desenvolvida nos exercícios anteriores e crie a classe `Biblioteca`.



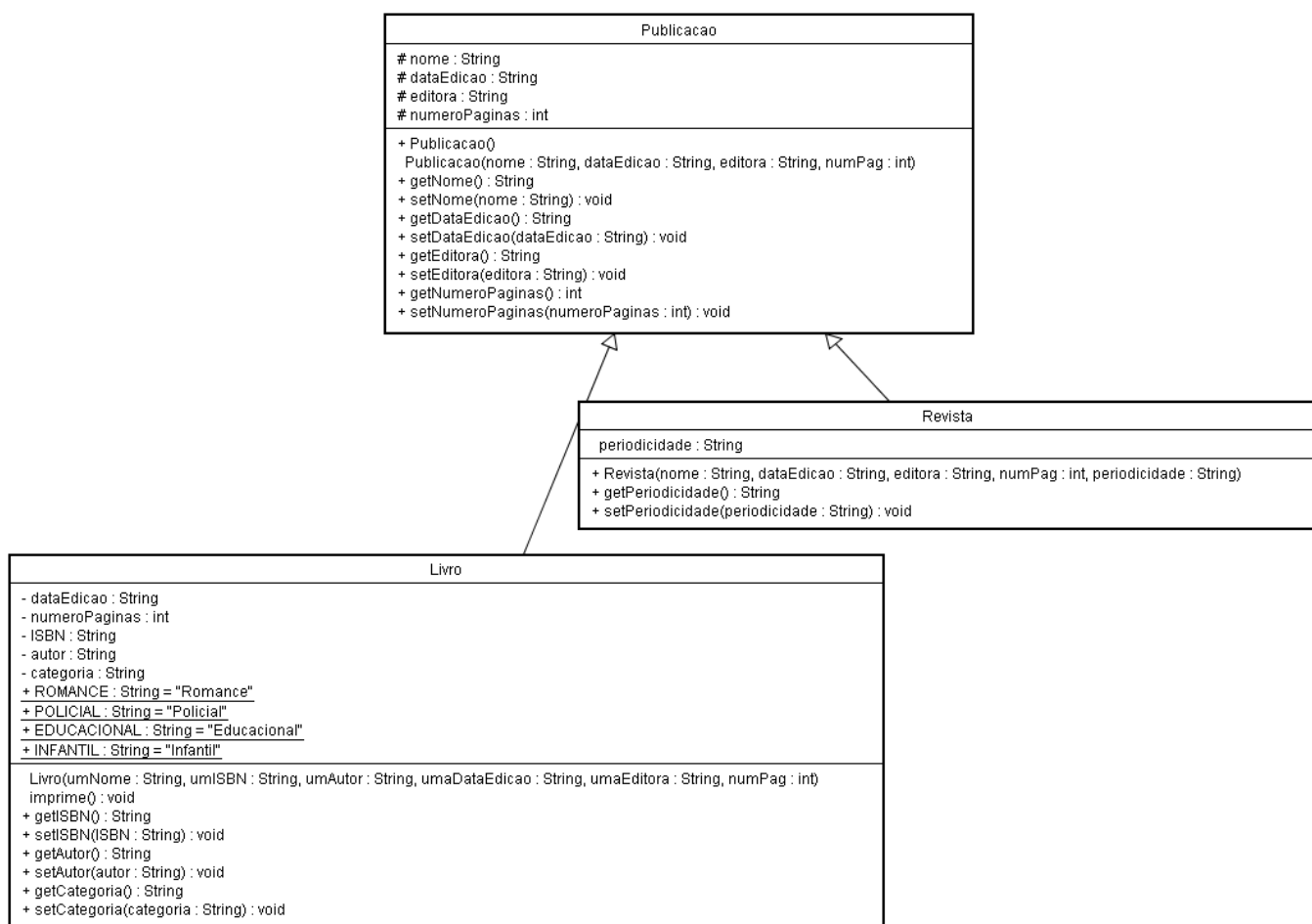
Repare que não é necessário implementar métodos de remoção de livros do acervo da biblioteca ou verificar se um livro pertence ao acervo.

O método `addLivro()` deve adicionar o livro recebido como parâmetro ao `array[]` de livros da classe `Biblioteca`. O método `imprimeAcervo()` deve imprimir o nome de todos os livros da biblioteca.

Para saber o que está acontecendo na chamada dos métodos de sua classe, dentro do método `addLivro()`, além de adicionar o livro no `array[]` de livros da classe `Biblioteca`, imprima o status da operação, dizendo se o `Livro` foi adicionado ao acervo da biblioteca, ou se o acervo já estava com sua capacidade máxima e portanto o livro não foi adicionado.

Para testar sua implementação crie a biblioteca com capacidade máxima para 5 livros, crie 6 livros e tente adicionar as instâncias da classe `Livro` a biblioteca.

7) Implemente o seguinte modelo de classes reutilizando as classes que já foram criadas nos exercícios anteriores. Lembre-se dos detalhes técnicos na utilização de construtores com herança.



8) Agora vamos adaptar a classe `Biblioteca` para que ela tenha um acervo de publicações e não de livros.

- ➡ Substitua o atributo `Livro[] livros` da classe `Biblioteca` para `Publicacao[] acervo`.
- ➡ Substitua o método `addLivro(Livro livro)` da classe `Biblioteca` pelo método `addPublicacao(Publicacao publicacao)`.

9) Torne a classe `Publicacao` abstrata, definindo um método `public void imprime()` abstrato nesta classe. Desta forma, todas as suas sub-classes terão que obrigatoriamente implementar o método abstrato `imprime`.

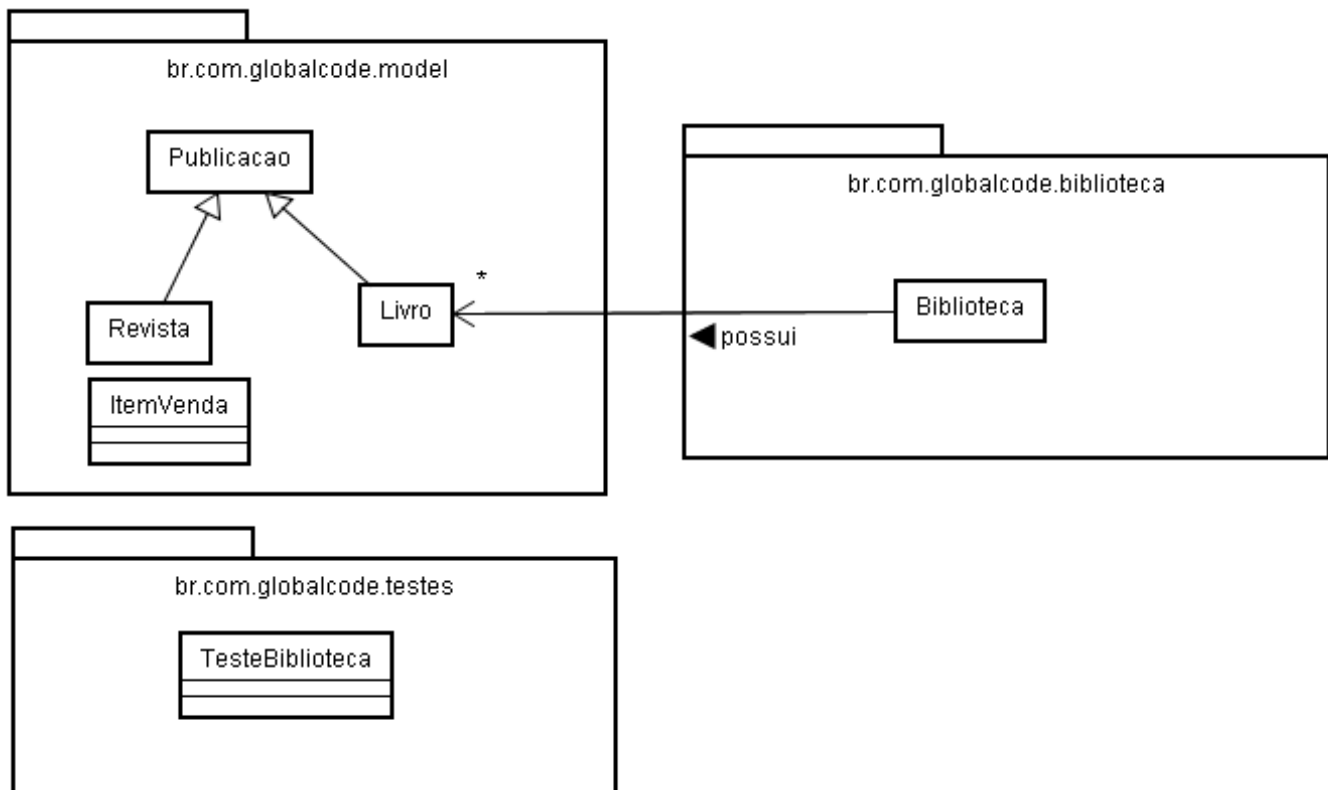
10) Se o método `imprime` na classe `publicação` for declarado com os modificadores `abstract` e `public`, e na classe `Revista`, que estende a classe abstrata `Publicacao`, o método `imprime` for declarado sem que seja colocado nenhum modificador de acesso (`public`, `protected` ou `private`) o que irá acontecer quando tentarmos compilar a classe `Revista`?

11) Para que um livro possa ser vendido por uma livraria ele terá que implementar a interface `ItemVenda`.

```
1
2 public interface ItemVenda {
3     public double getPreco();
4     public void setPreco(double preco);
5     public long getCodigo();
6     public void setCodigo(long codigo);
7 }
```

Faça com que a classe `Livro` implemente esta interface.

12) Organize as classes `Livro`, `Revista`, `Publicacao`, `Biblioteca` e `TesteBiblioteca` de acordo com o seguinte diagrama de pacotes.



13) Crie a classe `CapacidadeMaximaException` que será utilizada para indicar que a `Biblioteca` não pode mais aceitar publicações no seu acervo pois o acervo já está com sua capacidade máxima de publicações, ou seja, o `array[]` de objetos do tipo `Publicacao` já está cheio.

Até o momento foi impressa uma mensagem de erro para indicar que uma publicação não poderia mais ser adicionada, o que na prática não funcionaria, visto que os usuários dificilmente tem acesso a console, por isto, vamos substituir as linhas de impressão de mensagem no método `addPublicacao(Publicacao p)` por um lançamento de exception.

Faça as alterações necessárias a classe `TesteBiblioteca` e teste sua aplicação.