

Tarefa 1

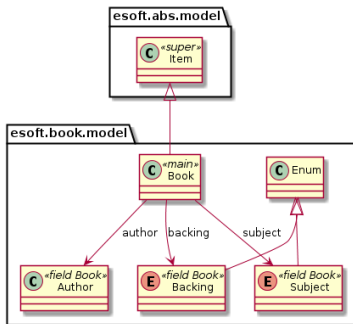
Classe Book

Arthur Vieira de Lima Gomes

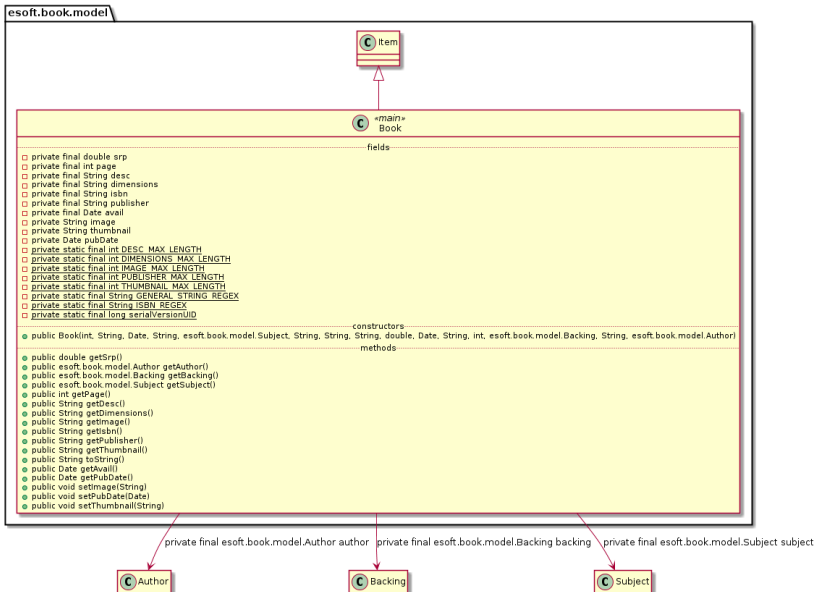
August 5, 2024

Classe Book

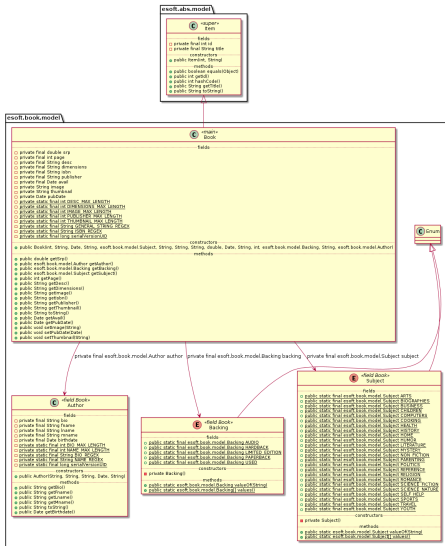
- Implementar a classe **Book**.



Classe Book



Classe Book



Construtor da Classe Book

O construtor da classe Book inicializa um novo objeto Book com os seguintes parâmetros:

- id: Identificador único do livro (herdado da classe pai).
- title: Título do livro (herdado da classe pai).
- pubDate: Data de publicação do livro.
- publisher: Nome do editor do livro.
- subject: Assunto do livro (do tipo Subject).
- desc: Descrição do livro.
- thumbnail: URL da imagem em miniatura do livro.
- image: URL da imagem de capa do livro.
- srp: Preço de venda sugerido do livro.
- avail: Data de disponibilidade do livro.
- isbn: Número ISBN do livro.
- page: Número de páginas do livro.
- backing: Tipo de encadernação do livro (do tipo Backing).
- dimensions: Dimensões do livro.
- author: Autor do livro (do tipo Author).

Além disso, ele faz uma chamada para o construtor da superclasse Item, passando os parâmetros `id` e `title`.

Validações no Construtor

O construtor realiza diversas validações para garantir a integridade dos dados:

- `pubDate`: Validado para não ser nulo.
- `publisher`: Validado para estar no formato correto e não exceder o comprimento máximo.
- `subject`: Validado para não ser nulo.
- `desc`, `thumbnail`, `image`, `dimensions`: Validados para não excederem os comprimentos máximos permitidos.
- `avail`: Validado para não ser nulo.
- `isbn`: Validado para estar no formato correto de ISBN.
- `srp`, `page`: Validados para serem valores positivos.
- `backing`: Validado para não ser nulo.
- `author`: Validado para não ser nulo.

Atribuição de Valores

Após as validações, os valores são atribuídos aos atributos correspondentes do objeto:

- `this.pubDate = pubDate;`
- `this.publisher = publisher;`
- `this.subject = subject;`
- `this.desc = desc;`
- `this.thumbnail = thumbnail;`
- `this.image = image;`
- `this.srp = srp;`
- `this.avail = avail;`
- `this.isbn = isbn;`
- `this.page = page;`
- `this.backing = backing;`
- `this.dimensions = dimensions;`
- `this.author = author;`

Tarefa 1

- Clonar o Repositório:
- `git clone ssh://esoftXX@inf300.ic.unicamp.br
/home/students/esoftXX/inf323/inf323-t1.git`
- Resolva a tarefa
- Submissão da Tarefa

Tarefa 1

- Implementar a classe **Book**. Garantir que:
 - `pubDate`, `avail`, `subject`, `backing`, e `author` não sejam nulos, utilizando `Validator.validateNotNull()`.
 - `publisher`, `desc`, `thumbnail`, `image`, `dimensions` passem por validações de formato e tamanho, com `Validator.validateString()` e `Validator.validateStringMaxLength()`.
 - `isbn` siga um formato específico para ISBN-10 ou ISBN-13, utilizando `Validator.validateString()` com uma expressão regular (`ISBN_REGEX`).
 - `srp` e `page` sejam valores positivos, utilizando `Validator.validatePositive()`.
- Implementar testes unitários na classe `BookTest` para validar o comportamento da classe `Book` e suas validações.

A nota da tarefa 1 será baseada em dois critérios principais:

- O sucesso do processo de build no Jenkins, que inclui testes e compilação.
- A porcentagem de cobertura dos testes unitários.

A cobertura de testes é avaliada usando a ferramenta **JaCoCo**. Esta ferramenta é integrada ao processo de build e gera um relatório detalhado que mostra a cobertura de testes para cada classe do projeto.

A nota será atribuída com base na porcentagem de cobertura de testes da classe Book, conforme indicado pelo relatório de JaCoCo.

Observação: Você pode acessar o relatório de cobertura de testes do JaCoCo de duas maneiras:

- Através da workspace da tarefa no Jenkins. Por exemplo,
<https://inf300.ic.unicamp.br/INF323/job/esoft99-tarefa1/ws/inf323-t1/target/site/jacoco/esoft.book.model/index.html>
- Localmente, no diretório target criado durante o processo de construção com Maven.
<inf323-t1/target/site/jacoco/esoft.book.model/index.html>

- Igual ou acima de 80% de cobertura de testes: **nota máxima (10 pontos)**
- Igual ou acima de 70% de cobertura de testes: **9 pontos**
- Igual ou acima de 60% de cobertura de testes: **8 pontos**
- Igual ou acima de 50% de cobertura de testes: **7 pontos**
- Igual ou acima de 40% de cobertura de testes: **6 pontos**
- Igual ou acima de 30% de cobertura de testes: **5 pontos**
- Menos de 30% de cobertura de testes: **nota 4 pontos**
- Se o código não passar nos testes: **nota 3**
- Se o código não compilar: **nota 1**
- Se o código não for alterado: **nota 0**