**Usar ENUMs em caixa alta (FROYO)**

**Duas subclasses da mesma superclasse**

**Sem construtores em algumas classes**

**Os dados devem ser validados por cada classe**

**Não está claro por ano deve ser atributo de classe**

**Atributos final devem estar em CAIXA ALTA**

**Métodos sem implementação em READER (deveria ter algo implementado – esse trabalho é sobre herança e reaproveitamento de código)**

**Não fez**

**13. Usar Protected acessando diretamente os atributos na classe derivada**

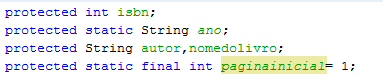
**Métodos sem implementação em READER (deveria ter algo implementado – esse trabalho é sobre herança e reaproveitamento de código**

***Importante***: Para item abaixo deve ser copiado trechos do código que cumprem o requisito e explicado, se não for aparente, o porquê o requisito é cumprido. Sejam bem explícitos. Deve ser indicado também o arquivo da classe em que está o trecho do código. Eu avaliarei o código do Github a partir desse documento para confirmá-lo e também para detectar possíveis erros. **Quem não seguir o que está indicado aqui, não terá o projeto avaliado e perderá a atividade.**

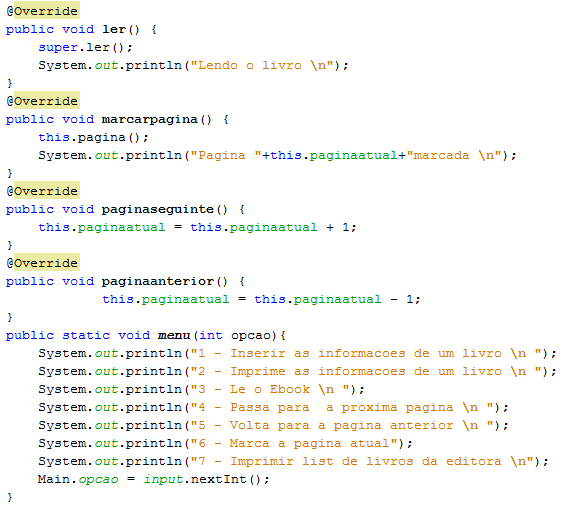
**Requisitos de implementação**

***Todos os atributos e funções membros devem estar relacionados a classe***

1. Pelo menos 4 atributos
2. 4 atributos da superclasse Livros *(Arquivo (Biblioteca)Livro.java)*



1. Pelo menos 4 funções membros sem incluir get e set
2. Funções de Kindle que herda de Reader *(Arquivo (Tablet)Kindle.java)*



**Requisitos de implementação**

1. Cinco classes: uma superclasse e duas subclasses, e duas classes relacionadas ao projeto

Interface tela que Reader implementa e Kindle herda de Reader

*Arquivo (Tablet)Tela.java*

Superclasse Reader que implementa Tela *Arquivo (Tablet)Reader.java*



Subclasse Kindle que estende de Reader *Arquivo(Tablet)Kindle.java*



Superclasse Livro *Arquivo (Biblioteca)Livro.java*

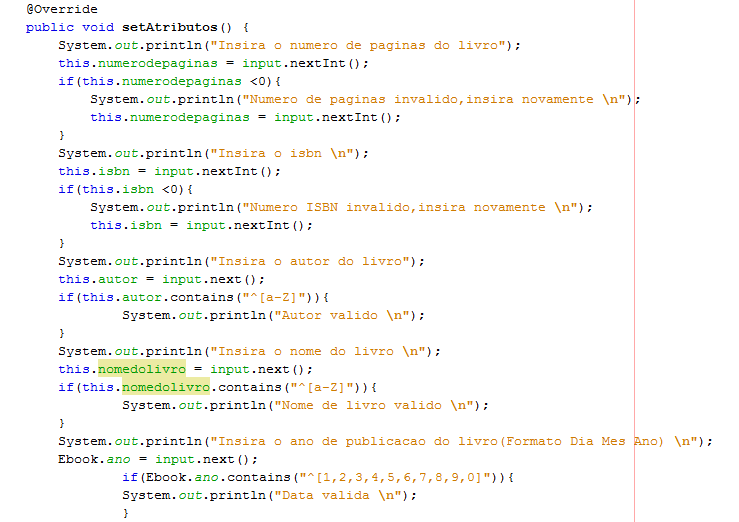


Subclasse Ebook que estende de Livro *Arquivo (Biblioteca)Ebook.java*

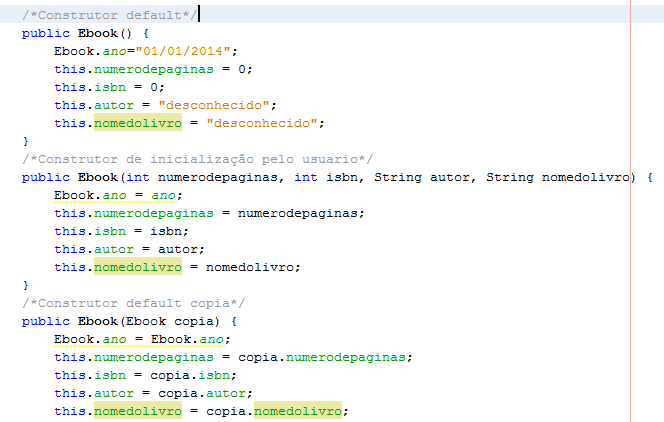


1. Sempre usar o super para o máximo de reaproveitamento de código
2. Diagrama de classes (obrigatório salvar também o png do diagrama no gitHub)
3. Todos os atributos devem ser inicializados. Fez validação de dados?

Validação dos dados *Arquivo (Biblioteca) Ebook.java*



1. Dois construtores *Arquivo (Biblioteca) Ebook.java*



1. Um atributo static.  *Arquivo (Biblioteca) Ebook.java*

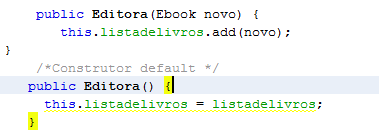


1. Um atributo const static *Arquivo (Biblioteca) Livro.java*

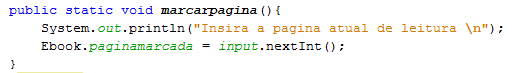


*Correta modelagem dos statics?*

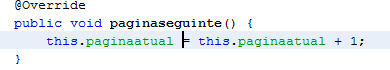
1. Um array *Arquivo (Biblioteca) Editora.java*



1. Método static deve ser chamado no main *Arquivo(Biblioteca) Ebook.java*

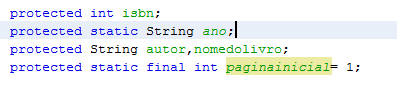


1. Sobrescrita de método: chamar dentro do método da classe. Usar o @override *Arquivo (Tablet) Kinde.java*



1. Usar Protected acessando diretamente os atributos na **classe derivada**

*Arquivo (Biblioteca) Livro.java*



1. Usar suas classes no main *Arquivo (Main) Main.java*



Opcionais que garantem pontos extras:

Trabalhar com ENUM(Feito) *Arquivo (Tablet) Kindle.java*



Trabalhar com pacotes (Feito)

