***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 – A classe Amazon tem 5 atributos.
2. Pelo menos 4 funções membros sem incluir get e set – Feito (isLogado(), mostrarCarrinho(), login() e logout()).

**Requisitos de implementação**

1. Cinco classes: uma superclasse e duas subclasses, e duas classes relacionadas ao projeto – Existem 5 classes, sendo duas delas derivadas de Produto, e um Enum.
2. Sempre usar o super para o máximo de reaproveitamento de código – Em todas as subclasses.
3. Diagrama de classes (obrigatório salvar também o png do diagrama no gitHub) – Feito.
4. Todos os atributos devem ser inicializados. Fez validação de dados? – Feito.
5. Dois construtores – Feito.
6. Um atributo static. Feito.
7. Um atributo const static – Feito.

*Correta modelagem dos statics?*

1. Um array – Sim. Na classe Amazon (variáveis contas e carrinho).
2. Método static – deve ser chamado no main – Na classe Amazon, acessando Amazon.CNPJ;
3. Sobrescrita de método: chamar dentro do método da classe. Usar o @override – Nas subclasses, o método mostrarInformacoes().
4. Usar Protected acessando diretamente os atributos na classe derivada – Feito,
5. Usar suas classes no main – Feito.

Opcionais que garantem pontos extras:

Trabalhar com ENUM – Feito. Enum Categoria.

Trabalhar com pacotes – Feito.