



## Lista 3: Modelagem de classes

Professor: Emerson Ribeiro de Mello

<http://docente.ifsc.edu.br/mello/poo>



### Nota:

- Antes de iniciar, sincronize seu repositório local com o seu repositório remoto “Listas de exercícios” do Github. É possível que o professor tenha feito um commit por lá após a entrega da lista de exercício anterior;
- Crie um diretório com o nome `lista-03` na raiz do repositório e coloque as soluções dessa lista dentro desse diretório;
- Edite o arquivo `Readme.md` que está na raiz do repositório e adicione um novo item com o nome `lista-03` e este deverá ter um link o qual deverá apontar para o subdiretório com o nome `lista-03`.

## 1 Modelagem de classes em UML – somente atributos

Modele uma classe, somente com os atributos, para cada um dos seguintes objetos. Para cada classe crie um arquivo `.png` e salve-o dentro do subdiretório `01-modelagem-uml`. Crie um arquivo `Readme.md` dentro do subdiretório `01-modelagem-uml` e nesse arquivo crie uma seção título 2 (ex: `## Livro`) para cada um dos seguintes objetos listados abaixo, escreva um parágrafo para indicar o contexto que considerou para o processo de abstração e referencie a respectiva figura PNG dentro de sua seção.

1. Livro
2. Disciplina
3. Curso
4. Robô explorador de planetas
5. Datagrama IPv4



### Nota:

Para fazer o diagrama de classes UML (e gerar o arquivo `.png`) use ferramentas como o site <https://draw.io> ou aplicativo desktop StarUML (<https://staruml.io>).

## 2 Modelagem de classes em UML – com atributos e métodos

Modele uma classe para representar um comutador de rede (*switch*). Identifique tudo que um comutador de rede é capaz de fazer (seus métodos) e tudo que define um comutador de rede (seus atributos). Salve o diagrama UML em um arquivo `.png` dentro do subdiretório `02-modelagem-comutador` e referencie essa imagem dentro do arquivo `Readme.md` que também deverá estar no subdiretório `02-modelagem-comutador`.

### 3 Implementação em Java da classe para representar comutador de rede

Crie um arquivo `.java` e implemente aquilo que foi modelado no Exercício 2. Salve esse arquivo dentro do subdiretório `03-implementacao-comutador`. É necessário pensar em comportamento adequado àquele existente em um comutador de rede real. Imagine que as instâncias dessa classe deverão simular os principais comportamentos de um comutador.

- É necessário prover pelo menos um método construtor, além do método construtor padrão;
- É necessário respeitar o conceito de encapsulamento, fazendo uso correto dos modificadores de acesso aos membros da classe;
- É necessário fazer pelo menos uma sobrecarga de método, mesmo se essa não tenha sido prevista na modelagem do Exercício 2.



#### Atenção:

- **Para receber nota 10:** Os commits para essa entrega deverão ser encaminhados (push) para o Github Classroom até o dia **13/03/2020**;
- **Para receber nota 5:** Os commits deverão ser encaminhados (push) para o Github Classroom até o dia **15/03/2020**;
- Qualquer entrega após o dia **15/03/2020** não será contabilizada e receberá nota 0.