

# Linguagem de Programação Orientada a Objetos I

Prof. Ana Karina D. Salina de Oliveira

Faculdade de Computação - UFMS

## Trabalho de LPOO

### 1. Descrição

Desenvolver um projeto em Java no Eclipse que simule um Sistema de Controle de Carteiras de Vacinação.

Fica opcional o uso de uma interface gráfica usando JavaFX para acessar as funcionalidades

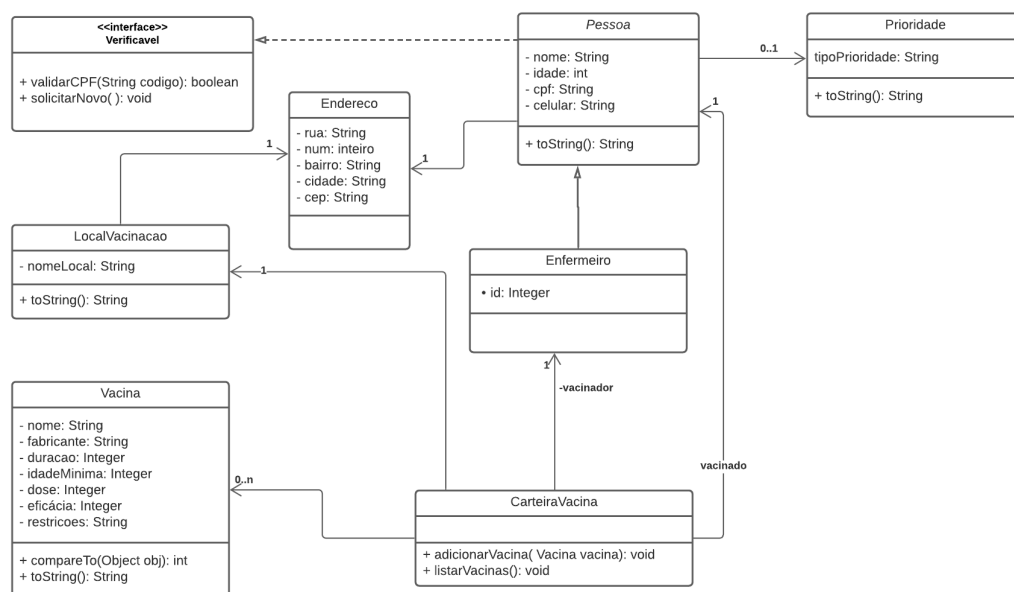
O sistema será composto pelas classes especificadas no Diagrama de Classes abaixo.

O sistema deverá permitir inclusão de Endereco, Local, Pessoa, Prioridade, Enfermeiro, Vacina e Carteira Vacina.

Também deverá permitir inserir novas informações, alterar e ler as informações dadas, utilizando os métodos get e set.

A saída deverá especificar em cada Carteira de Vacina:

o Enfermeiro que vacinou, a Pessoa Vacinada, o Local de Vacinação, listar as vacinas que recebeu em ordem alfabética.



### 2. Descrição das Classes e Interfaces

#### (a) Interface Verificável

- Método:

- a) boolean validarCPF(String codigo);

Este método deverá validar o CPF de Pessoa sempre que esse valor for cadastrado ou alterados. Se o valor retornado for false, então deverá chamar o método que solicita um novo valor a ser validado. Importante utilizar uma função real de validação de CPF.

- b) void solicitarNovo();

Este método deverá solicitar um novo CPF para Pessoa sempre que o valor validado retorna false. Quando o novo valor for fornecido, deverá chamar o método validar, para que seja verificado.

#### (b) Classe Endereço

- Atributos:  
`String rua; int numero; String bairro; String cidade; String cep;`
- 2 construtores:
  - a) Para endereço completo  
`public Endereco(String rua, int numero, String bairro, String cidade, String cep)`
  - b) Para endereço sem cep  
`public Endereco(String rua, int numero, String bairro, String cidade)`

#### (c) Classe Pessoa (Abstrata)

- Implementa a classe Verificavel
- Atributos privados:  
`String nome; Integer idade; String cpf; String celular;` A Pessoa deve ter uma Prioridade e um Endereco.
- 1 Construtor
  - a) `public Pessoa(String nome, Integer idade, String cpf, Endereco endereco, String celular, Prioridade prioridade)`  
obs: no construtor deve-se validar o CPF antes de atribuir o valor do cpf  
Então é preciso chamar o método para validar(`String codigo`)  
Se o CPF estiver correto, ele será atribuído,  
Caso contrário deverá solicitar um novo valor para o cpf
- Métodos:
  - a) *@Override*  
`public boolean validarCPF(String CPF)`  
este método terá que validar o CPF
  - b) *@Override*  
`public void solicitarNovo()`  
este método solicita um novo CPF, até que o valor do CPF informado esteja correto

#### (d) Classe Enfermeiro

- Atributos privados:  
`Integer id`  
`pessoa do tipo Pessoa`
- Estende a classe Pessoa
- 1 construtor com o id do Enfermeiro e os parâmetros de pessoa
- Método:
  - a) *@Override*  
`public String toString()`  
retorna uma String com o nome do Enfermeiro e o id do enfermeiro

#### (e) Classe Prioridade

- Atributos privados:  
`String tipoPrioridade`
- 1 construtor com a prioridade
- Método:
  - a) *@Override*  
`public String toString()`  
retorna uma String com a prioridade

(f) Classe LocalVacinação

- Atributos privados:  
String nomeLocal  
E endereço do tipo Endereco
- 1 construtor com a nomeLocal e endereço
- Método:  
a) *@Override*  
public String toString()  
retorna uma String com o nome do Local, a rua, o número, o bairro

(g) Classe Vacina

- Atributos privados:  
String nomeVacina  
String fabricante  
Integer duracao  
Integer idadeMinima  
Integer dose  
Integer eficacia  
String restricoes
- 1 Construtor com os atributos de vacina
- Métodos:  
a) toString Que retorna o nomeVacina  
b) compareTo que compara o nome das vacinas

(h) Classe CarteiraVacina

- Tem Vacina, LocalVacinao, vacinado e vacinador
- 1 Construtor com os atributos de vacina, do vacinado, do vacinador e do Local de Vacinao
- Métodos:  
a) adicionarVacina  
b) listarVacinas  
Este método deverá listar o nome do vacinado, o nome do vacinador, o Local de Vacinao, e os nomes das vacinas tomadas

(i) Classe Main

- Possui o método main
- Inserir pelo menos 7 Enderecos, 5 Pessoas, 2 Enfermeiros, 3 prioridades, 2 LocaisVacinao, 3 Vacinas, 5 carteirasVacina
- Inserir pelo menos 2 vacinas para cada pessoa
- Listar as vacinas cadastradas
- Listar os Locais de Vacinação
- Listar as carteiras de Vacina cadastradas

(j) Informações adicionais

- O trabalho poderá ser feito individual ou em dupla.
- O código deve ser comentado conforme necessidade de explicar certas funções e funcionalidades;

- O projeto deve utilizar obrigatoriamente os conceitos de Herança, Abstração, Polimorfismo, Encapsulamento e Interfaces;
- Usar coleções para armazenar as Vacinas na classe CarteiraVacina.
- Os modificadores de acesso das classes e atributos devem ser corretamente atribuídos;
- Entregar o projeto zipado com todos os códigos desenvolvidos e um arquivo txt contendo o nome dos alunos.  
No caso de grupo, a nota vai ser a mesma para todos os alunos. Basta que apenas um do grupo faça a entrega no Moodle. A data de entrega será definida no Moodle.
- Entregar um vídeo do trabalho executando, onde deverá explicar as principais funcionalidades implementadas no código. (Enviar o link de onde o vídeo estará armazenado no mesmo arquivo txt que contém o nome dos alunos do grupo). O vídeo deverá ter no mínimo 5 minutos e no máximo 15 minutos.
- A avaliação do trabalho vai ser realizada de acordo com a implementação entregue. Em caso de plágio de outro colega ou da internet (ou de alguma maneira não foi você que fez), é zero para o trabalho (todo o grupo).
- Quaisquer dúvidas no desenvolvimento do trabalho devem ser enviadas para o e-mail: [anakarina@facom.ufms.br](mailto:anakarina@facom.ufms.br).

**Bom trabalho!**