

Programação Orientada a Objeto



INSTITUTO FEDERAL
Triângulo Mineiro
Campus Paracatu

Exercícios
Lista 02

Professor: Edwar Saliba Júnior

Exercícios:

1. Desenvolva um *software* para controle de livros. Este *software* deverá ser capaz de armazenar N livros em memória principal. O seu *software* deverá possuir a classe Livro, com os seguintes atributos: título (*String*), autor (*String*), editora (*String*), ano da publicação (*int*), edição (*int*) e isbn (*String*). Para facilitar a utilização do *software*, por parte do usuário, crie uma interface amigável e de fácil entendimento e utilização. Crie telas que permitam as operações básicas: incluir, excluir, alterar e consultar.
2. Necessita-se de um *software* para controle de funcionários. Este *software* deverá possuir as seguintes classes: Funcionário, com seguintes atributos: cpf (*String*), nome (*String*), salario (*float*), identidade (*String*) e filhos[] (*Filho*); e também a classe Filho, com os seguintes atributos: cpf (*String*), nome (*String*), número da certidão de nascimento (*String*) e data de nascimento (*Date*), conforme Figura 1. Lembre-se, um funcionário poderá ter diversos filhos. Construa um *software* com uma estrutura que permita o cadastro de diversos funcionários e com uma interface amigável para os utilizadores. Crie telas que permitam as operações básicas: incluir, excluir, alterar e consultar.

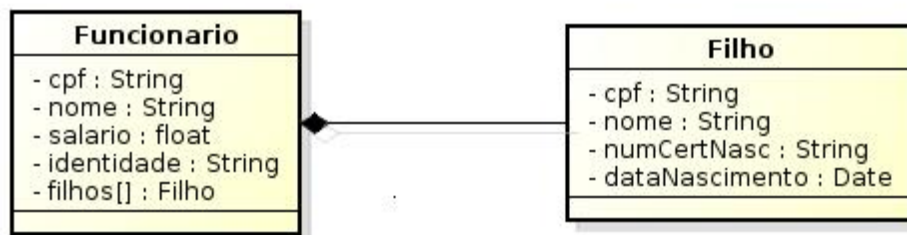


Figura 1: Diagrama de Classe - Funcionário - Filho.

3. Construa um *software* para controle de cobrança e entrega de produtos. Este *software* deverá possuir duas classes: Cliente, com os seguintes atributos: cpf (*String*), nome (*String*), valor (*float*), data (*Calendar*), endereço de cobrança (*Endereço*) e endereço de entrega (*Endereço*) e também a classe *Endereço*, com os seguintes atributos: logradouro (*String*), número (*int*), complemento (*String*), bairro (*String*), município (*String*), estado (*String*) e cep (*String*), conforme mostrado na Figura 2. Seu *software* deverá ser capaz de cadastrar, em memória principal quantos Clientes o usuário quiser. Para facilitar a sua utilização, construa uma interface simples com as operações básicas.

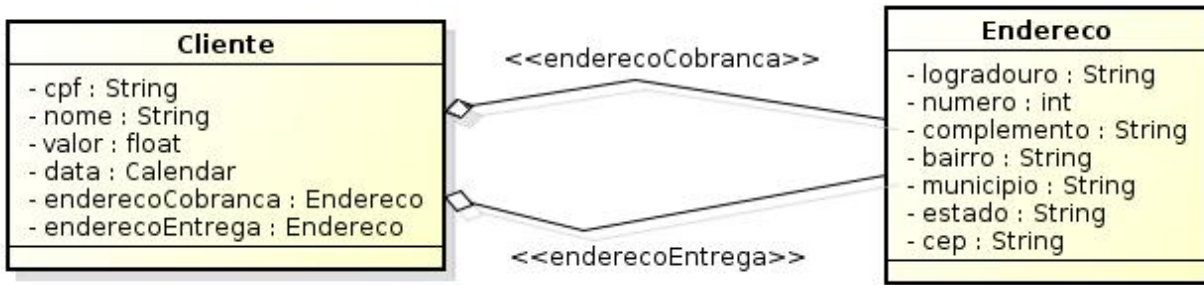


Figura 2: Diagrama de Classe - Cliente - Endereço.

4. Em um Sistema de Agendamento de Consultas de uma Clínica Médica, considere as classes Médico, Consulta e Paciente. A classe Médico possui os seguintes atributos: crm do tipo (*int*) e nome (*String*), a classe Paciente possui: cpf (*String*), nome (*String*), endereço (*String*) e telefone (*String*). Já a classe Consulta possui: médico (*Médico*), paciente (*Paciente*), data (*LocalDate*) e hora (*LocalTime*), conforme mostrado na Figura 3. Construa um *software* que seja capaz de registrar, em um SGBD PostgreSQL, as consultas, os médicos e os pacientes. Para que seu software fique organizado, então, utilize as classes de gerenciamento (vulgarmente conhecida como “fichários”) e também o conceito de MVC. Construa interfaces intuitivas para facilitar a utilização do *software* por parte do usuário.

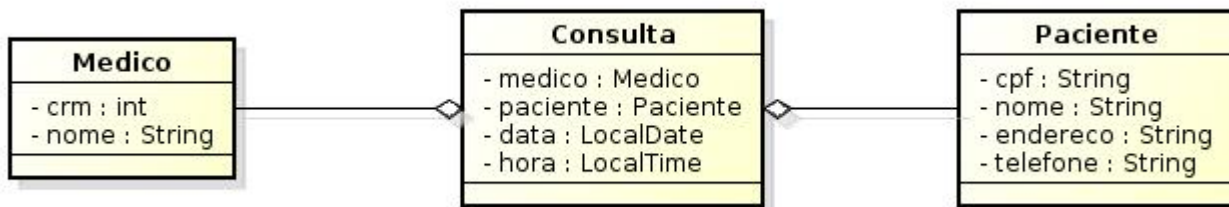


Figura 3: Diagrama de Classe - Médico - Paciente - Consulta.

Observação:

Em Java, antes da versão 8, haviam duas maneiras de se trabalhar com datas, ou seja, podíamos usar o tipo *Date* já ultrapassado ou o tipo *Calendar*, este, um pouco mais recente, mas ainda assim, meio chato de se trabalhar. Já na versão 8 do Java, foi incorporado o tipo *LocalDate*, ou seja, uma nova maneira de se trabalhar com datas em Java.

Para que você esteja ciente e saiba trabalhar com estas três maneiras existentes, os exercícios 2, 3 e 4 contemplam o uso destes três tipos de “datas” que a linguagem Java possui.

Para auxiliá-los neste primeiro momento, deixo-lhes três referências para serem estudadas:

FERREIRA, R.; AQUILES, A. **Conheça a Nova API de Datas do Java 8**. Disponível em: <<http://blog.caelum.com.br/conheca-a-nova-api-de-datas-do-java-8/>>. Acesso em: 05 mar. 2015.

LIMA, L. E. **Lidando Com Datas e Horas em Java**. Disponível em: <<http://www.tecnoclasta.com/2009/06/08/como-lidar-com-datas-e-horas-em-java/>>. Acesso em: 05 mar. 2015.

SALIBA JÚNIOR, E. **Exemplo de Utilização das Classes Calendar e Date**. Disponível em: <<http://www.esj.eti.br/Apostilas/Programacao/Java/DataEmJava.pdf>>. Acesso em: 05 mar. 2015.