

Introdução à Programação 2 - Lista 1

2020.1

Prof. Leandro M. Nascimento
Prof. Vanilson A. A. Burégio

Estoque (pontuação total 2,0 pontos)

1) Escreva as seguintes classes em Java. Você tem a liberdade de definir os tipos dos atributos, então pense bem para usar um tipo adequado ao propósito (ex.: atributo com o nome 'produto' deve ser do tipo Produto que você definiu):

a) Produto:

- Deverá conter os atributos privados: `codigo`, `nome`.
- Deverá conter um construtor sem parâmetros e outro contendo os parâmetros (`codigo`, `nome`).
- Deverá conter o método:
 - Método que altera o nome do produto
 - `mudarNome(nome)`

b) Estoque:

- Deverá conter os atributos privados: `produto`, `qtdAtual` e `qtdMinima`.
- Deverá conter um construtor sem parâmetros e outro contendo os parâmetros (`produto`, `qtdAtual` e `qtdMinima`).
- Deverá conter os métodos:
 - Método que altera o valor da quantidade mínima
 - `mudarQtdMinima(qtdMinima)`
 - Método soma a1o valor atual o valor passado
 - `reporEstoque(qtd)`
 - Método subtrai do valor atual o valor passado
 - `darBaixa(qtd)`
 - Método que escreve os valores atuais dos atributos
 - `relatorio()`

c) Escreva uma classe `TesteEstoqueMain` que vai realizar as seguintes operações:

- Instanciar os objetos:
 - `produto01`
 - `codigo: 1`
 - `nome: Teclado`
 - `produto02`
 - `codigo: 2`

- nome: Mouse
- estoque1
 - produto: produto01
 - qtdAtual: 20
 - qtdMinima: 15
- estoque2
 - produto: produto02
 - qtdAtual: 30
 - qtdMinima: 10

d) Execute as operações na seguinte ordem:

- Dar baixa em 10 unidades de estoque1;
- Fazer a reposição de 19 unidades de estoque2;
- Executar o método relatorio das instâncias estoque1 e estoque2;
- Mudar o nome do produto01 para "Teclado virtual";
- Executar o método relatorio da instância de estoque responsável pelo atributo produto01;

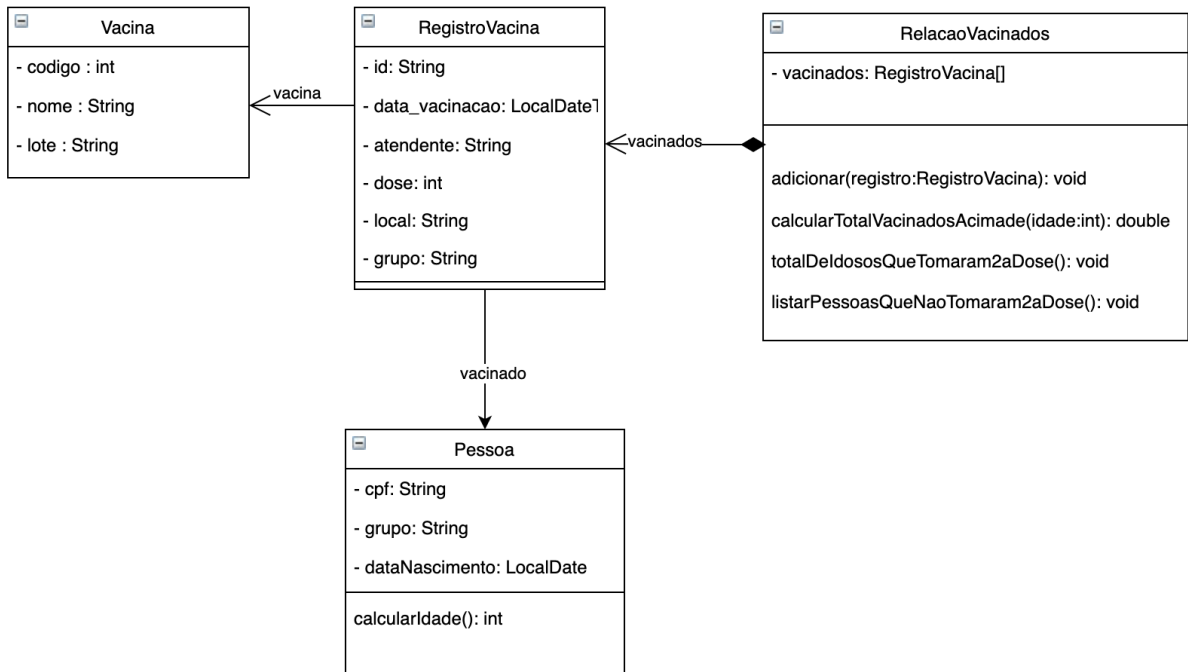
java.time (pontuação total 3,0 pontos)

2) Estude o pacote `java.time.*` e, usando as classes dele, escreva um programa que:

- a) (0,3 pontos) Pegue uma data que o usuário digitou no teclado e transforme em um objeto do tipo apropriado que será usado em outras partes do seu programa
- b) (0,3 pontos) Imprima a diferença (em dias) da data de hoje para a data que o usuário digitou no teclado
- c) (0,3 pontos) Some 110 dias à data digitada pelo usuário e imprima qual a nova data obtida
- d) (0,3 pontos) Altere o mês da data informada para ser igual ao mês de agosto
- e) (0,3 pontos) Imprima a data alterada no formato “dd-MMM-YYYY”. Para isso você deve usar a classe `java.time.format.DateTimeFormatter`.
- f) (1,5 ponto) Nesta mesma classe que você escreve seu programa, crie um método que recebe duas datas (`java.time.LocalDate`) e calcula a quantidade de dias úteis (excluindo sábados e domingos) contidos entre essas duas datas. Para testar o método mencionado, peça que o usuário forneça duas datas usando o teclado e então calcule a quantidade de dias úteis entre essas datas através da invocação do método.

Classes e relacionamentos (pontuação total 5,0 pontos)

3) Com base no diagrama UML abaixo, implemente as classes em Java.



- **(0,5) Vacina**
 - implemente os métodos get/set para cada atributo
 - valores possíveis para o atributo **nome**:
 - CORONAVAC
 - OXFORD
- **(0,5) Pessoa**
 - implemente os métodos get/set para cada atributo
 - método `calcularIdade()`: retorna a idade da Pessoa com base no atributo `dataNascimento`
- **(0,5) RegistroVacina:**
 - implemente os métodos get/set para cada atributo
 - valores possíveis para o atributo **grupo**:
 - IDOSOS
 - TRABALHADORES DA SAÚDE
 - OUTRAS PRIORIDADES
 - valores possíveis para o atributo **dose**: 1 ou 2
 - valores possíveis para o atributo **local**:
 - DRIVE THRU - UFRPE - UNIVERSIDADE RURAL
 - DRIVE THRU GERALDÃO
 - CENTRO DE VACINAÇÃO PARQUE DE EXPOSIÇÃO

- CENTRO DE VACINAÇÃO COMPAZ ARIANO SUASSUNA
- DRIVE THRU PARQUE DE EXPOSIÇÃO
- DRIVE THRU ARRUDA

- **(1,5) RelacaoVacinados:**

- implemente os métodos get/set para o atributo vacinados
- método *adicionar(RegistroVacina registro)*: adiciona um registro de Vacina ao objeto RelacaoVacinados
- método *calcularTotalVacinadosAcimaDe(idade:int)*: efetua a soma de todas as pessoas acima da idade informada como parâmetro que foram vacinadas nas 2 doses
- método *listarPessoasQueNãoTomaram2aDose()*: void
- *listarPessoasQueNãoTomaram2aDose()*: imprima as pessoas que só tomaram a 1a dose e indique a data da próxima vacina somando 21 dias à data em que a mesma tomou a 1a dose

OBS: Siga as tabulações e alinhamentos conforme exemplo. Observe que os valores numéricos estão alinhados à direita, enquanto os textuais estão alinhados à esquerda. Use o método `System.out.printf(...)` para atingir esse resultado de forma rápida e organizada. Vide <https://bit.ly/2L6olzV> ou <https://bit.ly/37GEvxb>. Vídeo com exemplos de printf: <https://bit.ly/39OKr9R>.

Tabela 1. Exemplo de impressão dos registros de vacina das pessoas que não tomaram a 2a dose

CPF	Nome	Idade	Data Indicada p. 2a Dose
0333787383	José Maria	67	13/05/2021
3830333787	Maria José	69	16/05/2021
7870333383	José Silva	71	15/05/2021
3787303383	Maria Silva	66	21/05/2021
Total de vacinados acima de 65 anos: 5			
Total de vacinados que não tomaram a 2a. dose: 4			
Total de vacinados que tomaram a 2a. dose: 1			

- **(1,0) Classe contendo método main:**

- A. Escreva uma classe `TesteVacinaMain` que vai realizar as seguintes operações:
 - Instanciar 5 objetos do tipo `Pessoa` contendo os dados apresentados na **Tabela 1**
 - Instanciar 2 objetos do tipo `Vacina` para representar os dois possíveis valores dos atributo mencionados anteriormente (`CORONAVAC`, `OXFORD`)

- Instanciar 6 objetos do tipo RegistroVacina contendo os dados apresentados na **Tabela 1** e associando-os diretamente aos objetos do tipo Pessoa e Vacina criados nos itens anteriores através da invocação de métodos do tipo set. Faça com que uma das pessoas instanciadas tenha dois objetos do tipo RegistroVacina associados, referentes às duas doses
- Instanciar 1 objeto do tipo RelacaoVacinados e adicione os 6 objetos do tipo RegistroVacina mencionados anteriormente

B. Uma vez instanciados os objetos do Item A:

- Crie um método que imprime os dados da **Tabela 1** e execute-o no final do método main. Lembre-se que, para imprimir todos esses dados, você precisa implementar corretamente todos métodos da classe RelacaoVacinados