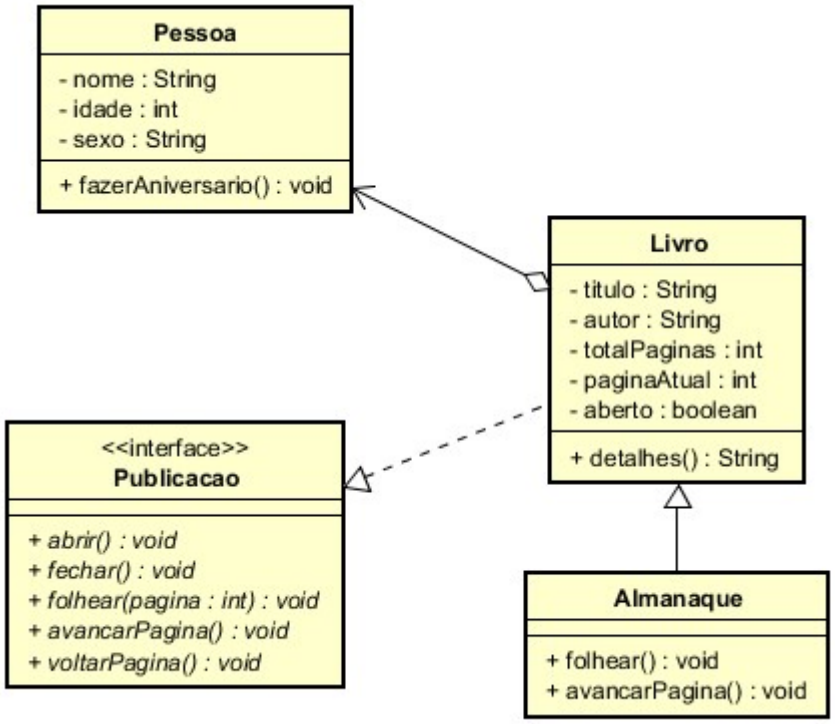
TREINA RECIFE
DESENVOLVIMENTO GERENCIAL E PROFISSIONAL

Lista de Exercícios Módulo 2 – Orientação a Objetos

Janeiro/2024

1. QUESITO

Implemente o modelo abaixo.



a. O método fazerAniversario() deve incrementar a idade de Pessoa.

b. Cada livro só pode ser lido por uma Pessoa por vez.

c. Todo Livro, inicialmente, tem o seu atributo **aberto** marcado como falso, indicando que está fechado, ou não foi iniciada a leitura. Consequentemente, a **paginaAtual** iniciará com o zero.

d. O método detalhes() deve retornar todas as informações do livro, inclusive o **nome** da Pessoa que está lendo o livro, separadas por vírgula, sem espaço.

   Ex.: "Turma da Mônica,Maurício de Souza,25,7,true,Mateus"

e. Os métodos da Publicação devem ser observados com cuidado, pois devem levar em consideração as regras básicas da lógica de leitura:

   i. Para ler um livro, navegando nas páginas, ele deve estar aberto;

   ii. Só se pode navegar por páginas válidas, respeitando os limites inicial e final do livro;

   iii. Só se avança ou volta uma página por vez, caso contrário utilizamos o método folhear().

   iv. Ao navegar pelas páginas, é importante sempre registrar a página atual;

   v. Uma Pessoa só pode ler um livro por vez (aberto). Apesar de poder ter vários em seu nome; e

   vi. Fechar um livro apenas habilita a possibilidade da mesma Pessoa poder abrir outro livro, mas não devem ser perdidas as informações de posição/página.

f. Um Almanaque segue algumas regras diferenciadas:

   i. As páginas são avançadas de 2 em 2, por ser dividido em folhas duplas;

Av. Conde da Boa Vista, nº 1410, sala 16, empresarial Palmira 2, Boa Vista, Recife – PE
Contatos: (81) 9.8215.2121 | contato@treinarecife.com.br | www.treinarecife.com.br

ii. Só é possível folhear esse tipo de publicação pelo índice, localizado na página 2.

2. QUESITO

A partir do modelo anterior, crie uma classe Principal que deverá testar o modelo proposto:

a. Crie duas Pessoas;

b. Crie três livros, associando uma Pessoa a cada um deles;

c. Crie 2 Almanaques, também assaciado a Pessoa;

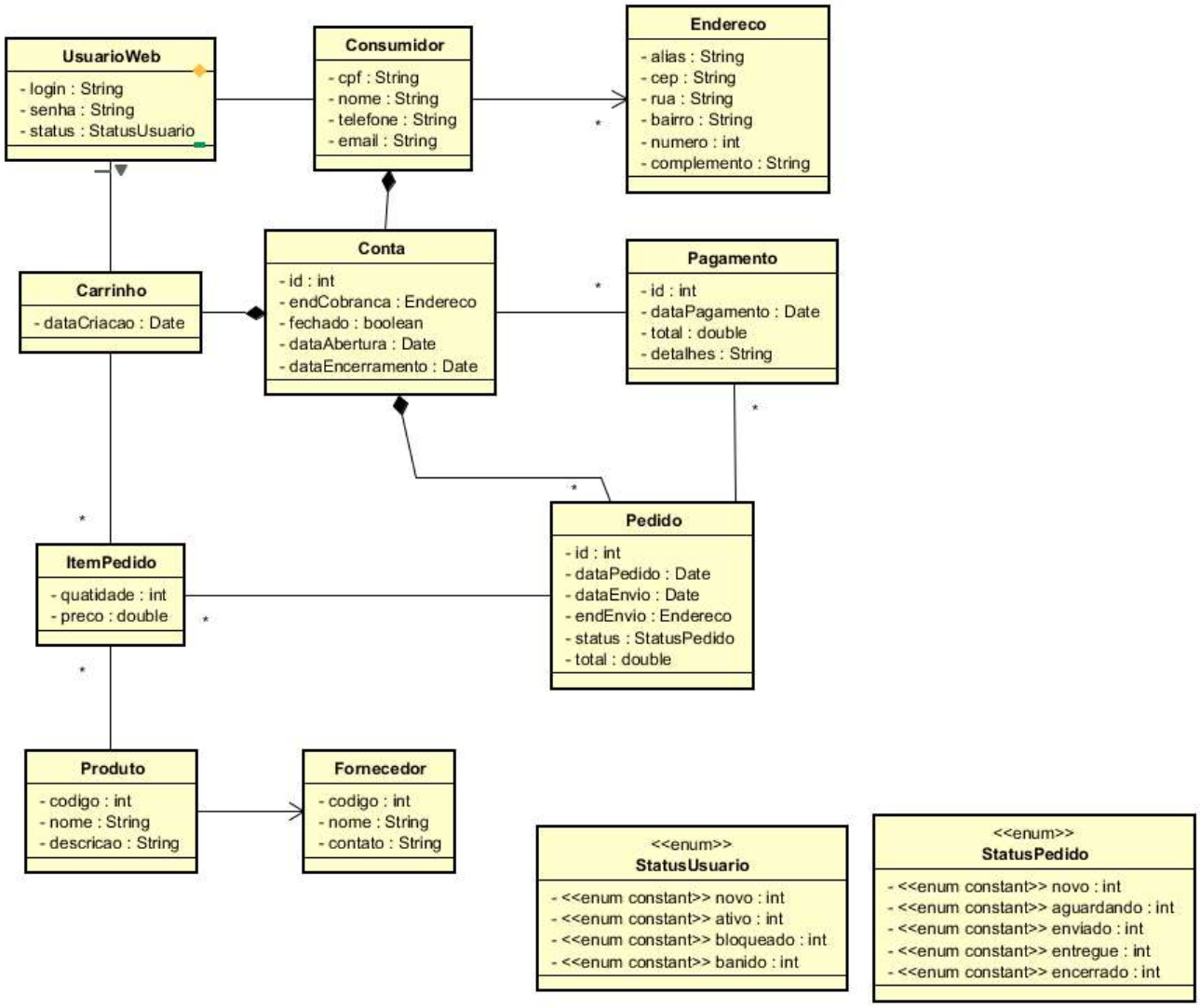
d. Imprima a lista de publicações (5) utilizando o método detalhes();

e. Navege pelas páginas dos livros (para frente, para trás e folheando aleatoriamente), lembrando de abrir e fechar o livro sempre que necessário.

f. Imprima a lista de publicações (5) utilizando o método detalhes();

3. QUESITO

Implemente o modelo abaixo.



Av. Conde da Boa Vista, nº 1410, sala 16, empresarial Palmira 2, Boa Vista, Recife – PE
Contatos: (81) 9.8215.2121 | contato@treinarecife.com.br | www.treinarecife.com.br

Neste diagrama do pacote de Entidades são utilizados os Enums, um tipo específico do Java, que nos permite criar listas (constantes).

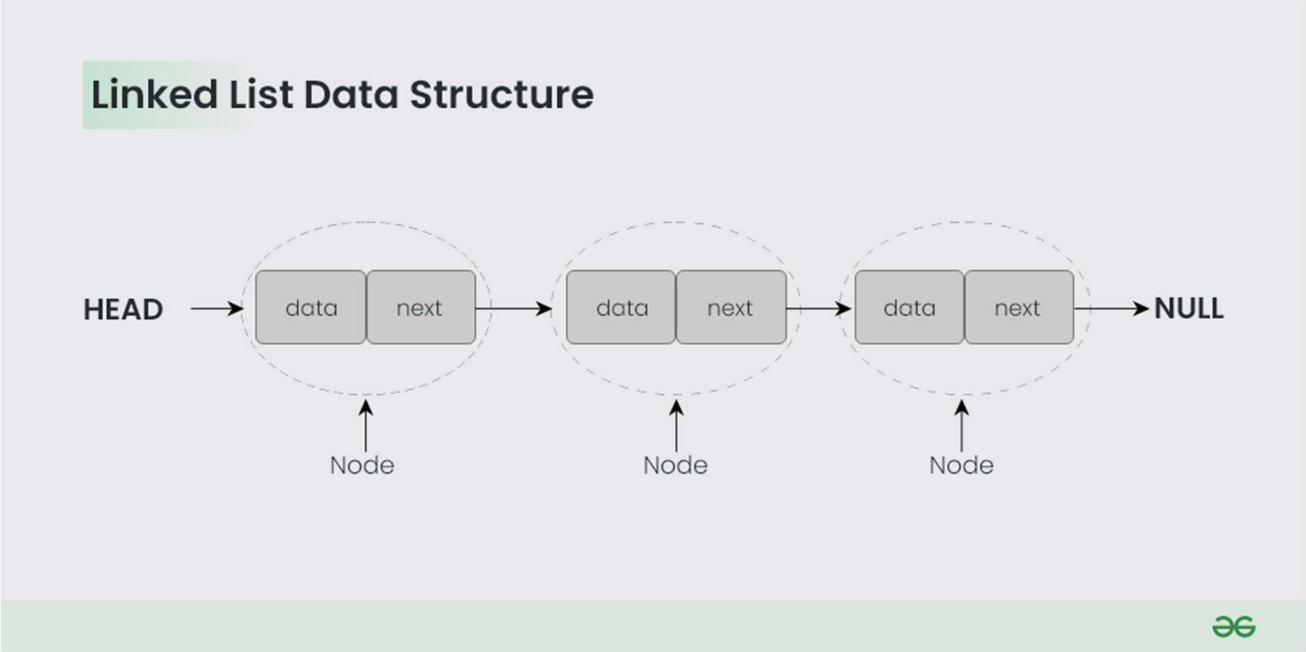
Nas associações sem navegabilidade definida, implemente da forma que lhe convier a relação, lembrando apenas que, pelo menos, uma das entidades deve referencias a outra.

Av. Conde da Boa Vista, nº 1410, sala 16, empresarial Palmira 2, Boa Vista, Recife – PE
Contatos: (81) 9.8215.2121 | contato@treinarecife.com.br | www.treinarecife.com.br

4. QUESITO

Vamos desafiar nossos neurônios?

Você já ouviu falar em Lista Encadeada? É uma estrutura de dados linear, onde os elementos não são armazenados de forma sequencial/contínua na memória (como os Arrays). Os elementos em uma lista encadeada são ligados usando ponteiros como mostrado na figura abaixo:



De forma simples, uma Lista Encadeada consiste em "Nós" (nodes) contendo um campo de dados (data) e uma referência (next) ao próximo nó da lista.

Se você quiser conhecer um pouco mais sobre Linked Lists e outras Estruturas de Dados, em inglês:

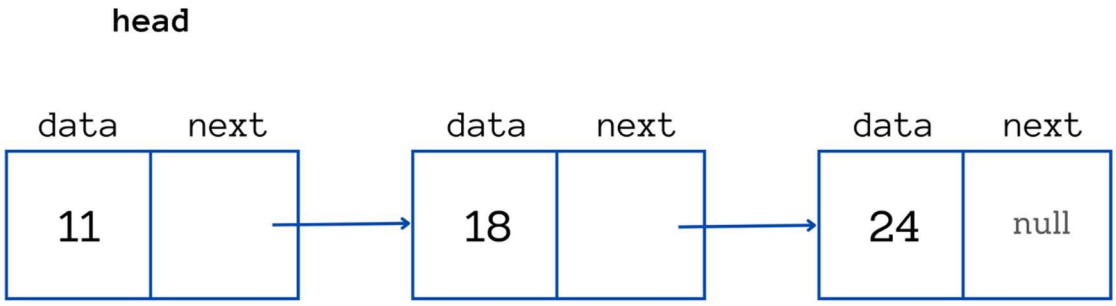
https://www.geeksforgeeks.org/learn-data-structures-and-algorithms-dsa-tutorial/

O desafio agora é criar a sua própria implementação dessa estrutura de dados. Para dar um empurrãozinho, vamos ajudar a identificar os componentes (classes):

1. Nó/Node, atributos:
    a. dado: armazena o objeto na lista. Para facilitar os primeiros paços, recomendo usar um objeto predefinido, como Integer, por exemplo. Ou seja, dado seria do tipo Integer.
    b. próximo: armazena uma referência ao próximo nó da lista. Se estiver em branco (null), significa que é o último elemento (rabo/tail). Ele literalmente represent ao próximo nó, ou seja, ele é do próprio tipo nó/node.
2. ListaEncadeada, atributos:
    a. cabeça/head: uma referência ao primeiro elemento da lista. Quando em branco/null, significa que a lista está vazia.

Av. Conde da Boa Vista, nº 1410, sala 16, empresarial Palmira 2, Boa Vista, Recife – PE
Contatos: (81) 9.8215.2121 | contato@treinarecife.com.br | www.treinarecife.com.br

b. rabo/tail: uma referência ao último elemento da lista. Assim como o head, quando está vazio/null, significa que a lista está vazia. Para uma lista com um único elemento, head e tail apontam para o mesmo nó.



Você pode incluir outros elementos para facilitar o trabalho, como um atributo tamanho/size na ListaEncadeada, que deve ser incrementado ao adicionar um novo elemento.

Lembrando que uma lista não servirá de nada se não possuir os métodos de acesso a ela, como:

ListaEncadeada, métodos (mínimos):

add (Integer elemento): adiciona um novo elemento no final da lista;

remove (Integer elemento): exclui o elemento da lista;

get(int index): retorna o elemento na posição indicada por "index";

clean(): limpa a lista.

Existem muitos outros métodos que podem ser implementados. Uma forma mais padrão seria implementar a interface List do Java. Nela estão protocolados vários métodos que uma lista do Java deveria ter.

```
public class ListaEncadeada implements List<Integer> {
```

Por fim, recomendo que você tente resolver o problema por conta própria ANTES de sair buscando tutorias na internet. Esse exercício vai exigir muito dos seus conhecimentos de lógica e solução de problemas, mas vai te dar ferramentas para resolver muitos outros problemas.

Caso queira reduzir seu esforço mental, vou deixar uma referência para um dos muitos tutorias que encontrei na web: aqui. Novamente, recomendo que você tente por conta própria por algumas semanas, pelo menos.

Se você achou esse desafio interessante, você pode continuar evoluindo com outras Estruturas de Dados, criando suas próprias implementações como exercício. No site recomendado GeeksForGeeks tem algumas estruturas interessante.

Boa diversão!

Av. Conde da Boa Vista, nº 1410, sala 16, empresarial Palmira 2, Boa Vista, Recife – PE
Contatos: (81) 9.8215.2121 | contato@treinarecife.com.br | www.treinarecife.com.br