

## CURSO CIÊNCIA DA COMPUTAÇÃO PROGRAMAÇÃO ORIENTADA A OBJETOS

### LISTA 2 - EXERCÍCIOS

1. Qual é o significado do diagrama mostrado abaixo:



a) Existe uma relação unidirecional entre as classes Pessoa e Celular, sendo que objetos da classe Celular possuem acesso aos objetos da classe Pessoa.

b) Existe uma relação unidirecional entre as classes Pessoa e Celular com multiplicidade igual a 1 em ambos os lados.

c) Existe uma relação unidirecional entre as classes Pessoa e Celular com multiplicidade de 1 para N, no lado da classe Pessoa e da classe Celular, respectivamente.

d) Existe uma relação bidirecional entre as classes Pessoa e Celular com multiplicidade igual a 1 em ambos os lados.

e) Existe uma associação bidirecional entre as classes Pessoa e Celular. A classe Pessoa possui acesso à classe Celular.

2. Em programação orientada a objetos é comum e necessário que objetos estejam associados, justamente para poderem modelar de maneira mais próxima o problema que se deseja resolver. Dentro deste contexto, assinale a alternativa incorreta:

a) Uma associação ou relação representa uma ligação entre classes de objetos. Ela pode ser sem direção, unidirecional e bidirecional

b) As associações devem representar a ligação entre classes de objetos, sendo necessário definir as multiplicidades ou cardinalidades existentes entre os objetos.

c) Há associações falsas e outras verdadeiras. As verdadeiras ocorrem sempre quando um objeto precisa ser criado.

d) Na associação unidirecional, somente um dos objetos participantes da relação possui acesso ao outro objeto participante da relação.

e) A associação unidirecional é representada graficamente por uma linha cheia entre as classes de objetos. Na extremidade da linha, coloca-se uma seta para informar a direção da associação.

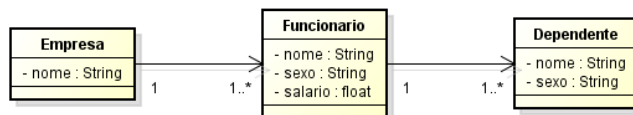
3. Sobre os princípios relacionados à programação orientada a objetos, assinale a correta:

- a) Coesão, nele deve-se evitar misturar funcionalidades ou mesmo concentrar tarefas. Cada método ou função e classe deve ser projetada de maneira a ter responsabilidades bem definidas.
- b) Acoplamento ou dependência é o princípio que determina a assinatura de um método.
- c) Confiança é o princípio da boa relação com a classe que se deseja associar. Ele é importante pois garante que sejam implementados todos os mecanismos de segurança necessários para uma classe.
- d) Despacho é o princípio que determina que uma classe seja projetada com o menor grau de dependência com outras classes. Com isso, facilita-se à evolução do programa.
- e) Modularização não é um princípio da orientação a objetos.

4. Na linguagem de programação Java, as associações podem ser implementadas utilizando? Assinale a alternativa correta:

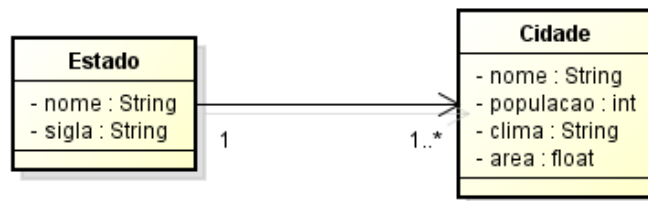
- a) Um vetor, exemplo clássico de alocação de memória dinâmica. O problema desta estrutura está na quantidade de memória alocada ser fixa.
- b) Um vetor, exemplo clássico de alocação de memória estática. O problema desta estrutura está na quantidade de memória alocada ser variável.
- c) Um vetor, exemplo clássico de alocação de memória estática. Esta estrutura exige do desenvolvedor controles para saber quando está lotado ou quando existe o acesso a uma posição não alocada, por exemplo. Uma alternativa está no uso de Coleções em Java. As coleções, com por exemplo *ArrayList*, são estruturas mais robustas, com controles mais sofisticados e com diversos métodos disponíveis na sua API, que tornam o seu uso mais facilitado e transparente.
- d) Somente é possível implementar as associações ou relações em Java, utilizando uma estrutura do tipo vetor.
- e) As relações em Java não podem ser implementadas, elas são muito complexas.

5. Analise o diagrama de classes da UML abaixo e assinale a alternativa correta:



- a) Há duas associações bidirecionais, uma entre a classe Empresa e Funcionário e outra entre Funcionário e Dependente.
- b) Há somente uma associação unidirecional entre a classe Empresa e Funcionário e entre Funcionário e Dependente não há associação.
- c) Há duas associações bidirecionais, uma entre a classe Funcionário e Empresa e outra entre Dependente e Funcionário.
- d) Há duas associações sem direção com multiplicidades ou cardinalidades iguais a 1 de um lado e 1..\* do outro lado.
- e) Há duas associações unidirecionais, uma entre a classe Empresa com direção à Funcionário e uma outra envolvendo Funcionário com direção à Dependente.

6. Considere um estado e, associado a ele, existem diversas cidades associadas, conforme diagrama da UML abaixo:



Fazer um programa orientado a objetos que:

- Crie classes para modelar a situação acima e uma classe Main para entrada e saída de dados. Na classe Main, procure fazer chamadas aos métodos responsáveis pelo cálculo ou geração de dados;
- Crie um objeto do tipo estado e a ele associe diversas cidades. Após, mostre os dados do estado (ele deve mostrar também os dados das cidades associadas);
- Para o estado criado, retorne qual é a cidade que possui a maior população;
- Apresente quantas cidades possuem clima igual a nome do clima (temperado, equatorial, tropical, etc). O nome do clima deve ser passado como parâmetro de entrada;
- Mostre a média das áreas das cidades.

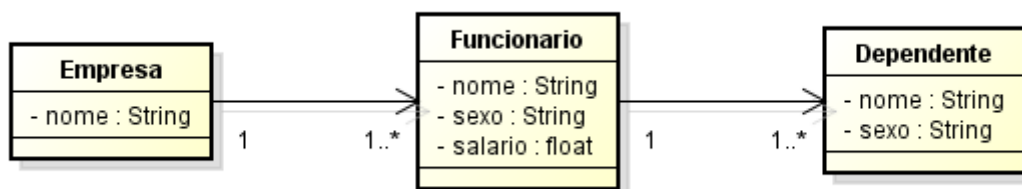
7. Considere que em um evento há vários participantes e diversos palestrantes. Crie um programa orientado a objetos que:

- Mostre o total de participantes e palestrantes associados ao evento;
- Crie e apresente uma lista de participantes que nasceram antes de 03/10/2020;
- Crie e apresente uma lista com palestrantes do sexo masculino.

8. Considere que em um curso de computação há vários alunos e diversos professores. Crie um programa orientado a objetos que:

- Mostre o total de alunos associados ao curso;
- Mostre o total de professores associados ao curso;
- Crie uma lista dos alunos que não moram em Lages;
- Mostre o total de professores com titulação igual a doutorado.

9. Considere que em uma empresa há diversos funcionários e para cada um funcionário tem-se um conjunto de dependentes, conforme a representação a seguir:



Faça um programa orientado a objetos em Java que:

- Defina atributos, métodos, construtores, *toString*, *sets*, *gets* e qualificadores adequados para todas as classes;
- Utilize *ArrayList* para fazer a ligação entre as classes ou objetos;
- Para uma dada empresa, mostre o total de funcionários com sexo igual a feminino;
- Para uma dada empresa e para um dado funcionário, informe a quantidade de dependentes associados ao funcionário em questão;
- Apresente o salário médio dos funcionários de uma empresa.

**10.** Considere um único curso e, associado a ele, existem diversas turmas associadas e a cada turma há diversos alunos associados.

Fazer um programa orientado a objetos que:

- a) Crie classes para modelar a situação acima e uma classe Main para entrada e saída de dados. Na classe Main, procure fazer chamadas aos métodos responsáveis pelo cálculo ou geração de dados;
- b) Crie o curso, mostre as turmas e, para cada turma, os alunos associados;
- c) Apresente a turma que possui o maior número de alunos;
- d) Crie um método de pesquisa. Informe como parâmetro o nome da turma e a matrícula do aluno. Caso o método encontre, apresente todos os dados do aluno.
- e) Cria uma lista contendo os dados da turma "Programação" com sexo igual a feminino.