

INSTITUTO FEDERAL  
GOIÁS

INSTITUTO FEDERAL DE GOIÁS  
CÂMPUS GOIÂNIA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
PROGRAMAÇÃO ORIENTADA A OBJETO I

Nome do Aluno: João Manoel Leite Ribeiro Nogueira Data: 06/11/2016

Prof. Renan Rodrigues de Oliveira

## **Trabalho: Revisão POO**

1. Considerando os conceitos sobre Orientação a Objetos:

a) Explique a seguinte afirmação aplicável a projetos de software OO: “Programe para uma interface, não para uma implementação”.

Temos que programar para a interface, no sentido de encapsular o código ao máximo possível, a fim de que possamos futuramente reaproveitar o código em outros projetos. Não é recomendável programar para a implementação pois o código nesse caso fica específico do projeto em questão. Fazendo assim, seguiremos as melhores práticas de desenvolvimento, garantindo a reutilização do código. Por exemplo, o atributo `listaRodas` de uma classe `Carro` deve ser privado, assim o código usado para as rodas de um carro pode ser reutilizado no caso da implementação de uma classe `Caminhonete`.

b) O que é uma associação de classes? Quais seus principais elementos?

Uma associação de classes é um relacionamento estrutural que especifica que objetos de um tipo são conectados a outro tipo.

Uma associação pode ter os seguintes elementos:

- Nome da Associação: descreve a natureza da associação;
- Papel: define o papel específico neste relacionamento;
- Direção de Leitura: indica como a associação deve ser lida;
- Multiplicidade: define quantos objetos estarão conectados a uma instância de uma associação.

c) Conceitue e diferencie os relacionamentos entre classes de agregação e composição.

### Agregação:

Indicada para representar um relacionamento entre “parte” e “todo”, onde o “todo” é formado por partes.

Este relacionamento é caracterizado pela parte poder existir sem o todo, ou seja, a parte deve existir antes que o vínculo seja realizado.

- Um objeto da classe parte integrante pode existir sem o todo, sendo que este último apenas agrega as partes já existentes;
- Tempo de vida da classe “parte” independente do tempo da classe “todo”.

### Composição:

Indicada para representar um relacionamento entre “parte” e “todo”, onde o “todo” é formado por partes.

Este relacionamento é caracterizado pela parte poder existir somente compondo o todo, ou seja, a parte deve ser criada dentro do vínculo.

- A classe que compõe (parte) possui o mesmo tempo de vida da classe composta (todo);
- Se a classe composta morrer, suas partes também morrerão.

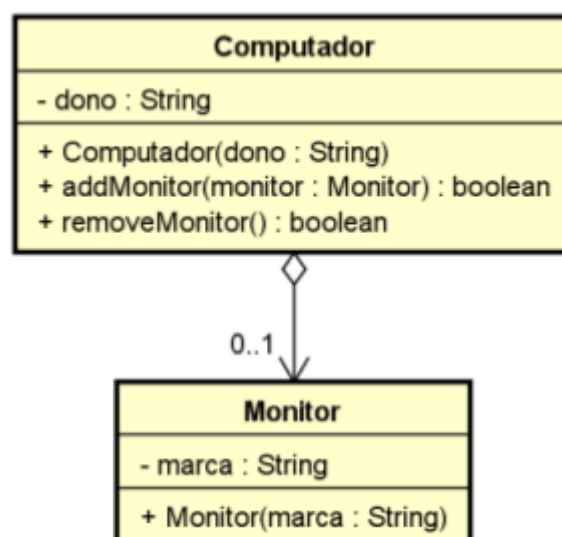
d) Explique a seguinte afirmação na implementação de relacionamentos entre classes: “É de responsabilidade do desenvolvedor prover métodos para vínculo, substituição e/ou remoção da parte”.

O desenvolvedor deve prover esses métodos visando a garantia da multiplicidade da associação entre as classes.

e) Considere os relacionamentos de agregação e composição. Para cada um destes relacionamentos, crie um modelo que represente uma relação de duas classes com as seguintes multiplicidades:

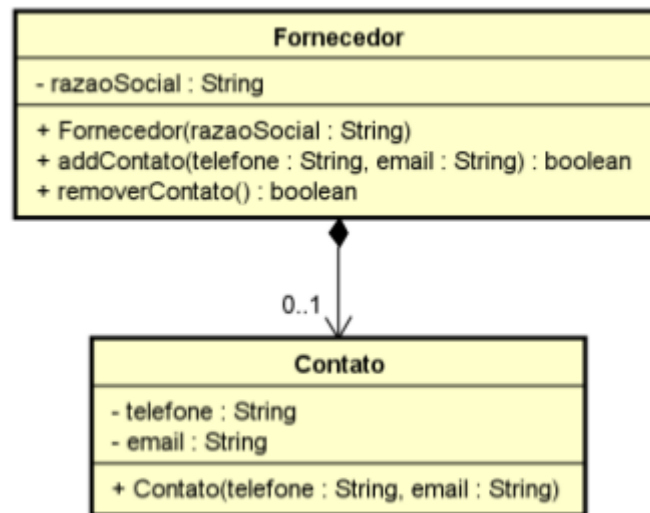
(0..1)

Agregação:



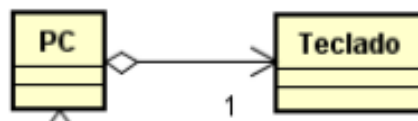
(0..1)

Composição:



(1)

Agregação:



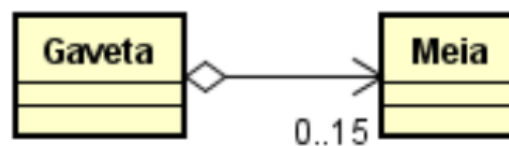
(1)

Composição:



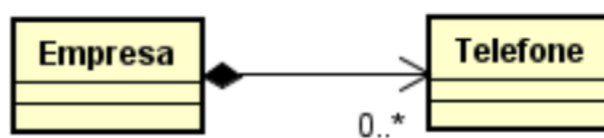
(0..\*)

Agregação:



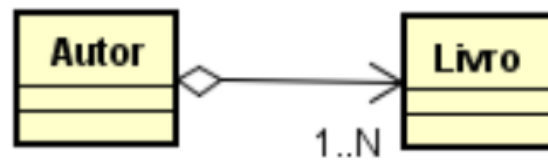
(0..\*)

Composição:



(1..N)

Agregação:



(1..N)

Composição:

