

Nome: Eduardo Henrique de Almeida Azidonio

Matrícula: 2020000315

Disciplina: Linguagens de Programação

## 2ª lista de Exercícios de Linguagens de Programação

1. A maioria das linguagens imperativas não possui classes e métodos. Como você usaria uma linguagem imperativa como Pascal ou C para simular as ideias de classes e métodos?

2. A maioria das linguagens atuais não dá suporte à herança múltipla diretamente. Tente elaborar, pelo menos dois problemas nos quais a herança múltipla é útil.

3. Ainda usando uma linguagem imperativa como C, como você simularia o mecanismo de herança? É possível simular múltipla nessa linguagem?

4. Ada possui formas de definir tipos abstratos. Discuta porque ela não é considerada uma linguagem orientada a objetos.

5. Argumente, sob o ponto de vista prático, porque as linguagens orientadas a objetos se tornaram tão populares e no que elas favoreceram a reutilização.

Resposta:

1. Em C, uma classe pode ser descrita na declaração de objeto como uma variável comum, e para validar isso, uma forma de implementação do objeto, é ser um tipo de variável criado a partir dos tipos primitivos/já existentes. Dessa forma, uma classe de um objeto pode ser definida como um typedef de uma struct. Dentro desta struct, podem ser

criadas variáveis diversas, de quaisquer tipos. Essas variáveis constituem os parâmetros da classe.

Além dos parâmetros, tem os métodos e para isso, utiliza ponteiros de uma função, e eles irão conter os endereços das funções a serem executadas (implementadas em lugares convenientes do código-fonte).

2) 1. Consideremos uma classe chamada nome que armazena somente uma palavra, e outra classe chamada sobrenome, em uma lista, caso tivermos dois nomes iguais teríamos que ter uma subclasse, logo nome sobrenome, seria a duplicação útil.

2. Na classe animal, da qual são derivadas de duas subclasse (Terrestre e Aquática), onde Terrestre tem atributos (anda na terra) e Aquática (nada na água), derivar de uma outra classe (sapo) era derivada.

3) Pode simular herança simples em C com muita facilidade, no entanto, isso depende de utilizar variáveis de tipo (typecast).  
Sim.

4) Antes de 2005, a linguagem Ada não era formalmente considerada orientada a objetos, apesar de possuir formas para definir tipos abstratos. Com a versão Ada 2005, incluiu mais conceitos usados na linguagem Java (orientado a objetos) e foi a primeira linguagem, juntamente a Objeto a ser padronizada internacionalmente.

5) Porque elas são simples e leves, de fácil manutenção. E tem o benefício da reutilização de código, já que a complexidade dos sistemas cada vez mais ampla, o tempo de

desenvolvimento iria aumentar muito mais, e  
tudo o que a matéria de linguagem de programação bus-  
ca, são linguagens claras de fácil entendimento, aprendizado  
e leitura.