Faculdade Dom Bosco de Porto Alegre
Bacharelado em Sistemas de Informação
Paradigmas de Linguagens – 2019/II
Prof. Filipo Mór – www.filipomor.com

Trabalho de Aula 5

13 de Setembro de 2019

1) Monte um quadro comparativo entre os paradigmas de programação procedural/imperativa e orientado a objetos. Cite exemplos de situações onde cada um seja mais adequado ou vantajoso.
2) Explique o Princípio de Substituição de Liskov. Cite exemplos.
3) Explique o padrão de projeto orientado a objetos S.O.L.I.D.
4) Qual a importância do princípio da abstração?
5) Faça uma comparação entre os conceitos de “herança” e “composição”. Descreva exemplos de cada.
6) Explique o conceito de “vinculação dinâmica” existente nas linguagens de programação orientadas a objetos.
7) Segundo SEBESTA, quais questões de projeto devem ser consideradas para linguagens de programação orientadas a objetos?
8) Compare o suporte a programação orientada a objetos nas linguagens Smaltalk, C++, Java e C#. Faça um quadro comparativo e comente os pontos principais.
9) Considere as funcionalidades básicas para um sistema de lista encadeada (cria lista, insere enodo, remove nodo, busca nodo). Crie dois códigos, um em uma linguagem imperativa/procedural, e o segundo em uma linguagem orientada a objetos, e faça uma análise das semelhanças e diferenças. Considere uma análise assimptótica do custo computacional, consumo de memória e tempo absoluto de execução.
10) Defina e dê ao menos um exemplo:
    a. Método virtual.
    b. Método abstrato.
    c. Classe abstrata.
    d. Classe aninhadora.
    e. Protocolo de mensagens de um objeto.
    f. Função amiga (friend) em C++.
    g. Boxing (encaixotamento).
    h. Como a versão pai de um método herdado sobrescrito em uma subclasse pode ser chamada em tal subclasse em C#?
    i. Uma situação de programação na qual a herança múltipla tenha uma vantagem significativa sobre as interfaces.
    j. Por que permitir que uma classe implemente múltiplas interfaces em Java e em C# não cria os mesmos problemas que a herança múltipla em C++ cria.