

Avaliação P1 – Programação Orientada à Objetos

Aluno: _____

Parte 1 – Prova Teórica

Avaliação a ser desenvolvida individualmente, sem consulta, a caneta. Respostas ilegíveis ou não passíveis de interpretação serão desconsideradas na correção.

1. (1.0 ponto) No que tange o Paradigma Orientado à Objetos (POO). Diferencie Classes de Instâncias. (min 3 linhas)

2. (1.0 ponto) Defina encapsulamento no POO ? Descreva os qualificadores de acesso disponíveis em C++ e como eles acabam interferindo no comportamento de atributos e métodos de uma classe. (min 3 linhas)

3. (1.5 pontos) Em relação à Herança. Explique o conceito e apresente detalhadamente pelo menos duas de suas vantagens. (min 5 linhas)

4. (1.5 pontos) Quanto a questão anterior: Apresente um exemplo de 3 níveis com menos 2 construtores para cada classe (sobrecarga, desconsiderando o *default*). No main, instancie 5 objetos de tipos variados, realizando a chamada do construtor. (escreva em C++, somente código)

5. (1.5 pontos) O que são classes abstratas e qual a sua relação com a Herança ? (min 3 linhas)

6. (1.5 pontos) Na Herança, a ordem de chamada dos construtores e destrutores é distinta. Justifique o porquê, apresentando também qual ordem as chamadas ocorrem.

7. (2.0 pontos) Verifique as afirmações abaixo em relação Programação Orientada a Objetos, colocando **V** ou **F**:
 - 1) () A programação orientada a objetos tem como principais objetivos reduzir a complexidade no desenvolvimento de software e aumentar sua produtividade.
 - 2) () Em C++, classes podem possuir quantos métodos construtores e destrutores forem necessários.
 - 3) () Variáveis membro de uma classe com o qualificador *static* são únicas e reservadas nos objetos, não sendo possíveis serem acessadas externamente, mesmo quando qualificadas como públicas.
 - 4) () Por meio do mecanismo de sobrecarga, vários métodos de uma mesma classe podem ter o mesmo nome, desde que suas listas de parâmetros sejam diferentes. No entanto, devem obrigatoriamente apresentar o mesmo tipo de retorno.
 - 5) () Na herança, as subclasses tornam-se mais específicas em relação à superclasse, podendo adicionar novos atributos e métodos que serão acessíveis ao longo de toda a estrutura hierárquica.
 - 6) () No uso de herança entre uma classe base e uma derivada, tanto construtores como destrutores são primeiro executados na classe base e depois na derivada.
 - 7) () Em uma Herança de 3 níveis hierárquicos: é possível possuir várias instâncias de objetos em qualquer nível. Uma forma de polimorfismo é utilizar um ponteiro do tipo do primeiro nível hierárquico, e utilizá-lo para representar qualquer instância em qualquer nível hierárquico das subclasses, exceto uma instância do primeiro nível.
 - 8) () Ao derivar uma classe usando herança do tipo **protected**, os membros **private** da classe base são todos **protected**.
 - 9) () No paradigma orientado à objetos a comunicação entre objetos se dá através de troca de mensagens.
 - 10) () Herança múltipla é descrita na orientação à objetos como um mecanismo onde uma classe derivada herda a partir de duas classes bases, e possui características de ambas. Porém, nenhuma Linguagem de Programação implementa esse mecanismo devido ao auto nível de ambiguidade.