

Colono

Discente		Nota
CCR	Programação OO - 2024/1	8,03
Data	17/06/2024	
Docente	Valdemar Lorenzon Junior	

Orientações Gerais:

Lêa todas as questões e orientações com atenção, a correta interpretação do enunciado das questões faz parte da avaliação. Não serão permitidas perguntas sobre as questões, em caso de dúvida ou interpretação para solução, faça complemento na sua resposta com a sua argumentação na própria resolução da prova. A avaliação é individual e sem consulta a nenhum tipo de dispositivo ou material. Assinale as respostas finais a caneta de cor preta ou azul na folha. Respostas a lápis não serão consideradas quando solicitado revisão de prova. Não devem ser destacadas as folhas que estiverem grampeadas.

De 1 a 5: 1 ponto cada, uma única alternativa correta:

1. Dada uma interface, como um exemplo a seguir, não havendo declarações de encapsulamento e sobre características de atributos podemos afirmar:

```
interface Interface {
    int valor = 0;
    void setValor ( int valor );
    int  getValor ( );
}
```

- ☒ Os métodos são todos do tipo públicos e abstratos e os atributos públicos e final;
- ☐ Os métodos são todos do tipo protegidos e abstratos e os atributos públicos e final;
- ☐ Os métodos são todos do tipo públicos e os atributos privados e final;
- ☐ Os métodos são todos do tipo privados e abstratos e os atributos privados e final.

2. No que tange a herança, uma classe pai, uma classe intermediária e uma classe concreta pode ser entendidas respectivamente como:

- ☒ Uma classe no topo ou no meio da hierarquia de classes, uma classe abstrata e uma classe que se pode instanciar objetos;
- ☐ Uma classe na base da hierarquia de classes, uma classe instanciável e uma classe abstrata;
- ☐ Uma classe que serve para extends de outras , uma classe instanciável e uma classe abstrata;
- ☐ Uma classe instanciável, uma classe abstrata e uma classe abstrata.

abstrata
↓

3. O polimorfismo de sobrecarga é conceituado como:

- ☒ () A capacidade que há de vários métodos possuírem o mesmo nome e assinaturas diferentes;
- ☐ () A possibilidade de sobreescrever um método herdado da classe pai numa classe descendente;
- ☐ () A possibilidade de sobreescrever um método na classe pai de um método criado em uma classe descendente;
- ☐ () A capacidade que há de vários métodos possuírem o mesmo nome e retorno e assinaturas diferentes.

4. Métodos getters e setters são aplicações do encapsulamento para:

- ☐ () Garantir a privacidade de atributos privados numa classe;
- ☒ () Garantir a proteção do estado interno do objeto;
- ☐ () Garantir a proteção de atributos e métodos públicos do objeto;
- ☐ () Proteger o objeto através de chamadas a apenas métodos públicos.

5. Em Java, o que melhor descreve o uso de polimorfismo em conjunto com generics?

- ☐ () Generics permitem que métodos possam ser sobrecarregados para aceitar diferentes tipos de parâmetros em tempo de execução, facilitando o polimorfismo dinâmico;
- ☒ () Polimorfismo com generics permite que classes ou métodos possam operar com qualquer tipo especificado em tempo de compilação, promovendo a reutilização de código sem a necessidade de conversões explícitas;
- X ☐ () Polimorfismo com generics impede o uso de herança e interfaces, forçando a utilização de tipos específicos apenas em tempo de compilação;
- X ☐ () Generics em Java garantem que apenas instâncias de uma superclasse específica podem ser adicionadas a uma coleção, excluindo suas subclasses.

6. V ou F (1 ponto cada) alternativa.

- (✓) ☒ () Uma referência de uma classe superior ^{down cast} pode ser instanciada com o construtor de uma classe descendente e um método descendente pode ser acessado por esta referência através do uso de casting;
- (F) ☐ () O polimorfismo de sobrecarga ^{return} (não) é possível utilizando-se apenas uma mudança de retorno de métodos de mesmo nome e assinatura;
- (✓) ☒ () O polimorfismo é a capacidade que o OO tem para que o objeto responda a uma mensagem de acordo com os dados externos ao objeto;
- (F) ☐ () O encapsulamento é um pilar da OO que garante uma resposta do objeto de forma mais adequada a sua especialização;
- (F) ☐ () Abstração, encapsulamento, herança e interfaces são os 4 pilares que garantem a OO.