Revisão prova II

Disciplina: Programação Orientada a Objetos

Professor: Clênio Silva

1. Quando utilizamos o encapsulamento em atributos no Java, precisamos criar alguns métodos para atribuir e recuperar o valor do atributo de um objeto. Com base nisso, como ficaria a definição dos métodos para atribuir e recuperar o valor de um atributo **cpf** que foi encapsulado? Escreva aqui os métodos e explique a diferença entre eles.

2. Qual das alternativas a seguir é utilizada como um modificador de acesso para encapsular atributos, isto é ocultar o acesso direto através de outras classes?
    a) abstract
    b) public
    c) static
    d) private

3. O que são construtores em Java, qual a sua finalidade, como podemos definir um construtor, e qual a diferença de um construtor vazio com um que recebe argumentos?
4. Crie um projeto com o nome **"revisaoProvaII".** Dentro do projeto crie um pacote com o nome **"exercicio4"**, lembrando que um pacote deve ser sempre criado dentro da pasta **src (source)**. Crie a classe a seguir dentro do pacote mencionado:

```
class Conta {
    String titular;
    int numero;
    double saldo;
```

- Para evitar que os atributos dessa classe se tornem visíveis para outra classe devemos encapsulá-los.
- Criei para cada atributo os métodos de atribuição e recuperação.
- Toda vez que formos instanciar um objeto da classe **Conta**, é necessário informar um titular logo no momento de instanciação, do contrario não será possível instanciar um objeto dessa classe. Para isso crie um construtor com passagem de argumento, o qual receberá o titular da conta.
- Também é uma exigência do sistema que ao criar uma conta apresente uma mensagem com o nome do titular, ex: **"Conta criada para Fulano"**.
- Crie uma classe com o nome Aplicação. Essa classe irá ser uma classe executável onde você deverá criar 5 contas distintas.

5 No java quando herdamos um método, podemos alterar o seu comportamento. Podemos reescrever esse método (**override**). Sendo assim vamos práticar um pouco de reescrita de métodos.

- Crie um novo pacote com o nome **"exercicio5"** dentro do projeto **"revisaoProvaII".**
- Dentro do pacote, crie uma classe Funcionario contendo os atributos (nome, cpf, salario e idFuncionario). Já sabemos com base nos atributos quais são os tipos deles, nesse caso iremos usar 3 tipos, String, int e double.
- Crie também uma classe Gerente contendo os atributos (senha, numeroFuncionarioGerenciados).
- Implemente uma herança entre as duas classes, lembre do principio (um fulano é um ciclano, portanto ciclano é a classe pai e fulano herda de ciclano).
- Dentro da classe Funcionario implemente o método **setBonificacao**(). Esse método atribuirá 10% ao salario do funcionario.
- Na classe Gerente implemente a reescrita do método setBonificacao alterando o comportamento para atribuir 15% ao salario.

6 O polimorfismo é a capacidade de um objeto poder ser referenciado de várias formas (cuidado, polimorfismo não quer dizer que o objeto fica se transformando, muito pelo contrário, um objeto nasce de um tipo e morre daquele tipo, o que pode mudar é a maneira como nos referimos a ele). Sendo assim vamos resolver um problema usando polimorfismo.

- Crie um novo pacote chamado "exercicio6" dentro do projeto **"revisaoProvaII".**
- Crie uma classe **Pessoa** com atributos nome e matricula.
- Crie uma classe uma Classe **Aluno** com os atributos media e quantidadeDisciplinas.
- Crie uma classe **Professor** com os atributos os atributos salario, quantidadeAulas e turno.
- Implemente a herança observando o principio de (um fulano é um ciclano, portanto ciclano é a classe pai e fulano herda de ciclano).
- Na classe pai implemente o método **getInformacoes()**, esse método irá imprimir os atributos da classe.
- Usando o conceito de reescrita de métodos (*override*), implemente nas classes filhas o método **getInformacoes()** alterando o seu comportamento para imprimir os atributos de cada classe junto com os herdados.
- Crie uma classe **MostraInfo**. Nessa classe iremos implementar um método que recebe como argumento uma **Pessoa**. Esse método deve ser **mostra(Pessoa pessoa)**, ele irá realizar uma chamada para o método **getInformacoes()** do argumento passado como referência na assinatura do método.
- Crie uma classe **AppPolimorfismo**. Essa classe será uma classe executável, ou seja terá o método **main**. Crie uma instância para **Aluno** e outra para **Professor**.

- Crie também uma instância da classe **MostraInfo** e usando o objeto criado, faça chamadas para o método **mostra()** passando o objeto criado para Professor e Aluno.