

Exercício Herança em Java

1. Crie um projeto no eclipse/netbeans chamado TESTAFUNCIONARIOS e:

a. Defina uma classe FUNCIONARIO para modelar os funcionários do banco, sabendo que todo funcionário possui nome e salário. Inclua os getters e setters dos atributos.

b. Crie uma classe para cada tipo específico de funcionário herdando da classe FUNCIONARIO. Suponha somente três tipos específicos de funcionários: GERENTES, TELEFONISTAS e SECRETÁRIAS. Os gerentes possuem um nome de usuário e uma senha para acessar o sistema do banco. As telefonistas possuem um código de estação de trabalho. As secretárias possuem um número de ramal.

c. Teste o funcionamento dos três tipos de funcionários criando um objeto de cada uma das classes: GERENTE, TELEFONISTA e SECRETARIA.

d. Suponha que todos os funcionários possuam uma bonificação de 10% do salário. Acrescente um método CALCULABONIFICACAO na classe FUNCIONARIO para calcular essa bonificação.

e. Altere a classe TESTAFUNCIONARIOS para imprimir a bonificação de cada funcionário além dos dados que já foram impressos. Depois, execute o teste novamente.

f. Suponha que os gerentes possuam uma bonificação maior que os outros funcionários. Reescreva o método CALCULABONIFICACAO na classe GERENTE. Depois, execute o teste novamente.

g. Defina na classe FUNCIONARIO um método MOSTRADADOS para imprimir seus dados na tela.

h. Reescreva o método MOSTRADADOS nas classes GERENTE, TELEFONISTA e SECRETARIA para acrescentar a impressão dos dados específicos de cada tipo de funcionário.

i. Modifique a classe TESTAFUNCIONARIO para utilizar o método MOSTRADADOS.

j. Defina um construtor com parâmetros na classe FUNCIONÁRIO, que receba como argumento todos os atributos da classe. Defina os demais construtores nas demais classes. Crie os objetos novamente na classe TESTAFUNCIONARIOS usando os respectivos construtores.

2 - Elabore uma classe ContaBancaria, com os seguintes detalhes:

Atributos:

cliente – String
numConta – int
saldo – double

Métodos:

sacar – o saldo não pode ficar negativo
depositar

Agora acrescente ao projeto duas classes herdadas de ContaBancaria: ContaPoupanca e ContaEspecial, com as seguintes características a mais:

- Classe ContaPoupanca
 - Atributo: diaRendimento – int
 - Método: calcularNovoSaldo – recebe a taxa de rendimento da poupança e atualiza o saldo
- Classe ContaEspecial
 - Atributo: limite – float

- Método: redefinição do método sacar, permitindo saldo negativo até o valor do limite

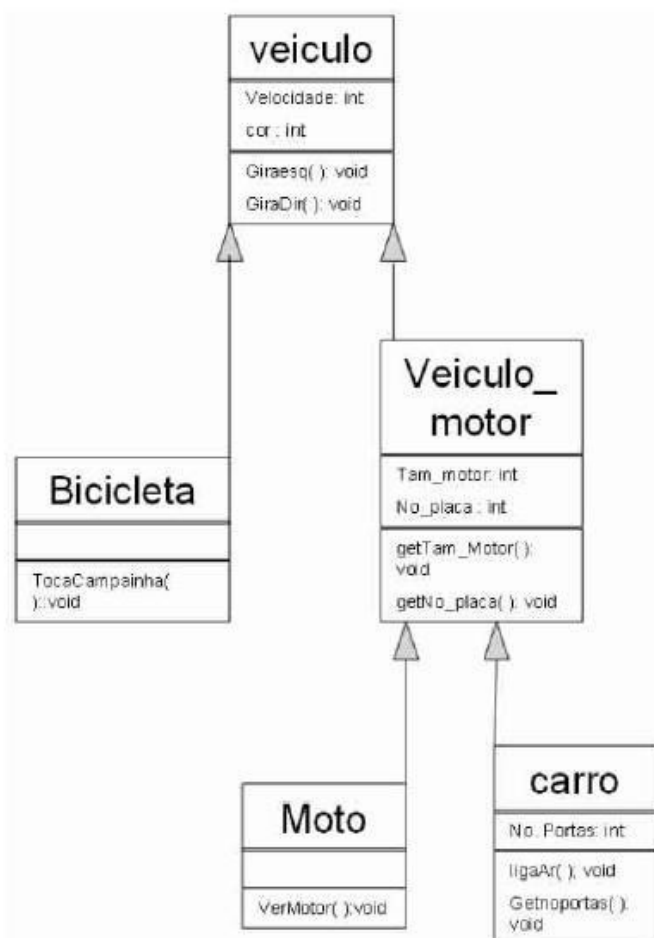
Após a implementação das classes acima, você deverá implementar uma classe appContas.java, contendo o método main, com as seguintes operações:

- Criar uma instância de ContaBancaria, uma de ContaPoupanca e uma de ContaEspecial
- Setar valores dos atributos de cada uma
- Depositar valores em cada conta
- Sacar valores de cada conta
- Mostrar os dados das contas
- Mostrar o novo saldo de cada Conta – (se for ContaPoupanca, calcular novo saldo a partir do rendimento antes de exibir o novo saldo)

3. Crie uma classe IMÓVEL, que possui um código, endereço e preço. Defina um construtor para a classe IMÓVEL passando esses três atributos como argumento. Crie uma classe NOVO que herda de IMÓVEL e possui um preço adicional. Crie métodos de acesso e impressão deste preço adicional. Crie também uma classe USADO, que herda de IMÓVEL e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto. Crie objetos do tipo Imóvel NOVO e USADO. Conforme o objeto, imprima o valor final do imóvel. **Use a técnica de sobrescrita (ou redefinição) de métodos.**

4. Implemente os diagramas de classes abaixo. Defina construtores para as superclasses. Atente-se às indicações de sobrescrita (ou redefinição) de métodos. Crie um método principal e instancie diferentes classes.

a)



b)

