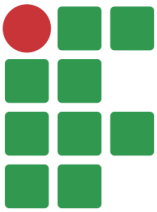


INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade II - 4 - Herança

Lista Única

Instruções

Responda às questões abaixo. Pode usar este próprio documento. Questões práticas devem ser anexadas separadamente.

As primeiras 5 questões valem 1. As questões 6 a 11 valem 2.

Questões

1. O que é herança e quais os benefícios que ela traz para o desenvolvimento de software?

   R: Um dos pilares da programação orientada a objetos, herança é útil na reutilização de software, pois podemos criar uma classe que contém dados e comportamentos de uma classe já existente e podemos aprimorá-la com novas capacidades.

2. Qual a diferença entre classe básica e classe derivada?

   R: Classe básica: A classe original, a classe que será herdada, pode ser chamada de superclasse.

   Classe derivada: Será a nova classe, aproveitará os membros da classe original, pode ser chamada de subclasse.

3. Qual a diferença entre uma classe básica direta e indireta?

   R: Direta: É quando uma classe herda diretamente essa classe básica.

   Indireta: É quando uma classe herda uma classe que herdou de uma outra classe básica em um determinado nível.

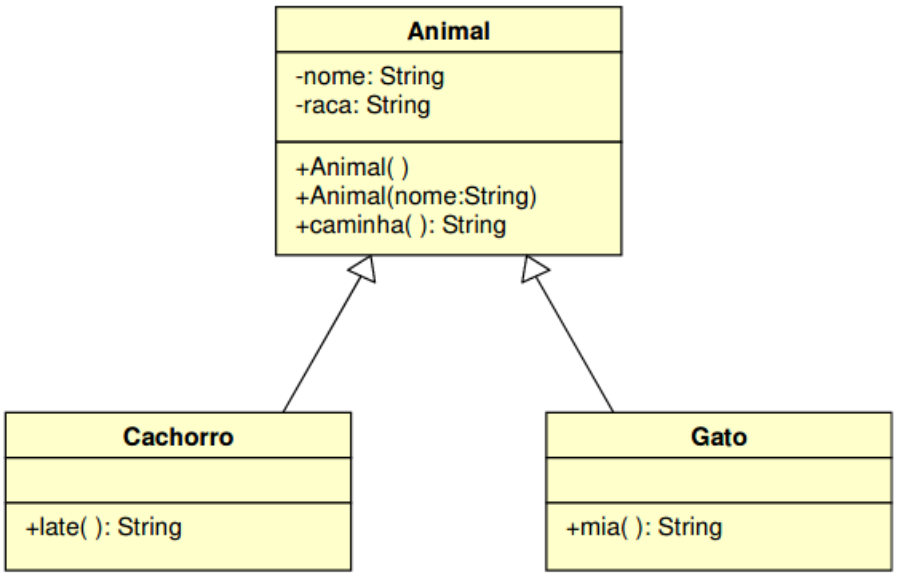
4. Em termos de relacionamento entre classes, qual a diferença entre a composição e a herança?

   R: Composição é um relacionamento do tipo tem um e a herança é um relacionamento do tipo é um.

5. Como o modificador protected se diferencia dos modificadores public e private? Em que contexto o protected deve ser usado?

   R: Ele é usado em classes básicas e oferece um nível intermediário de acesso. São semelhantes aos membros private, mas, quando a classe for herdada, os membros protected dessa classe básica podem ser acessados pela classe derivada, porém não acessíveis de fora da classe.

6. Implemente o diagrama de classes abaixo:



7. Crie uma classe Imovel que possui como atributos um endereço e um preço. Forneça métodos get e set para esses atributos. Em seguida, crie uma classe ImovelNovo que herda Imovel e possui um adicional no preço. Forneça um método get e um método set para esse atributo adicional. Crie também uma classe ImovelVelho que herda Imovel e possui um desconto no preço. Forneça um método get e um método set para esse atributo desconto. As classes ImovelNovo e ImovelVelho devem fornecer também um método getPreco que sobrescreve o método getPreco da classe básica Imovel e considera, respectivamente, o adicional ou desconto. Mostre que suas classes estão funcionando corretamente por meio de exemplos na função main.

8. Crie uma classe chamada Pessoa que tenha como atributo protected o nome da pessoa. Em seguida, crie duas outras classes chamadas PessoaFisica e PessoaJuridica que herdam da classe Pessoa. A classe PessoaFisica terá como atributos privados o CPF e o nome da pessoa, enquanto a classe PessoaJuridica terá como atributos privados o CNPJ, a razão social e o nome fantasia. Note que o atributo nome da PessoaFisica e nome fantasia da PessoaJuridica são herdados de Pessoa, isto é, é o atributo nome de Pessoa. Crie métodos get e set para todos os atributos das três classes. A classe básica Pessoa deve ter uma função amiga que sobrecarrega o operador << para imprimir o nome da pessoa na tela. Mostre que suas classes estão funcionando corretamente por meio de exemplos na função main.

9. Crie uma classe chamada Funcionario que herde da classe PessoaFisica da questão 8. Essa classe deverá ter como atributos privados a matrícula, o salário base do funcionário, a carga horária mensal (quantidade de horas mensais) e a quantidade de horas trabalhadas no mês. Além disso, a classe terá um método público chamado calculaSalarioBruto que não terá nenhum parâmetro e deverá ser capaz de calcular e retornar o salário bruto através da seguinte equação: salarioBase * quantidadeHorasTrabalhadas/cargaHorariaMensal. Por fim, crie métodos get e set para os atributos. Note que a quantidade de horas trabalhadas não poderá superar a carga horária mensal e nem ser inferior a 0. Garanta isso dentro da classe. A classe Funcionario deve ter uma função amiga que sobrecarrega o operador << para imprimir os dados do funcionário na tela. Mostre que suas classes estão funcionando corretamente por meio de exemplos na função main.

10. Crie uma classe chamada Cliente que herde da classe PessoaFisica da questão 8. Essa classe deverá ter atributos privados que armazenem um telefone e um endereço. Crie métodos get e set para esses atributos. A classe Cliente deve ter uma função amiga que sobrecarrega o operador << para imprimir os dados do cliente na tela. Mostre que suas classes estão funcionando corretamente por meio de exemplos na função main.

11. Crie uma classe chamada Empresa que herde da classe PessoaJuridica da questão 8. Essa classe deverá ter uma lista de funcionários e uma outra lista de clientes (pode ser array de tamanho fixo). Crie métodos para adicionar funcionários e clientes (não precisa se preocupar em excluir). Crie um método para imprimir os funcionários e outro para imprimir os clientes. Crie também um método chamado calcularFolhaDePagamento que deverá calcular o salário bruto de todos os funcionários e retornar o total a ser gasto com os funcionários. Mostre que suas classes estão funcionando corretamente por meio de exemplos na função main.