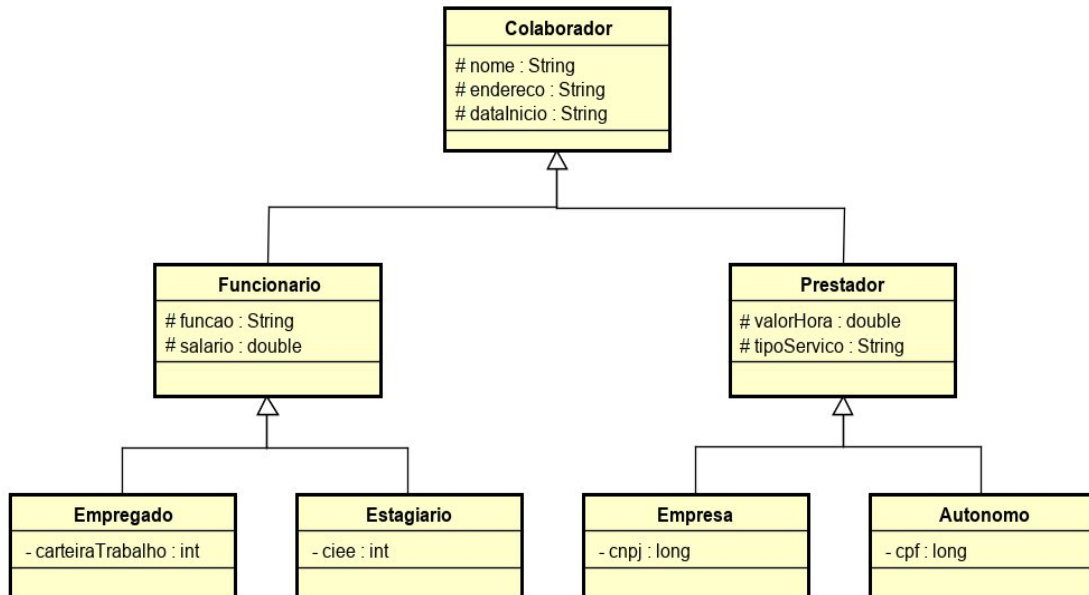


Lista de Exercícios 6

1. Desenvolva:

- Construa a classe **Data** com os seguintes métodos:
 - Construtores recebendo todos os parâmetros (dia, mês e ano).
 - Construtor recebendo como parâmetros os valores para mês e ano, e então chamar o construtor anterior passando o dia com o valor padrão: 1.
 - Construtor recebendo somente o valor do dia.
 - Chamar o construtor completo, passando por parâmetro o valor do dia recebido, além dos valores padrão: 01 para mês e 2019 para ano.
 - Construtor não recebendo parâmetros.
 - Chamar o construtor completo, passando os valores:
 - Dia = 01;
 - Mês = 01;
 - Ano = 2019.
 - Método **exibirData()** que retorna uma **String** com os valores de Data, no seguinte formato:
 - dia/mês/ano.
- Construa uma classe chamada **Feriado** que deve herdar da classe **Data**.
 - Um atributo chamado “descricao” que deve ser do tipo String;
 - Construtor que não recebe parâmetros
 - Chama o construtor-pai correspondente
 - Inicializa descricao com “Feriado”
 - Construtor recebendo uma String para a descrição
 - Chama construtor pai-correspondente;
 - Atribui valor recebido à variável-membro descricao
 - Construtor recebendo valores para dia, mês, ano e descrição
 - Chama construtor pai-correspondente
 - Atribui valor recebido à variável-membro descricao
 - Método **exibirData()** que retorna uma String com os valores de Data no seguinte formato:
 - dia/mês/ano – descrição.

2. Seguindo a modelagem, desenvolva o exemplo:



- Crie todos os métodos getters, setters e construtores para cada classe do modelo.
- Para cada classe, crie métodos toString que retornam uma variável do tipo String, a qual concatena todos os atributos da classe.
- Crie métodos para imprimir os atributos da classe.
- Crie uma classe de teste que irá instanciar todos os tipos de subclasses (empregado, estagiário, empresa e autonomo).

3. Crie uma classe Pessoa, que possui um nome (String), nascimento (String), rg (int) e cpf (int). Nesta mesma classe, implemente um método construtor que receba por parâmetro todos os atributos de pessoa. Também implemente um método que exiba as informações de uma pessoa. **Obs.: Não utilize o método toString() para exibir as informações de pessoa.**

Crie uma classe Aluno que é uma Pessoa. O aluno possui uma matrícula (int) e, como o Aluno é uma Pessoa, então também possui todos os atributos de pessoa. Implemente um método construtor em aluno que receba todos os atributos do aluno (nome, nascimento, rg, cpf e matrícula). Sendo que os dados nome, nascimento, rg e cpf devem ser passados para o construtor da classe Pessoa, e a matrícula deve ser inicializada no próprio construtor de Aluno. Ainda na classe aluno, implemente um método que imprima todas as informações do aluno.

Crie uma classe AlunoTeste onde deverá:

- ser criado um objeto do tipo pessoa, com suas informações;
- ser invocado o método para imprimir as informações do objeto pessoa;
- ser criado um objeto aluno, com suas informações; e
- ser invocado o método para imprimir as informações do objeto aluno.

4. Desenvolva um sistema para gerenciar diferentes veículos de uma empresa. Em conversa com o diretor da empresa, ele mencionou que os veículos podem ser carros motos e caminhões. O carro é identificado por uma marca, modelo, ano de fabricação e quantidade de portas. Todos estes dados devem ser inicializados no momento da criação do objeto. Além disso, deve ser possível calcular o IPVA para o carro. O cálculo do IPVA é dado por 3,5% do valor do carro. Também deve ser possível exibir todos os detalhes do carro, inclusive o valor do IPVA dele.

A moto é identificada por uma marca, modelo, ano de fabricação e tipo de guidão (esportivo, cross, customizado, entre outros). Todos estes dados devem ser inicializados no momento da criação da moto. Além disso, deve ser possível calcular o IPVA da moto que, diferentemente do carro, o cálculo do IPVA para a moto é de apenas 2% do valor da moto. A exibição de todos os detalhes da moto deve ser possível por meio de uma funcionalidade.

Por fim, há o caminhão que é identificado por uma marca, modelo, ano de fabricação e capacidade de carga. Todos estes dados devem ser inicializados no momento da criação do caminhão. Em relação ao cálculo do IPVA, deve-se considerar 1,5% do valor do caminhão. Também deve ser possível apresentar todos os dados do caminhão em tela.

Para realizar a instanciação dos objetos, bem como a invocação de seus métodos, utilize uma classe de testes.

Obs.: Note que as classes de carro, moto e caminhão possuem vários atributos e métodos comuns. Consequentemente, estes atributos e métodos podem ser implementados em uma classe genérica (pai) as classes citadas.

5. Você foi contratado para desenvolver um sistema de uma imobiliária. Em conversa com o gerente desta imobiliária ele lhe disse que a princípio a imobiliária trabalha apenas com dois imóveis: imóvel novo e imóvel usado.

O imóvel novo possui um valor base, valor adicional e endereço, que é constituído de um logradouro, número, bairro, cidade e unidade federativa. Além disso, o gerente mencionou que o sistema deve calcular o valor final do imóvel novo e exibi-lo em tela, além de todos os dados do imóvel (valor base, valor adicional e os dados do endereço). O cálculo para o imóvel novo é a soma do valor base e do valor adicional.

Por outro lado, também há o imóvel usado que é constituído do valor base, valor de desconto e endereço, sendo este último formado pelo logradouro, número, bairro, cidade e unidade federativa. Também deve ser possível calcular o valor final do imóvel usado e exibi-lo em tela. O cálculo para o imóvel usado é dado pela subtração do valor base e valor de desconto.

Para criar objetos e invocar os métodos a fim de testá-los, crie uma classe de testes.

6. Desenvolva um sistema para calcular os salários de diferentes funcionários de uma organização. Em conversa com o RH foi lhe informado que há três tipos de funcionários: administrativo, desenvolvedor e gerente.

O funcionário administrativo possui um nome, registro, salário base e um adicional. Todas estas informações devem ser inicializadas no momento da criação deste funcionário. Além disso, deve ser possível calcular o salário do funcionário administrativo, que somará o salário base adicional recebido por ele. Por fim, deve ser possível exibir todos os dados do funcionário administrativo, como nome, registro, salário base, adicional e salário final.

Por outro lado, o desenvolvedor possui um nome, registro, salário base, número de projetos concluídos e uma bonificação por projetos concluídos. Todas estas informações devem ser inicializadas no momento da criação do desenvolvedor. O cálculo do salário do desenvolvedor se dá pela soma do salário base com a multiplicação do número de projetos concluído com o valor que o desenvolvedor recebe por cada projeto concluído. Exemplo: se o salário base é de R\$10.000,00, o desenvolvedor ganha uma bonificação de R\$1.500,00 por projeto concluído e ele finalizou 2 projetos no mês, então o salário final deste desenvolvedor é de $10.000 + (1.500 * 2) = R\$13.000,00$. Também deve ser possível exibir todos os dados do desenvolvedor, como nome, registro, salário base, número de projetos concluídos, bonificação de projetos concluídos e salário final.

Por fim, há o gerente que possui um nome, registro, salário base, total de lucro da empresa e percentual de participação nos lucros da empresa. Todas estas informações devem ser inicializadas no momento da criação do gerente. O cálculo do salário do gerente é definido pela soma do salário base com o percentual que ele ganha sobre os lucros da empresa. Exemplo: se o salário base é de R\$10.000,00, a empresa obteve lucro de R\$500.000,00 e o percentual de participação do gerente é de 1%, então o salário final deste gerente é de $10.000 + (500.000 * 1\%) = R\$15.000,00$. Além disso, deve ser possível exibir todos os dados do desenvolvedor, como nome, registro, salário base, número de projetos concluídos, bonificação de projetos concluídos e salário final.