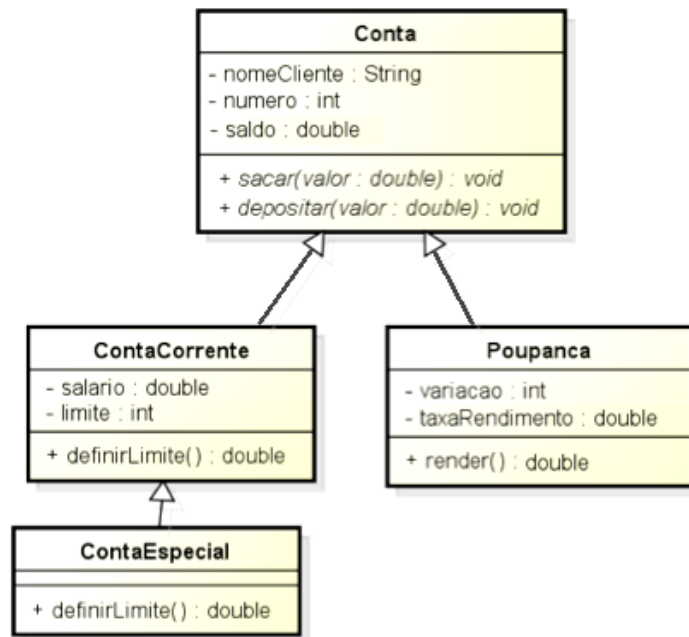


<p>Universidade Federal da Paraíba</p> <p>Centro de Informática</p>	<p>Laboratório de LP1</p> <p>Semestre: 2019.1</p> <p>Professor: Derzu Omaia</p>
---------------------------------------------------------------------	---------------------------------------------------------------------------------

### **ROTEIRO 3**

#### **ASSUNTO: POO – Herança e Polimorfismo e Exceções**

- 1) O que é polimorfismo? Dê um exemplo (com código) de um método polimórfico.
  
- 2) (a) Crie uma classe abstrata `IConta` com os métodos `void sacar(double valor)` e `void depositar(double valor)` sendo virtuais puros (abstratos).  
 (b) Crie uma classe `Conta` que herda de `IConta` e que contenha os atributos `nomeCliente`, `salarioMensal`, `numeroConta`, `saldo` e `limite`, e os métodos para obter e alterar esses atributos (métodos `get` e `set`). Além disso, essa classe possui as seguintes características:
  - Os valores dos atributos `nomeCliente`, `salarioMensal`, `numeroConta` e `saldo` são configurados no construtor da classe.
  - O método `void definirLimite()`, define o valor do atributo `limite` como 2 vezes o valor de `salarioMensal`.  
 (c) Crie uma classe `ContaEspecial` que herda da classe `Conta` e sobrescreve o método `definirLimite()` como 3 vezes o valor de `salarioMensal`.  
  
 (d) Crie um método `main` que cria um objeto da classe `Conta`, um objeto da classe `ContaEspecial` e invoca os métodos `sacar`, `depositar` e `definirLimite`.
  
- 3) Em um grande banco existe um sistema de gerenciamento de contas. Considere que o seguinte diagrama de classes é uma versão simplificada dele.



**Observações gerais:** Os atributos herdados devem ser `protected` os demais `private`.

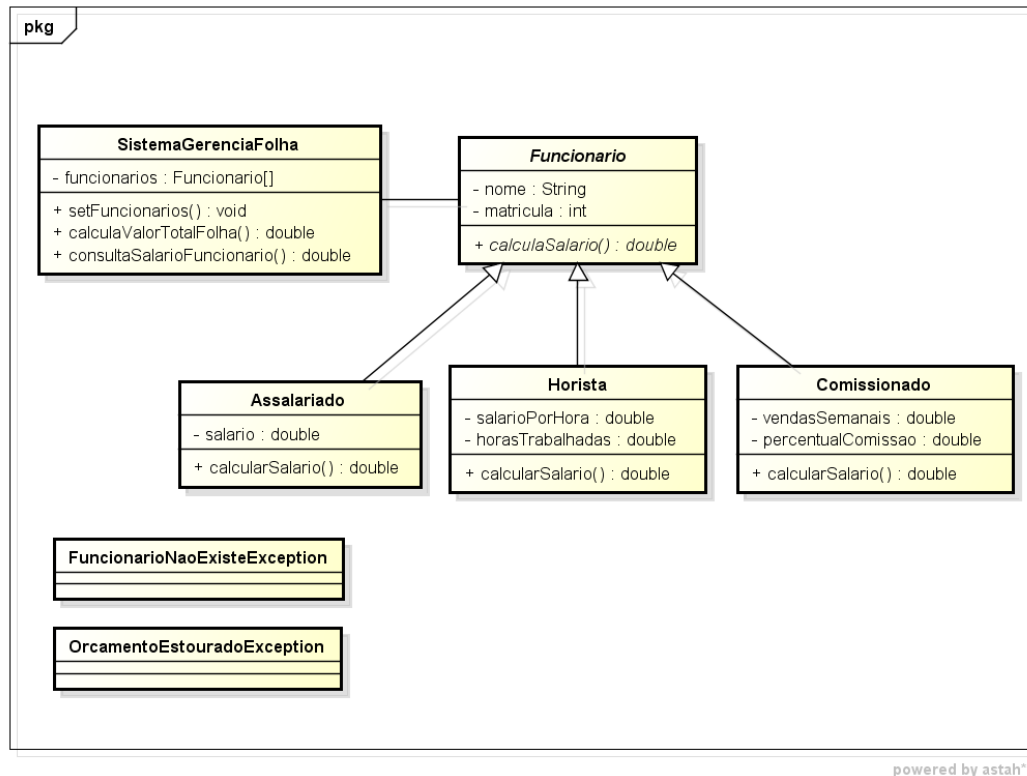
**(a)** Crie os métodos `get` e `set` de todos os atributos de todas as classes. Crie também os construtores de todas as classes, estes devem receber como parâmetro todos os atributos da sua classe, e, caso herdem, os atributos do pai também. Nos construtores os atributos devem ser inicializados e, quando possível, o construtor do pai deve ser chamado.

**(b)** Na classe `Conta` os métodos `sacar` e `depositar` devem alterar o valor do atributo `saldo` de forma apropriada. O método `sacar` deve possuir um `if` que exiba uma mensagem de saldo indisponível quando o valor a ser sacado é maior que o saldo disponível.

**(c)** Na classe `Poupanca`, o método `render` deve retornar o rendimento da poupança, baseado no `saldo` e na `taxaRendimento`. Se a `variação` da poupança for 51 o rendimento deve ser baseado apenas na taxa de rendimento, se for 1 a `taxaRendimento` deve ser acrescida de 0.5%.

**(d)** Na classes `ContaCorrente` e `ContaEspecial`, o método `definirLimite` deve retornar o duas vezes e quatro vezes, respectivamente, o salário do cliente.

4) Considere um sistema para calcular a folha de pagamento dos funcionários de uma empresa (conforme diagrama abaixo). A empresa possui três tipos de funcionários: os **Assalariados** que recebem salários fixos independentemente do número de horas trabalhadas, os **Horistas** que são pagos da mesma forma e recebem horas extras (1,5 vezes sua taxa de salário por hora) por todas as horas trabalhadas além das 40 horas normais e os **Comissionados** que recebem um salário base mais um percentual sobre suas vendas.



Nesse sistema usamos a classe abstrata **Funcionario** que possui **nome** e **matricula** (e seus respectivos métodos **get** e **set**) e um método abstrato **calcularSalario()** para representar o conceito geral de um funcionário. Também definimos as classes **Assalariado**, **Horista** e **Comissionado** que herdam de **Funcionario** e representam cada um dos três tipos de funcionários.

Em **SistemaGerenciaFolha** o método **consultaSalarioFuncionario** deve receber o nome do funcionário como parâmetro. Se o funcionário não existir **FuncionarioNaoExisteException** deve ser lançada. No método **calculaValorTotalFolha()** deve lançar uma exceção se o valor da folha for maior que o orçamento máximo. Defina o valor do orçamento máximo no construtor de **SistemaGerenciaFolha**.

- Implemente a classe abstrata **Funcionario**.
- Implemente as classes **Assalariado**, **Horista** e **Comissionado**.
- Implemente a classe **SistemaGerenciaFolha**.
- Implemente um método **main** para testar as classes criadas.

Para todas as questões, considere os atributos e funções definidos no diagrama de classes acima.

