

<p>Universidade Federal da Paraíba</p> <p>Centro de Informática</p> <p>Departamento de Informática</p>	<p>Linguagem de Programação I</p> <p>Semestre: 2019.2</p> <p>Professor: Tiago Maritan</p>
--	---

2ª Lista de Exercícios

Entrega: 17/03/2020

1) (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`.

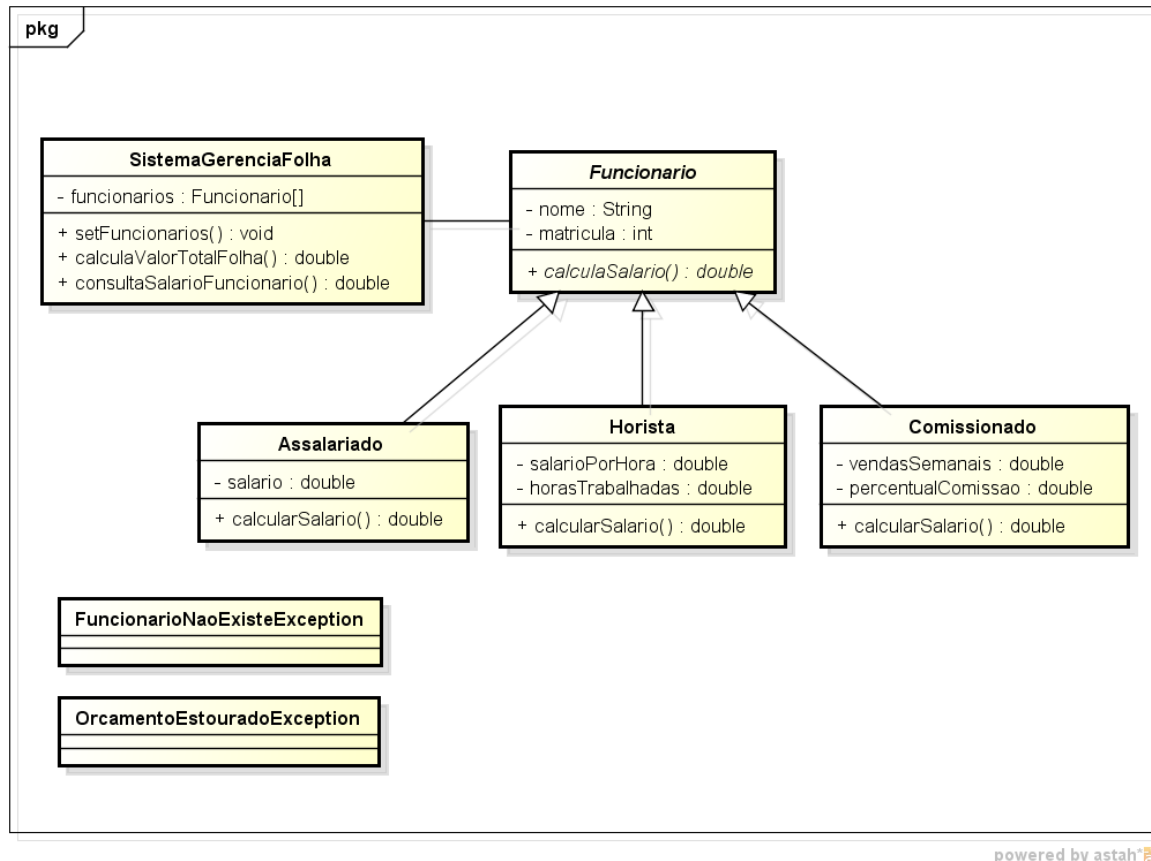
2) Considere o sistema para criar contas criado na questão 01.

(a) Implemente uma classe de exceção `SaldoNaoDisponivelException`. Essa classe deve herdar da classe `exception`.

(b) Implemente uma nova versão do método `sacar()` na classe `Conta` que deve lançar uma exceção do tipo `SaldoNaoDisponivelException`, quando o valor a ser sacado é maior que o saldo disponível.

(c) Crie um método `main`, que deve criar um objeto da classe `Conta`, chamar o método `sacar` (da classe `Conta`) e capturar a exceção `SaldoNaoDisponivelException`. Ao capturar a exceção, o programa deve imprimir uma mensagem informando o problema.

3) 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.



powered by astah

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.

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

Considere os atributos e métodos definidos no diagrama de classes acima.

4) Crie um classe **TestaValidaNumero** com um metodo "`void validaNumero(int num)`", que lança as seguintes exceções de acordo com o valor do parâmetro **num**:

- menor ou igual a 0: lança a exceção **ValorAbaixoException**;
- maior que 100 e menor que 1000: lança a exceção **ValorAcimaException**;
- maior ou igual a 1000: lança a exceção **ValorMuitoAcimaException**;

Escreva o código das classes de exceção **ValorAbaixoException**, **ValorAcimaException** e **ValorMuitoAcimaException**

6) Crie um programa em C++ que represente algumas informações para gerenciamento de um restaurante. Esse programa deve conter as classes `Pedido`, `MesaDeRestaurante` e `RestauranteCaseiro`. Algumas recomendações sobre a criação dessas classes são:

- A classe `Pedido` representa o pedido de um único item do cardápio, e contém os seguintes atributos: o número, a descrição, a quantidade comprada de um item e o preço do item;
- A classe `MesaDeRestaurante` deve conter atributos para representar um conjunto de instâncias da classe `Pedido`, representando os pedidos feitos na mesa. Ela deve conter também um método `adicionaAoPedido()` que adiciona um novo pedido ao conjunto de pedidos feitos; o método `zeraPedidos()` que cancela todos os pedidos feitos, isto é, faz com que a quantidade de pedidos seja zero para cada item; e o método `calculaTotal()`, que calcula o total a ser pago por aquela mesa.
- A classe `RestauranteCaseiro` deve ter várias atributos que são instâncias da classe `MesaDeRestaurante`, para representar suas mesas separadamente.
- A classe `RestauranteCaseiro` também deve ter um método `adicionaAoPedido()` que adicionará uma quantidade a um item de uma mesa. Esse método deverá chamar o método `adicionaAoPedido()` da mesa à qual o pedido está sendo adicionado.
- A classe `RestauranteCaseiro` também deve conter um método `calculaTotalRestaurante()` que faz o cálculo do total arrecado no restaurante, somando os pedidos realizados em todas as mesas.

Obs: Você pode usar arrays e criar um número predeterminado e imutável de instâncias de `Pedidos` e de `MesaDeRestaurante`.

5) Implemente um programa que faz a cópia do conteúdo de um arquivo (Ex: "teste.txt") para outro (Ex: "teste\_bkp.txt"), usando as classes e métodos vistos em sala de aula.