

# Programação Orientada à Objetos

4<sup>o</sup> Sem. ADS – FATEC Campinas

Prof. Alexandre Ferreira

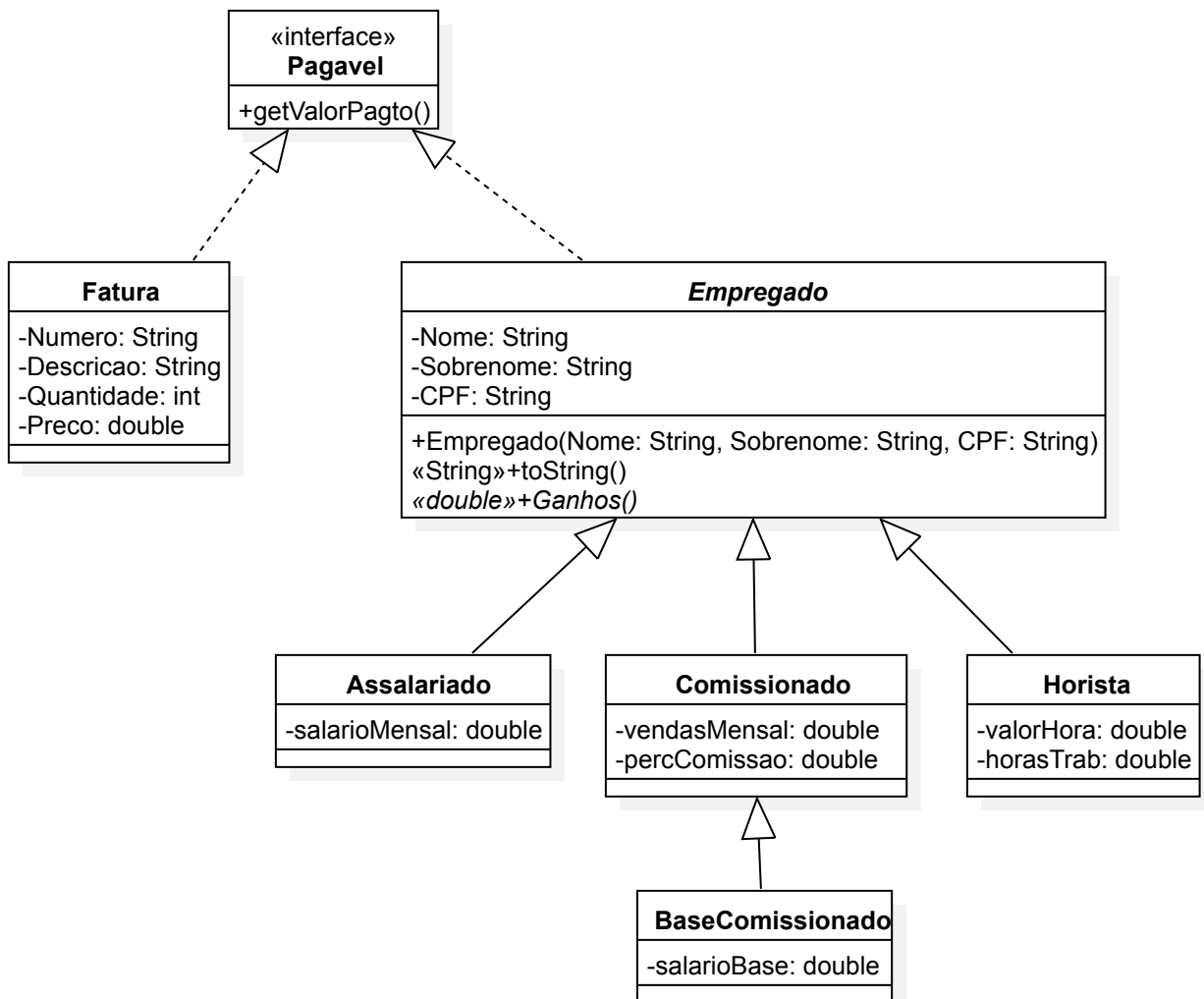
2<sup>a</sup> prova - 01/12/2020

**Instruções:** Os exercícios devem ser implementados nas linguagens Java e Python. A entrega da prova via MS Teams deve ser feita até dia 05/12/2020 às 23:59. A entrega deve conter os arquivos de forma separada. A prova em dupla. Boa prova!

1. (5,0 pontos) Considere o seguinte cenário:

*Uma empresa faz pagamentos mensais aos seus funcionários. Os funcionários são de quatro tipos: os assalariados recebem um salário mensal fixo, independente do número de horas trabalhadas; os funcionários remunerados por hora recebem a hora trabalhada e horas extras (isto é, 1,5 vez o valor da hora) por todas as horas trabalhadas que ultrapassarem 160 horas; os funcionários comissionados recebem uma percentagem de suas vendas; e os funcionários assalariados com comissão recebem um salário-base mais uma percentagem de suas vendas. Para o período de pagamento em curso, a empresa decidiu recompensar os funcionários assalariados com comissão, adicionando 10% aos seus salários-base. A empresa quer escrever um algoritmo em Java que efetue seus cálculos de folha de pagamento de forma polimórfica.*

O diagrama de classe em UML abaixo mostra a hierarquia de herança do algoritmo polimórfico de folha de pagamento de funcionário. Todas as classes devem implementar os métodos *get* e *set* para seus respectivos atributos além de sobrescrever os métodos *toString* e *Ganhos* apropriadamente.



Faça o que se pede:

- a Implemente o cenário descrito acima com um exemplo na classe de execução. Crie um array com diversos funcionários diferentes e calcule a folha de pagamento durante 1 ano. Adicione também algumas faturas nos pagamentos.
  - b Inclua o atributo `dataNascimento` na classe `Empregado`. Considerando que o pagamento é processado uma vez por mês, crie um array para armazenar diversos funcionários. Em um loop, calcule a folha de pagamento para cada objeto e acrescente um bônus de R\$ 200,00 ao valor da folha de pagamento da pessoa, se o mês corrente for aquele de seu aniversário.
2. **(3,0 pontos)** Escreva um programa em Python que simule um ecossistema contendo apenas dois tipos de criaturas: **ursos** e **peixes**. Ambos os animais possuem o atributo **genero**. O ecossistema consiste em um rio representado por uma lista de uma dimensão onde cada posição contém um objeto da classe `Animal`, que pode ser `Urso`, um `Peixe` ou `Null`.

Em cada instante de tempo, peixes e ursos tentam se mover aleatoriamente para as células adjacentes ou permanecem onde estão. Se dois animais do mesmo tipo tentam se mover para o mesmo lugar e são de generos diferentes, eles permanecem onde estão e uma nova instância do animal em questão é adicionada aleatoriamente na lista no lugar de uma célula nula. Caso contrário, eles permanecem onde estão mas nenhum novo objeto é criado. Se um peixe e um urso colidem (independente do genero), então o peixe morre, isto é, sua posição se torna nula.

Mostre na tela a posição dos animais no rio em cada instante de tempo. Crie um exemplo com, pelo menos, 10 instantes de tempo diferentes.
3. **(2,0 pontos)** Um usuário da lista Python Brasil enviou um email com o título “Herança múltipla: você já precisou disso em Python?”. Diversas respostas foram dadas. Elabore um email de resposta direcionada ao grupo Python Brasil em que você contribua **justificando o uso de herança múltipla**.