Desafios

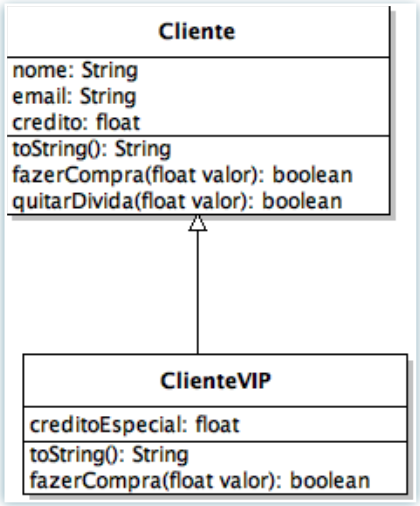
Herança e Polimorfismo

Fundamentos de JAVA

Portal Professor Isidro

Clientes da Loja

As Lojas P&I possuem agora, além do modelo Pré-Pago de compras para seus clientes comuns, uma nova modalidade: Clientes VIP. Independente do tipo de cliente, sempre há um crédito disponível na loja que permite com que cada um faça suas compras. Obviamente o cliente que estiver com seu saldo zerado não poderá fazer mais compras. Caso queira continuar comprando, o cliente pode "quitar suas dívidas" adicionando valores ao seu saldo remanescente (tipo uma conta de celular mesmo). Porém para o Cliente VIP há um saldo extra (o famoso "fiado" do interior). Mesmo que não haja saldo, porém havendo limite, o cliente VIP pode efetuar suas compras. Observe o Modelo abaixo.

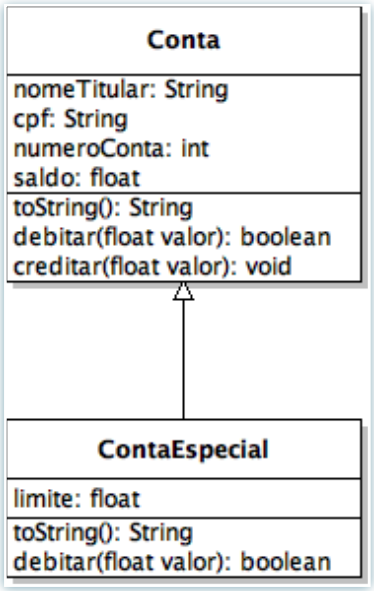


Implemente as classes descritas e uma aplicação exemplo para testar as compras tanto dos clientes normais quanto dos clientes VIP.

PI-Bank

O PI-Bank é um banco via Internet que oferece operações de depósito e saque de valores para as contas de seus clientes e utilizem os valores administrados em serviços de jogos online tais como STEAM, Nuuvem, entre outros.

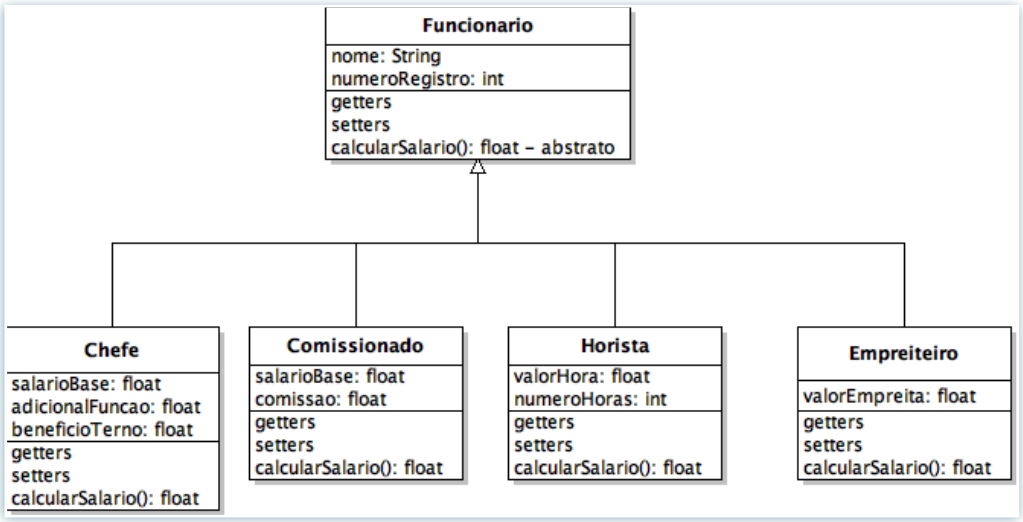
Dependendo da demanda (volume de movimentação do Conta, saldo disponível, entre outros), uma Conta pode ser considerada ESPECIAL e, por isso, possui um limite de crédito adicional, o que o permite ter seu saldo negativo (desde que haja limite na conta), exatamente como mostra o modelo abaixo.



Desse modo, implemente as classes java descritas e também uma aplicação PIBankTeste para que você possa validar suas resoluções.

Folha de Pagamento

A empresa ACME de Construção de Silos Anti Nucleares para o Apocalipse Zumbi está reestruturando seu sistema de folha de pagamentos e, por conta disso, reorganizou o organograma empresarial. Agora são 4 categorias de funcionários (Chefe, Trabalhador Horista, Comissionado - para vendas e Empreiteiro), nos quais há regras distintas para cálculo do salário.



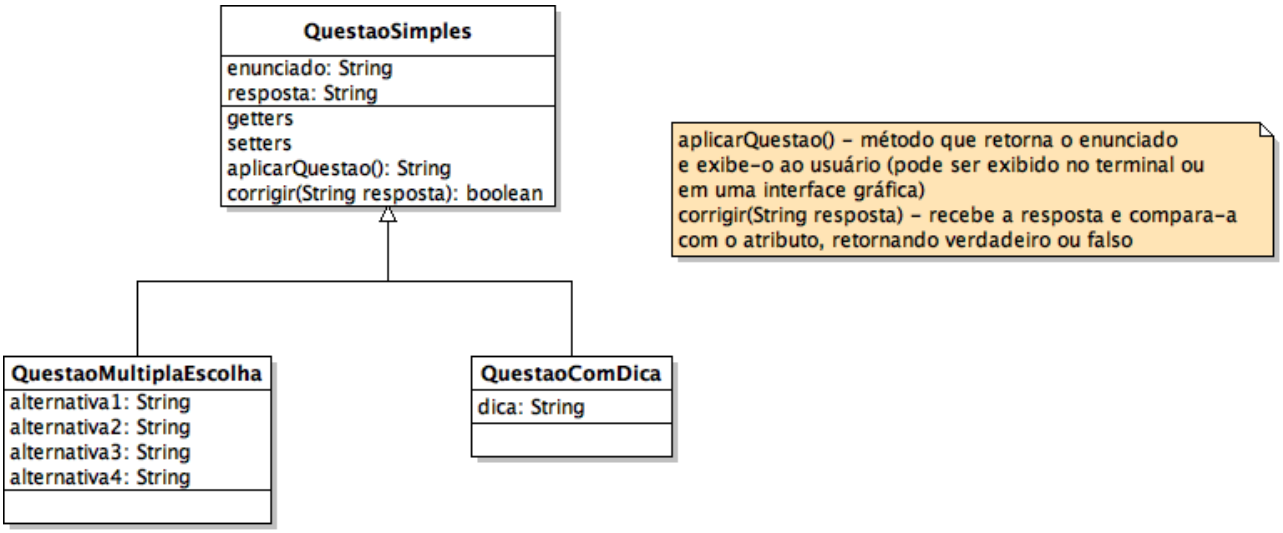
A regra para cálculo do salário é simples e rápida:

> Chefe: Salario Base + SalarioBase * adicional Funcao/ 100 + beneficio Terno
>
> Comissionado = salario Base + salarioBase * comissao / 100;
>
> Horista = valor Hora * numero Horas
>
> Empreiteiro = valor Empreita

Desse modo, implemente o modelo proposto, bem como uma aplicação que gere os Recibos de Pagamento de cada funcionário. Lembre-se: a lista de funcionários deve ser única!

UniversiPI

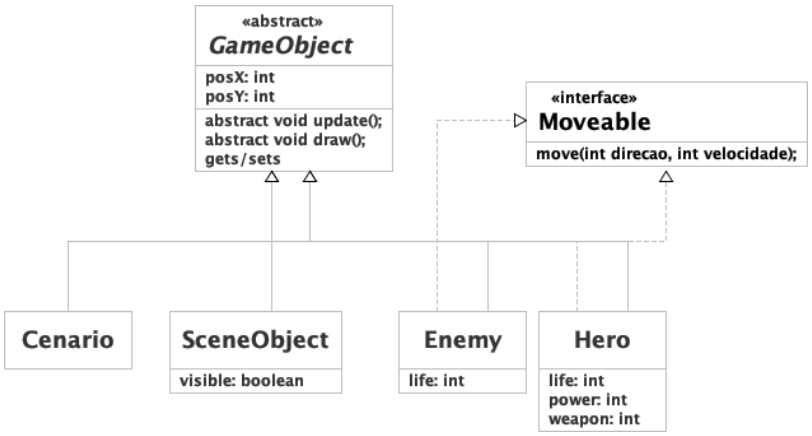
A UniversiPI, Universidade Tecnológica do Professor Isidro decidiu ampliar a forma de avaliação de seus alunos, criando um novo sistema de correção automática de simulados. O aluno recebe uma questão, a responde e o próprio sistema já corrige em função de uma resposta previamente cadastrada. Como é uma universidade Tecnológica, sua base de correção é binária: ou está certo, ou está errado. Existem 3 tipos de questões para um simulado da UniversiTS: Questão Simples, Questão com Dica ou Questão Múltipla Escolha. Exatamente como o modelo abaixo.



Por fim, crie as classes, implemente os métodos e crie um aplicativo AplicaProva que cria uma lista de questões, aplica-as e mostra a nota final do usuário.

Ultimate Game

O Jogo Ultimate Game é um jogo 2D super bacana que tem 1 única fase infinita (quase um - argh - Minecraft). Entretanto ele está só no conceito. Precisa ser implementado. Dessa forma, o diagrama de classes que representa os objetos do jogo está descrito a seguir:



A Classe abstrata GameObject representa todos os objetos do jogo (Cenário, objetos de cena, herói e inimigos). Cada classe-filha tem suas particularidades, entretanto todos devem implementar os métodos *update* (que faz a atualização das regras do jogo) e *draw* (que desenha o objeto). Os objetos Enemy e Hero implementam a interface Moveable que tem o método move (que indica a direção que ele vai se mover e a velocidade). Ex: direcao 0 significa que ele não sai do lugar, direção 1 ele vai para a direita, direção -1 vai para a esquerda, direção 2 vai para cima e direção -2 vai para baixo. Dessa forma, de acordo com a direção, a coordenada (x ou y) muda.

O que isso muda? Todos os objetos que implementam a interface precisam, em seus métodos *update* chamar o método *move* obrigatoriamente.

Claro que, por ser um teste, você só precisa implementar as classes e emitir mensagens nos métodos.

Faça uma aplicação **Jogo** que instancie vários desses objetos e invoque os métodos UPDATE e DRAW de cada um.