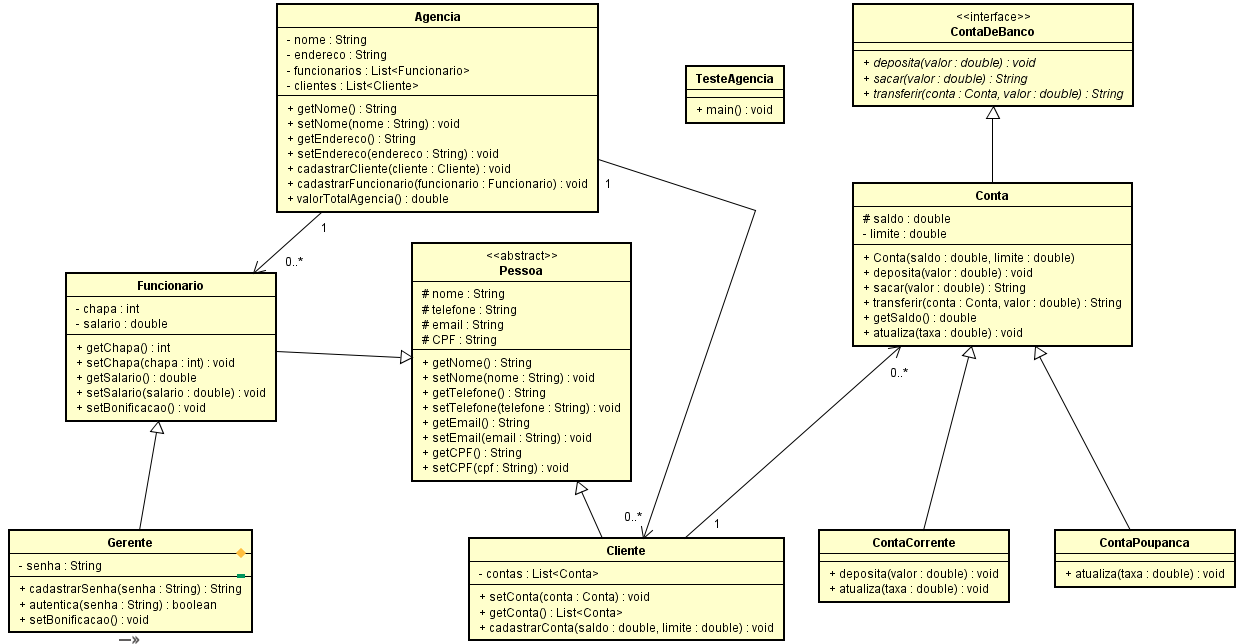
Utilize o diagrama de classes abaixo para desenvolver as questões:



1. (1,00). Criar a classe **Pessoa** conforme a imagem.
   1. A classe deve ser abstrata.
   2. Os atributos devem ser protegidos (protected).
   3. Fazer os métodos get e set correspondente aos atributos.
2. (1,00). Criar a classe **Funcionario** conforme imagem.
   1. Funcionário é um tipo de Pessoa.
   2. Os atributos são privados e os métodos públicos (conforme imagem).
   3. O **método construtor** da classe deve criar um número randômico de 0 a 1000 para o atributo chapa.
   4. O método **setSalario** só atribui valor ao atributo salário se o valor for maior que zero.
   5. Todo fim de ano o funcionário do banco tem um acréscimo de 10% de bonificação no salário, e para isso desenvolva o **método setBonificacao** na classe.
3. (1,00). Criar a classe **Gerente** conforme imagem.
   1. Gerente é um tipo de funcionário.
   2. Criar um **método cadastrarSenha** que recebe uma string alfa-numérica e retorna a mensagem “senha cadastrada com sucesso” caso a senha seja maior que 5 caracteres ou a mensagem “Senha deve conter no mínimo 5 caracteres” (só atribuir o valor ao atributo senha se satisfazer a condição).
   3. Criar um **método autentica** que verifica se a senha recebida é igual a senha do atributo, retorna true caso a senha seja a mesma e false caso a senha seja diferente.
   4. Sobrescreva o **método setBonificacao** para quando executado recebe um acréscimo de 20% de bonificação no salário.
4. (1,00). Crie uma interface chamada “**ContaDeBanco**”, que deve conter os métodos: deposita, sacar e transferir, conforme imagem.
5. (1,00). Criar a classe **Conta** conforme imagem.
   1. Definir quais são atributos privados e quais são protegidos, conforme imagem.
   2. Criar um **método construtor** que recebe: saldo, limite.
   3. Implementar o método **deposita**, que deve adicionar o valor passado ao saldo da conta já existente.
   4. Implementar o **método sacar** que deve retirar o valor do saldo somente quando o valor passado for menor que a soma do (saldo + limite) da conta e retornar a mensagem “OK” quando o valor for sacado e “Valor indisponível” quando não for possível realizar o saque.
   5. Implementar o **método transferir**, que deve sacar o dinheiro da conta atual e depositar o dinheiro na conta passada como parâmetro. Retorne a mensagem “OK”, caso a transferência tenha sido realizada com sucesso.
   6. Criar o **método atualiza** que recebe uma taxa e atualiza essa conta de acordo com a taxa fornecida, onde no saldo é adicionado essa taxa. (**fórmula: saldo = saldo + (saldo \* taxa)**)
6. (1,00). Crie uma classe **Cliente** conforme imagem.
   1. Cliente é um tipo de Pessoa.
   2. Criar o método **setConta** que recebe um objeto conta e adiciona na lista de contas do cliente.
   3. Criar o **método cadastrarConta**, conforme imagem que recebe como parâmetros saldo e limite, para criar a conta e vincular à conta ao cliente.
7. (1,00). Criar a classe **ContaCorrente** conforme imagem.
   1. ContaCorrente é um tipo de Conta.
   2. Deve sobrescrever o método “**deposita**”, onde todo depósito bancário de uma conta corrente deve descontar a taxa bancária de 10 centavos.
   3. Deve sobrescrever o método “**atualiza**”, onde o valor atualizado é o dobro da taxa passada (seguir a fórmula da questão 5.f, levando em consideração que a taxa é em dobro).
8. (1,00). Criar a classe **ContaPoupanca** conforme imagem.
   1. ContaPoupanca é um tipo de Conta.
   2. Deve sobrescrever o método “atualiza”, onde o valor atualizado é o triplo da taxa passada (seguir a fórmula da questão 5.f, levando em consideração que a taxa é o triplo agora).
9. (1,00). Criar a classe **Agencia** conforme imagem.
   1. Implementar o **método** **cadastrarCliente** que recebe um cliente e insere na lista de clientes da agência.
   2. Implementar o **método cadastrarFuncionario** que recebe um funcionário e insere na lista de funcionários da agência.
   3. Criar o **método valorTotalAgencia** que retorna a soma do saldo de todas as contas dos clientes da agência.
10. (1,00). Criar a classe TesteAgencia.
    1. Criar o método main e realizar o teste do método “valorTotalAgencia” da classe Agencia e para tal, crie os objetos necessários.

**Boa sorte!**