**Universidade Federal de Alagoas - Campus Arapiraca**

**Curso de Ciência da Computação**

**Laboratório de Programação 2**

**Programação Orientada a Objetos**

**Prof. Thiago Bruno Melo de Sales**

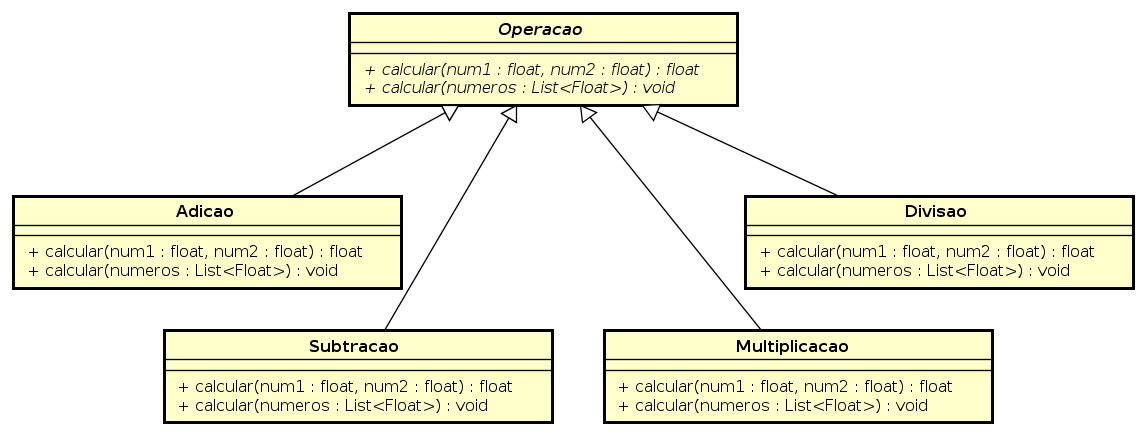
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lista 3 - Polimorfismo**

**1.** Uma determinada empresa deseja implementar um sistema de autenticação para os seus funcionários, que possui como características relevantes nome, cpf, e-mail, registro único, data de nascimento, função e senha para efetuar login. O gerente da empresa, por questão de facilidade de acesso ao sistema, deseja que os funcionários possam realizar login tanto através do e-mail e senha quanto através do registro e senha. Implemente a classe Funcionario e os métodos de *realizarLogin* com as devidas particularidades exigidas pelo gerente da empresa.

**2.** Considerando a empresa do exemplo anterior, o gerente agora deseja realizar uma nova modificação no sistema. Dependendo do tipo de funcionário, o acesso ao sistema ocorre de forma diferenciada, ou seja, se a função do funcionário for “Gerente” ele terá acesso a um determinado conjunto de funcionalidades, da mesma forma, se o campo de função estiver preenchido com “Atendente” as suas funcionalidades no sistema serão restritas. Sendo assim, para o teste inicial o gerente gostaria que um funcionário ao realizar login dependendo da sua função fosse exibida uma mensagem “nome\_do\_funcionario é gerente!” ou “nome\_do\_funcionario é atendente!”.

**3. Considerando o diagrama de classes apresentado abaixo:**

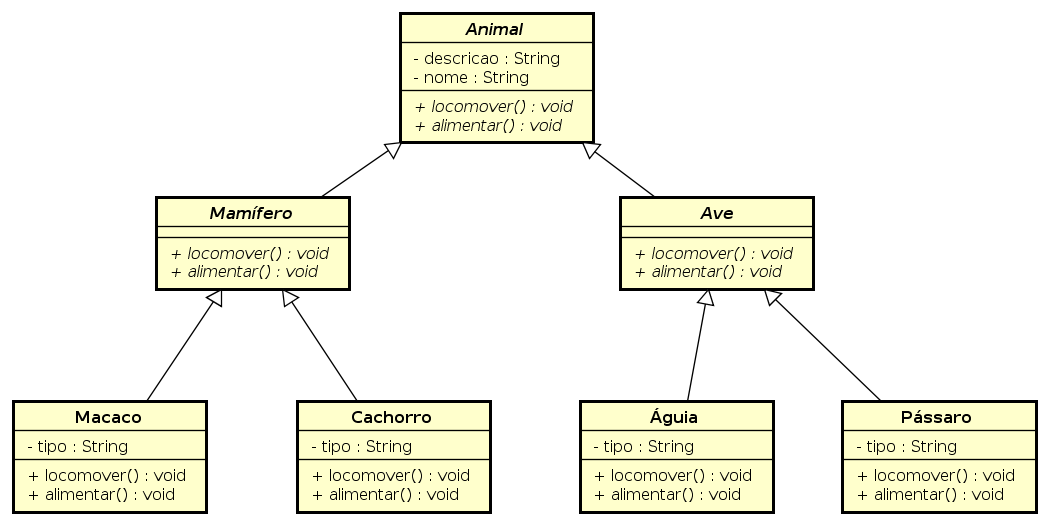


**a)** Implemente o diagrama de classes, incluindo todos os métodos getters e setters.

**b)** Para cada uma das operações, deve-se ler uma sequência de números, até que o número zero seja inserido. O número zero é um indicador de parada e não deve ser considerado para as operações. Se o usuário tiver digitado 2 números, deve ser invocado o método com dois argumentos, assim como se o usuário digitar mais de 2 números deve ser invocado o método que possui uma lista como argumento.

**c)** Modifique as operações para corresponder à operação de cada subtipo.

**4.** No diagrama exemplificado abaixo, vamos modelar o reino animal. Podemos perceber que as classes *Animal*, *Mamífero* e *Ave* são classes abstratas, assim como seus dois métodos locomover() e alimentar(), dessa forma, não é possível termos uma instância destas classes, apenas dos seus subtipos concretos que são Macaco, Cachorro, Águia e Pássaro.



**a)** Implemente o diagrama de classes apresentando, incluindo todos os métodos getters e setters necessários.

**b)** Os métodos getters de *Animal*, *Mamífero* e *Ave* devem ser implementados com as seguintes mensagem “nome\_animal é um animal”, “nome\_animal é um mamífero” e “nome\_animal é uma ave”, respectivamente.

**c)** Sobrescreva os métodos locomover e alimentar para atender as particularidades de cada animal.

**5.** Uma empresa necessita realizar cálculos de Folha de Pagamento (FP) baseados nos tipos de empregados que possui. Implemente e construa um diagrama de classes do sistema de FP:

Utilizar uma superclasse <abstract> *Empregado*. Esta classe possui: *NOME*, *CPF*, *RG* entre outros atributos dependendo do tipo de empregado. E possui um método abstrato chamado *calcularGanho*. São tipos de Empregado:

* **Chefe**: empregado pago com um salário fixo semanal independente do número de horas trabalhadas. O retorno do método ganho desta classe é o próprio valor atribuído ao atributo *salarioMensal***.**
* **TrabalhadorComissionado**: empregado pago com um salário fixo mais uma porcentagem sobre as vendas. O retorno do método ganho desta classe é o resultado da operação sobre os valores atribuídos aos atributos desta classe da seguinte forma: *salario* + (*comissao* \* *quantidade*).
* **TrabalhadorPecaProduzida**: empregado pago pelo número de itens produzidos. O retorno do método ganho desta classe é o resultado da operação sobre os valores atribuídos aos atributos desta classe da seguinte forma: (*remuneracão* \* *quantidade*).
* **TralhadorHora**: empregado que é pago por hora e que recebe um adicional por hora extra trabalhada. O retorno do método ganho desta classe é o resultado da operação sobre os valores atribuídos aos atributos desta classe da seguinte forma: *salarioHora* \* *horas*.

Segue diagrama detalhado para a resolução desta questão.

