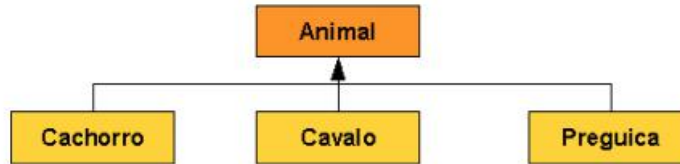


## Lista de Exercícios sobre Polimorfismo

1- Crie uma hierarquia de classes conforme abaixo com os seguintes atributos e comportamentos (observe a tabela), utilize os seus conhecimentos e distribua as características de forma que tudo o que for comum a todos os animais fique na classe Animal:

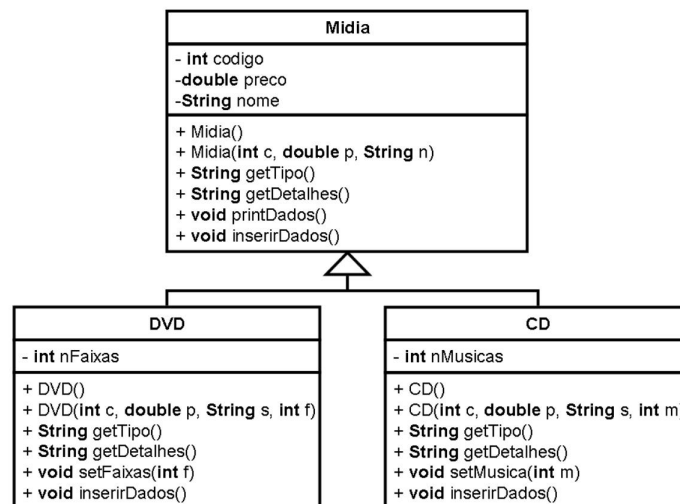


Cachorro	Cavalo	Preguica
Possui Nome	Possui Nome	Possui Nome
Possui Idade	Possui Idade	Possui Idade
Deve emitir som	Deve emitir som	Deve emitir som
Deve correr	Deve correr	Deve subir em árvores

2- Implemente um programa que crie os 3 tipos de animais definidos no exercício anterior e invoque o método que emite o som de cada um de forma polimórfica, isto é, independente do tipo de animal.

Fonte: Centro Universitário Estácio - Prof. Newton Gomes. Disponível em <https://sites.google.com/site/profnewtonjava/exercicioheranca>

2) Uma loja que vende CD e DVDS deseja construir um cadastro com dos seus produtos. Para tanto, foi elaborado o diagrama de classes dado na Figura 1.1.

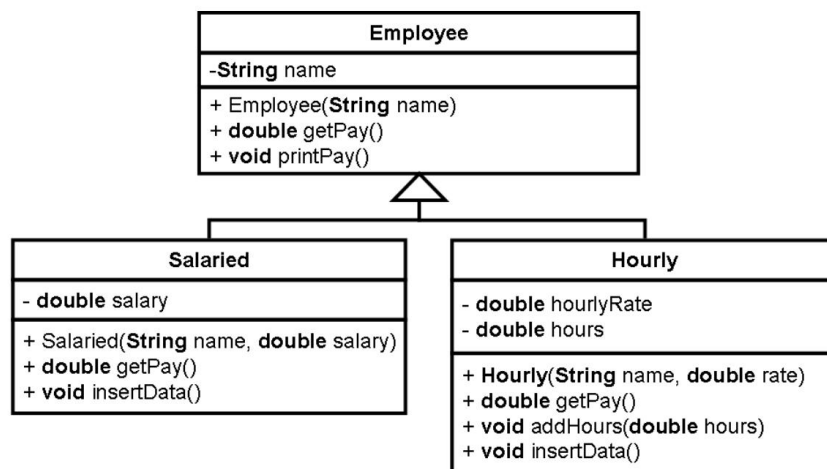


A Tabela a seguir fornece uma descrição dos métodos que deverão ser elaborados para cada uma das classes.

Método	Descrição
getTipo()	Retorna uma String com o nome da classe.
getDetalhes()	Retorna uma String com as informações contidas nos campos.
printDados()	Imprime as informações contidas nos campos da classe. Para tanto, usa dois métodos para recuperar estas informações: getTipo() e getDetalhes(). Estas funções por sua vez são polimórficas, ou seja, seu tipo retorno varia de acordo com a classe escolhida, tal que este método é sobreposto nas subclasses.
inserirDados()	Insere os dados necessários para se preencher os campos de um objeto de uma dada classe. Seu comportamento é polimórfico.

Descrição dos métodos a serem implementados. Além dos métodos descritos na Tabela, deverão ser criados os métodos get e set correspondentes para retorna e modificar o conteúdo dos campos, respectivamente, bem como os construtores com e sem parâmetros de cada classe. Criar um programa que simule o uso de um cadastro de CD e DVDs.

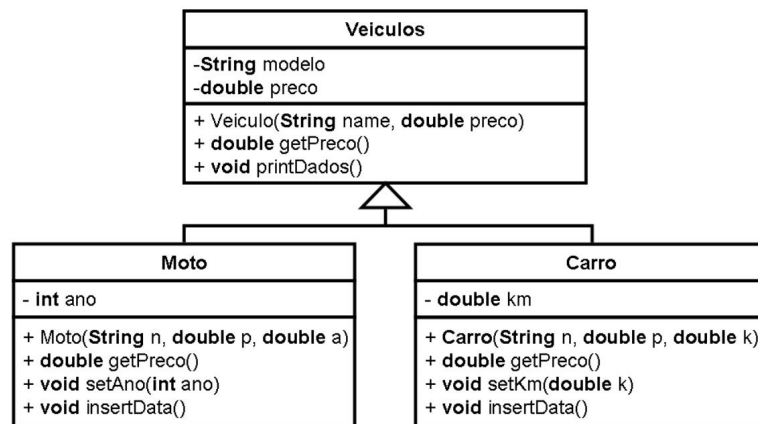
3) simular a operação da folha de pagamento de uma empresa. Como existem dois tipos de funcionários (Assalariado - Salaried e Horista - Hourly). O diagrama UML com as classes necessárias para realizar tal simulação é apresentado na Figura que se segue:



Criar um programa que pede para o usuário definir se deseja inserir um empregado assalariado (Salaried) ou horista (Hourly) e depois preenche os campos de forma adequada. Para determinar, em tempo de execução, se um dado objeto é pertencente a uma dada classe é necessário usar a palavra-chave instanceof tal como dado no código da Figura.

Após isso, deseja-se conhecer o gasto total da empresa com a folha salarial. Para tanto, será necessário conhecer o quanto cada empregado ganha a partir do método getPay().

4) simular a operação de um cadastro de veículos de uma revenda. Existem dois tipos de veículos: Carro, e Moto. O diagrama UML com as classes necessárias para realizar tal simulação é apresentado na Figura

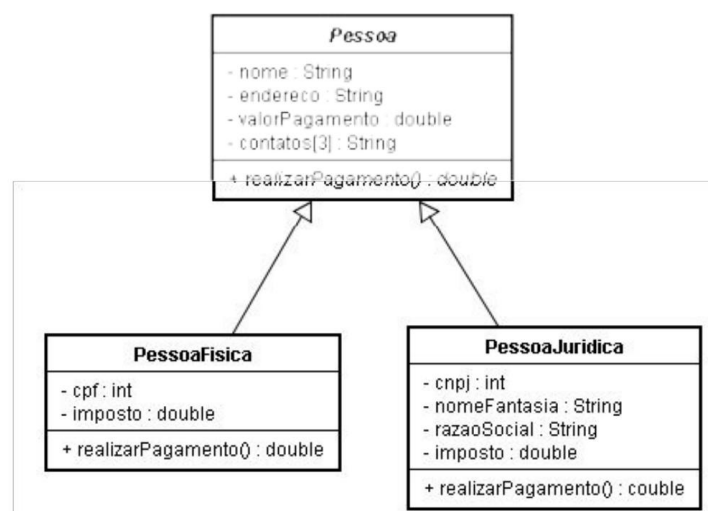


Deseja-se construir um programa que:

Item (A): O usuário decide se deseja inserir uma moto ou um carro. Depois insere as informações necessárias e finalmente imprime o relatório com as características de cada veículo contido no cadastro.

Item (B): Após o usuário inserir as informações deverá ser realizado um levantamento acerca das motos e carros. Primeiro será obtido o total preços dos veículos contidos no cadastro. Depois, é verificado e uma moto tiver no campo ano um valor maior ou igual que 2008 seu valor será reajustado em 10%. Se um carro tiver um valor no campo km um valor maior que 100000, então, seu valor será reduzido em 8%. Por fim, um novo cálculo acerca do total de preços é realizada e impressa.

5) Implementar o seguinte diagrama



Detalhes:

- Classe Pessoa e seu método realizarPagamento são abstratos;
- Atributo contatos[] é do tipo String e corresponde a um Array de 3 posições;
- Atributo imposto corresponde a uma porcentagem que será diminuída sobre o valorPagamento (10% para PF e 20% para PJ
- Estes valores devem ser inicializados no construtor da classe);
- O método realizarPagamento retorna o valor do pagamento diminuído da porcentagem do imposto;

6) Crie uma classe para representar uma conta corrente, com métodos para depositar uma quantia, sacar uma quantia e obter o saldo. Para cada saque será debitada também uma taxa de operação equivalente à 0,5% do valor sacado. Crie, em seguida, uma subclasse desta classe anterior para representar uma conta corrente de um cliente especial. Clientes especiais pagam taxas de operação de apenas 0,1% do valor sacado. Faça testes com as duas classes e verifique seus resultados.

7) Crie uma hierarquia de classes de domínio para uma loja que venda livros, CDs e DVDs.

Sobrescreva o método toString() para que imprima:

- Para livros: nome, preço e autor;
- Para CDs: nome, preço e número de faixas;
- Para DVDs: nome, preço e duração.

Evite ao máximo repetição de código utilizando a palavra super no construtor e no método sobrescrito. Em seguida, crie uma classe Loja com o método main() que adicione 5 produtos diferentes (a sua escolha) a um vetor e, por fim, imprima o conteúdo do vetor.

7b) Modifique o código do programa anterior, da seguinte forma:

- Adicione um atributo que represente o código de barras do produto (é um valor obrigatório e, portanto, deve ser pedido no construtor);
- Sobrescreva o método equals() retornando true se dois produtos possuem o mesmo código de barras;
- Na classe Loja, implemente um simples procedimento de busca que, dado um produto e um vetor de produtos, indique em que posição do vetor se encontra o produto especificado ou imprima que o mesmo não foi encontrado;
- No método Loja.main(), após a impressão do vetor (feita na questão 7a), escolha um dos 5 produtos e crie duas novas instâncias idênticas a ele: uma com o mesmo código de barras e outra com o código diferente. Efetue a busca deste produto no vetor utilizando as duas instâncias e verifique o resultado.