

MATA55 - PROGRAMAÇÃO ORIENTADA A OBJETOS

Aula 18 e 20: Herança, classe abstratas, interface,
poliformismo e sobre-escrita.

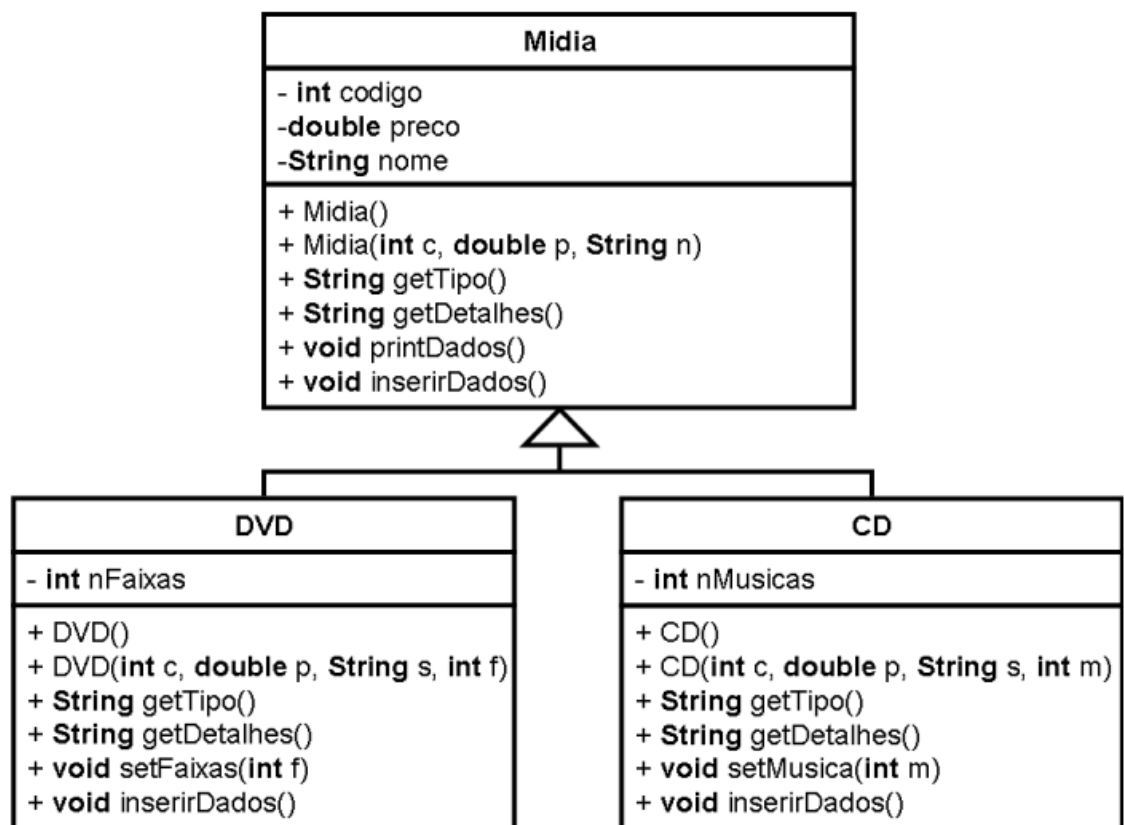
Prof. Felipe Fernandes

19 e 26 outubro de 2022

1. Escreva uma classe **ObjetoGeometrico** que representa um objeto de duas dimensões. Esta classe deve ter métodos para inicializar o objeto, mostrar seus dados e calcular e retornar a sua área e perímetro. Usando esta classe como base, escreva as classes herdeiras **Circulo** (contendo duas coordenadas para o centro e um raio), **Retangulo** (contendo dois valores para os lados) e **Triangulo** (contendo três valores para os lados), que sobreponham os métodos descritos em **ObjetoGeometrico**.
2. Uma biblioteca possui em seu acervo livros e periódicos. Cada livro tem título e situação; cada periódico tem título e número do volume. Os periódicos não podem sair da biblioteca, mas os livros sim. Durante o empréstimo, a situação de um livro é emprestado, quando ele é devolvido, retorna à situação de disponível. Implemente um programa em JAVA que controle o acervo da biblioteca e o empréstimo de livros. Seu código deve ser enxuto e utilizar herança. Dote cada classe de um construtor, de métodos necessários para o encapsulamento e para implementar o comportamento descrito acima.
3. Implemente um sistema em java para um banco que fornece a seus clientes duas opções de contas bancárias: conta corrente e conta poupança. Ambas têm um identificador, uma agência, um cliente (CPF, nome, data de nascimento) e um saldo. Ambas possibilitam realizar saques e depósitos. A conta corrente não cobra taxas sobre o saque. A conta poupança cobra uma

taxa de 5% sobre o valor sacado. A conta poupança rede a uma taxa fixa de 4% sobre o saldo do mês. Em ambas as contas, o cliente pode solicitar um histórico de transações a partir de uma data. São transações para conta corrente: saque e depósito. São transações para conta poupança: saque, depósito e rendimentos.

4. Uma loja que vende CD e DVDS deseja construir um cadastro dos seus produtos. Os dados são fornecidos pelo usuário via teclado. Para tanto, foi elaborado o diagrama de classes da Figura abaixo.

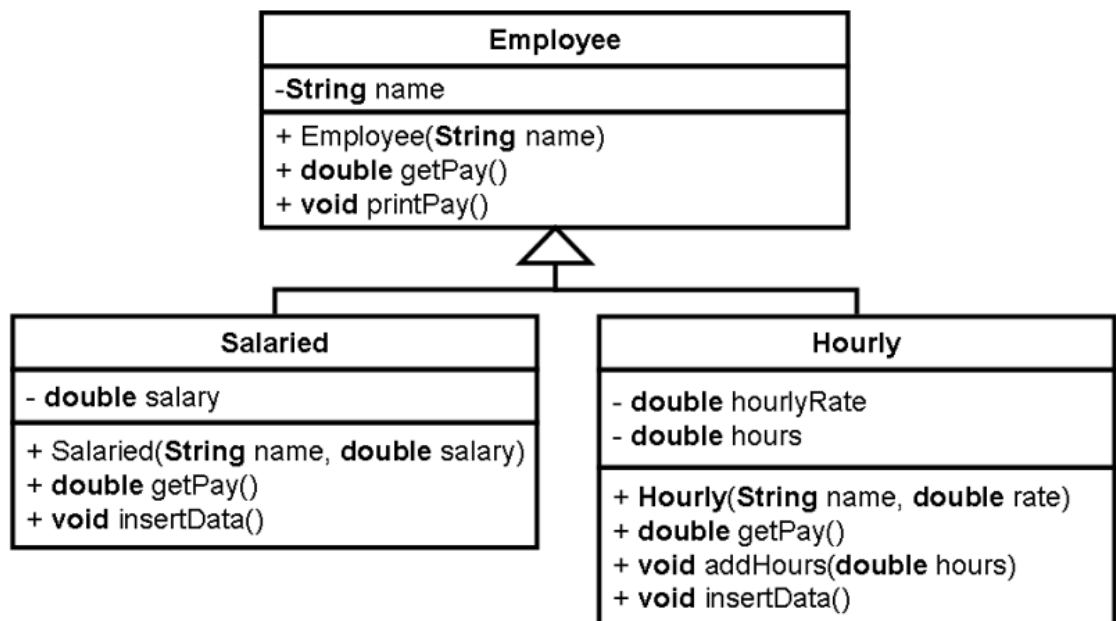


- **getTipo()** - Retorna uma String com o nome da classe.
- **getDetalhes()** - Retorna uma String com as informações contidas nos campos.

- **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.

Além dos métodos descritos acima, 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.

5. Implementar um programa em Java que simule a operação da folha de pagamento de uma empresa. A empresa tem vários funcionários, dos quais destaca-se duas categorias: Assalariado (Salaried) e Horista (Hourly). O diagrama com as classes necessárias para realizar tal simulação é apresentado abaixo.



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. O programa deve perguntar ao usuário se ele deseja inserir um empregado assalariado (Salaried) ou horista (Hourly). Em seguida, o programa deve solicitar as informações para preencher os campos de forma adequada. Ao final, 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()**.