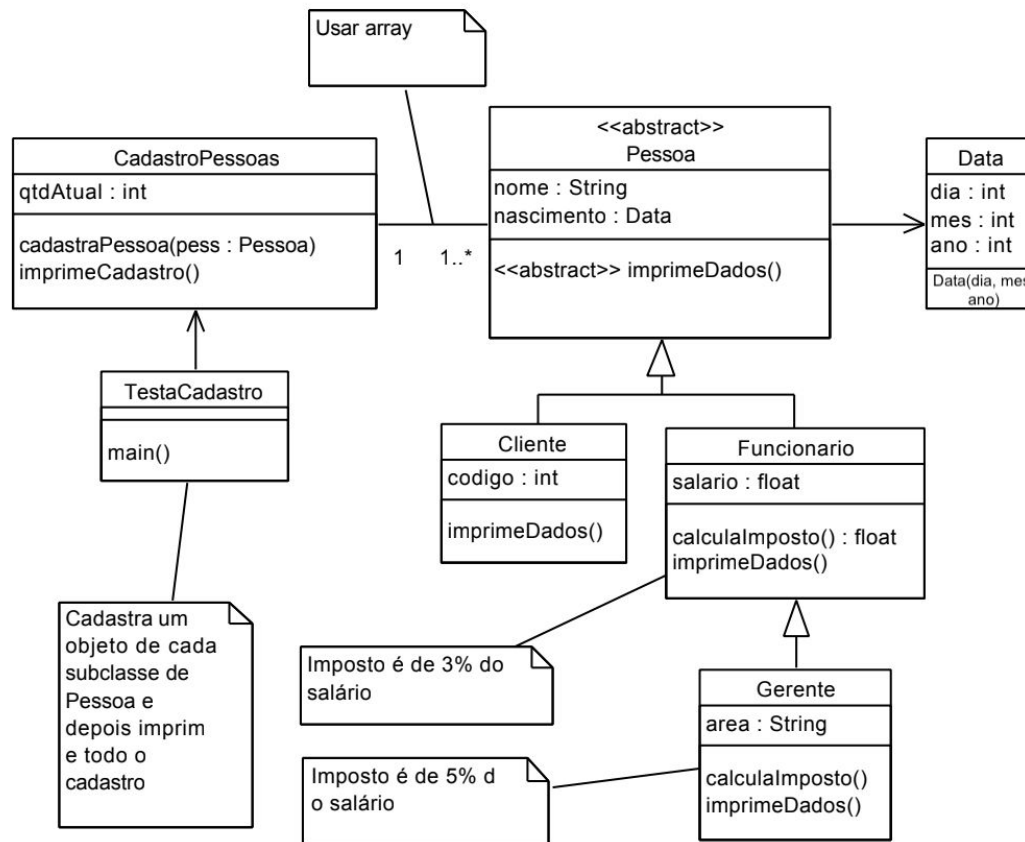


1. Implemente o seguinte diagrama:



- Implemente em java uma classe abstrata chamada **FiguraGeometrica**, que possui dois métodos abstratos: `calcularArea`, `calcularPerimetro`, crie as classes filhas para representar as figuras geométricas: círculo, retângulo e quadrado, que herdam da **FiguraGeometrica**. Crie os atributos necessários para obter a área e o perímetro de cada figura, por exemplo: círculo possui um raio e as outras classes os tamanhos dos lados. Aplique polimorfismo sobrescrevendo os métodos da classe abstrata, conforme a regra de calculo do perímetro e da área de cada figura especificamente.
- Explique os principais conceitos da Programação Orientada por Objetos: encapsulamento, herança, composição e polimorfismo. Dê um exemplo de código para cada um deles em qualquer linguagem de programação.
- 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.

5. Modifique o código do programa anterior, da seguinte forma:
 - a. Adicione um atributo que represente o código de barras do produto (é um valor obrigatório e, portanto, deve ser pedido no construtor);
 - b. Sobrescreva o método `equals()` retornando `true` se dois produtos possuem o mesmo código de barras;
 - c. 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;
 - d. No método `Loja.main()`, após a impressão do vetor, 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
6. Crie uma hierarquia de herança que um banco possa utilizar para representar dois tipos de conta: poupança e conta corrente. Todos os clientes deste banco podem depositar e sacar dinheiro de suas contas.

A classe **`Conta`** deve possuir um atributo que represente o saldo da conta. Este atributo deve ser inicializado através de um construtor parametrizado que valide o valor enviado como parâmetro. Devem ser criados métodos para mostrar o saldo, para crédito e para débito na conta. Crie um *getter* e um *setter* para o atributo.

A classe **`Poupanca`** deve possuir um atributo relacionado à variação (rendimento), com métodos *getter*, *setter* e construtor. Crie também um método `CalculaRendimento()`, que informa o valor do saldo multiplicado pela taxa de rendimento.

A classe **`ContaCorrente`** deve incluir um atributo que represente a taxa cobrada por cada transação de crédito/débito, com *getter*, *setter* e construtor. Sobrescreva os métodos de crédito e débito para descontar o valor de tal taxa a cada transação bem sucedida.

Adicione o comportamento polimórfico conforme descrito:

- a. Os métodos de crédito, débito e saldo em conta devem ser abstratos na classe base;
- b. Crie um *driver* com um vetor de referências para `Conta`, cada referência deve ser relativa a um dos objetos de cada classe concreta da hierarquia
 - i. Invoque os métodos de débito e crédito de cada objeto apontado.

- c. Percorra o vetor e determine em tempo de execução qual é o tipo de cada conta:
 - i. Se for uma poupança, calcule seu rendimento através do método *CalculaRendimento()*;
 - ii. Se for uma conta corrente, apenas mostre seu saldo.