



O Diagrama de Classes acima tem exemplos do uso de herança , classe abstrata e também interface . Para começar, uma descrição breve do que se trata o sistema.

O diagrama apresenta um sistema que irá abordar o básico de uma faculde. Este básico compreende a existência de professores, alunos, diretores, coordenadores, secretarias e também contas bancárias para as pessoas.

Sabendo, de forma geral, do que se trata de um sistema, vejamos um pouco mais detalhado sobre cada classe.

- **Pessoa:** classe geral para definir o que todos os tipos de pessoas têm em comum, desde os atributos até os métodos. Como se trata de uma classe que não precisa ser instanciada e pode conter assinaturas de métodos, ou seja, a declaração de métodos sem sua respectiva implementação (métodos abstratos), ela será implementada como uma classe abstrata.
- **PessoaFisica:**
 - Subclasse de Pessoa.
 - Esta classe é definida para conter atributos e métodos comuns para as demais classes que serão Pessoas Físicas. Da mesma forma que Pessoa, essa classe não precisa ser instanciada, portanto, pode ser implementada também como classe abstrata.
- **PessoaJuridica:**
 - Subclasse de Pessoa.
 - Esta classe com o mesmo propósito de PessoaFisica, entretanto para as classes que serão vistas como Pessoas Jurídicas. Da mesma forma, pode ser implementada como uma classe abstrata.
- **Faculdade**
 - Subclasse de PessoaJuridica.
 - Esta classe irá representar qualquer faculdade. Ou seja, seus objetos instanciados serão diferentes faculdades. Conterá em si atributos e métodos herdados de Pessoa e PessoaJuridica, possivelmente tendo de implementar algum método abstrato que tenha sido declarado em uma de suas superclasses.
- **Banco**
 - Subclasse de PessoaJuridica.
 - Similar à classe Faculdade, esta classe representará qualquer banco, ou seja, seus objetos instanciados serão diferentes bancos. Além de seus atributos e métodos específicos, também herdará os de suas superclasses.
- **Aluno**
 - Subclasse de PessoaFisica.
 - É a classe que representará toda entidade do tipo aluno. Ou seja, todo novo aluno criado no sistema será um objeto instanciado dessa classe. Terá

seus atributos e métodos específicos, e herdará atributos e métodos das classes Pessoa e PessoaFisica .

- **Funcionario**

- Subclasse de PessoaFisica .
- Esta é outra classe "geral". Conterá em si todos os atributos e métodos comuns a qualquer tipo de funcionário. Como existem variados tipos de funcionário, não faz sentido instanciar esta classe , portanto, pode ser implementada como classe abstrata . Podem existir métodos comuns a todos os funcionários, cujas implementações serão diferentes em cada tipo de funcionário. Portanto, nesta classe podem existir também métodos abstratos .

- **Professor**

- Subclasse de Funcionario .
- Classe para representar todos os professores. Ou seja, cada novo professor introduzido ao sistema será um objeto instanciado dessa classe. Além de seus atributos e métodos específicos, herdará aqueles de Pessoa , PessoaFisica e Funcionario .

- **Gerente**

- Subclasse de Funcionario .
- Exato mesmo caso da classe Professor .

- **Presidente**

- Subclasse de Funcionario .
- Implementa Diretoria .
- Classe cujos objetos serão os presidentes de alguma empresa (faculdade ou banco). Herda atributos e métodos das classes Pessoa , PessoaFisica e Funcionario . Ao mesmo tempo toma para si métodos da interface Diretoria , os quais irá implementar.

- **Diretor**

- Subclasse de Funcionario .
- Implementa Diretoria .
- Exato mesmo caso de Presidente .

- **Coordenador**

- Subclasse de Funcionario .
- Implementa Diretoria .
- Mesmo caso das classes Presidente e Diretor .

- **SecAcademica**

- Subclasse de `Funcionario` .
- Implementa `Secretaria` .
- Similar às classes `Presidente` , `Diretor` e `Coordenador` . A maior diferença está na interface a qual implementa, que neste caso é `Secretaria` .
- **SecFinanceira**
 - Subclasse de `Funcionario` .
 - Implementa `Secretaria` .
 - Mesmo caso de `SecAcademica` .
- **Diretoria**
 - Interface que declara todos os métodos comuns a classes que sejam parte de alguma entidade de diretoria. É importante lembrar que interfaces não possuem somente métodos , pois podem possuir outros componentes. Além disso, seus métodos são abstratos por padrão e, por isso, não precisam ser declarados com a palavra reservada `abstract` .
- **Secretaria**
 - Mesmo caso da interface `Diretoria` . É importante pontuar também que uma interface **não é** uma classe .
- **Conta**
 - Classe que conterá os atributos e métodos comuns a todas as contas bancárias. Como se trata de uma classe "geral", não precisa ser instanciada , portanto, pode ser implementada como classe abstrata .
- **ContaCorrente**
 - Subclasse de `Conta` .
 - Classe que representa todas as contas correntes que existirão no sistema. Deverá ter seus atributos e métodos específicos e herdará aqueles de `Conta` .
- **ContaPoupanca**
 - Subclasse de `Conta` .
 - Mesmo caso da classe `ContaCorrente` .

Após a descrição mais detalhada do sistema é importante pontuar que **nenhum** desenvolvedor irá perceber de cara como vai implementar todas as classes . Ou seja, à medida em que o sistema é construído (primeiro com um diagrama, e depois na programação), o desenvolver (a construção do diagrama é um trabalho para um Analista de Software) vai percebendo quais classes deverão ser abstratas ,

quais deverão ser interfaces , e assim por diante. **Não é uma tarefa trivial** e depende muito de experiência para que os sistemas sejam melhor arquitetados.