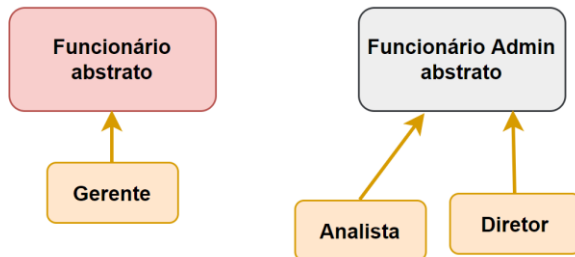


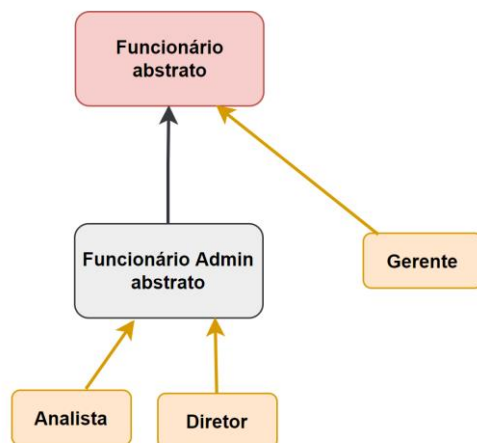


Kotlin não permite fazer heranças múltiplas por padrão (assim como java), mas existe um jeito de fazer uma **classe ter dois supertipos apenas alterando a hierarquia** (isso depende do caso). Bom, atualmente todo mundo ou é **funcionário** ou **funcionárioAdmin** (tem **membros** específicos para login e autenticação):



A ideia da hierarquia para “herança múltipla” é fazer o **funcionário admin** herdar de **funcionário**, agora toda **implementação concreta** pode decidir em que **nível** ela **pertence**. As **implementações concretas** podem ser do nível admin permitindo **login e autenticação** e ainda sim **manterem status de funcionários**.

Já as **implementações concretas** que não possuem **membros** de admin se mantêm no nível de **funcionário** podendo ser **compatível** com qualquer operação de funcionário sem saber qualquer coisa relacionado a **autenticação e login**:



```
abstract class FuncionarioAdmin(  
    nome: String,  
    cpf: String,  
    salario: BigDecimal,  
    val senha : String  
) : Funcionario(nome = nome, cpf = cpf, salario = salario){
```

****Membros abstratos de classes abstratas não precisam ser implementados por outras classes abstratas que os herdam****