


Kotlin

Herança e Polimorfismo




Softblue
cursos online

Tópicos Abordados



- Herança
- Construtores e a herança
- Sobrescrevendo métodos
- Classes abstratas
- Interfaces

Herança em Kotlin



- Toda classe em Kotlin herda de *Any* direta ou indiretamente
 - A classe *Any* fornece os métodos *equals()*, *hashCode()* e *toString()*
- Nenhuma classe pode ser herdada, a não ser que o modificador *open* seja aplicado
- Kotlin só permite a herança direta de uma classe

```
open class Veiculo {  
    //...  
}  
  
class Automovel : Veiculo() {  
    //...  
}
```

Subclasse

Construtores e a Herança



- Se uma superclasse define um construtor primário, a subclasse deve chamá-lo

```
open class Veiculo(var ano: Int) { }  
  
class Automovel(ano: Int) : Veiculo(ano) {  
    var modelo: String = ""  
  
    constructor(ano: Int, modelo: String) : this(ano) {  
        this.modelo = modelo  
    }  
}
```

Se o construtor primário não existir, os construtores secundários devem usar a expressão `super()` para chamar os construtores da superclasse

Sobrescrevendo Métodos



- Apenas métodos definidos com o modificador *open* podem ser sobrescritos
- A subclasse deve usar o modificador *override* ao fazer a sobrescrita

```
open class Veiculo {  
    open fun ligar() {  
        print("ligando o veiculo")  
    }  
}  
  
class Automovel : Veiculo() {  
    override fun ligar() {  
        super.ligar()  
        print("ligando o automóvel")  
    }  
}
```

Chama o método da superclasse

Classes Abstratas



- Uma classe abstrata não pode ser instanciada
- Pode ter métodos abstratos, que devem ser implementados por uma subclasse concreta

```
abstract class Veiculo {  
    abstract fun ligar()  
}  
  
class Automovel : Veiculo() {  
    override fun ligar() {  
        print("ligando o automóvel")  
    }  
}
```

Não tem implementação

Interfaces



- Interfaces são criadas através da expressão *interface*
- Podem ter métodos abstratos e também implementar métodos
- Não podem ter properties (a não ser que elas sejam abstratas)
- Uma classe pode implementar várias interfaces

```
interface Recortavel {  
    fun recortar() {  
        //...  
    }  
    fun destacar() {  
        //...  
    }  
}
```

→ Método abstrato

→ Método implementado