



# Herança Múltipla

**Prof. Fábio Procópio**

**Prof. João Nascimento**



2

## Relembrando...

- Na [aula passada](#), iniciamos os estudos sobre Herança, considerada um dos pilares da Orientação a Objetos
  - Nos exemplos que vimos, as subclasses herdavam apenas métodos e atributos de uma única classe. Mas, se precisássemos que as subclasses fossem filhas de mais de uma superclasse?
- Nesta aula, continuaremos nossos estudos, no entanto falando sobre **Herança Múltipla**, isto é, subclasses que são filhas de mais de uma superclasse.





# Introdução – 1 de 2

- **Herança múltipla** ocorre sempre que uma **subclasse possui duas ou mais superclasses**, ou seja, é filha de mais de uma classe;
- Permite **combinar características de várias superclasses** já existentes como sendo um ponto de partida para a definição de novas classes;
- A vantagem da herança múltipla é a **maior capacidade de especificação** de classes e a maior oportunidade de **reutilização**;
- Uma classe que possui mais de uma superclasse é denominada **classe de junção**.



4

## Introdução – 2 de 2



**Celular**



**Tablet**



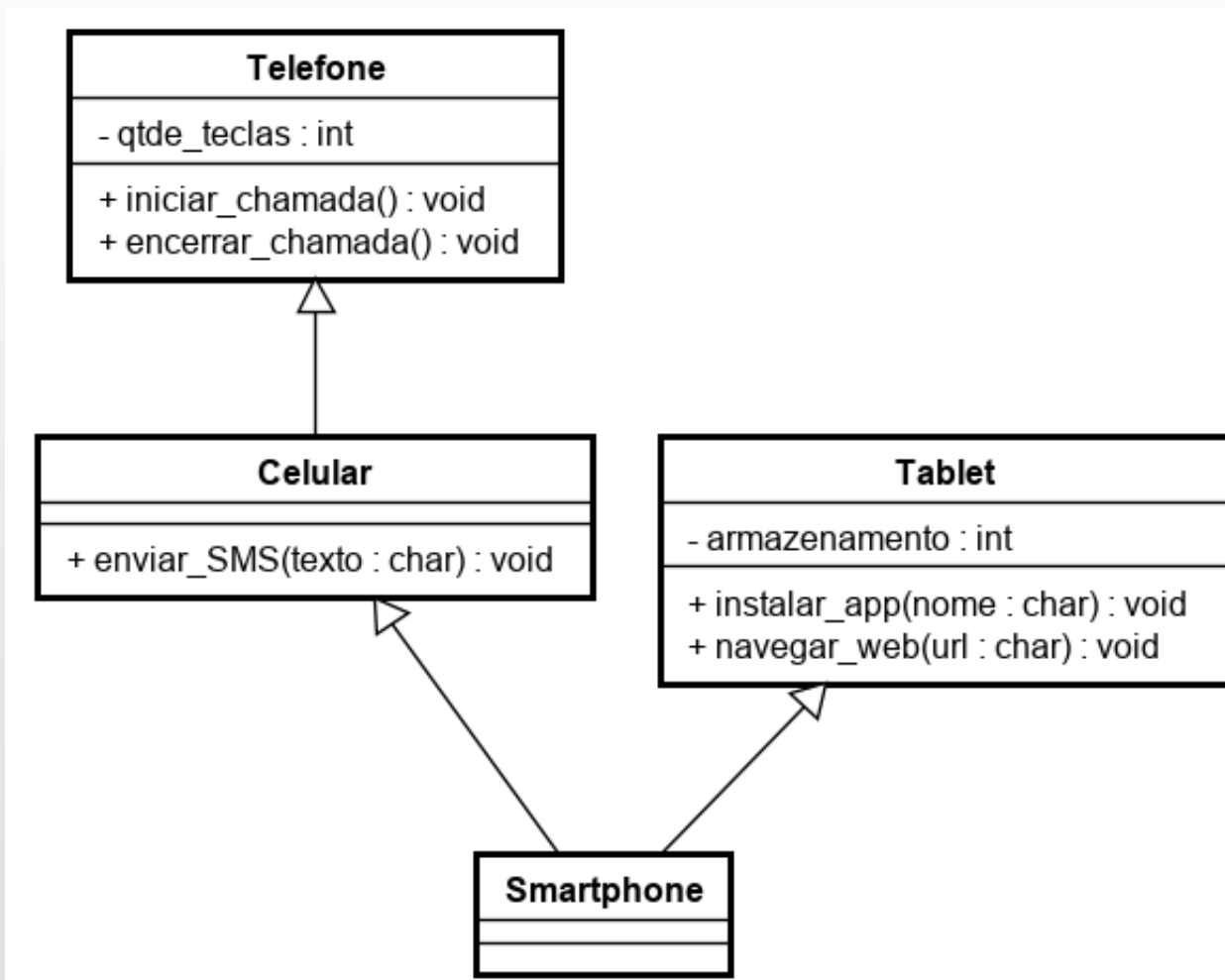
**Smartphone**



5

## Exercício Resolvido 01 – 1 de 4

➤ Vamos implementar o modelo abaixo.





## Exercício Resolvido 01 – 2 de 4

```
class Telefone:
    def __init__(self, qtde_tecclas):
        self.qtde_tecclas = qtde_tecclas

    def iniciar_chamada(self):
        print("Discando...")

    def encerrar_chamada(self):
        print("Ligação encerrada.")
```

```
from Telefone import *

class Celular(Telefone):
    def __init__(self, qtde_tecclas):
        Telefone.__init__(self, qtde_tecclas)

    def enviar_SMS(self, texto):
        print(texto)
        print("Mensagem enviada com sucesso.")
```



## Exercício Resolvido 01 – 3 de 4

```
class Tablet:
    def __init__(self, armazenamento):
        self.armazenamento = armazenamento

    def instalar_app(self, nome):
        print("Instalando aplicativo...")
        print("App {} instalado com sucesso.".format(nome))

    def navegar_web(self, url):
        print("Conectando em {}...".format(url))
```

```
from Celular import *
from Tablet import *

class Smartphone(Celular, Tablet):
    def __init__(self, qtde_tecclas, armazenamento):
        Celular.__init__(self, qtde_tecclas)
        Tablet.__init__(self, armazenamento)
```



## Exercício Resolvido 01 – 4 de 4

```
from Smartphone import *

teclas = int(input("Informe quantas teclas possui seu smartphone: "))
armaz = int(input("Informe a capacidade de armazenamento do smartphone: "))

smart = Smartphone(teclas, armazen)

msg = input("Digite sua SMS: ")
smart.enviar_SMS(msg)

url = input("Digite a URL que deseja acessar: ")
smart.navegar_web(url)

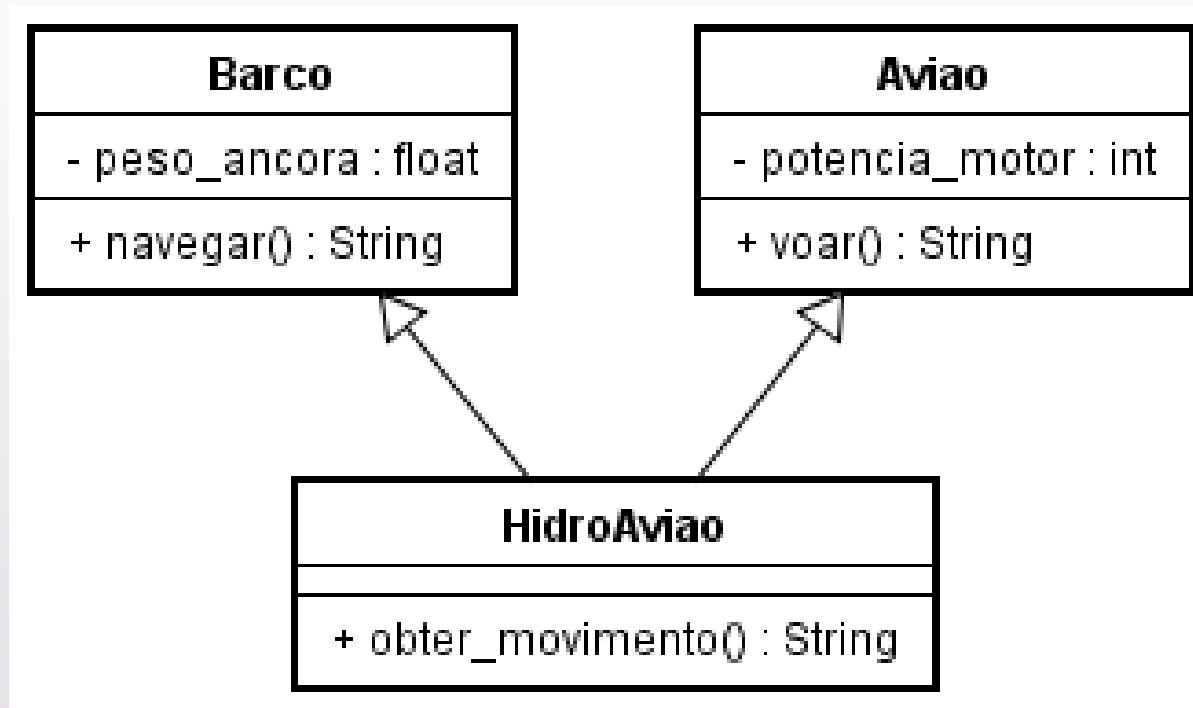
app = input("Qual aplicativo deseja instalar? ")
smart.instalar_app(app)
smart.iniciar_chamada()
smart.finalizar_chamada()
```





# Exercício de Fixação 01

- Implemente o modelo a seguir:



Método **navegar()**  
Retorna a *string* "Navega"

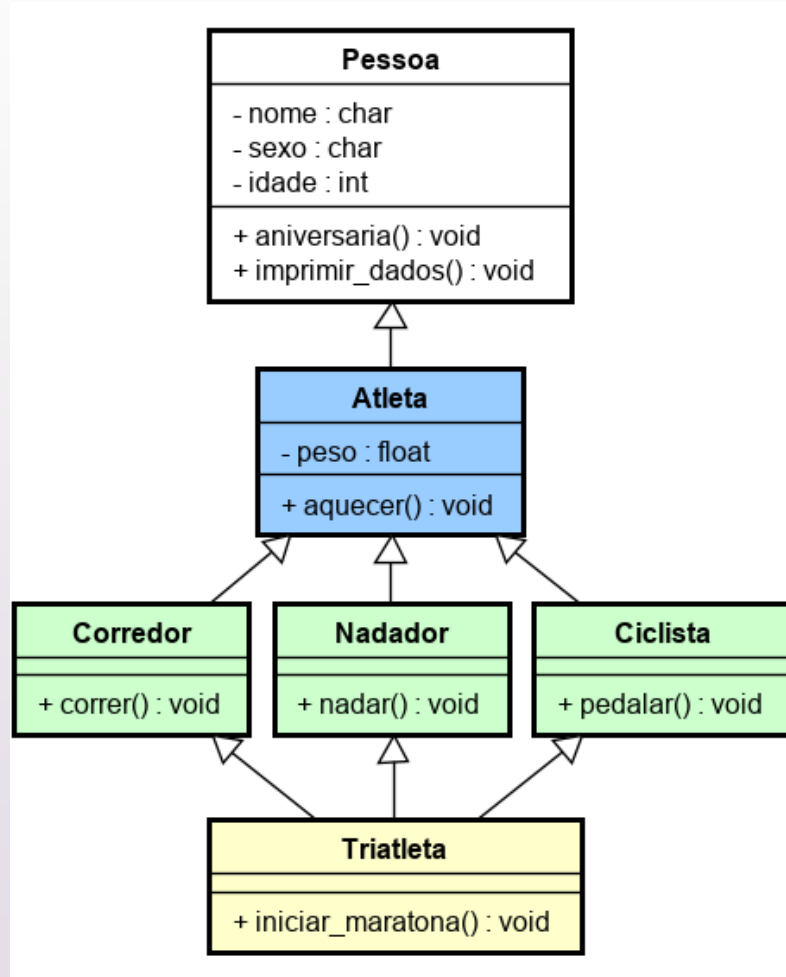
Método **voar()**  
Retorna a *string* "Voa"

Método **obter\_movimento()**  
Retorna a *string* "Navega e Voa"



## Exercício de Fixação 02

- Implemente o modelo a seguir:



Método **aquecer()**  
Imprime a *string* "Aquecendo..."

Método **correr()**  
Imprime a *string* "Correndo..."

Método **nadar()**  
Imprime a *string* "Nadando..."

Método **pedalar()**  
Imprime a *string* "Pedalando..."

Método **iniciar\_maratona()**  
Imprime a *string* "Atleta iniciou a maratona."



# Canal no Youtube

Em nosso canal no Youtube, você pode acessar uma playlist que preparamos com várias vídeo-aulas falando sobre o assunto. Veja:

➡ [Herança Múltipla](#)



# Referências

1. MANSSOUR, Isabel. **Herança Múltipla**. Disponível em: <http://www.inf.pucrs.br/manssour/LinguagemC++/HerancaMultipla.pdf>. Acessado em: 27 ago. 2019.