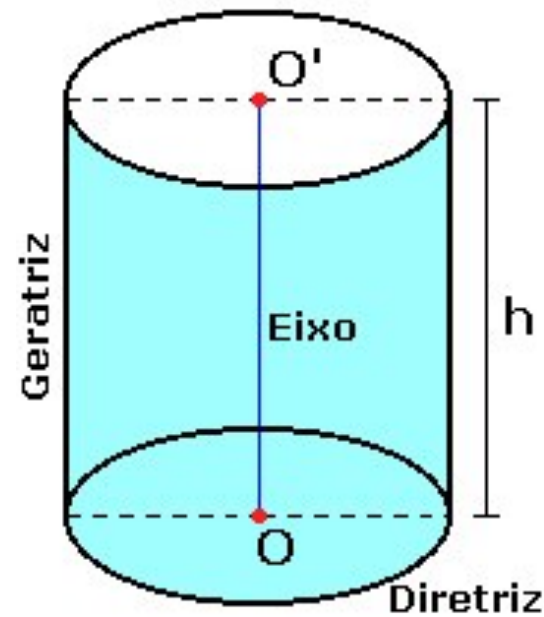


Herança e Polimorfismo

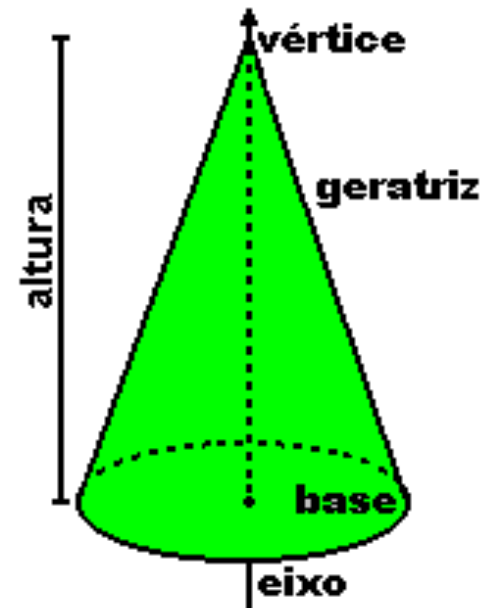
Prof. Me. Eugênio Júlio M. C. Carvalho
eugeniojulio@uol.com.br

FIGURAS GEOMETRICAS

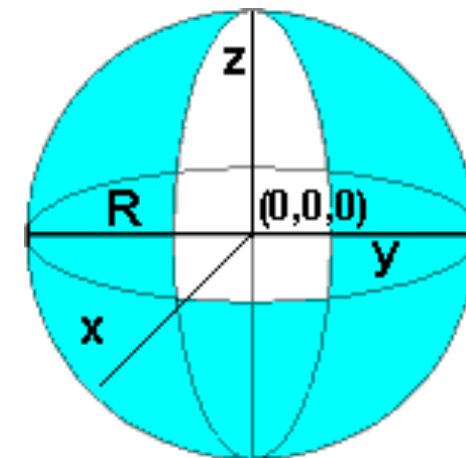
Cilindro



Cone



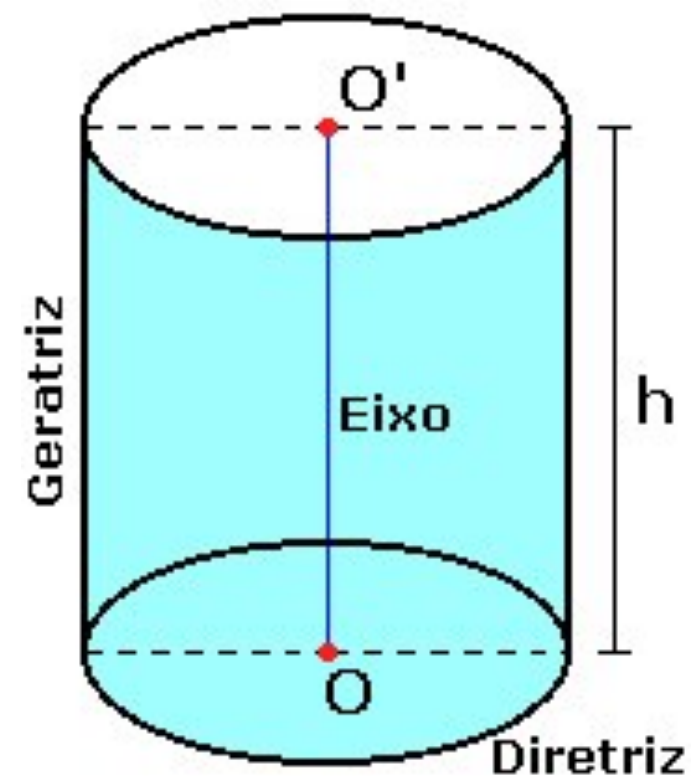
Esfera



FIGURAS GEOMETRICAS

CARACTERISTICAS DO OBJETO CILINDRO

- ✦ RAIO
- ✦ ALTURA
- ✦ CALCULAR ÁREA LATERAL
- ✦ CALCULAR ÁREA TOTAL
- ✦ CALCULAR VOLUME
- ✦ TIPO DA FIGURA



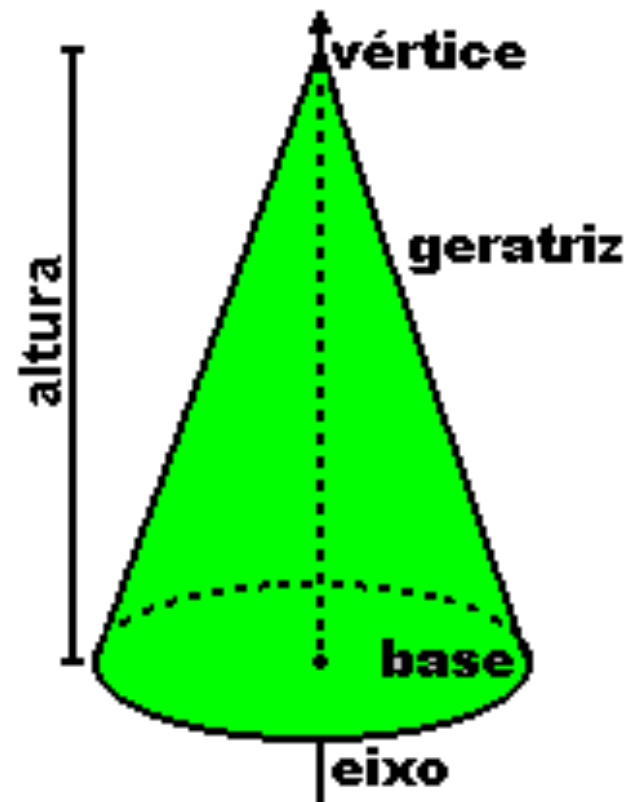
CLASSE CILINDRO

Cilindro
<ul style="list-style-type: none">- raio : float- altura : float
<ul style="list-style-type: none">+ setRaio(raio : float) : void+ getRaio() : float+ setAltura(altura : float) : void+ getAltura() : float+ calcularAreaLateral() : float+ calcularAreaTotal() : float+ calcularVolume() : float+ getTipoDaFigura() : String

FIGURAS GEOMETRICAS

CARACTERISTICAS DO OBJETO CONE

- ✦ RAIO
- ✦ ALTURA
- ✦ CALCULAR GERATRIZ
- ✦ CALCULAR ÁREA LATERAL
- ✦ CALCULAR ÁREA TOTAL
- ✦ CALCULAR VOLUME
- ✦ TIPO DA FIGURA



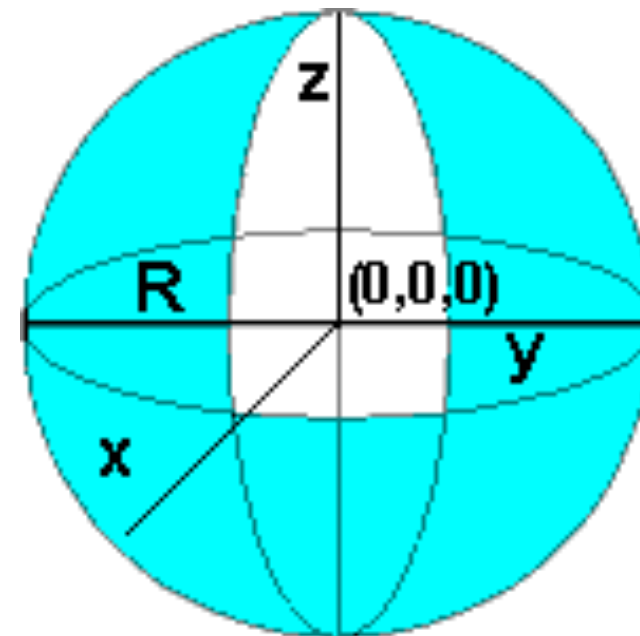
CLASSE CONE

Cone
<ul style="list-style-type: none">- raio : float- altura : float
<ul style="list-style-type: none">+ setRaio(raio : float) : void+ getRaio() : float+ setAltura(altura : float) : void+ getAltura() : float+ calcularGeratriz() : float+ calcularAreaLateral() : float+ calcularAreaTotal() : float+ calcularVolume() : float+ getTipoDaFigura() : String

FIGURAS GEOMETRICAS

CARACTERISTICAS DO OBJETO ESFERA

- ✦ RAIO
- ✦ CALCULAR ÁREA TOTAL
- ✦ CALCULAR VOLUME
- ✦ TIPO DA FIGURA



CLASSE ESFERA

Esfera
– raio : float
+ setRaio(raio : float) : void + getRaio() : float + calcularAreaTotal() : float + calcularVolume() : float + getTipoDaFigura() : String

CLASSES

Esfera
– raio : float
+ setRaio(raio : float) : void + getRaio() : float + calcularAreaTotal() : float + calcularVolume() : float + getTipoDaFigura() : String

Cilindro
– raio : float – altura : float
+ setRaio(raio : float) : void + getRaio() : float + setAltura(altura : float) : void + getAltura() : float + calcularAreaLateral() : float + calcularAreaTotal() : float + calcularVolume() : float + getTipoDaFigura() : String

Cone
– raio : float – altura : float
+ setRaio(raio : float) : void + getRaio() : float + setAltura(altura : float) : void + getAltura() : float + calcularGeratriz() : float + calcularAreaLateral() : float + calcularAreaTotal() : float + calcularVolume() : float + getTipoDaFigura() : String

CLASSES

As três classes apresentadas possuem características comuns:

Atributo – raio

Métodos – obter (get) e atribuir (set) raio;
calcular área total;
calcular volume;
obter (get) o tipo da figura.

CLASSES

O atributo raio e os métodos obter (get) e atribuir (set) raio, possuem códigos de execução idênticos em ambas as classes, portanto existe uma duplicação destes entre elas.

CLASSES

As três classes representam objetos com características comuns.

A Programação Orientada a Objeto possui um conceito que evita esta duplicação:

A Herança.

HERANÇA

A herança é o mecanismo do paradigma orientado a objetos que permite compartilhar atributos e operações entre classes, baseados em um relacionamento hierárquico.

HERANÇA

É a propriedade pela qual podemos criar classes que se ampliam, a partir de definições básicas de classes mais simples e genéricas para classes mais complexas e específicas.

HERANÇA

A herança, na prática, significa a possibilidade de construir objetos especializados, que herdam as características de objetos mais generalistas.

HERANÇA

Ou ainda, a herança é uma forma de reutilização de código a medida que podemos aproveitar os atributos e métodos das classes já existentes, para gerar novas classes mais específicas, que aproveitarão os recursos das classes hierarquicamente superior.

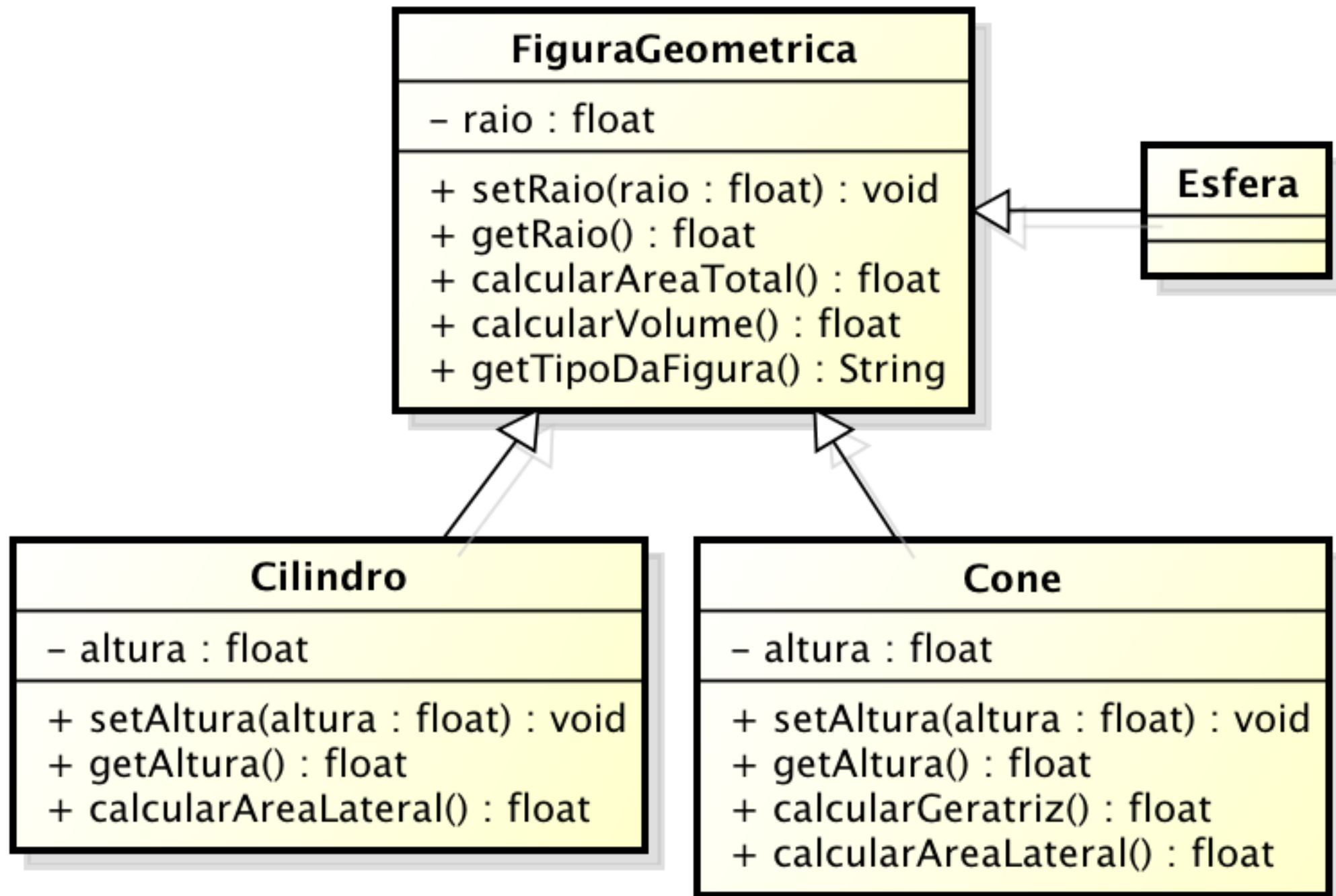
CLASSES

Esfera
<u>- raio : float</u>
<u>+ setRaio(raio : float) : void</u> <u>+ getRaio() : float</u> <u>+ calcularAreaTotal() : float</u> <u>+ calcularVolume() : float</u> <u>+ getTipoDaFigura() : String</u>

Cilindro
<u>- raio : float</u> <u>- altura : float</u>
<u>+ setRaio(raio : float) : void</u> <u>+ getRaio() : float</u> <u>+ setAltura(altura : float) : void</u> <u>+ getAltura() : float</u> <u>+ calcularAreaLateral() : float</u> <u>+ calcularAreaTotal() : float</u> <u>+ calcularVolume() : float</u> <u>+ getTipoDaFigura() : String</u>

Cone
<u>- raio : float</u> <u>- altura : float</u>
<u>+ setRaio(raio : float) : void</u> <u>+ getRaio() : float</u> <u>+ setAltura(altura : float) : void</u> <u>+ getAltura() : float</u> <u>+ calcularGeratriz() : float</u> <u>+ calcularAreaLateral() : float</u> <u>+ calcularAreaTotal() : float</u> <u>+ calcularVolume() : float</u> <u>+ getTipoDaFigura() : String</u>

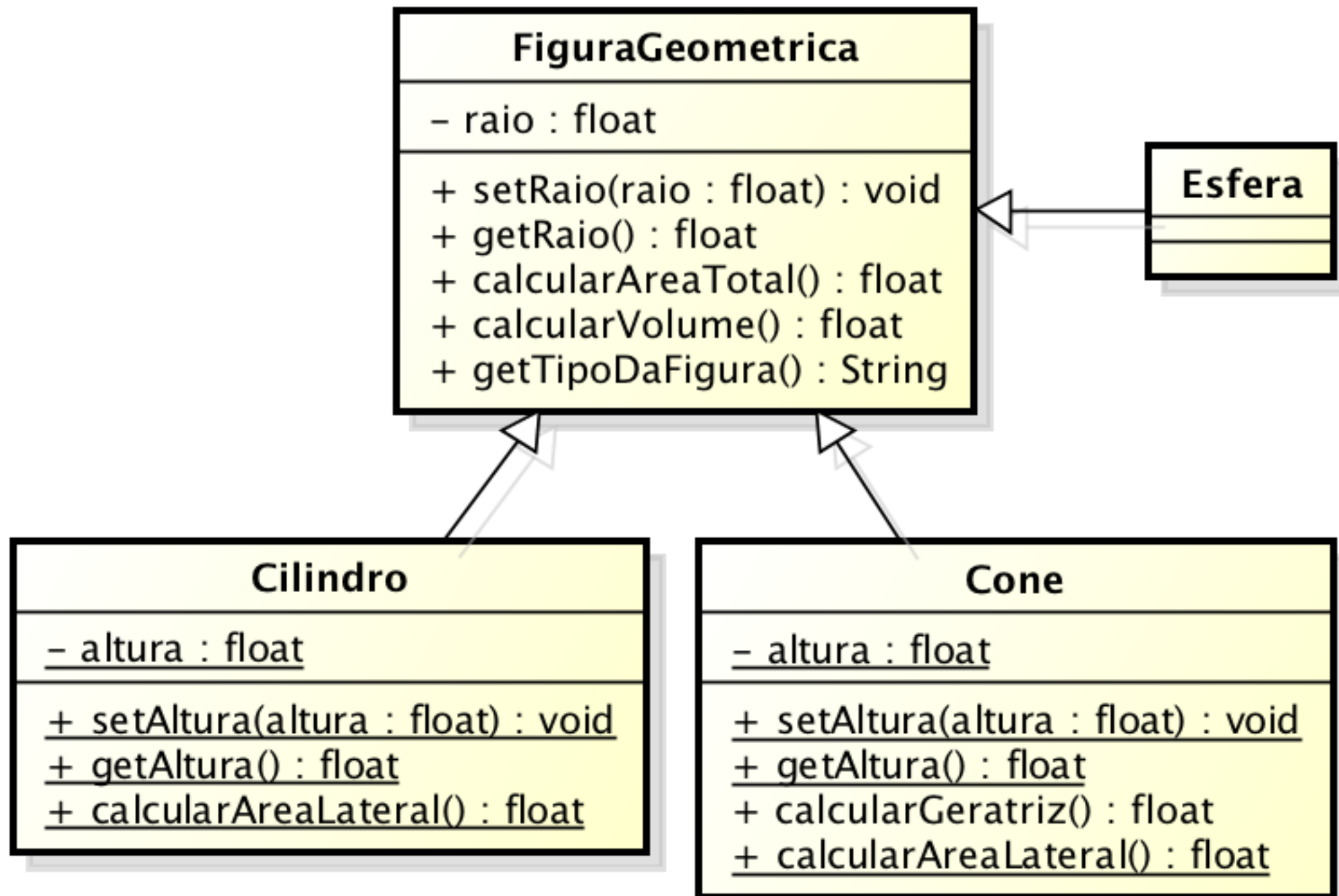
CLASSES COM HERANÇA



CLASSES COM HERANÇA₃

Observando novamente as classe com herança notamos que as classes Cilindro e Cone ainda possuem elementos comuns – **altura, obter (get) e atribuir (set) altura e calcular área lateral.**

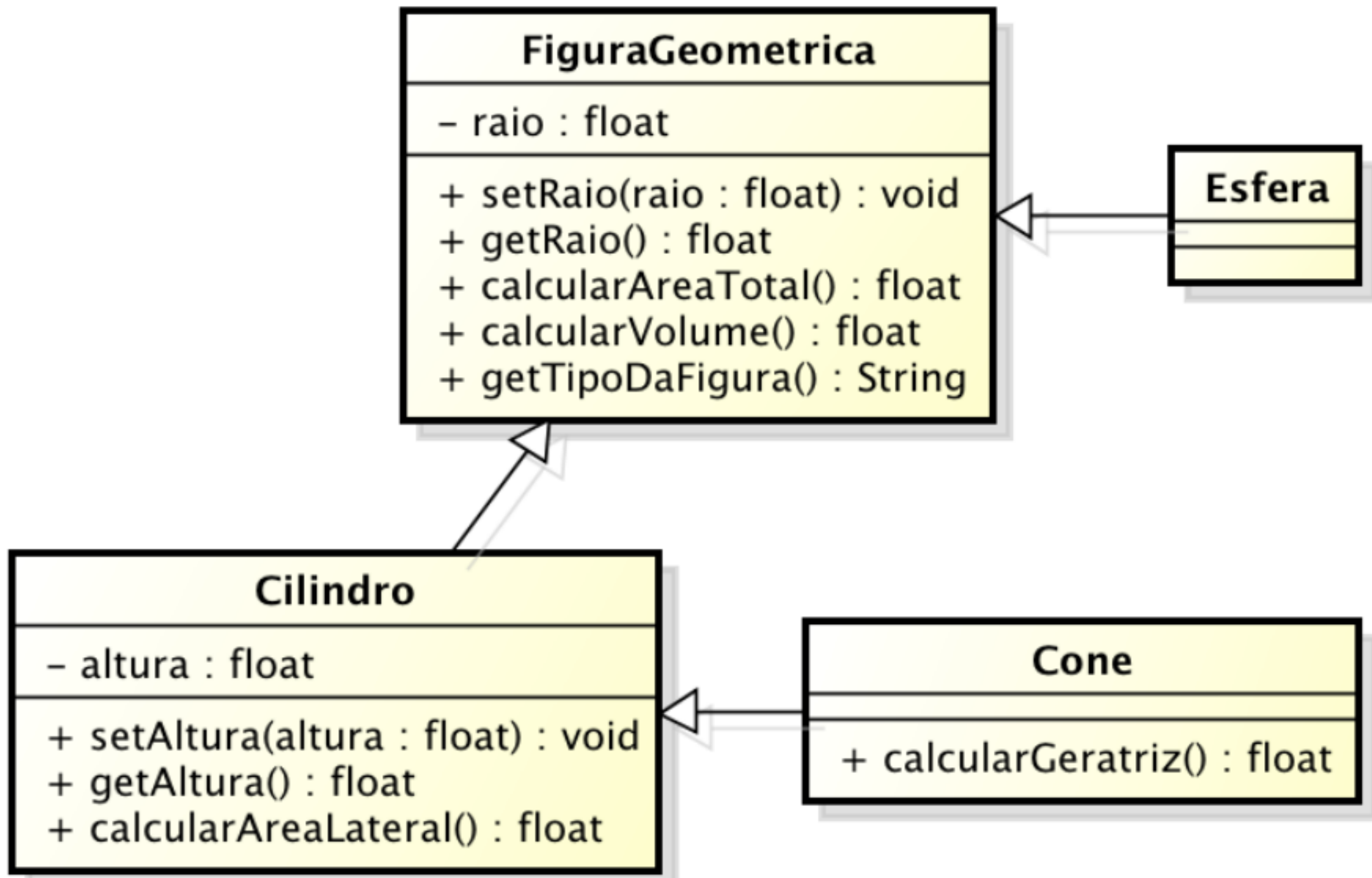
CLASSES COM HERANÇA



CLASSES COM HERANÇA₃

Matematicamente o cone é uma restrição de um cilindro. Portanto podemos novamente aplicar o conceito de herança e ampliarmos os elementos.

CLASSES COM HERANÇA



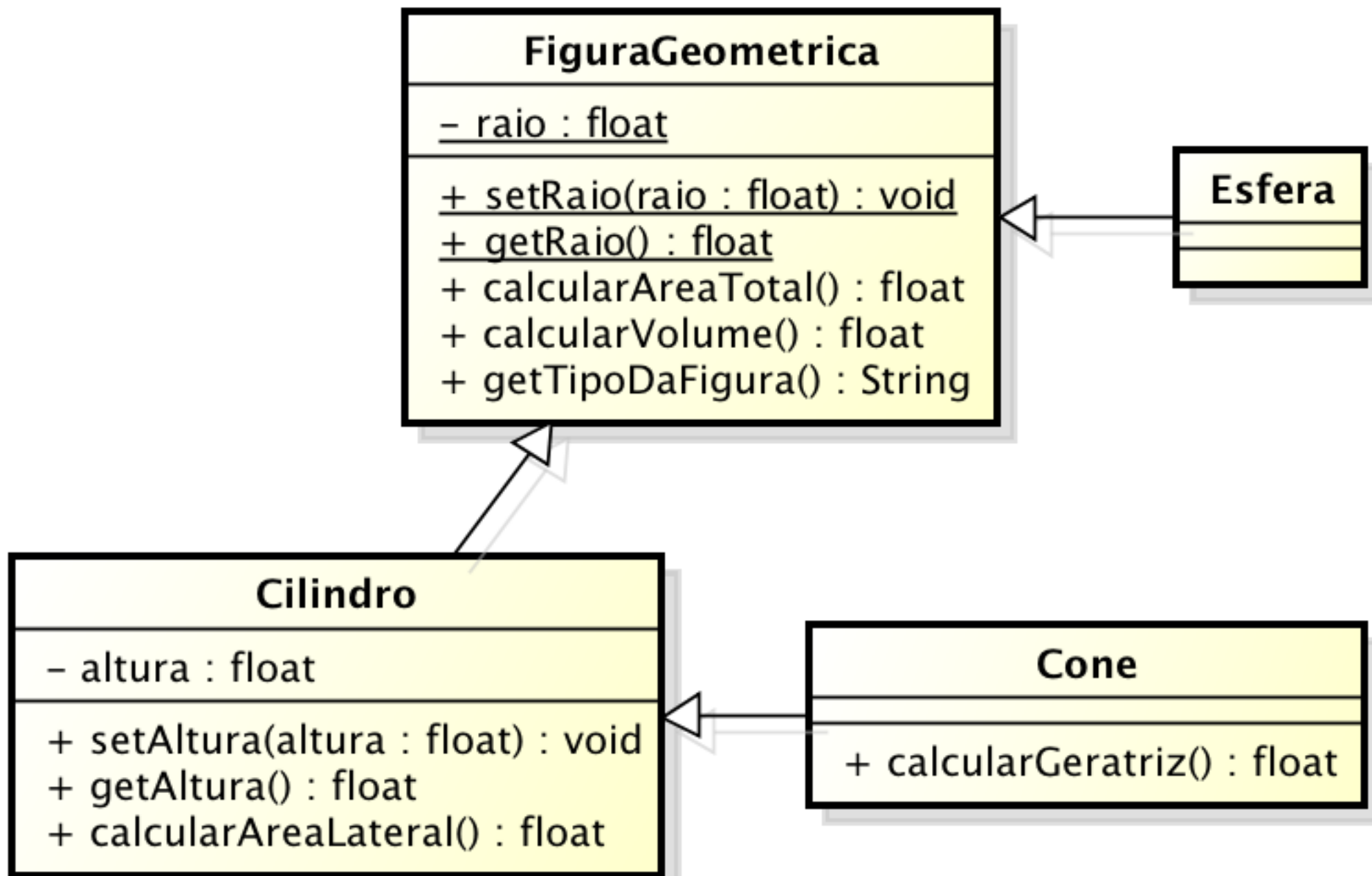
CLASSES COM HERANÇA₃

O atributo **raio** e os métodos **obter (get) raio** e **atribuir (set) raio** – cabeçalho e definição, são elementos comuns as três classes e portanto serão implementados na classe pai e herdados pelas classes filhos, evitando assim a duplicação destes códigos.

CLASSES COM HERANÇA₃

Porém uma mudança em um método na classe pai acarreta uma mudança na classe filho – herança gera classes fortemente acopladas.

CLASSES COM HERANÇA



CLASSES COM HERANÇA₃

Observando as classes notamos que os **métodos volume, tipo da figura e área total** também são comuns – **cabeçalho**, porém a forma de cálculo destes são diferentes em cada uma das classe.

CLASSES COM HERANÇA₃

Quando no processo de herança existem métodos que possuem o mesmo cabeçalho – mesmo **nome e lista de parâmetros**, porém com comportamento diferente, observa-se o conceito de ***Polimorfismo***.

POLIMORFISMO

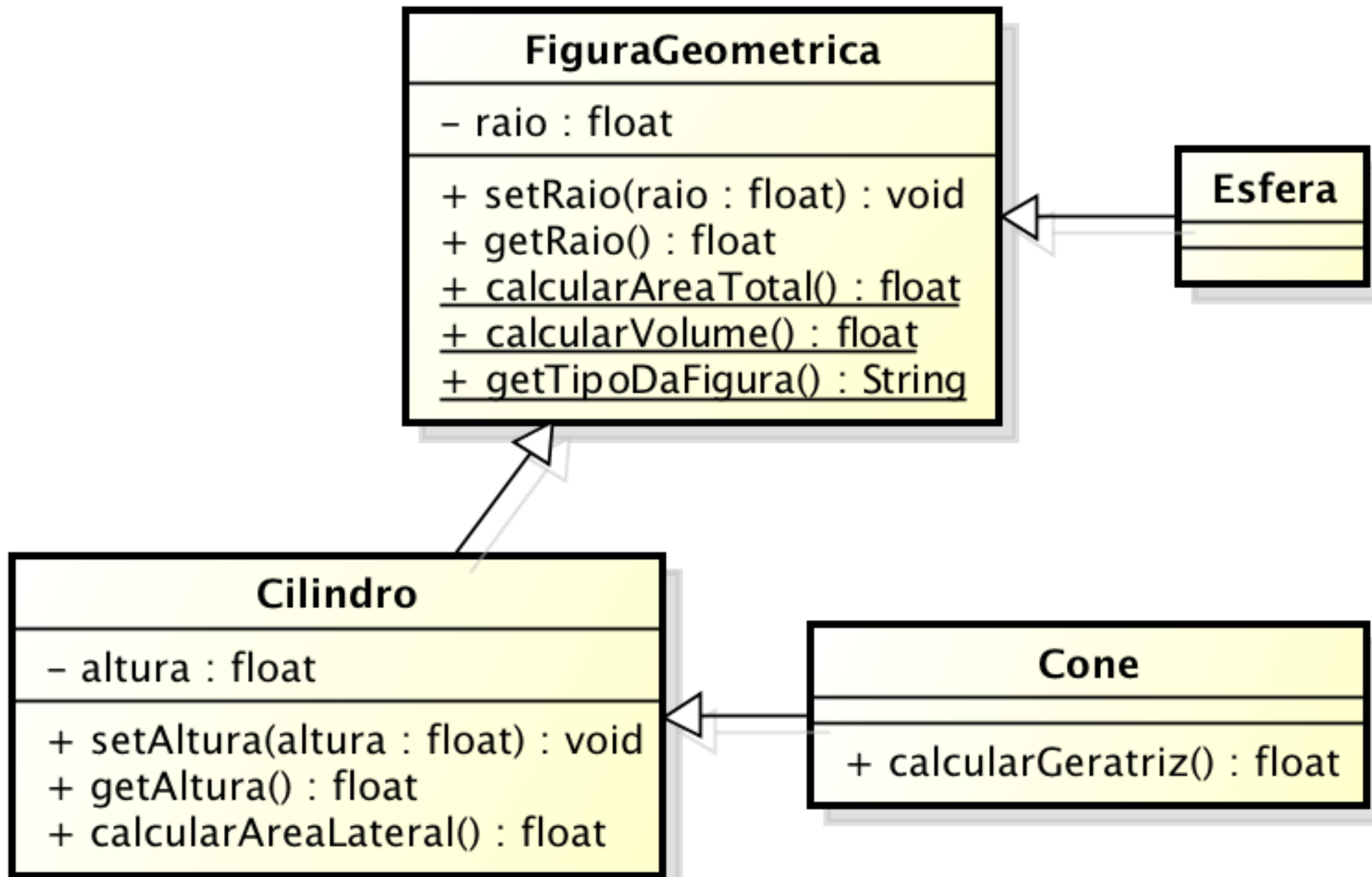
A palavra *polimorfismo* significa muitas formas, e representa o fato de uma determinada característica ser diferente em cada filho.

Polimorfismo significa que a mesma operação pode se comportar de forma diferente em classes diferentes.

POLIMORFISMO

Para os métodos **volume**, **tipo da figura** e **área total** observa-se que estes possuem o mesmo cabeçalho, mesmo ***nome e lista de parâmetros***, porém o forma de cálculo do volume, tipo da figura e da área total nas classes esfera, cilindro e cone ocorrem de forma diferente. Portanto aplica-se a eles o conceito de polimorfismo em suas várias formas.

POLIMORFISMO

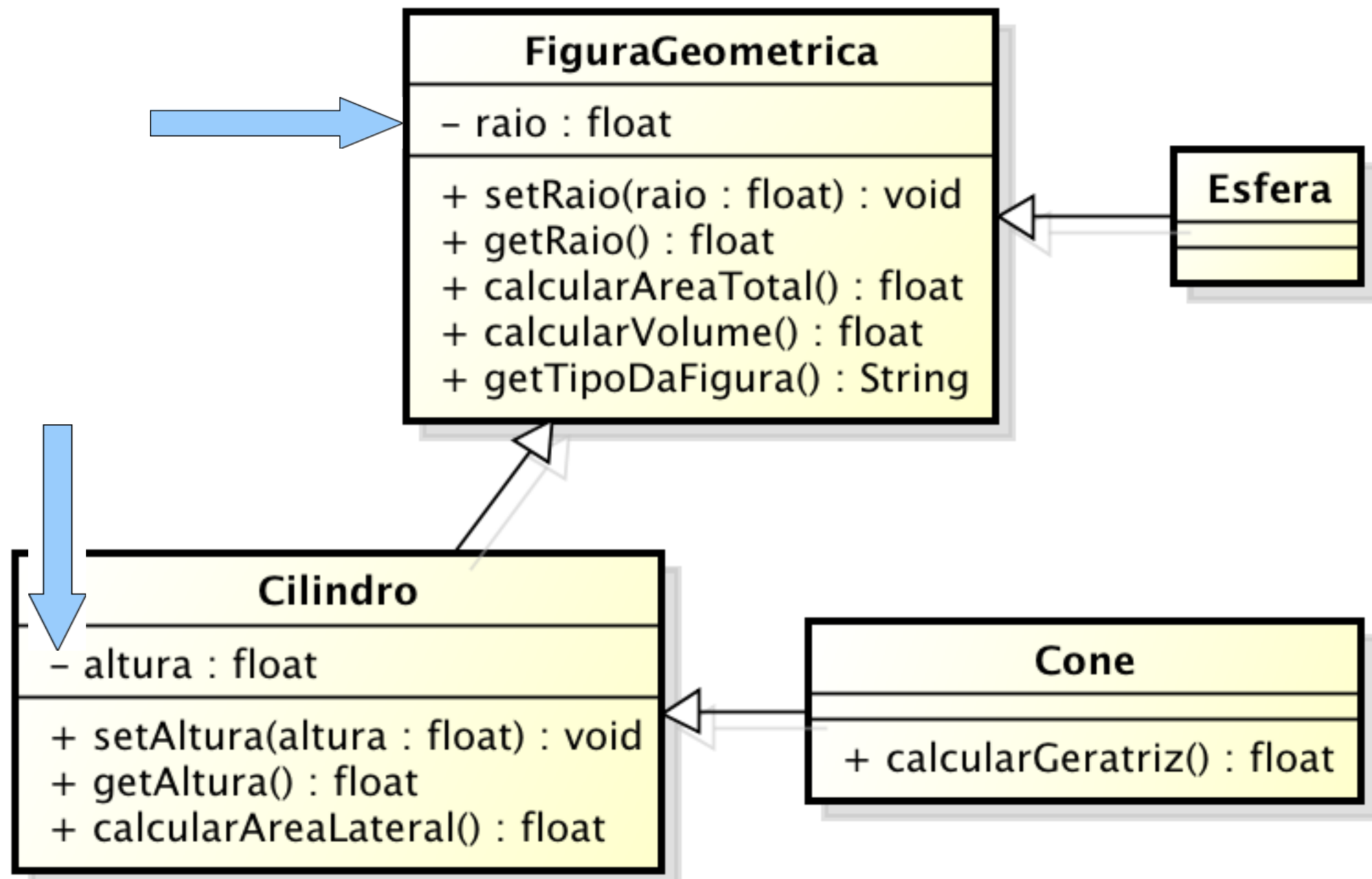


CLASSES COM HERANÇA₃

A visibilidade dos atributos raio e altura das classes Figuras Geométricas e Cilindro são privadas (-), portanto o acesso a estes somente é possível através dos métodos obter (get) e atribuir (set).

As classes filhos não podem acessar diretamente os atributos das classes pais quando estes são privados.

CLASSES COM HERANÇA

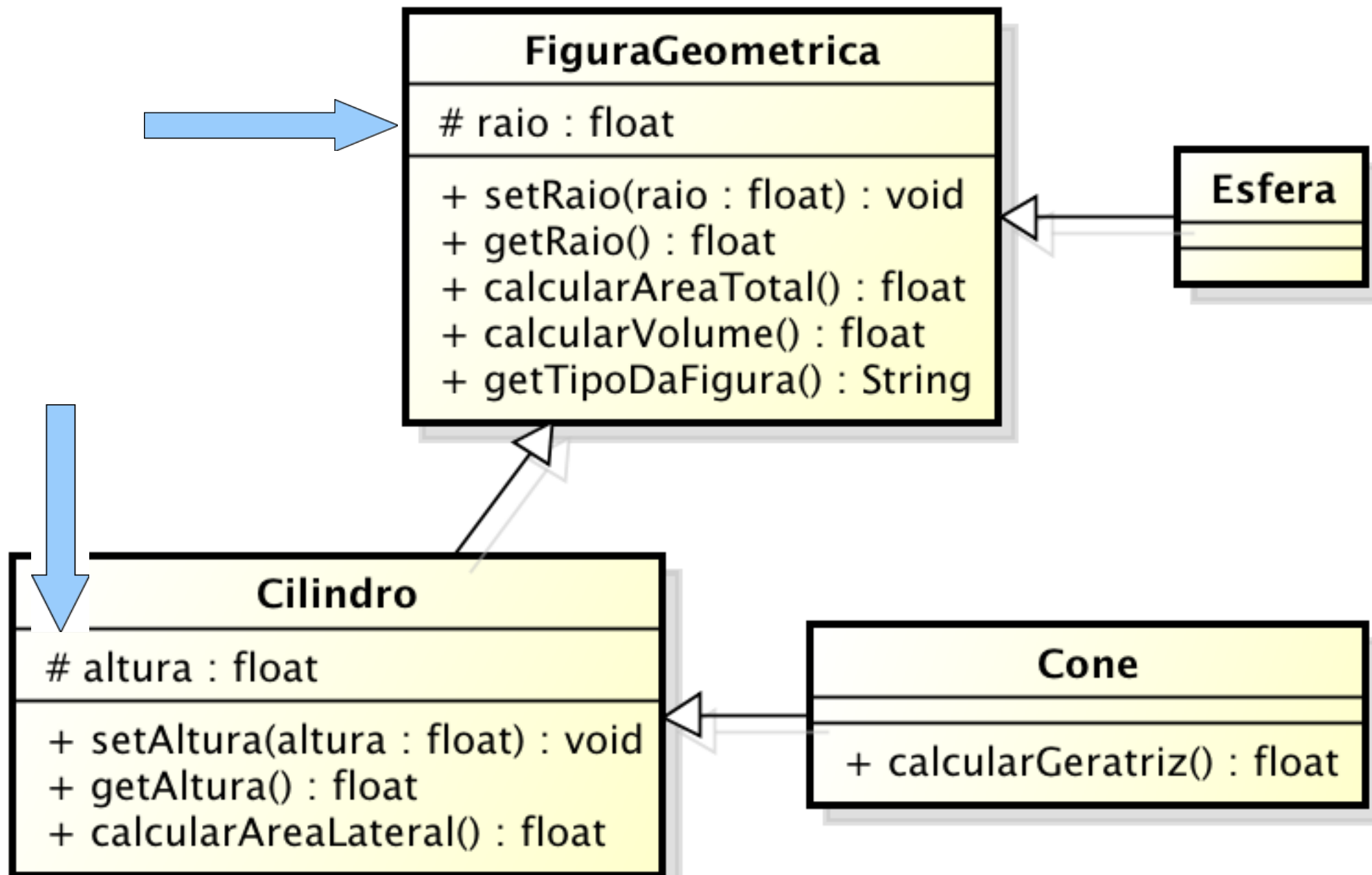


CLASSES COM HERANÇA₃

Mudando a visibilidade dos atributos raio e altura das classes Figuras Geométricas e Cilindro para *protegidas (#)*, as classes filhos poderão acessar estes de forma direta e não somente através dos métodos obter (get) e atribuir (set).

As classes filhos podem acessar diretamente os atributos das classes pais quando estes são protegidos.

CLASSES COM HERANÇA



HERANÇA COM INTERFACE

