

# Classes – Herança Prática

Prof. Gabriel Barbosa da Fonseca

Email: [gbfonseca@sga.pucminas.br](mailto:gbfonseca@sga.pucminas.br)

# Sintaxe

Supondo que temos uma classe **mamífero** com os atributos **peso e cor**, e com o método **caminhar**, criaremos uma classe **cachorro** que herda de mamífero.

```
class Mamifero{  
    public:  
    string cor;  
    float peso;  
    void caminhar(int metros){  
        cout<<"Caminhando "<<metros<<" metros";  
    }  
};
```

```
class Cachorro : Mamifero{  
    public:  
    void latir(){  
        cout<<"\nAU AU!";  
    }  
};
```

# Sintaxe

A sintaxe para a declaração de uma classe filha é a seguinte:

```
class classe_derivada : [<acesso>] classe_base {  
    //corpo da classe derivada  
}
```

```
class Cachorro : Mamifero{  
  
    public:  
    void latir(){  
        cout<<"\nAUAU!";  
    }  
};
```

Repare que temos o operador ":" ( dois pontos ) como elo entre as duas classes. Este operador promove o "parentesco" entre as duas classes quando é usado na declaração de uma classe derivada.

# Sintaxe

O termo [<acesso>] é opcional, mas se estiver presente deve ser public, private ou protected. Ele define o grau de visibilidade dos membros da classe base quando a classe derivada precisar acessá-los. Exemplos:

```
class Cachorro : public Mamifero{
```

```
};
```

```
class Cachorro : private Mamifero{
```

```
};
```

```
class Cachorro : protected Mamifero{
```

```
};
```

# Política de acesso

Quando uma classe herda outra, os membros da classe base são incorporados como membros da classe derivada.

O acesso dos membros da classe base à classe derivada é determinado pelo especificador de acesso: **public**, **private** e **protected**.

Por "default" (padrão) temos o **private**.

# Política de acesso

- Classe base herdada como **public**:
  - Membros públicos (public) da classe base: É como se **copiássemos** os membros da classe base e os colocássemos como "public" na classe derivada. No final, eles permanecem como públicos.
- Membros privados (private) da classe base: Os membros estão presentes na classe, porém **ocultos como privados**. Desta forma as informações estão presentes, mas só podem ser acessadas através de **funções públicas ou protegidas da classe base**.
- Membros protegidos (protected) da classe base: Se tivermos membros protegidos (protected) na classe derivada, eles se comportam como se tivessem sido **copiados para a classe derivada como protegidos (protected)**.

# Política de acesso

- Classe base herdada como **private**:
  - Membros públicos (public) da classe base: Os membros se comportam como se tivessem sido **copiados como privados (private)** na classe derivada.
- Membros privados (private) da classe base: **Os membros estão presentes na classe, porém ocultos como privados.**
- Membros protegidos (protected) da classe base: Os membros se comportam como se tivessem sido **copiados como privados (private)** na classe derivada.

# Política de acesso

- Classe base herdada como **protected**:
  - Membros públicos (public) da classe base: Se comportam como se tivéssemos **copiado-os como protegidos** (protected) na classe derivada
- Membros privados (private) da classe base: **Os membros estão presentes na classe, porém ocultos como privados.**
- Membros protegidos (protected) da classe base: Se comportam como se estivéssemos **copiado-os como protegidos** (protected) na classe derivada.



# Política de acesso

Material de estudos complementar:

[https://pt.wikibooks.org/wiki/Programar\\_em\\_C%2B%2B/Heran%C3%A7a](https://pt.wikibooks.org/wiki/Programar_em_C%2B%2B/Heran%C3%A7a)