

# LINGUAGEM DE PROGRAMAÇÃO C++

# C++

## Revisão

**Controle de acesso nos Métodos e Atributos da classe:**

**public:** elementos acessíveis internamente e externamente a classe;

**private:** elementos acessíveis somente no interior da classe;

**protected:** elementos acessíveis internamente a classe e acessíveis também as classes filhas.

# C++

## Revisão

### Forma geral da classe

```
class nome_da_classe
```

```
{
```

```
    descrição da classe;
```

```
};
```

# Métodos Estáticos e Normais

**Quando um método é static ou normal na classe pai, caso a classe filho faça uma chamada a este, sempre será referenciado o método da classe pai.**

# C++

## MÉTODOS VIRTUAIS PUROS

Forma geral:

**virtual** tipo **nomeDoMétodo**(argumentos) = 0;

Métodos virtuais puros não possuem sua implementação descrita junto da sua declaração. Portanto a forma de agir deste método não é definido na classe.

# C++

## MÉTODOS VIRTUAIS PUROS

Uma classe com um ou mais métodos virtuais puros é uma *classe abstrata* e nenhum objeto desta classe pode ser criado.

Uma classe abstrata pode ser usada somente como uma interface e/ou como uma classe base para outras classes herdarem.

# C++

## FORMA GERAL PARA HERANÇA

```
class nomeClasseFilho : acesso nomeClassePai  
{  
    descrição da classe;  
};
```

onde **acesso**: **public**, **protected** e **private**.



## FORMAS DE ACESSO DA CLASSE NA HERANÇA:

**private** – todos os membros public e protected da classe básica tornam-se membros private da classe derivada;

**public** – todos os membros public e protected da classe básica tornam-se public e protected na classe derivada

**protected** – todos os membros public e protected da classe básica tornam-se os membros protected da classe derivada



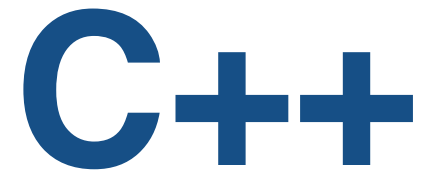
# C++

## HERANÇA

**Para que uma classe filho seja instanciada é necessário o acionamento do construtor da classe pai antes que se conclua a execução do construtor da classe filho.**

**A ordem de criação das instâncias das classes é primeiro os pais depois o filho.**

**A ordem de destruição das instâncias das classes é primeiro o filho depois os pais.**



## **CONSTRUTOR DO FILHO**

**Forma geral:**

**construtorClasseFilho(lista de argumentos) :**

**construtorClassePai(lista de argumentos),**

**atributo1(valor1),atributo2(valor1),...,atributoN(valorN)**

**{**

**descrição da classe;**

**}**

# C++

## FUNÇÕES VIRTUAIS

Forma geral:

**virtual** tipo nomeDaFunção (*argumentos*);

Um método virtual possui referências diferentes no pai e no filho. Assim podemos escolher quais dos métodos desejamos usar: o do pai ou o do filho.

# VARIÁVEL POLIMÓRFICA

**Existe a possibilidade de se acessar os elementos de uma classe filho, através de variáveis do tipo da classe pai.**

**Este conceito é chamado de variável polimórfica.**

# VARIÁVEL POLIMÓRFICA

**Na linguagem de programação C++ as variáveis polimórficas podem ser criadas de duas formas:**

- ✓ através de um ponteiro da classe pai;**
- ✓ através de uma referência da classe pai.**