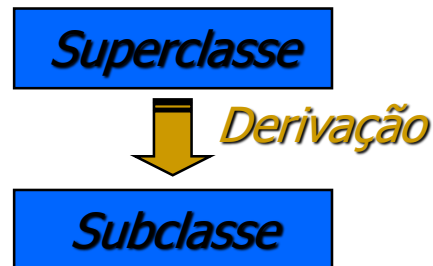


Paradigma Orientado a Objetos

Profª Maria Adelina Raupp Sganzerla
Paradigmas de Linguagens de Programação
Ulbra Gravataí – 2016/2

Herança

- Reutilização de código e possibilidade da incorporação de novas funcionalidades a este
- Subclasse (classe derivada) herda variáveis e métodos públicos da superclasse (classe mãe)



Herança

- Categorias de controle de acesso:
 - **Público:** uma subclasse pode utilizar todas as entidades da classe
 - **Privado:** define entidades ocultas, ou seja, a subclasse não as enxerga
 - **Protegido:** permite que as subclasses enxerguem as entidades protegidas como públicas; para as demais classes, tais entidades são privadas

Herança em C++

- Definição 1: todos os membros públicos da superclasse são também membros públicos da subclasse

```
class <subclasse>: public <superclasse> {  
    <declarações subclasse>  
};
```

- Definição 2: todos os membros públicos da superclasse são herdados como membros privados da subclasse

```
class <subclasse>: <superclasse> {  
    <declarações subclasse>  
};
```

Herança

- Tipos de herança:

- **Simples**

- Subclasse possui apenas uma classe mãe (superclasse)
- Relações expressas através de árvores de derivação

- **Múltipla**

- Subclasse possui mais de uma classe mãe
- Relações expressas através de grafos
- Exemplo:

```
class <subclasse>: public <superclasse_1>, public <superclasse_2> {  
    <declarações subclasse>  
};
```