

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Currais Novos

Programação Orientada a Objetos

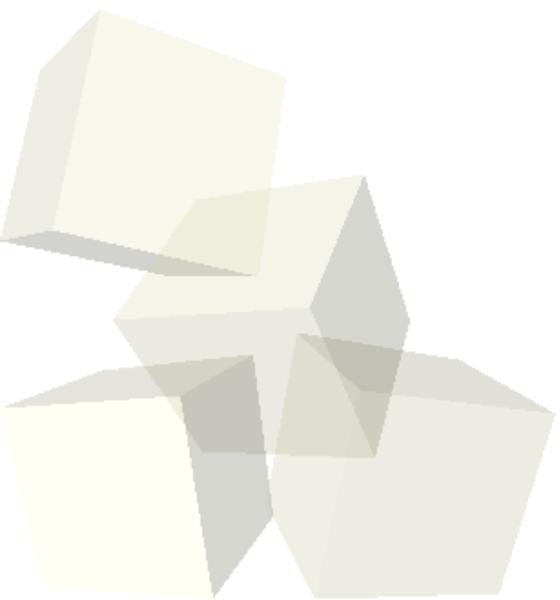
Aula IV – Herança

Prof.: Bruno E. G. Gomes
IFRN



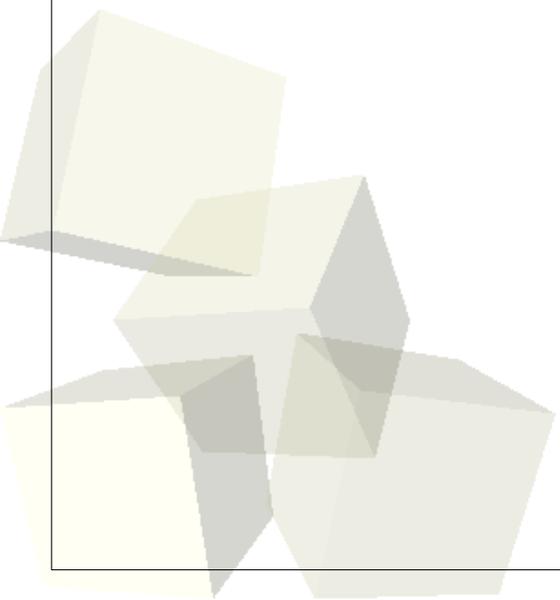


- O termo herança tem significado semelhante em diversos contextos em nossa vida
 - ◆ Por exemplo, na biologia





- Definir o conceito de herança em POO
- Implementar o relacionamento de herança em C++



- Definição de uma classe a partir de outra
 - ◆ Classe que herda recebe atributos e métodos da “*classe mãe*”
 - ◆ relacionamento “*é um tipo de*” entre classes/objetos
- Herança promove:
 - ◆ Reuso de código
 - ◆ Melhor distribuição de responsabilidades entre classes
 - ◆ Possibilidade de substituição dinâmica de objetos sob herança

Exemplo

abstrato

Transporte

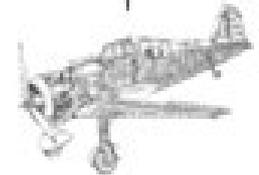
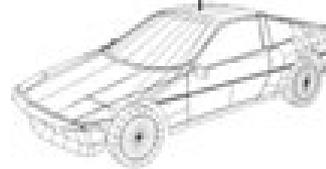
• *capacidade*

Aquático

Terrestre

• *número de rodas*

Aéreo



Barco

Automóvel

• *cor*

• *número de portas*

• *placa*

Avião

concreto

Questões sobre o exemplo anterior

- 1) Quais são os atributos da classe Terrestre? E da classe Automóvel?
- 2) Defina atributos para a classe Aéreo.
- 3) Defina atributos para a classe Avião.
- 4) Crie dois métodos para a classe terrestre (que não sejam *get* e *set*)
- 5) Declare ao menos 1 método para a classe automóvel.

- Diversas figuras geométricas são polígonos:
 - ♦ *Triângulo, quadrado, retângulo, pentágono, hexágono, ...*
- Elas possuem propriedades em comum:
 - ♦ *lados, vértices, ângulos, etc.*
- Também possuem comportamento comum:
 - ♦ *Cálculo de área, obtenção do número de lados, etc.*
- Polígono é um conceito abstrato
 - ♦ Representação concreta: triângulo, quadrado, etc.
- *Um triângulo é um tipo de polígono, portanto a classe triângulo pode herdar as características básicas e o comportamento de um polígono.*

Termos utilizados para herança

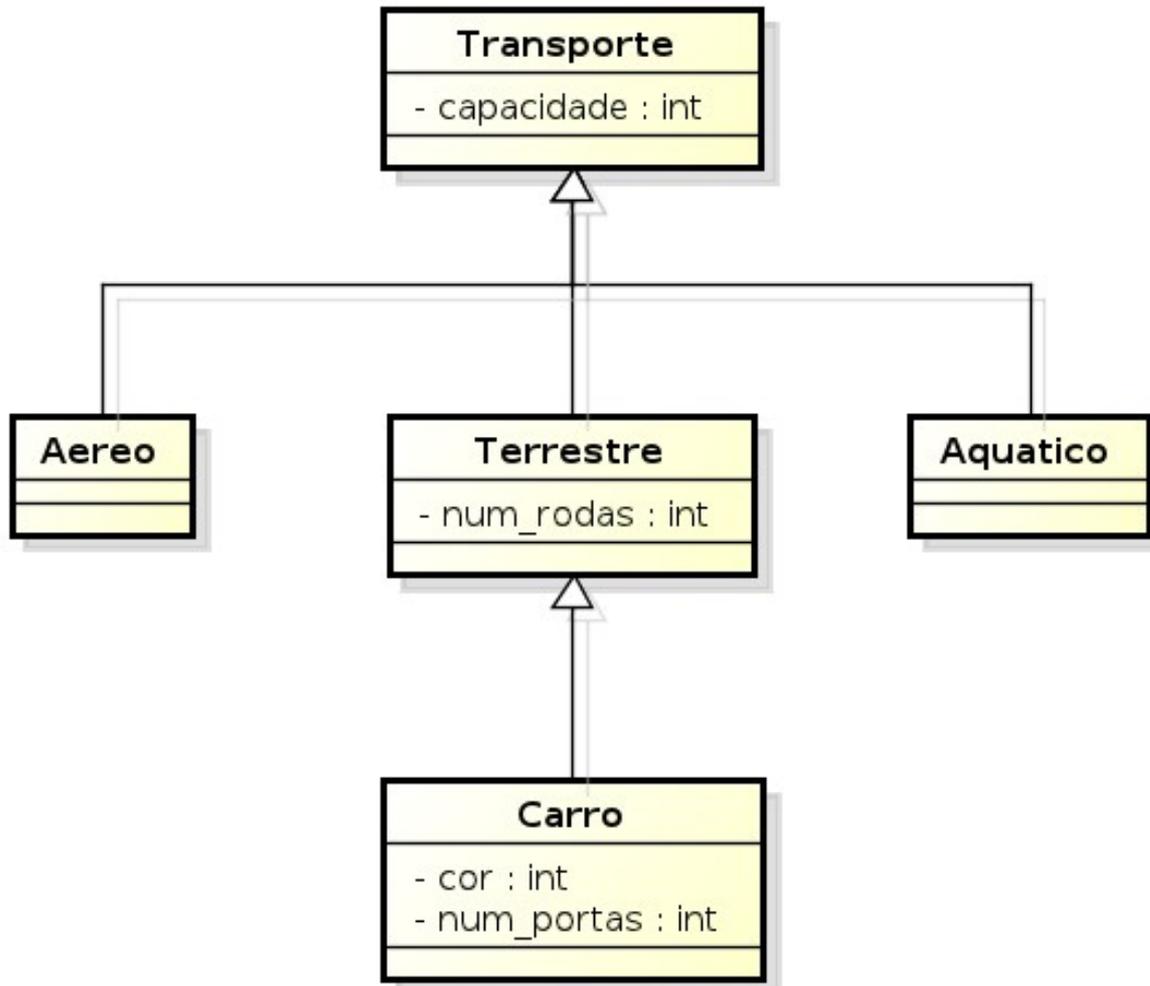
Classes que fornecem herança	Classes que herdam de outras
Superclasse	Subclasse
Mãe	Filha
Tipo	Subtipo

■ Outros

- ◆ Classe raiz – classe no topo da hierarquia de herança
- ◆ Classe folha – classe sem filhas

Herança em UML

- Representação da herança:
 - ♦ Uma seta com um triângulo branco na ponta, apontando da subclasse para a sua classe mãe.



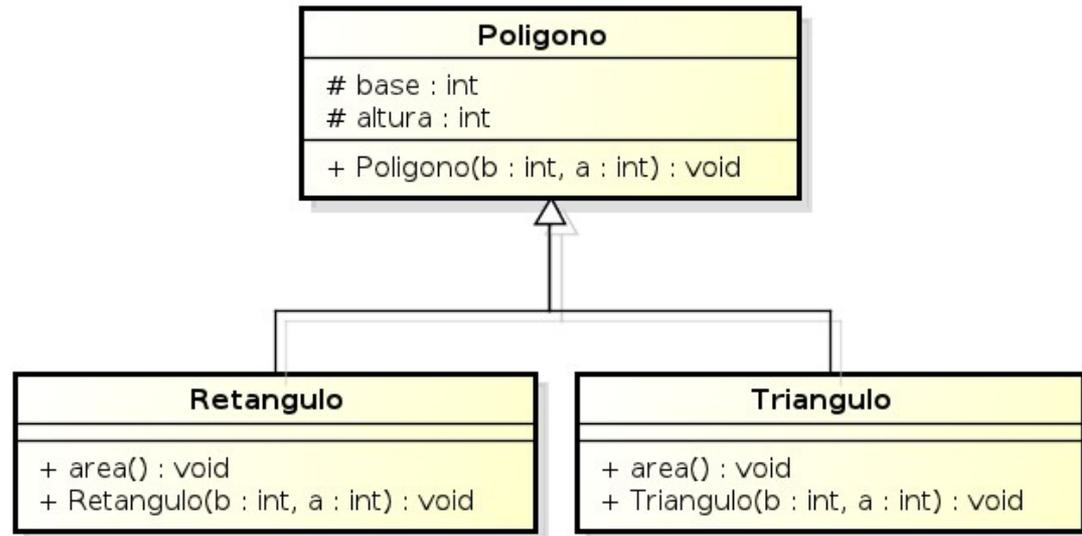
- Para herdar de uma outra classe, faça:

class <nome> **:** <acesso> <nome_superclasse>

- Acesso protegido (*protected*):
 - ♦ Membros da classe são acessíveis apenas por ela e suas subclasses
 - ♦ Em UML, é representado por #
- Na declaração, <acesso> define o nível *mínimo* de acesso aos membros da *superclasse*:
 - ♦ *public* – todos os membros terão o mesmo acesso na subclasse;
 - ♦ *protected* – todos os membros públicos (*public*) são convertidos para protegidos (*protected*) na subclasse;
 - ♦ *private* – todos os membros serão privados na subclasse.



Exemplo: classe Retângulo

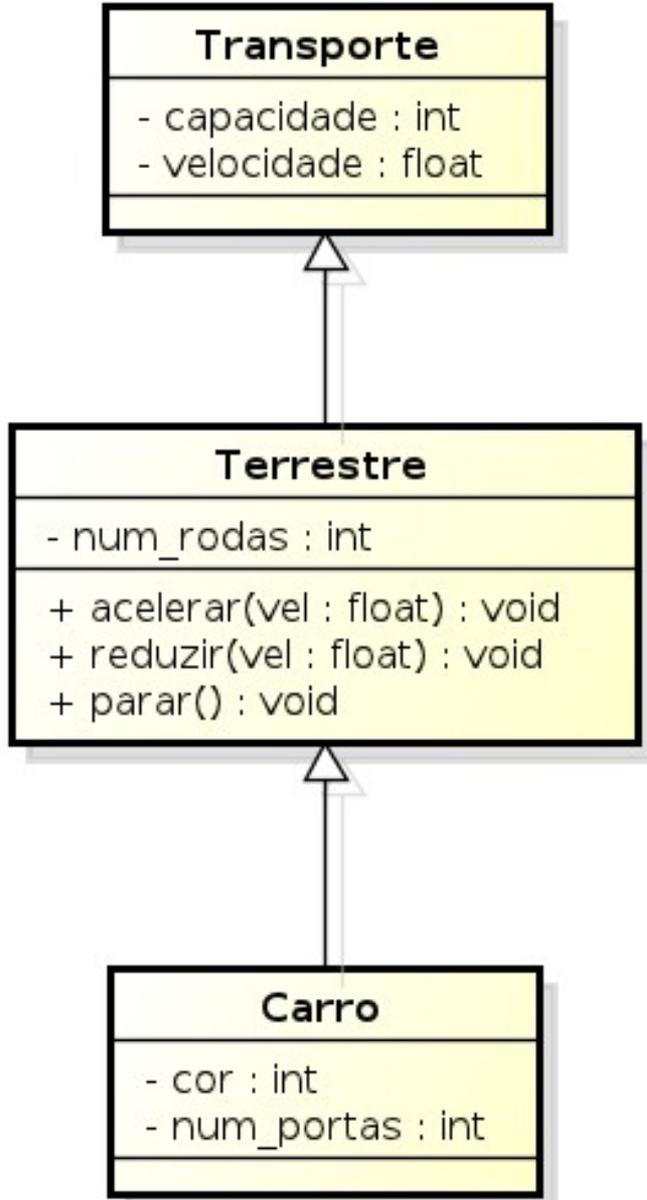


```
class Poligono {  
protected:  
    int base, altura;  
public:  
    Poligono(int base, int alt);  
};
```

```
class Retangulo : public Poligono {  
public:  
    Retangulo(int base, int alt);  
    int area();  
};
```

```
class Triangulo : public Poligono{  
public:  
    Triangulo(int base, int alt);  
    int area();  
};
```





- Implemente em C++ as classes descritas no diagrama
- **Métodos:**
 - *Acelerar* – aumenta a velocidade até o valor “vel”
 - *Reduzir* – reduz a velocidade até o valor “vel”
 - *Parar* – reduz a velocidade até o veículo parar (velocidade = 0.0)
- **Observações:**
 - *capacidade* – número máximo de pessoas que podem ser transportadas
 - *velocidade* deve ser inicializada com o valor 0.0;
 - Em cada método, a velocidade deve ser reduzida ou aumentada de 5 em 5 km/h até atingir o valor desejado.