

Associação entre classes

Mateus Raeder



Associação entre classes

- Um objeto é a representação de uma entidade do mundo real
- Instanciar é criar um novo objeto
- Por exemplo, Floribela é uma instância da classe Pessoa
- No BlueJ, como criamos uma instância de uma classe?
clicando com o direito na classe e escolhendo new...

Logo, um novo objeto é criado com a palavra reservada **new**

Associação entre classes

- Vamos imaginar que temos uma classe chamada Dado:

```
public class Dado{  
    private int lados;  
    public Dado(int nLados){  
        lados = nLados;  
    }  
    public void jogaDado(){  
        System.out.println("Jogando o dado...");  
    }  
}
```

- Como criamos um novo objeto do tipo Dado?

Associação entre classes

- Objetos diferentes podem se conhecer, podem conversar uns com os outros
- **Mas como objetos podem conversar?**
 - Imagine 2 pessoas. Para elas conversarem, eles devem se conhecer antes. Logo, para dois objetos conversarem, eles devem se *conhecer*
 - Vamos imaginar, então, que a Floribela quer jogar o dado
 - Dois objetos diferentes, de duas classes diferentes
 - A classe Dado possui um método que chama-se *jogaDado()*

- **Mas como a Floribela pode acessar este método?**
- Somente quando ela *conhecer* o objeto dado, ou seja, quando ela tiver uma **instância da classe Dado**
- Assim, vamos poder dizer que uma pessoa tem um dado

Associação entre classes

- Mais do que isso, uma Pessoa (classe) tem um atributo do tipo Dado (classe), e este atributo (um objeto) deve ser declarado e inicializado
- Logo, na classe Pessoa teremos:

```
public class Pessoa  
{  
    private String nome;  
    ...  
    private Dado meuDado;  
    ...  
}
```

Declaração de
uma instância da
classe Dado,
associando a
classe Dado à
classe Pessoa

Associação entre classes

- Para que este atributo se torne utilizável, devemos inicializá-lo, informando ao Java que trata-se de um novo objeto
- Assim, devemos utilizar o new, conforme dito anteriormente
- No construtor de Pessoa, teremos então:

```
public Pessoa()  
{  
    ...  
    meuDado = new Dado(6);  
    ...  
}
```

Inicialização
do objeto

Construtor da classe
Dado pede o
número de lados!!

Associação entre classes

- Como o atributo meuDado já foi inicializado, e trata-se de um objeto da classe Dado, ele possui todos os atributos e métodos desta classe
- A pessoa então, pode acessar todos estes métodos e atributos de um objeto dado. A pessoa tem um dado e pode jogá-lo.
- Como este acesso é realizado?

Através do operador . (ponto)

```
public Pessoa(){  
    meuDado = new Dado(6);  
}
```

```
public void jogaDado(){  
    meuDado.jogaDado();  
}
```

Significa que quero
acessar (executar) o
método chamado
jogaDado() do objeto
meuDado

Associação entre classes

- Uma pessoa não precisa ter necessariamente 1 dado

```
public class Pessoa
{
    private Dado meuDado1;
    private Dado meuDado2;
    private Dado meuDado3;
    ...

    public Pessoa(...)
    {
        meuDado1 = new Dado(6); //cria um dado de 6 lados
        meuDado2 = new Dado(13); //cria um dado de 13 lados
        meuDado3 = new Dado(6); //cria outro dado de 6 lados
        ...
    }
}
```

Associação entre classes

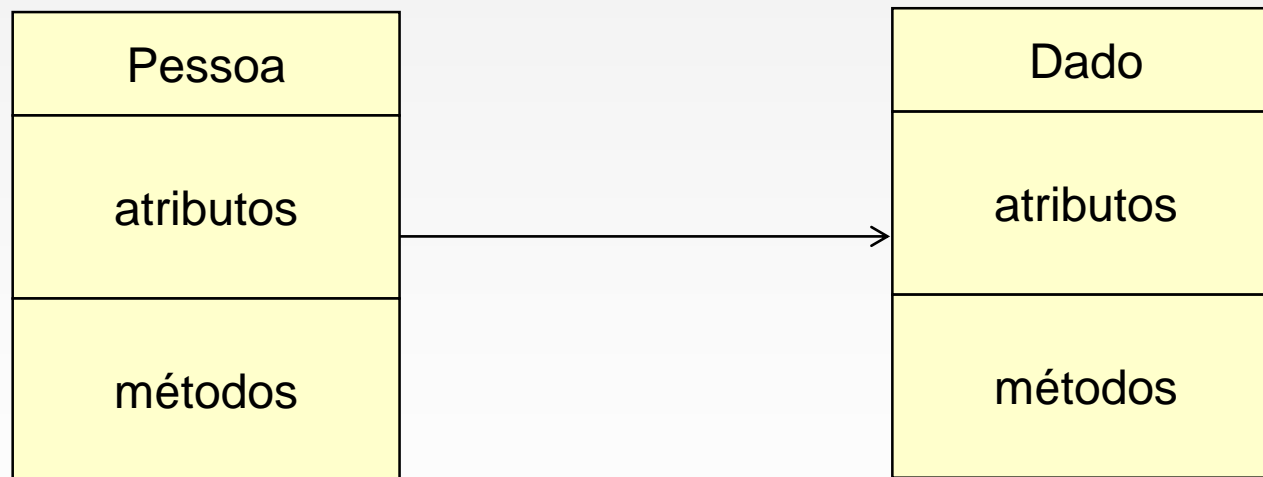
- A pessoa então, pode jogar os dados de acordo com sua vontade

```
public Pessoa(...)
{
    meudado1 = new Dado(6); //cria um dado de 6 lados
    meudado2 = new Dado(13); //cria um dado de 13 lados
    meudado3 = new Dado(6); //cria outro dado de 6 lados
    ...
}

public void jogarDados()
{
    meuDado1.jogaDado();
    meuDado2.jogaDado();
    meuDado3.jogaDado();
}
```

Associação entre classes

- O diagrama UML ilustra o relacionamento entre as classes da seguinte maneira:



O objeto da classe Pessoa conhece o objeto da classe Dado,
mas o contrário não é verdade

Associação entre classes

- Uma associação é um relacionamento estrutural entre duas classes, significando que os objetos de uma classe estão ligados aos objetos de outra
- O objeto pessoa está ligado ao objeto dado

