



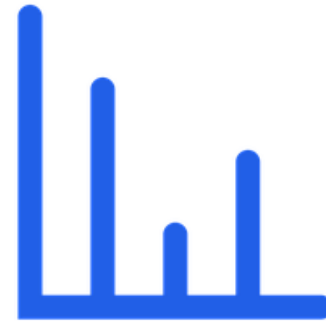
Atributo do tipo Classe

Prof^a. Rachel Reis
rachel@inf.ufpr.br



Problema

O IBGE precisa de uma aplicação para armazenar os seguintes dados dos moradores de Curitiba: nome, endereço, ocupação e data de nascimento.





Problema

- **Usando o conceito de abstração, quais classes devem ser criadas para esta aplicação?**

OIBGE precisa de uma aplicação para armazenar os seguintes dados dos **moradores de Curitiba**: **nome**, **endereço**, **ocupação** e **data de nascimento**.



Moradores de Curitiba

- Tipo dos atributos:
 - Nome → tipo String
 - Ocupação → tipo String
 - Endereço → tipo ??
 - Data de nascimento → tipo ??



Classe Endereço

- Exemplo de atributos:
 - nome da rua
 - número
 - bairro
 - cidade
 - estado
 - cep



Classe Data

- Atributos
 - dia
 - mês
 - ano



Classes

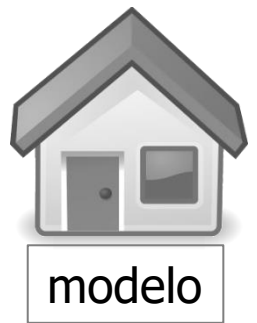
```
public class Endereco{ ... }
```

```
public class Data{ ... }
```

```
public class Pessoa{ ... }
```

```
public class Principal{  
    public static void main (String args[]){  
        ...  
    }  
}
```

```
public class Endereco{  
    // Atributos  
    private String nomeRua;  
    private int numero;  
    private String bairro;  
    private String cidade;  
    private String estado;  
    private String cep;  
  
    // Construtor  
    public Endereco(String nomeRua, int numero)  
    {  
        this.setNomeRua (nomeRua) ;  
        this.setNumero (numero) ;  
    }  
    ...  
}
```




```
public class Data{
```

```
    // Atributos
```

```
    private byte dia;
```

```
    private byte mes;
```

```
    private short ano;
```

```
    // Construtor completo
```

```
    public Data(byte dia, byte mes, short ano)
    {
```

```
        this.setDia(dia);
```

```
        this.setMes(mes);
```

```
        this.setAno(ano);
```

```
    }
```

```
    ...
```

```
}
```

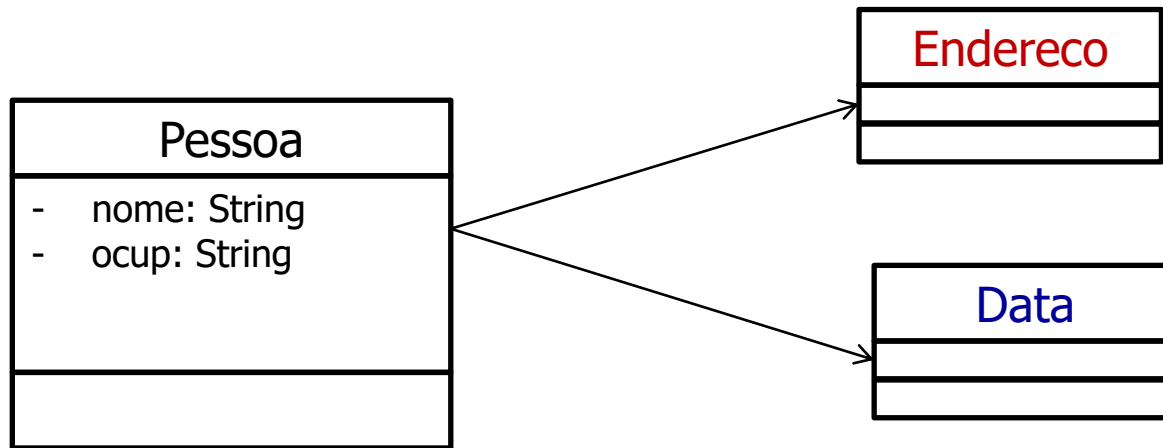


modelo



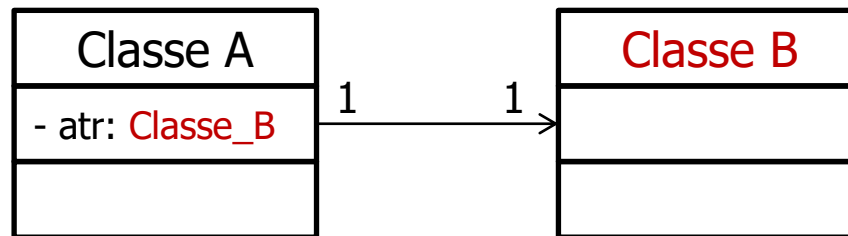
Associação

- Como associar as classes Endereco e Data à classe Pessoa?



Associação - UML

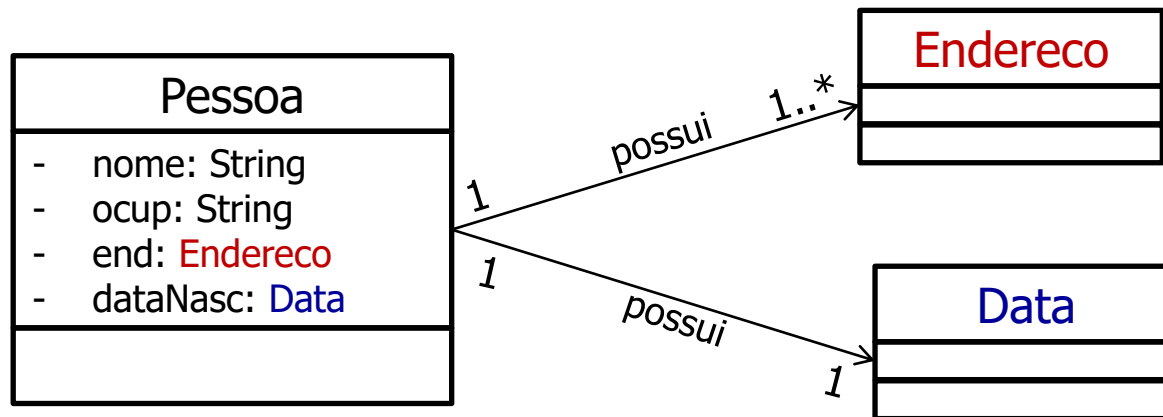
- Associação: indica que uma classe depende de outra.
 - Associação forte: a classe A possui um ou mais atributos do tipo da classe B



- Multiplicidade 1..1: indica que **apenas um** objeto da **classe B** se relaciona com os objetos da classe A.

Diagrama UML

- Qual a multiplicidade das classes Pessoa-Endereço e Pessoa-Data?



```
public class Pessoa{  
    // Atributos  
    private String nome;  
    private String ocup;  
    private Endereco end;  
    private Data dataNasc;  
  
    // Construtores 1 (padrão) e 2 (completo)  
    public Pessoa() {}  
    public Pessoa(String nome, String ocup,  
                   Endereco end, Data dataN) {  
        this.setNome(nome) ;  
        this.setOcup(ocup) ;  
        this.setEnd(end) ;  
        this.setDataNasc(dataNasc) ;  
    }  
    ...  
}
```



modelo

```
public class Pessoa{
```

```
...
```

```
// Outros métodos
```

```
public void imprimir()
```

```
{
```

```
    System.out.println("Nome: " + this.getNome());
```

```
    System.out.println("Ocupacao: " +  
                        this.getOcup());
```

```
    System.out.printf("Rua: %s, %d \n",  
                      this.end.getNomeRua(),  
                      this.end.getNumero());
```

```
    System.out.printf("Nascimento: %d/%d/%d",  
                      this.dataNasc.getDia(),  
                      this.dataNasc.getMes(),  
                      this.dataNasc.getAno());
```

```
}
```

```
}
```



modelo

```
public class Principal{
```

```
    public static void main (String args[]) {
```

```
        Pessoa p1 = new Pessoa();
```

```
        Data nasc = new Data();
```

```
        Endereco end = new Endereco();
```

```
        p1.setNome("Hulk");
```

```
        p1.setOcup("Herói");
```

```
        nasc.setDia((byte) 30);
```

```
        nasc.setMes((byte) 10);
```

```
        nasc.setAno((short) 1985);
```

```
        p1.setDataNasc(nasc);
```

—————> Concretiza associação

```
        end.setNomeRua("A");
```

```
        end.setNumero(50);
```

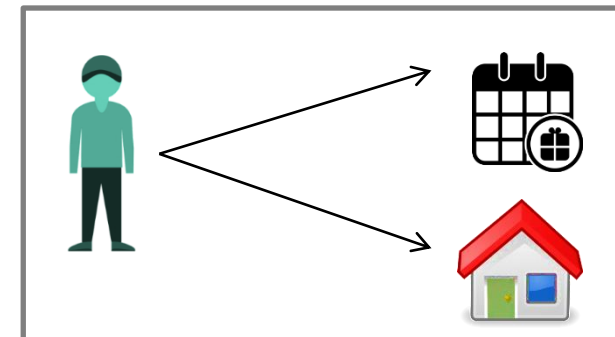
```
        p1.setEnd(end);
```

—————> Concretiza
associação

```
        p1.imprimir();
```

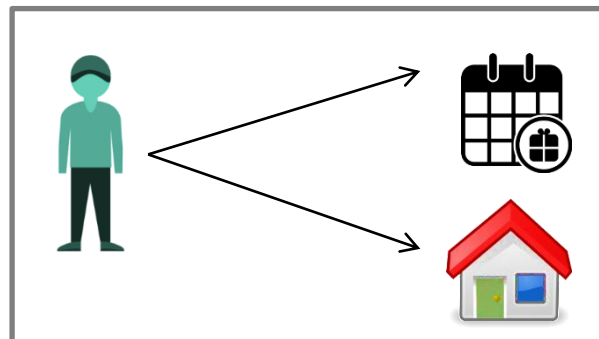
```
    }
```

```
}
```



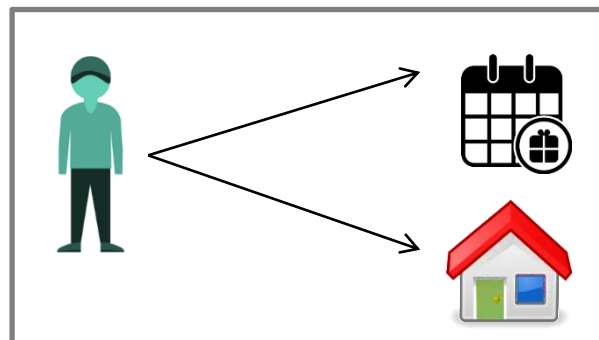
```
public class Principal{  
    public static void main (String args[]) {  
        Pessoa p1 = new Pessoa();  
        p1.setNome("Hulk");  
        p1.setOcup("Herói");  
        p1.setData(new Data((byte)30, (byte)10, (short)1985));  
        p1.setEnd(new Endereco("A", 50));  
        p1.imprimir();  
    }  
}
```

Concretiza associação




```
public class Principal{  
    public static void main (String args[])  
    {  
        Endereco end = new Endereco("A", 50) ;  
        Data nasc = new Data ((byte)30, (byte)10,  
                                (short)1985) ;  
  
        Pessoa p1 = new Pessoa("Hulk", end, "heroi",  
                                nasc) ;  
        p1.imprimir() ;  
    }  
}
```

Concretiza associação



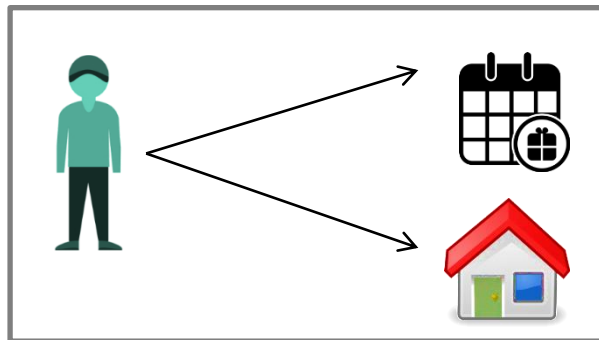
Outro exemplo de construtor

```
public class Pessoa{  
    // Atributos  
    ...
```

```
➡ // Construtor 3  
    public Pessoa(String nome, Endereco end,  
                  String ocup, byte d, byte m, short a)  
    {  
        this.setNome(nome) ;  
        this.setEnd(end) ;  
        this.setOcup(ocup) ;  
        this.dataNasc = new Data((byte)d, (byte)m,  
                                (short)a) ;  
    }  
    ...  
}
```

```
public class Principal{  
    public static void main (String args[])  
    {  
        Endereco end = new Endereco("A", 50) ;  
        Pessoa p1 = new Pessoa("Hulk", end,  
                                "heroi", (byte)30, (byte)10, (short)1985) ;  
        p1.imprimir() ;  
    }  
}
```

Concretiza associação





Para praticar...

- Escreva um construtor 4 para a classe Pessoa, de forma que os campos abaixo sejam passados como parâmetro. Em seguida, refaça a função main() da solução 4.
 - nome
 - nome da rua, número, bairro, cidade, estado, cep
 - ocupação
 - dia, mês e ano de nascimento