



# Classes e Relacionamentos

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

# [ Classe Pública em Java ]

- Declaração de uma classe pública (***public***) deve ser feita em um arquivo de mesmo nome
  - Com a extensão “**.java**”
- Modificador ***public*** permite acesso / chamadas de qualquer outra classe

```
public class Carro {  
    String cor;  
    int velocidadeAtual;  
  
    void acelerar() {}  
    void frear() {}  
}
```

Arquivo Carro.java

# [ Corpo de Classes e Métodos ]

- O corpo de uma classe é delimitado por chaves { }
- O corpo de um método também é delimitado por chaves { }

```
public class Carro {  
    String cor;  
    int velocidadeAtual;  
  
    void acelerar() {...}  
    void frear() {...}  
}
```

# [ Assinatura de Métodos ]

- Modificador + Tipo de Retorno + Nome do Método + Lista de Parâmetros
  - **Modificador:** public | private | protected | ...
  - **Tipo de Retorno:** void | int | String | ...
  - **Nome:**  
acelerar | frear | ...
  - **Parâmetros:**  
delimitado por parênteses

```
public class Carro {  
    private String cor;  
    private int velocidadeAtual;  
  
    public void acelerar() {}  
    public void frear() {}  
}
```


# [ Visibilidade em UML ]

- Pública (+)
  - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
  - Somente a classe ou sub-classes terão acesso
- Privada (-)
  - Somente a classe terá acesso

# [ Visibilidade UML x Java ]

Pessoa
# nome - email
+ enviarMensagem()

```
public class Pessoa {  
  
    protected String nome;  
    private String email;  
  
    public void enviarMensagem() {  
        ...  
    }  
}
```



# Associação e Herança

# [ Associação e Composição ]

- Uma classe pode ter referências para objetos de outras classes
  - Exemplo, para saber quando despertar, um relógio tem que saber a hora





# [ Associação em Java ]

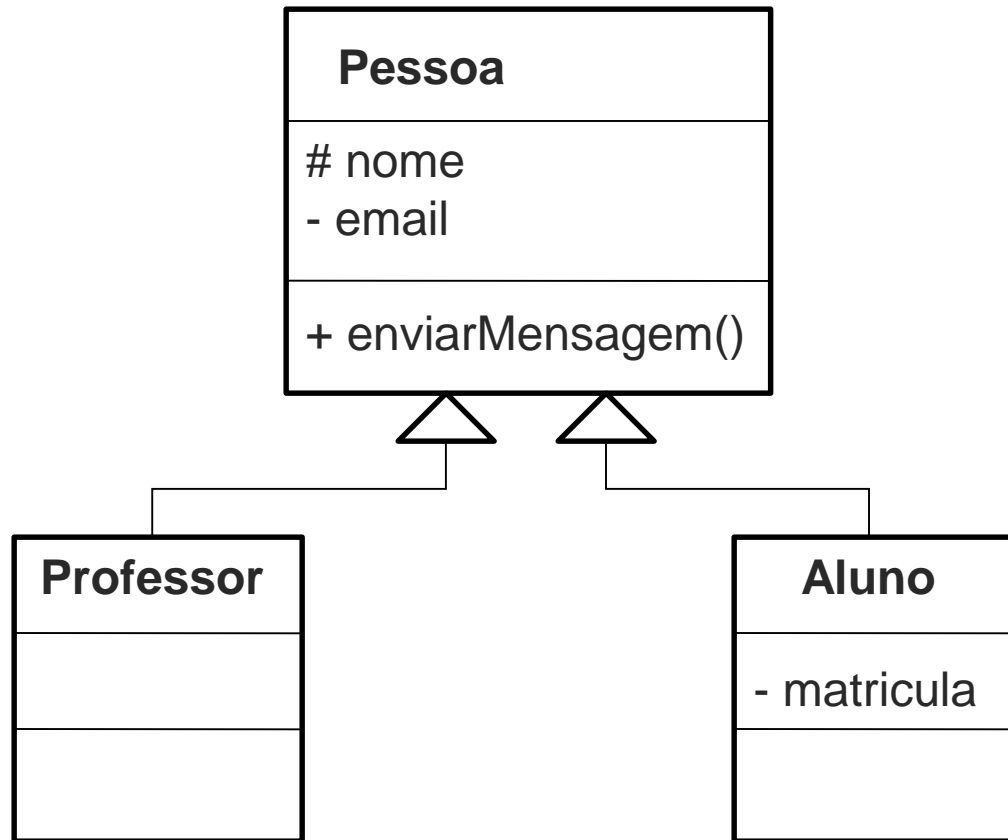
```
public class AlarmClock {  
  
    protected Time alarm;  
  
    public void setAlarm(Time1 time) { alarm = time; }  
    ...  
}
```



# [ Especialização / Generalização ]

- Identificar super-classe (geral) e sub-classes (especializadas)
  - Semântica “é um”
  - Tudo que a classe geral pode fazer, as classes específicas também podem
- Todos os atributos e métodos definidos na classe-mãe são herdados pelas classes-filhas

# [ Herança em UML ]



# [ Herança em Java ]

```
// Pessoa.java
public class Pessoa {
    protected String nome;
    private String email;
    public void enviarMensagem() { ... }
}
```

```
// Aluno.java
public class Aluno extends Pessoa {
    private String matricula;
}
```

```
// Professor.java
public class Professor extends Pessoa {
}
```

# [Membros Protegidos]

- Podem ser acessados por sub-classes mesmo que estejam em pacotes diferentes

```
public class Pessoa {  
    protected String nome;  
    private String email;  
    public void enviarMensagem() { ... }  
}
```



**Acessível às classes  
Aluno e Professor**

```
public class Aluno extends Pessoa { ... }
```

```
public class Professor extends Pessoa { ... }
```

# [ Bibliografia da Aula ]

- DEITEL, H. M.; DEITEL P. J. **Java: Como Programar**, 8a. Edição. Pearson, 2010.
  - Capítulo 8 Classes e Objetos
  - Capítulo 9 Herança