



# Classes Aninhadas

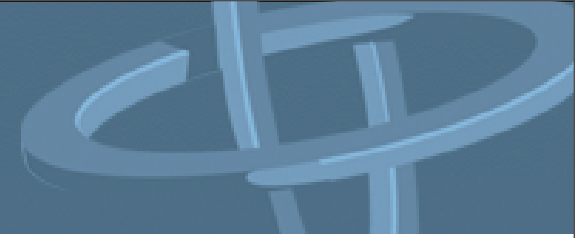


Programação Orientada a Objetos

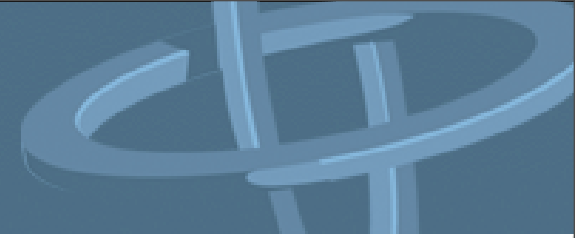


middleware laboratory

Renato Cerqueira  
Departamento de Informática - PUC-Rio

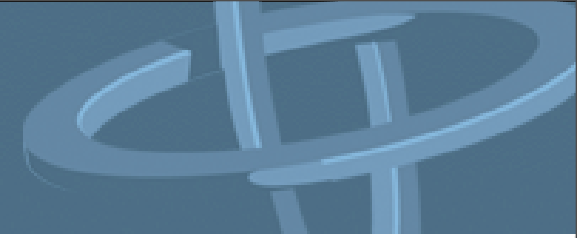


- *é uma classe declarada como membro de outra classe*
- motivação
  - classes cuja a finalidade é exclusivamente auxiliar na implementação de uma outra classe
  - classes que precisam ter acesso a toda a implementação de uma outra classe
- permite que classes e interfaces sejam estruturadas e aninhadas em grupos logicamente relacionados
- permite conectar de forma simples e efetiva objetos logicamente relacionados
- diversas linguagens OO oferecem mecanismos com objetivos semelhantes



- é considerada parte (membro) da classe envolvente
  - pode ser membro *public*, *private*, *protected* ou *package*
- compartilha um relacionamento de *confiança* com a classe envolvente
  - cada uma pode acessar todos os membros da outra
- uma classe aninhada pode ser
  - *estática*: aninhamento de tipos
  - *dinâmica*: aninhamento de instâncias (objetos)

# Exemplo de Aninhamento Estático



```
package p;
```

```
public class A {  
    public static class B {  
        ...  
    }  
}
```

---

```
p.A a = new p.A();  
p.A.B b = new p.A.B();
```

# Exemplo de Aninhamento Estático

```
public class BankAccount {  
    private long number;  
    private long balance;  
    public static class Permissions {  
        public boolean canDeposit, canWithdraw, canClose;  
    }  
    ...  
}
```

---

```
BankAccount.Permissions perm = ac.permissionsFor(owner);
```

# Exemplo de Aninhamento Dinâmico

```
class A {  
    int i;  
    class B {  
        void g() {...}  
    }  
    void f() {...}  
}
```

