Programação Orientada a Objectos

Sérgio Lopes

• Identificar o conceito de programação orientada a objectos • Reconhecer os pilares base: herança, polimorfismo e encapsulamento Objectivos

Ver o problema como um conjunto de interacções entre vários objectos, que se relacionam para atingir um objectivo.

Abordagem Procedimental

Divisão por funções.

Identificar funções, as suas características e relações, dando pouca relevância aos dados.

Uso de variáveis globais, e ponteiros, para trocar informação entre funções. (Problemático!)

Abordagem OO

Identificação de objectos (entidades) das suas características (comportamento) e das relações existentes.

Os dados são importantes, os objectos são identificados a partir dos dados.

Caixas pretas que passam para o exterior apenas o necessário para que as relações funcionem.

Classes

Elemento base da programação orientada a objectos.

```
class MinhaClass {
    //atributos
    private int x;
    //Construtores e Métodos
    ...
}
```

Passagem de características (atributos e comportamento) aos descendentes.

Herança

Conceito de Herança

Só herança simples.

Classes descendentes herdam (têm acesso a) métodos e atributos da classe mãe.

Possibilita a reutilização de código (componentes) e refinação de características do problema.

Bicicletas (cores, rodas, travões, suspensão).

Exemplo

Objectos com múltipla personalidade!

Capacidade de uma variável assumir-se como qualquer um de vários tipos que partilhem a mesma hierarquia.

Polimorfismo

Polimorfismo

Uma variável declarada como sendo de um tipo pode conter valores (referências para objectos) de qualquer tipo descendente.

O comportamento irá adequar-se ao objecto real dentro da variável.

O acesso aos métodos é controlado pelo tipo de declaração.

Bicicletas ...

Exemplo

Controlo de acesso ao estado interno e ao comportamento interno dos objectos.

Encapsulamento

Encapsulamento

Processo que limita o acesso externo ao interior de um objecto. Sempre o acesso externo.

Esconder o estado interno (atributos e métodos) permite que a relação entre os objectos seja menor.

Tipos de Acesso

Acesso **Público**.

Acesso **Protegido**.

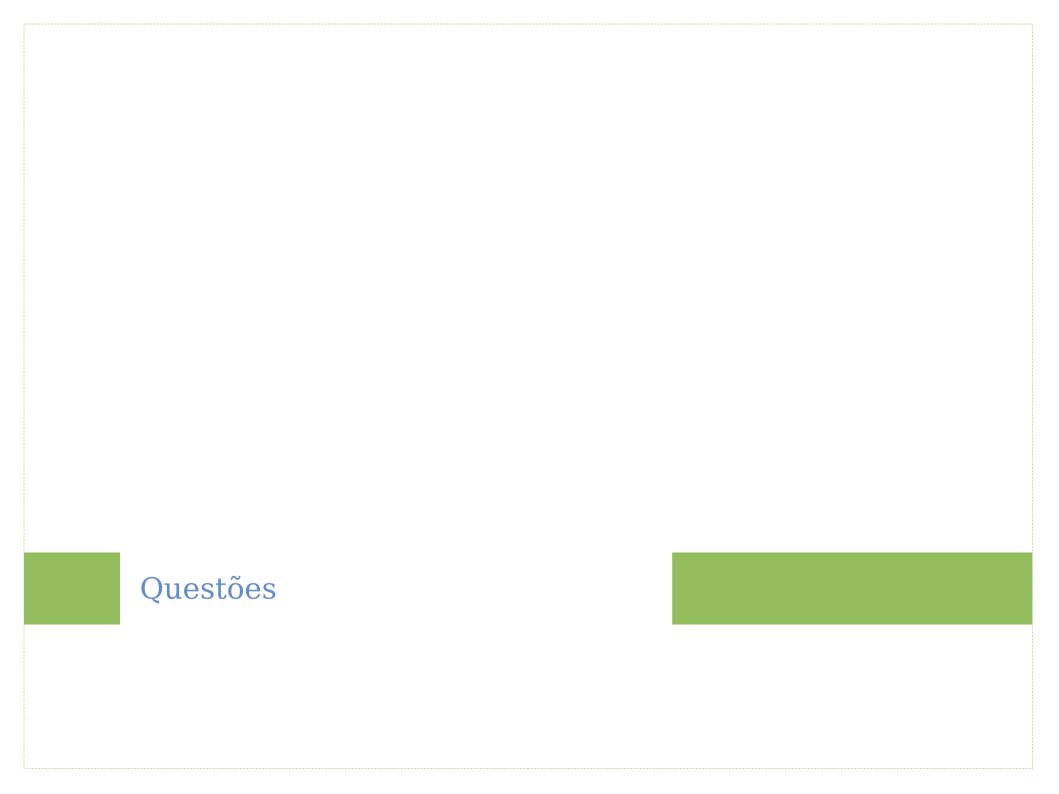
Acesso de Omissão/Package/Amigos.

Acesso Privado.

Instalação Eléctrica de uma Casa

- Lâmpadas, interruptores, circuitos, controlos de presença e luminusidade, controlos remotos, etc.

Exercício



• Herança como partilha de características.

• Polimorfismo em conjunto com a herança.

• Encapsulamento para esconder implementação.

Resumindo

Programação Orientada a Objectos torna a resolução dos problemas um processo natural, facilitando a obtenção de soluções melhores e mais adequadas.

Conclusão