

Singleton Pattern

SISMO - Sistemas e Mobilidade
<http://www.sismo.deinf.ufma.br>

Departamento de Informática / UFMA

Junho de 2008

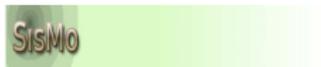


As vezes, apenas um objeto é tudo o que se precisa

- Veja os exemplos:
 - Pool de threads, caixas de diálogos
 - Objetos para manipulação de logging
 - Objetos que atuam como drivers de dispositivos (ex: impressoras)
- Ter mais de uma instância nesses casos pode ser uma fonte de problemas



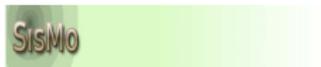
- Criando um objeto
 - **new** *MeuObjeto()*;
- Se a classe *MeuObjeto()* for uma classe pública, então podemos criar tantos objetos quanto quisermos usando o operador **new**
- Se a classe *MeuObjeto()* não for uma classe pública, então apenas classes no mesmo pacote podem instanciá-la. Mas ainda se pode instanciar mais de um objeto.



Singleton Pattern Passo a Passo II

```
Public MeuObjeto {  
    private MeuObjeto() { };  
}
```

- Essa classe não pode ser instanciada porque seu construtor é privado
- O único código que poderia instanciar essa classe teria que ser um método da própria classe
- É o caso do ovo e da galinha



Singleton Pattern Passo a Passo II

```
Public MeuObjeto {  
    private MeuObjeto() { };  
    public static MeuObjeto getInstance() {  
        return new MeuObjeto();  
    }  
}
```

- *getInstance()* é um método estático, isto é, um método que pode ser chamado diretamente pela classe
 - *MeuObjeto.getInstance();*

Singleton Pattern Passo a Passo III

```
Public Singleton {  
    private static Singleton single;  
    private Singleton() { };  
    public static Singleton getInstance() {  
        if (single == null) {  
            return new Singleton();  
        }  
        return single;  
    }  
}
```

Figura: Classe implementando o padrão *Singleton*



- Classificação
 - Propósito: Criacional
 - Escopo: Objetos
- Intenção
 - garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para essa instância.



- Quando usar o *Singleton*
 - Quando deve existir uma única instância de uma classe e o acesso à essa instância deve ser através de um ponto bem conhecido

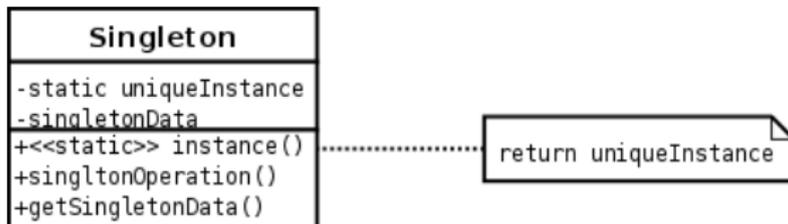


Figura: *Singleton - Estrutura*

- Singleton
 - Define uma operação *static instance()* que permite aos clientes acessarem sua única instância



- Os clientes acessam uma instância Singleton apenas através da operação *instance()* do Singleton



- Acesso controlado à instância única
- Redução do espaço de nomes (menos variáveis globais para armazenar instâncias únicas)
- Permite um número variável de instâncias (alterando o padrão para controlar esse número)

