

Aula Prática - 5 Padrões de Criação - Singleton

Intenção

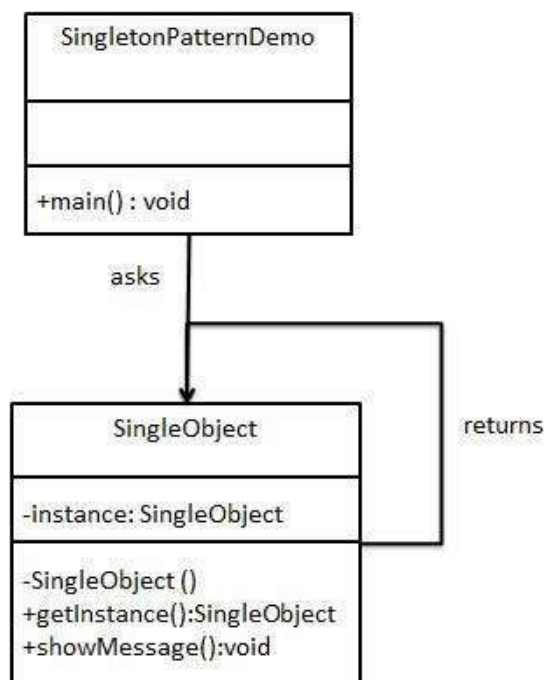
Garantir que uma classe possui somente uma instância e prover um ponto de acesso global a ela.

Usar este padrão quando...

- Tiver que haver exatamente uma instância de uma classe e ela tiver que estar acessível a todos num local bem definido;
- Quiser permitir ainda que esta classe tenha subclasses (construtor protected).

Vantagens e desvantagens

- Acesso controlado à instância:
 - A própria classe controla sua instância única.
- Não há necessidade de variáveis globais:
 - Variáveis globais poluem o espaço de nomes.
- Permite extensão e refinamento:
 - A classe Singleton pode ter subclasses.
- Permite número variado de instâncias:
 - Você pode controlar este número.
- Mais flexível do que operações de classe:
 - Usar membro static perde flexibilidade.



Passo 1

Criar uma classe Singleton

SingleObject.java

```
public class SingleObject {  
  
    //create an object of SingleObject  
    private static SingleObject instance = new SingleObject();  
  
    //make the constructor private so that this class cannot be  
    //instantiated  
    private SingleObject(){}  
  
    //Get the only object available  
    public static SingleObject getInstance(){  
        return instance;  
    }  
  
    public void showMessage(){  
        System.out.println("Hello World!");  
    }  
}
```

Passo 2

Obtenha o único objeto da classe singleton.

SingletonPatternDemo.java

```
public class SingletonPatternDemo {  
    public static void main(String[] args) {  
  
        //illegal construct  
        //Compile Time Error: The constructor SingleObject() is not visible  
        //SingleObject object = new SingleObject();  
  
        //Get the only object available  
        SingleObject object = SingleObject.getInstance();  
  
        //show the message  
        object.showMessage();  
    }  
}
```

Passo 3

Teste sua implementação!