



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br



Introdução ao padrão de projeto Singleton





Introdução ao padrão de projeto Singleton

O padrão de projeto Singleton é um dos padrões de projetos mais simples e conhecidos no mundo da programação.

O Singleton proporciona uma forma de ter um, e somente um, objeto de determinado tipo, além de um ponto de acesso global à este objeto.



Introdução ao padrão de projeto Singleton

Onde você iria usar o Singleton?



Introdução ao padrão de projeto Singleton

Onde você iria usar o Singleton?

Usamos o padrão singleton em casos como logging (logs) ou operações de bancos de dados, spoolers de impressão e muitos outros cenários em que seja necessário que haja apenas uma única instância de determinado objeto disponível para toda a aplicação.

Isso para evitar requisições conflitantes para o mesmo recurso.

Por exemplo podemos querer usar um único objeto de banco de dados para executar operações e manter a consistência dos dados, ou um objeto de uma classe para logging (log) para vários serviços, de modo que descarreguemos as mensagens de log em um arquivo particular, sequencialmente.



Introdução ao padrão de projeto Singleton

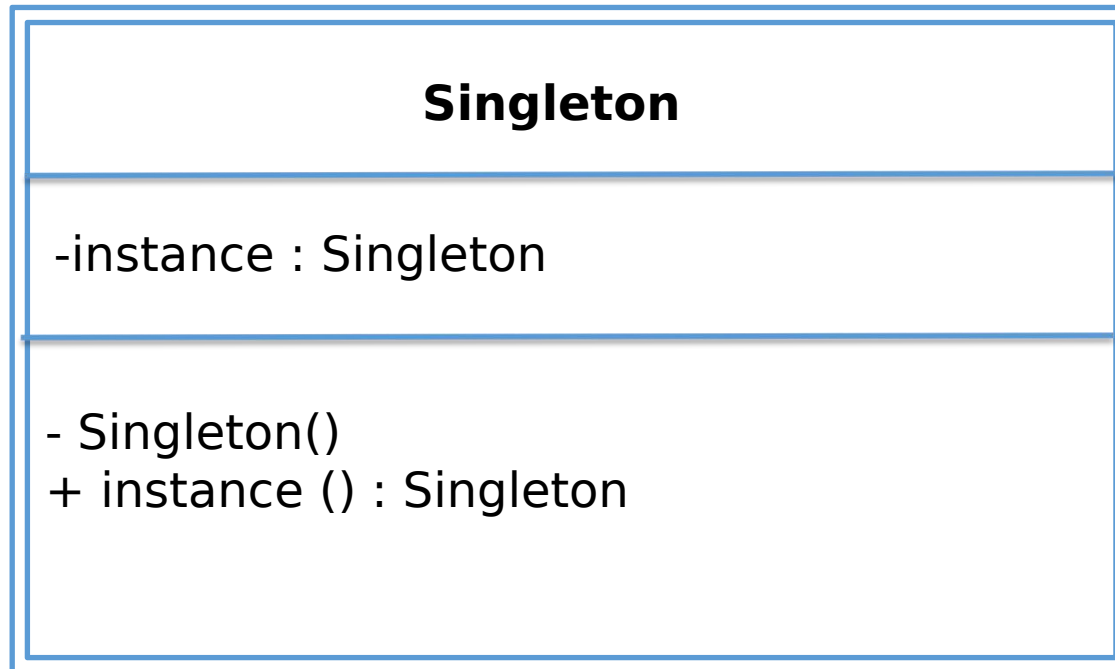
Em resumo, você irá usar o Singleton se suas intenções são:

- Garantir que um e somente um objeto de determinada classe seja instanciado;
- Oferecer um ponto de acesso para o objeto que seja global no programa;
- Controlar o acesso concorrente a recursos compartilhados;



Introdução ao padrão de projeto Singleton

Representação do diagrama UML para o Singleton



Uma maneira simples de implementar o Singleton é deixar o construtor privado e criar um método estático, público, que faça a inicialização do objeto.

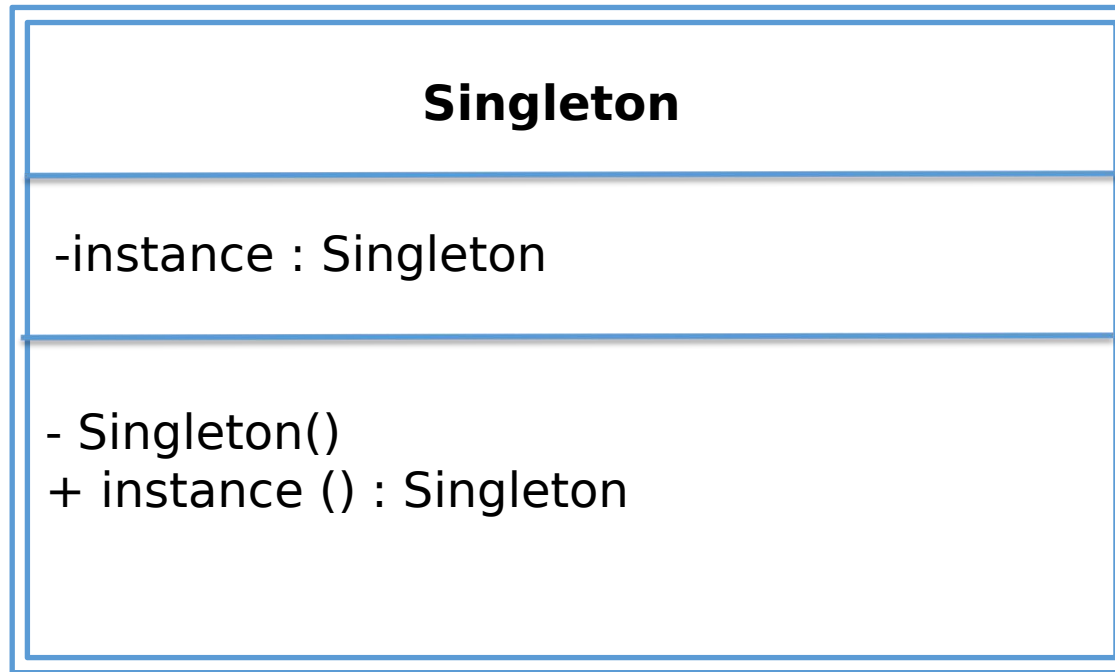
Desta forma o objeto é criado na primeira chamada e a classe devolverá este objeto já criado nas outras chamadas.

Em Python implementamos este padrão de um modo diferente já que não há opção de criar construtores privados.



Introdução ao padrão de projeto Singleton

Representação do diagrama UML para o Singleton



Uma maneira simples de implementar o Singleton é deixar o construtor privado e criar um método estático, público, que faça a inicialização do objeto.

Desta forma o objeto é criado na primeira chamada e a classe devolverá este objeto já criado nas outras chamadas.

Em Python implementamos este padrão de um modo diferente já que não há opção de criar construtores privados.

Na próxima aula iremos implementar o Singleton...



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br