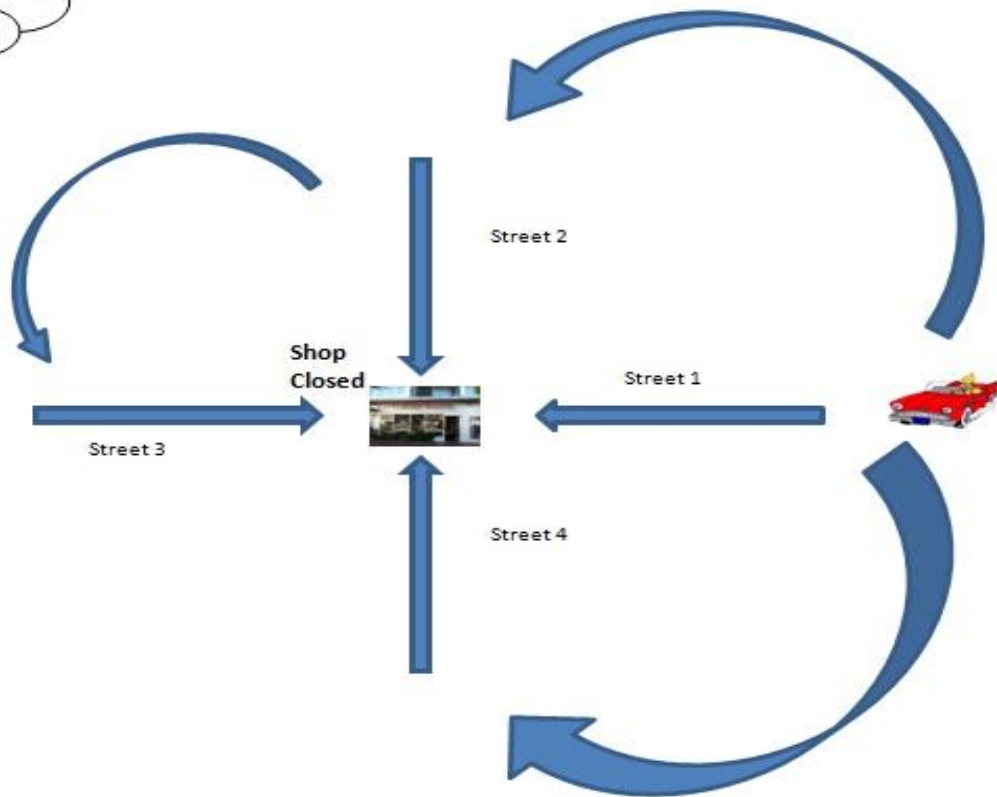


Padrões de Projeto: Singleton

- David Silva Alexandre
- José Silvino da Silva Neto

Introdução

O que é Singleton?



Only one instance of class exists in your entire application.

Singleton

- singleton : Singleton

- Singleton()

+ getInstance() : Singleton

- O atributo singleton que é do tipo da sua própria classe e é estático;
 - O construtor da classe Singleton() que é PRIVADO;
 - O método getInstance() é estático.
-



Aplicações

- Banco de Dados
 - Arquivo de Log
 - Objeto que representa um vídeo
 - Class loader
-

Implementação

Exemplo 1

```
public class Singleton
{
    private static Singleton uniqueInstance;

    private Singleton()
    {
    }

    public static synchronized Singleton getInstance()
    {
        if(uniqueInstance == null)
            uniqueInstance = new Singleton();

        return uniqueInstance;
    }
}
```

Exemplo 2

```
public class Singleton
{
    private static Singleton uniqueInstance = new Singleton();

    private Singleton()
    {
    }

    public static Singleton getInstance()
    {
        return uniqueInstance;
    }
}
```

Vantagens

- Acesso Central e Extensivo a Recursos e Objetos
-

Desvantagens

- Qualidade da Implementação
 - Difícil de Testar
 - Usado muitas vezes para substituir variáveis globais
 - Difícil de Implementar em Sistemas Distribuídos
-

Conclusão

Referências

- Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. Head First Design Patterns. O'Reilly Media, 2004.
 - Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 2010.
 - Dav Media
 - Coursera
-