

Singleton

"andro "antos Andrade

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Letro+Letr, nica
Graduação Tecnológica em Análise e esen!ol!imento de "istemas



"ingleton

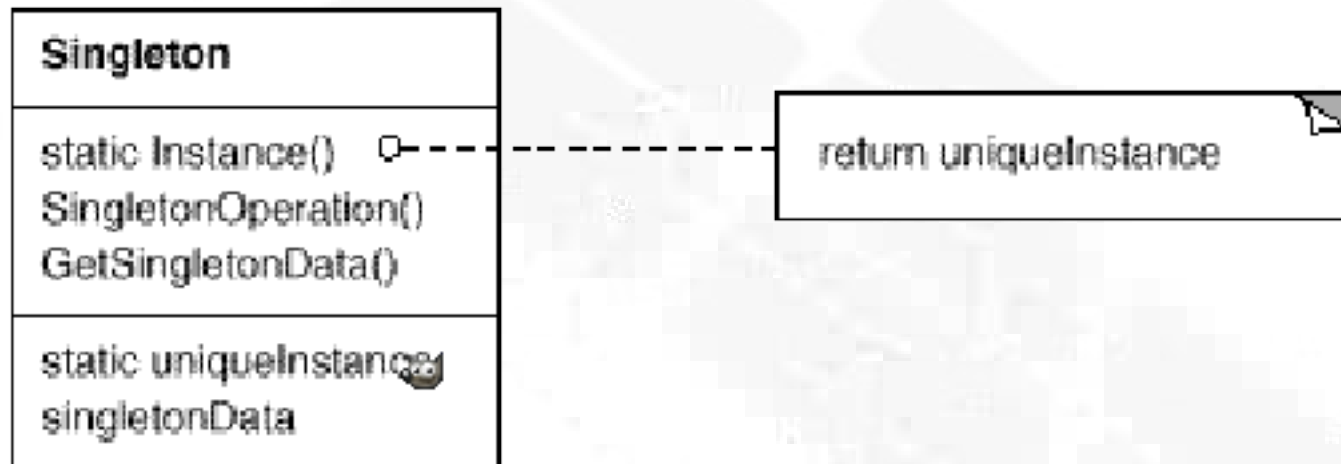


-
-
-
- *spooler*
- *window manager*
-
-

"ingleton



"ingleton



"ingleton

- *Singleton*

instance()

Instance()

"ingleton

Singleton
instance() *Singleton*

"ingleton

name space

Singleton

name space

Singleton

run-time

"ingleton

Singleton

instance()

"ingleton

instance()

"ingleton

```
class Singleton {  
public:  
    static Singleton* Instance();  
protected:  
    Singleton();  
private:  
    static Singleton* _instance;  
};
```

```
Singleton* Singleton::_instance = 0;  
  
Singleton* Singleton::Instance () {  
    if (_instance == 0) {  
        _instance = new Singleton;  
    }  
    return _instance;  
}
```

"ingleton

■

■

■

■

■

■

■

■

protected

instance()

_instance

su -+classe

Singleton

Singleton

Singletons

"ingleton

■

■

■

■

■

■

■

Singleton

instance()

instance()

Singleton

Singleton

run-time

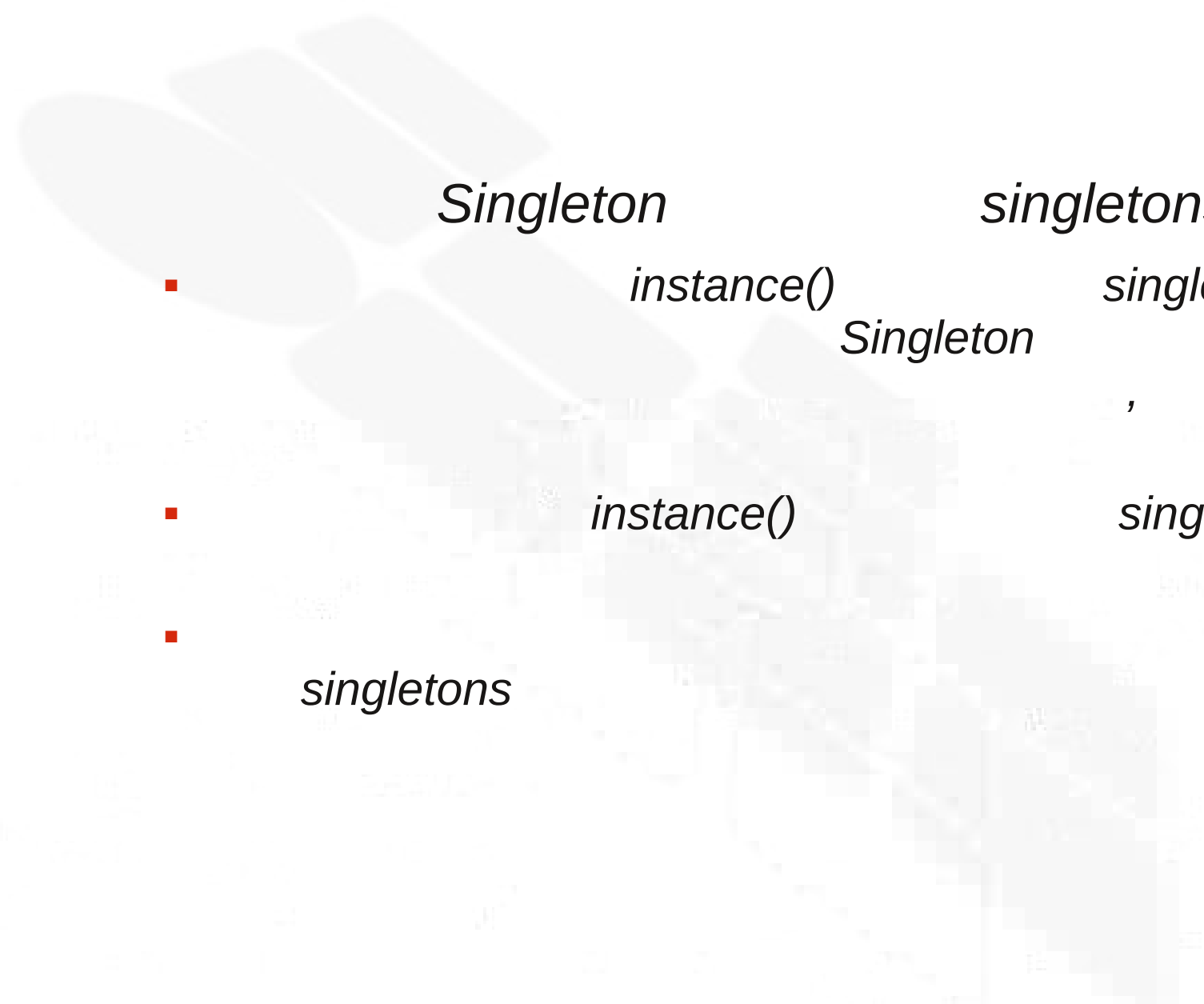
link

hard-coded

registro de

singletons

"ingleton



-
-
-
-

Singleton *singletons*

instance() *singleton*

Singleton

instance() *singleton*

singletons

"ingleton

Singleton

singletons

```
class Singleton {
public:
    static void Register(const char* name, Singleton*);
    static Singleton* Instance();
protected:
    static Singleton* Lookup(const char* name);
private:
    static Singleton* _instance;
    static List<NameSingletonPair>* _registry;
};
```

"ingleton

Singleton

singletons

```
Singleton* Singleton::Instance () {  
    if (_instance == 0) {  
        const char* singletonName = getenv("SINGLETON");  
        // user or environment supplies this at startup  
        _instance = Lookup(singletonName);  
        // Lookup returns 0 if there's no such singleton  
    }  
    return _instance;  
}
```

"ingleton

singleton

```
MySingleton::MySingleton() {  
    // ...  
    Singleton::Register("MySingleton", this);  
}
```

```
static MySingleton theSingleton;
```

"ingleton

```
class MazeFactory {
public:
    static MazeFactory* Instance();

    // existing interface goes here
protected:
    MazeFactory();
private:
    static MazeFactory* _instance;
};
```


"ingleton

```
MazeFactory* MazeFactory::Instance () {  
    if (_instance == 0) {  
        const char* mazeStyle = getenv("MAZESTYLE");  
  
        if (strcmp(mazeStyle, "bombed") == 0) {  
            _instance = new BombedMazeFactory;  
        } else if (strcmp(mazeStyle, "enchanted") == 0) {  
            _instance = new EnchantedMazeFactory;  
        }  
  
        // ... other possible subclasses  
  
        } else {          // default  
            _instance = new MazeFactory;  
        }  
    }  
    return _instance;  
}
```

"ingleton

-
-
- *InterView* *singleton*
widgets

"ingleton

-
-

Abstract Factory, Builder, Prototype

Singleton

"andro "antos Andrade

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Letro+Letr, nica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

