Azzolini Riccardo 2019-10-08

Singleton Design Pattern

1 Singleton

In alcuni casi, può essere utile creare una classe che abbia al massimo una singola istanza.

La soluzione a questo problema è data da un *design pattern* chiamato **singleton**. Per implementarlo, è necessario creare una classe con:

- tutti i costruttori dichiarati `private`;
- un riferimento statico alla singola istanza della classe;
- un metodo statico che restituisce un riferimento all'istanza (creando tale istanza la prima volta che viene invocato, e restituendo la stessa per tutte le invocazioni successive).

Oltre a questi elementi, la classe ne conterrà poi anche altri, che implementano le sue funzionalità.

1.1 Esempio

```
public class SingletonClass {
    private static SingletonClass singleInstance;
    private int innerVariable;

    private SingletonClass(int value) {
        innerVariable = value;
    }

    public static SingletonClass instantiate(int value) {
        if (singleInstance == null) {
            singleInstance = new SingletonClass(value);
        }
        return singleInstance;
    }

    public int getValue() {
        return innerVariable;
```

```
    }

    public void setValue(int value) {
        innerVariable = value;
    }
}

public class MainProgram {
    public static void main(String[] args) {
        SingletonClass instance1 = SingletonClass.instantiate(1);
        SingletonClass instance2 = SingletonClass.instantiate(2);

        System.out.println("Content of instance1 = "
                           + instance1.getValue());
        System.out.println("Content of instance2 = "
                           + instance2.getValue());

        instance1.setValue(3);
        System.out.println("Content of instance1 = "
                           + instance1.getValue());
        System.out.println("Content of instance2 = "
                           + instance2.getValue());

        if (instance1 == instance2) {
            System.out.println("instance1 and instance2 are aliases");
        }
    }
}
```

L'output del programma è:

```
Content of instance1 = 1
Content of instance2 = 1
Content of instance1 = 3
Content of instance2 = 3
instance1 and instance2 are aliases
```