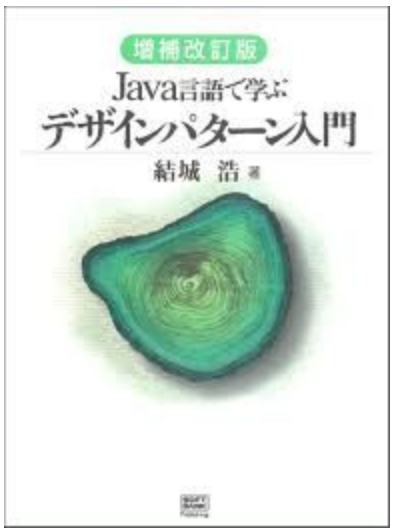
Java言語で学ぶデザインパターン入門

Singleton
たった１つのインスタンス



Singletonパターン

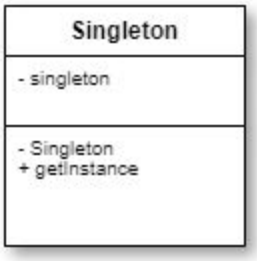
- 指定したクラスのインスタンスが絶対に１個しか存在しないことを保証したい
- インスタンスが１個しか存在しないことをプログラム上で表現したい

サンプルプログラム

クラス一覧

名前	解説
Singleton	インスタンスが１つしか存在しないクラス
Main	動作テスト用のクラス

クラス図



Singletonクラスのコンストラクタはprivateになっている。これは、Singletonクラス外からコンストラクタを呼び出すことを禁止するため。もし、new Singleton()という式がこのクラスの外にあったとしても、コンパイル時のエラーになる。

Singletonクラス

```
public class Singleton {
	private static Singleton singleton = new Singleton();
	private Singleton() {
		System.out.println("インスタンスを生成しました。");
	}
	public static Singleton getInstance() {
		return singleton;
	}
}
```

Singletonクラス

Mainクラス

```
public class Main {
	public static void main(String[] args) {
		System.out.println("Start.");
		Singleton obj1 = Singleton.getInstance();
		Singleton obj2 = Singleton.getInstance();
		if(obj1 == obj2) {
			System.out.println("同じインスタンスです。");
```

```
            } else {
                  System.out.println("同じインスタンスではありません。");
            }
            System.out.println("End.");
      }
}
```

Mainクラス

唯一のインスタンスはいつ生成されているか

プログラムの実行開始後、最初にgetInstanceメソッドを呼び出したときにSingletonクラスは初期化される。そして、そのときにstaticフィールドの初期化が行われ、唯一のインスタンスが生成される。