

## 2. フィールドを初期化する必要のないクラス

フィールドがあっても、初期化する必要がなければ、やはりコンストラクタは作成しません。そのかわり、自動生成されるデフォルトコンストラクタを使うことができます。次のNoConstructorクラスで試してみましょう。

### リスト 7-2-7: フィールドがあってコンストラクタがないクラス

```
package chapter7_4;
public class NoConstructor {
    private int number;
    private String str="***";

    public int getNumber() {
    }
    public String getStr() {
    }
}
```

直接代入による初期化 ①  
// 空文字で初期化 ②  
③  
④

このNoConstructorクラスには、コンストラクタはありません。

しかし、①、②のようにフィールドがあります。このうち、②は最初から "\*\*\*" を代入しています。フィールドでは、このような**コンストラクタを使わない初期化**も可能です。

また、③、④はフィールドのゲッターです

では、デフォルトコンストラクタを使ってインスタンスを生成し、ゲッターでフィールドの値を見てみましょう。

### リスト 7-2-8: コンストラクタで初期化しないフィールドの値

```
package chapter7_4;
public class Sample2 {
    public static void main(String[] args) {
        NoConstructor nc = new NoConstructor();
        System.out.println(nc.getNumber());
        System.out.println(nc.getStr());
    }
}
```

実行すると、次のように表示されます。

```
0
***
```