

# 第2章

# オブジェクトの生成と インスタンスの利用



オブジェクトの利用は、イメージが大事！  
手順を覚えて着実にマスター！

## 2-1 オブジェクトの生成

例題をとおして、オブジェクトの生成について学習します。また、複数のインスタンスの生成についても学習します。



### 例題 2-1

- ・プログラム例 1-2 で作成した Customer クラスの、インスタンスを生成するプログラムを作成しなさい。

Customer.java (プログラム例 1-2 で作成済)

```
1 class Customer {
2     int id;
3     String name;
4     void display() {
5         System.out.println("ID  :" + id);
6         System.out.println("氏名:" + name);
7     }
8 }
```

#### クラス定義

クラス名 : Example221

メソッド名 : main メソッド



## [プログラム例 2-1]

### Example221.java

```

1 class Example221 {
2     public static void main(String[] args) {
3         Customer cust = new Customer();
4     }
5 }

```

### 実行結果



・このプログラムは、実行してもなにも表示されません。




## [作業手順]

- ① Eclipse の「Java\_第2部」プロジェクトの「src」階層に、新規 Java クラスを作成します。

### 〈「新規 Java クラス」ウィンドウで入力する値〉


入力項目	入力値
ソース・フォルダー	Java_第2部/src
名前	Example221
修飾子	「パッケージ」にチェックを入れる
どのメソッド・スタブを作成しますか？	「public static void main(String[] args)」にチェックを入れる

※これ以外の項目には、入力しません。

- ② プログラム例 2-1 を参考にして、プログラムを入力します。入力後、Eclipse の  ボタンを押して、プログラムを保存します。



・保存後、「ローカル変数 cust の値は使用されていません」という警告メッセージが出ますが、問題ありません。

- ③ Eclipse の  ボタンを押して、プログラムを実行します。



## 例題 2-1 解説

### (1) オブジェクトの生成

#### ● インスタンス化

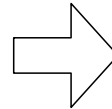
クラスからオブジェクトを生成することを、**インスタンス化**といいます。また、クラスから生成されたオブジェクトを**インスタンス**といいます。



#### インスタンス化

##### Customerクラス

```
class Customer {  
    int id;  
    String name;  
    void display() {  
        System.out.println("ID : " + id);  
        System.out.println("氏名 : " + name);  
    }  
}
```



##### Customerインスタンス

###### インスタンス変数

id	0
name	null

###### メソッド

display()

クラスからオブジェクトを生成する ⇒ インスタンス化



・クラスに定義したインスタンス変数やメソッドは、インスタンス化して初めて利用することができます。

インスタンス化の命令には、**new** を使います。

#### 記述方法 (インスタンス化)

```
class クラス名 {  
    public static void main(String[] args) {  
        参照型 参照変数名 = new 参照型();  
    }  
}
```

**new** の後に指定した参照型のクラスがインスタンス化されます。



・**参照型**とは、インスタンスを参照する変数に指定するデータ型です。  
・参照型には、作成済のクラス名を指定します。

参照変数には、インスタンスの参照を代入します。



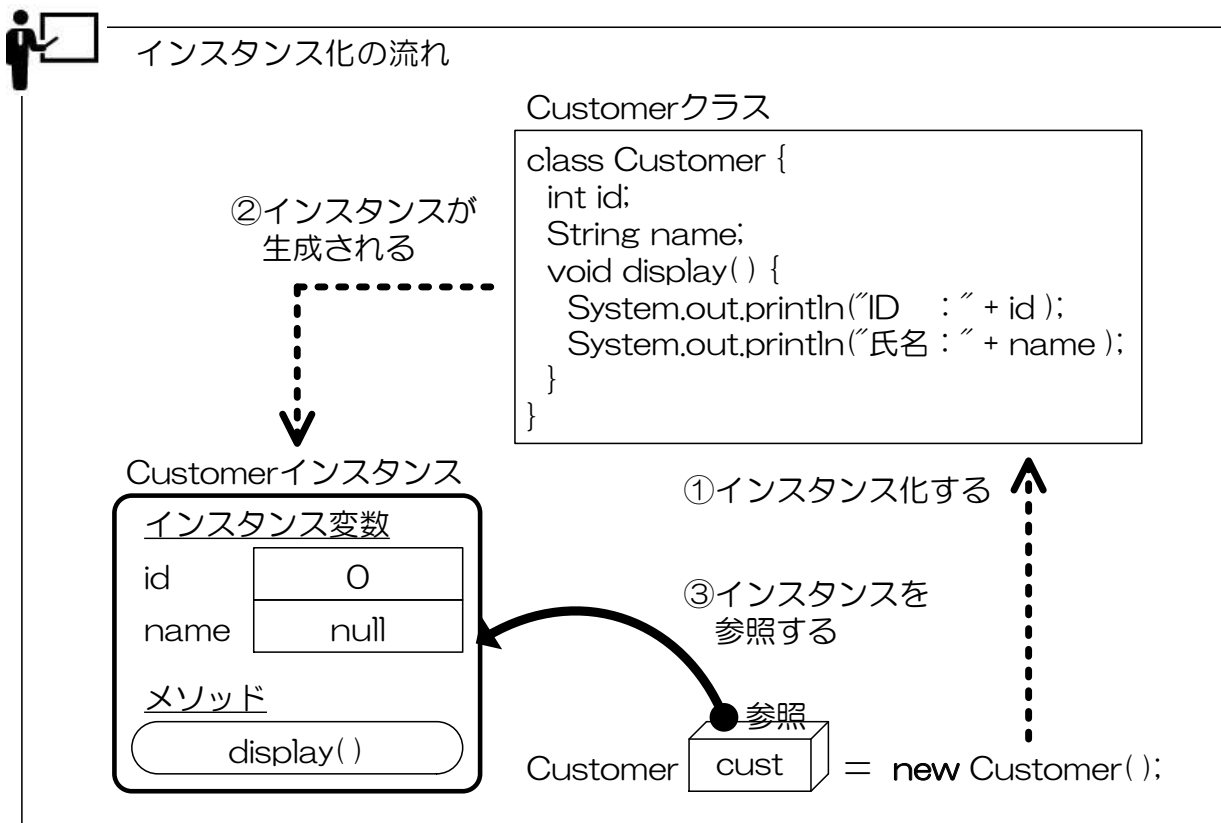
・コンピュータの内部では、インスタンスは主記憶装置上に記憶されます。  
・**参照**とは、主記憶装置上のインスタンスの位置情報 (アドレス) です。

プログラム例 2-1 では，インスタンス化は次に示す 3 行目です。

Example221.java (抜粋)

```
3 Customer cust = new Customer();
```

Customer クラス型の変数 cust に，Customer クラスのインスタンスの参照を代入しています。



- ・生成されたインスタンスの各インスタンス変数は，自動的に初期化されます。
- ・インスタンス化の際に代入される初期値は次のとおりです。

<インスタンス生成時の変数の初期値>

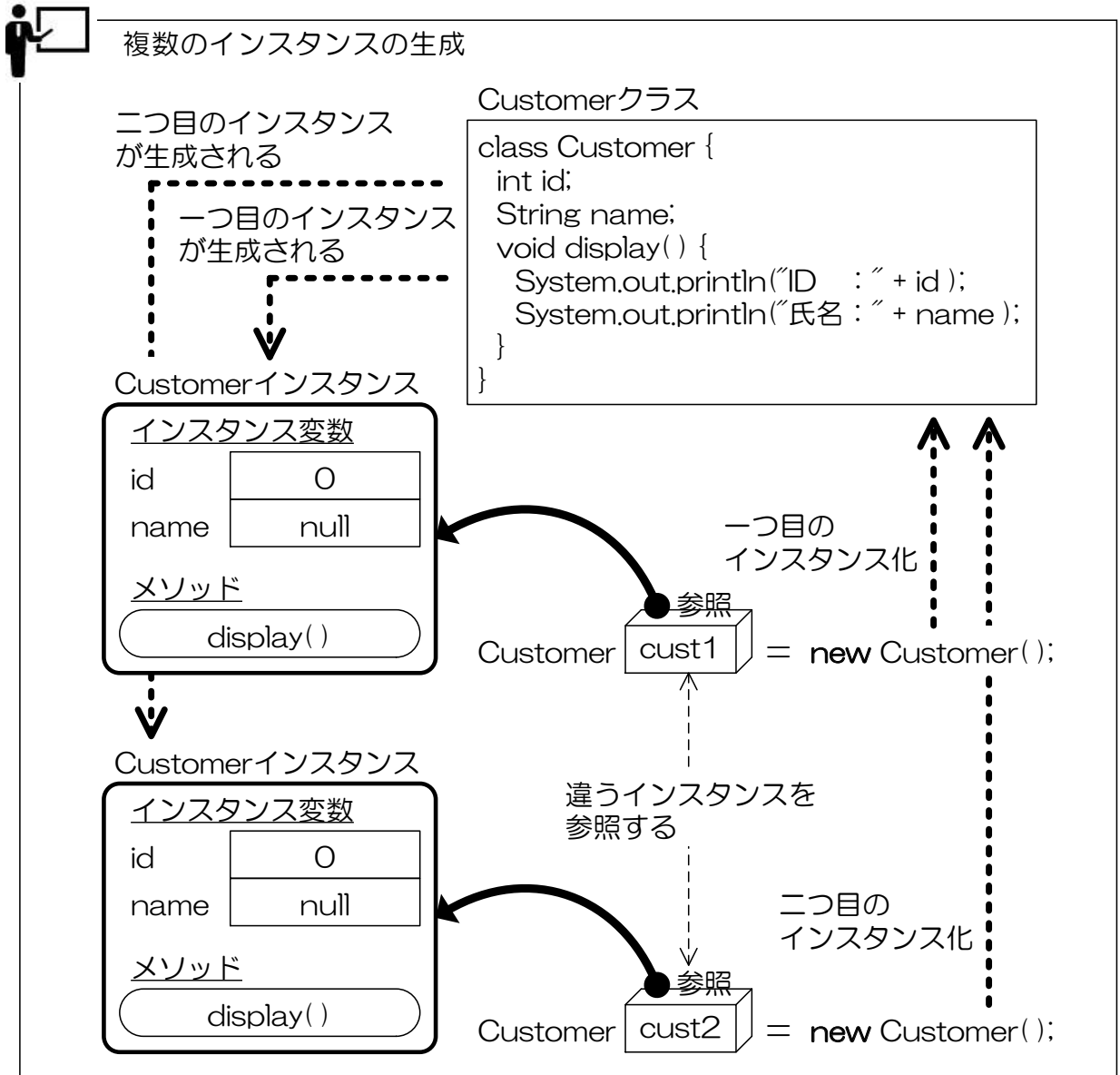
データ型	初期値
整数型 (byte, short, int, long)	0
浮動小数点数型 (float, double)	0.0
文字型 (char)	'\u0000'
真偽値型 (boolean)	false
参照型 (String など)	null

## (2) 複数のインスタンスの生成

一つのクラスから、複数のインスタンスを生成することができます。

### 記述例（複数のインスタンスの生成）

```
//一つ目のインスタンスの生成
Customer cust1 = new Customer();
//二つ目のインスタンスの生成
Customer cust2 = new Customer();
```



・生成された二つのインスタンスは、性質の異なる二つのオブジェクトとして扱うことができます。



・インスタンスを参照する変数に、他のインスタンスの参照を代入すると元の参照は消え、後から代入した参照が残ります（上書き）。



## 練習問題 2-1

- ・練習問題 1-2 で定義した Product クラスの、インスタンスを生成するプログラムを作成しなさい。

※保存後、文法エラー（コンパイルエラー）がないことを確認してください。

### Product.java（練習問題 1-2 で定義済）

```
1 class Product {
2     String no;
3     String name;
4     int price;
5     int quantity;
6     void display() {
7         int total = price * quantity;
8         System.out.println("ナンバー：" + no);
9         System.out.println("製品名  ：" + name);
10        System.out.println("価格    ：" + price + "円");
11        System.out.println("数量    ：" + quantity + "個");
12        System.out.println("合計金額：" + total + "円");
13    }
14 }
```

#### クラス定義

クラス名 : Practice221  
メソッド名 : main メソッド

#### main メソッドの処理条件

- ・Product クラス型の変数 pro に、Product クラスのインスタンスを生成してその参照を代入します。