

# 参照について

以下のようなコードをコンパイルし、実行したときの結果はどうなるでしょうか？

```
class Card {  
    int deposit;  
}  
  
class Test {  
    public static void main(String[] args) {  
        Card c1 = new Card();  
        Card c2 = new Card();  
        Card c3 = c1;  
        c1.deposit = 1000;  
        c2.deposit = 2000;  
        c3.deposit = 3000;  
        int sum = c1.deposit + c2.deposit + c3.deposit;  
        System.out.println("sum == " + sum);  
    }  
}
```

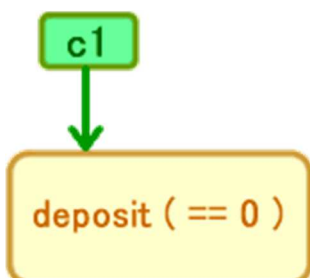
このソースで注意する点は、参照とインスタンスは別のものであるということです。

参照はインスタンスがメモリ上のどこにあるのかを指し示すのであって、インスタンス自身ではないのです。

それでは以下で参照とインスタンスの状態について見ていきましょう。

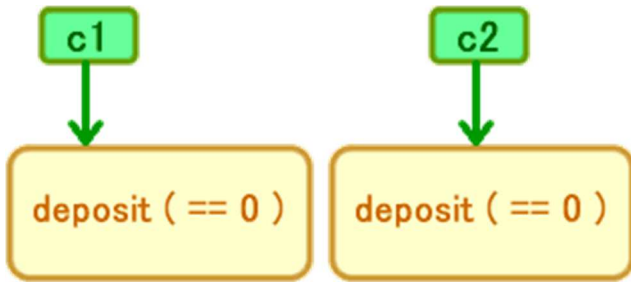
```
Card c1 = new Card();
```

インスタンスが生成され、そのインスタンスへの参照が c1 に格納されています。



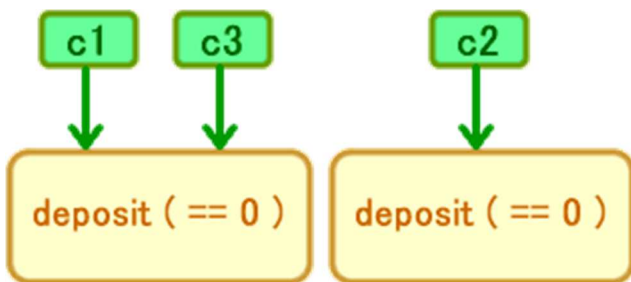
```
Card c2 = new Card();
```

別の新しいインスタンスが生成され、このインスタンスへの参照は c2 に格納されています。



```
Card c3 = c1;
```

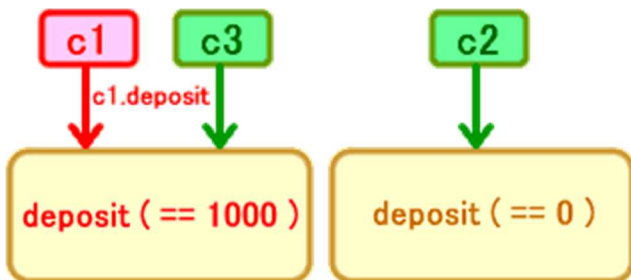
c1 に格納されていた参照が c3 にコピーされただけでインスタンスは生成されていません。  
その結果、c1 と c3 は同じインスタンスを指し示すことになります。



このような状態でインスタンスを操作するとどうなるのか、処理の流れを追ってみましょう。

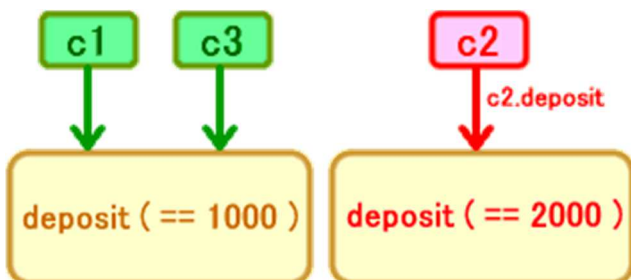
```
c1.deposit = 1000 ;
```

c1.deposit に 1000 を代入



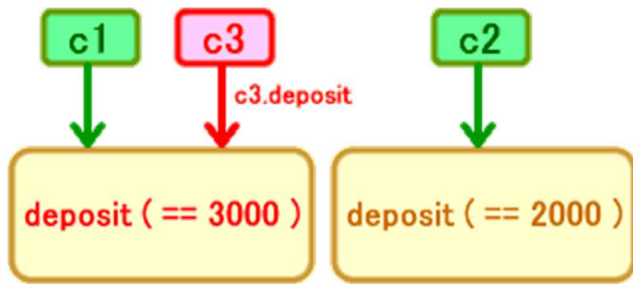
```
c2.deposit = 2000;
```

c2.deposit に 2000を代入



```
c3.deposit = 3000;
```

c3.deposit に 3000 を代入



c1 と c3 を通じて行った操作が同じインスタンスへ作用している様子が分かったかと思います。

c1.deposit と c3.deposit は同じものとなるので

```
int sum = c1.deposit + c2.deposit + c3.deposit;
```

の値は以下ようになります。

```
int sum = 3000 + 2000 + 3000;
```

そのため、実行結果は

```
>javac Test.java
```

```
>java Test
```

```
sum == 8000
```

となり、sum == 8000 と表示されます。