

4.8 練習問題

練習 4-1

次のような書籍クラス (Book クラス) があります。

```
1 import java.util.*;
2
3 public class Book {
4     private String title;
5     private Date publishDate;
6     private String comment;
7     :    // getter/setterの宣言は省略
8 }
```

Book.java

以下のすべての動作を実現するよう、Book クラスを改良してください。ただし、各フィールドに null が入っている場合を考慮する必要はありません。

- ① 書名と発行日が同じであれば等価なものと判定され、かつ、HashSet などに格納しても正しく利用できる。
- ② Book インスタンスを格納したコレクションに対して、「Collections.sort();」を呼び出すと、発行日が古い順に並び替えられる。
- ③ clone() を呼び出すと、深いコピーによる複製が行われる。

4.9 練習問題の解答

練習 4-1 の解答

```
1 import java.util.*;
2
3 public class Book implements Comparable<Book>, Cloneable {
4     private String title;
5     private Date publishDate;
6     private String comment;
7     public int hashCode() {
8         int r = 1;
9         r = 31 * r + publishDate.hashCode();
10        r = 31 * r + title.hashCode();
11        return r;
12    }
13    public boolean equals(Object o) {
14        if(this == o) {
15            return true;
16        }
17        if(o == null) {
18            return false;
19        }
20        if(!(o instanceof Book)) {
21            return false;
22        }
23        Book b = (Book) o;
24        if(!publishDate.equals(b.publishDate)) {
25            return false;
```

Book.java

```
26     }  
27     if(!title.equals(b.title)) {  
28         return false;  
29     }  
30     return true;  
31 }  
32 public int compareTo(Book o) {  
33     return this.publishDate.compareTo(o.publishDate);  
34 }  
35 public Book clone() {  
36     Book b = new Book();  
37     b.title = this.title;  
38     b.comment = this.comment;  
39     b.publishDate = (Date) this.publishDate.clone();  
40     return b;  
41 }  
42 :    // フィールド宣言、getter/setterの宣言は省略  
43 }
```

第5章

さまざまな種類のクラス

Java では一般的なクラスはもちろん、さまざまな種類の「クラスのようなもの」を利用して便利に開発を進めることができます。たとえばインタフェースがそれにあたりますが、数あるものの1つに過ぎません。

この章では、新しい種類のクラスについて学ぶとともに、クラスや型について理解を深めましょう。