

# INLOOP Softwaretechnologie, Collections I (Library with List)

HENRY HAUSTEIN

Datei Book.java

```
1  package collections1;
2
3  // Durch Nutzung des Interfaces Comparable<Book> ist die
4  // compareTo()-Funktion verfuegbar
5  public class Book implements Comparable<Book> {
6      private String isbn;
7      private String author;
8      private String title;
9
10     public Book(String isbn) {
11         this.isbn = isbn;
12         this.title = "";
13         this.author = "";
14     }
15
16     public Book(String isbn, String author, String title){
17         this.isbn = isbn;
18         this.author = author;
19         this.title = title;
20     }
21
22     public String getTitle() {
23         return title;
24     }
25
26     public String getIsbn() {
27         return this.isbn;
28     }
29
30     public String getAuthor() {
31         return this.author;
32     }
33
34     // Vergleich von Buechern = Vergleich der ISBNs
35     // fuer den Vergleich der ISBNs wird compareTo von Strings verwendet
36     public int compareTo(Book book) {
37         return this.getIsbn().compareTo(book.getIsbn());
```

```
38     }
39 }
```

Datei Library.java

```
1  package collections1;
2
3  // wird benoetigt fuer Collections.sort() und Collections.binarySearch()
4  import java.util.*;
5
6  public class Library {
7      private List<Book> stock;
8
9      public Library() {
10         this.stock = new LinkedList<Book>();
11     }
12
13     public boolean insertBook(Book newBook) {
14         stock.add(newBook);
15         Collections.sort(stock);
16         return true;
17     }
18
19     public Book searchForIsbn(String isbn) {
20         int index = Collections.binarySearch(stock, new Book(isbn));
21         // wird -1 als index zurueckgegeben, ist das Buch nicht vorhanden
22         if (index >= 0) {
23             return stock.get(index);
24         }
25         else {
26             return null;
27         }
28     }
29
30     public Collection<Book> searchForAuthor(String author) {
31         ArrayList<Book> authorList = new ArrayList<Book>();
32         // for-Schleife, funktioniert genau so wie in Python
33         // for listentry in list -> for (listentry : list)
34         // jedes Element der Liste wird einmal in listentry
35         // abgespeichert und innerhalb der for-Schleife
36         // kann dann auf listentry zugegriffen werden
37         for (Book b : stock) {
38             if (b.getAuthor().equals(author)) {
39                 authorList.add(b);
40             }
41         }
42         return authorList;
43     }
44 }
```