

# Softwaretechnologie, Übung 5

HENRY HAUSTEIN

## Aufgabe 1

(a) Mit Mengen

```
1  public class Buch implements Comparable<Buch> {
2      private String isbn;
3      private String autor;
4      private String title;
5
6      public Buch(String isbn, String autor, String title) {
7          this.isbn = isbn;
8          this.autor = autor;
9          this.title = title;
10     }
11
12     public Buch(String isbn) {
13         this.isbn = isbn;
14     }
15
16     public String toString() {
17         return isbn + " - " + title;
18     }
19
20     public String getISBN() {
21         return isbn;
22     }
23
24     public String getAuthor() {
25         return autor;
26     }
27
28     public String getTitle() {
29         return title;
30     }
31
32     public int compareTo(Buch buch){
33         return this.isbn.compareTo(buch.getISBN());
34     }
35
36     @Override
37     public boolean equals(Object obj){
```

```

38     if (obj instanceof Buch) {
39         Buch buch = (Buch) obj;
40         return isbn.equals(buch.isbn);
41     }
42     return false;
43 }
44
45 @Override
46 public int hashCode(){
47     return isbn.hashCode();
48 }
49 }
50
51 public class Bibliothek {
52
53     private Set<Buch> bestand = new TreeSet<>();
54
55     public boolean sortiertesEinfuegen(Buch neuesBuch){
56         //sortiert nach ISBN, automatisch sortiert
57         return bestand.add(neuesBuch);
58     }
59
60     public Buch sucheNachBuch(String isbn){
61         for (Buch buch : bestand) {
62             if (buch.getIsbn().equals(isbn)){
63                 return buch;
64             }
65         }
66     }
67
68     // Alternative mit Iterator
69     public Buch sucheNachISBN(String isbn) {
70         Iterator<Buch> iter = bestand.iterator();
71         while (iter.hasNext()) {
72             Buch buch = iter.next();
73             if (buch.getIsbn().equals(isbn)){
74                 return buch;
75             }
76         }
77     }
78
79     public Collection<Buch> sucheNachAutor(String autor){
80         Collection<Buch> authorBooks = new TreeSet<Buch>();
81         for (Book b : bestand) {
82             if (b.getAuthor().equals(autor)) {
83                 authorBooks.add(b);
84             }
85         }
86         return authorBooks;
87     }
88 }

```

(b) Quelltext

```
1  public Collection<String> getAllAuthors() {
2      Set<String> result = new HashSet<String>();
3      for (Buch b : bestand) {
4          result.add(b.getAuthor());
5      }
6      return result;
7  }
8
9  public Map<String, Set<Buch>> bestandNachAutor() {
10     Map<String, Set<Buch>> map = new TreeMap<String, Set<Buch>
        >>();
11     Collection<String> autorenliste = getAllAuthors();
12     for (String autor : autorenliste) {
13         Set<Buch> buecher = sucheNachAutor(autor);
14         map.put(autor, buecher);
15     }
16     return map;
17 }
18
19 // Alternative mit nur einer benoetigten Funktion
20 public Map<String, Set<Buch>> bestandNachAutor() {
21     Map<String, Set<Buch>> map = new TreeMap<String, Set<Buch>
        >>();
22     for (Buch buch : bestand) {
23         Set<Buch> buecher = map.get(buch.getAuthor());
24         if (buecher == null) {
25             buecher = new TreeSet<Buch>();
26             map.put(buch.getAuthor(), buecher);
27         }
28         // Call-by-reference
29         buecher.add(buch);
30     }
31 }
```

## Aufgabe 2

Datei Book.java

```
1  package collections3;
2
3  public class Book implements Comparable<Book>{
4      private String isbn;
5      private String author;
6      private String title;
7
8      public Book(String isbn, String author, String title){
9          if (isbn == null | author == null | title == null) {
10             throw new IllegalArgumentException("Fill the fucking
                arguments.");
11         }
12     }
13 }
```

```

11     } else {
12         this.isbn = isbn;
13         this.author = author;
14         this.title = title;
15     }
16 }
17
18 public Book(String isbn){
19     if (isbn == null) {
20         throw new IllegalArgumentException("Fill the fucking
21             arguments.");
22     } else {
23         this.isbn = isbn;
24         this.author = "";
25         this.title = "";
26     }
27 }
28
29 public String getIsbn() {
30     return this.isbn;
31 }
32
33 public void setIsbn(String isbn) {
34     this.isbn = isbn;
35 }
36
37 public String getAuthor() {
38     return this.author;
39 }
40
41 public void setAuthor(String author) {
42     this.author = author;
43 }
44
45 public String getTitle() {
46     return this.title;
47 }
48
49 public void setTitle(String title) {
50     this.title = title;
51 }
52
53 public String toString() {
54     return this.getTitle() + " by " + this.getAuthor() + " (ISBN "
55         + this.getIsbn() + ")";
56 }
57
58 public int hashCode(){
59     return isbn.hashCode();
60 }

```

```

60     @Override
61     public boolean equals(Object o) {
62         if (o instanceof Book) {
63             Book book = (Book) o;
64             return this.isbn.equals(book.getIsbn());
65         }
66         return false;
67     }
68
69     @Override
70     public int compareTo(Book book) {
71         return this.isbn.compareTo(book.getIsbn());
72     }
73 }

```

Datei Library.java

```

1  package collections3;
2
3  import java.util.*;
4
5  public class Library {
6      private Map<String, Set<Book>> stock;
7
8      public Library(){
9          stock = new TreeMap<String, Set<Book>>();
10     }
11
12     public Collection<Book> getStock(){
13         Set<Book> output = new TreeSet<Book>();
14         for (Map.Entry<String, Set<Book>> entry : stock.entrySet()) {
15             output.addAll(entry.getValue());
16         }
17         return output;
18     }
19
20     // alternativ, mit stock.values()
21     public Collection<Book> getStock() {
22         Collection<Book> books = new TreeSet<Book>();
23         for (Collection<Book> b : this.stock.values()) {
24             books.addAll(b);
25         }
26         return books;
27     }
28
29     public boolean insertBook(Book toInsertBook){
30         Set<Book> tempSet = new TreeSet<Book>();
31         tempSet.add(toInsertBook);
32         if(stock.containsKey(toInsertBook.getAuthor())){
33             Set<Book> savedSet = stock.get(toInsertBook.getAuthor());
34             if(savedSet.contains(toInsertBook)){
35                 return false;

```

```

36         } else {
37             savedSet.add(toInsertBook);
38             stock.put(toInsertBook.getAuthor(), savedSet);
39             //replace - stellt sicher, dass Autor schon da ist
40             stock.replace(toInsertBook.getAuthor(), savedSet);
41         }
42     } else {
43         stock.put(toInsertBook.getAuthor(), tempSet);
44     }
45     return true;
46 }
47 //inloop approved, aber sehr umstaendlich
48
49 // alterantiv
50 public boolean insertBook_2(Book book){
51     if (!stock.containsKey(book.getAuthor())){
52         stock.put(book.getAuthor(), new TreeMap<Buch>());
53     }
54     Set<Buch> booksOfAuthor = stock.get(book.getAuthor());
55     return booksOfAuthor.add(book);
56 }
57
58 // alternativ
59 public boolean insertBook(Book newBook) {
60     if (stock.containsKey(newBook.getAuthor())) {
61         // Autor gibt es bereits
62         Set<Book> authorBooks = stock.get(newBook.getAuthor());
63         boolean result = authorBooks.add(newBook);
64         stock.put(newBook.getAuthor(), authorBooks);
65         return result;
66     }
67     else {
68         // Autor muss noch hinzugefuegt werden
69         Set<Book> authorBooks = new TreeSet<Book>();
70         authorBooks.add(newBook);
71         stock.put(newBook.getAuthor(), authorBooks);
72         return true;
73     }
74 }
75
76 public Book searchForIsbn(String isbn){
77     for (Map.Entry<String, Set<Book>> entry : stock.entrySet()) {
78         for (Book book : entry.getValue()){
79             if (book.getIsbn().equals(isbn)){
80                 return book;
81             }
82         }
83     }
84     return null;
85 }
86

```

```

87 // alternativ
88 public Book searchForIsbn_2(String isbn){
89     for (Set<Buch> books : stock.values()){
90         for (Book book : books ) {
91             if (book.getIsbn().equals(isbn)){
92                 return book;
93             }
94         }
95     }
96     return null;
97 }
98
99 public Map<String,Set<Book>> listStockByAuthor(){
100     return stock;
101 }
102
103 public Collection<Book> searchForAuthor(String author){
104     Set<Book> output = new TreeSet<Book>();
105     if(!stock.isEmpty()){
106         if(stock.get(author) == null){
107             return output;
108         }
109         return stock.get(author);
110     } else {
111         return output;
112     }
113 }
114
115 // alternativ
116 public Collection<Book> searchForAuthor(String author) {
117     if (stock.get(author) == null) {
118         //alternativ: return Collections.emptySet();
119         return new TreeSet<Book>();
120     }
121     else {
122         return stock.get(author);
123     }
124 }
125 }

```