

Software-Engineering

Praktikum SS 2006

2. Komponenten-, Integrations- und Systemtest

1 Lernziele

Vertiefen der in der Vorlesung vermittelten Kenntnisse über dynamische, white- und black-box Testverfahren für sequentielle Programme sowie dem Testen im Software-Lebenszyklus.

2 Aufgabe

Schreiben Sie ein C-Programm, mit dem ein Bücherbestand verwaltet werden kann. Ein Buch hat die Kennzeichen ISBN, Autor(en), Titel, Verlag, Jahr. Es sollen folgende Funktionalitäten implementiert werden:

- Es können neue Bücher erfasst und in den Bestand aufgenommen werden.
- Es können alte Bücher (Schlüssel ist die ISBN) aus dem Bestand gelöscht werden.
- Zu einem gegebenen Autor können alle Bücher gefunden und ausgegeben werden.
- Der Bücherbestand kann in einer Datei abgespeichert und aus einer Datei geladen werden.

Die Benutzeroberfläche kann wie folgt aussehen:

```
neues Buch aufnehmen      --> (a)ppend
altes Buch löschen        --> (d)elete
Bücher zu Autor finden    --> (f)ind
Bücherbestand ausgeben    --> (p)rint
Bücherbestand abspeichern --> (s)ave
Bücherbestand laden       --> (l)oad
Programm beenden         --> (q)uit
```

ihre Auswahl?

Alternativ können Sie eine grafische Benutzeroberfläche erstellen. Verwenden Sie zum Verwalten der Bücher die Klasse `SearchTree` aus Aufgabe 1. Die Klasse ist so zu erweitern, dass der Baum durchlaufen werden kann (Iterator), um alle Bücher eines gegebenen Autors suchen zu können.

Implementieren Sie eine Klasse `Buch`, deren Objekte in den Knoten des Suchbaums abgelegt werden. Erstellen Sie ferner eine Klasse `Buchbestand`, die folgende Methoden bereitstellt:

- `addBook(Buch b)` fügt ein Buch dem Bestand hinzu
- `deleteBook(Buch b)` entfernt ein Buch aus dem Bestand
- `Buch getBook(string isbn)` liefert das Buch zur angegebenen ISB-Nummer
- `vector getBooksFromAuthor(string author)` liefert alle Bücher zu einem gegebenen Autor
- `saveBooksToFile(string filename)` schreibt den Bücherbestand in eine Datei (alternativ: Überladen des Ausgabeoperators `<<`)
- `loadBooksFromFile(string filename)` liest einen Bücherbestand aus einer Datei aus (alternativ: Überladen des Eingabeoperators `>>`)
- Definieren Sie geeignete Exception-Klassen.

Erstellen Sie Testtreiber/-Stubs und Testfälle, um die Korrektheit Ihrer Implementierung nachzuweisen. Führen Sie dazu folgende Tests durch:

- Komponententest
- Integrationstest
- Systemtest

Der Komponententest gilt als bestanden, wenn bei den einzelnen Komponenten und einer Zweigüberdeckung von 95% keine Fehlerwirkungen aufgetreten sind.

Zum Testat müssen Sie das Programm, die Testfälle und die Testarten im Detail erklären können. Ferner sollen Sie mittels `gcov` zeigen, dass durch Ihre Testfälle die angegebenen Überdeckungsquoten tatsächlich erreicht werden.

3 Literatur

- Spillner, Linz: Basiswissen Softwaretest. dpunkt.verlag
- Zeller, Krinke: Programmierwerkzeuge. dpunkt.verlag.
- Ottmann, Widmayer: Algorithmen und Datenstrukturen. BI Wissenschaftsverlag.
- Cormen, Leiserson, Rivest: Introduction to Algorithms. MIT Press.
- Aho, Hopcroft, Ullman: Data structures and algorithms. Addison-Wesley.