

Software-Engineering

Praktikum SS 2006

1. Überdeckungstest

1 Lernziele

Vertiefen der in der Vorlesung vermittelten Kenntnisse über dynamische, white-box Testverfahren für sequentielle Programme.

2 Aufgabe

Implementieren Sie eine Klasse `SearchTree` in C++. Die Klasse soll als Template-Klasse geschrieben sein, so dass Objekte einer beliebigen kanonischen Klasse gespeichert werden können (homogene Datenstruktur).

- anlegen eines initialen Suchbaums (Konstruktor)
- freigeben des belegten Speicherbereichs (Destruktor)
- einfügen eines Wertes (`insert`) – falls der Wert im Suchbaum bereits vorhanden ist, soll eine `DuplicateValueException` geworfen werden
- suchen eines Wertes (`find`)
- entfernen eines Wertes (`delete`) – falls der Wert nicht im Suchbaum vorhanden ist, soll eine `ValueNotFoundException` geworfen werden
- ausgeben als ASCII-Zeichenfolge (`output`) – Werte sollen sortiert ausgegeben werden, indem eine In-Order-Durchlaufstrategie implementiert wird

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

- Anweisungsüberdeckungstest, Testende bei Anweisungsüberdeckung = 100%
- Zweigüberdeckungstest, Testende bei Zweigüberdeckung = 100%
- minimale Bedingungsüberdeckung
- boundary interior test

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.