

## Buchindex

Bei Sach- und Fachbüchern findet sich häufig am Ende des Buches ein sogenanntes Stichwortregister, in dem für wichtige (alphabetisch sortierte) Fachbegriffe die Seitenzahlen ihres Auftretens aufgelistet sind. Für ein Informatik-Fachbuch könnte dieses Stichwortregister etwa folgendermaßen aussehen (Es sind nur einige Stichwörter dargestellt.):

Automat	56, 63
Baum	34, 78, 83
FIFO	10
LIFO	20
Liste	20, 65, 89, 95
Schlange	27, 29

Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt.

Der Durchlauf durch das Buch soll automatisch erfolgen, da davon auszugehen ist, dass Bücher heutzutage in elektronischer Form vorliegen.

Kommt ein Stichwort mehrfach im Buch vor, so müssen für dieses Wort die entsprechenden Seitenzahlen übernommen werden.

Mehrfaches Auftreten eines Stichwortes auf einer Seite wird nur einmal berücksichtigt.

Die Gesamtheit aller Stichwörter mit Seitenangaben soll in einer geeigneten Datenstruktur erfasst werden.

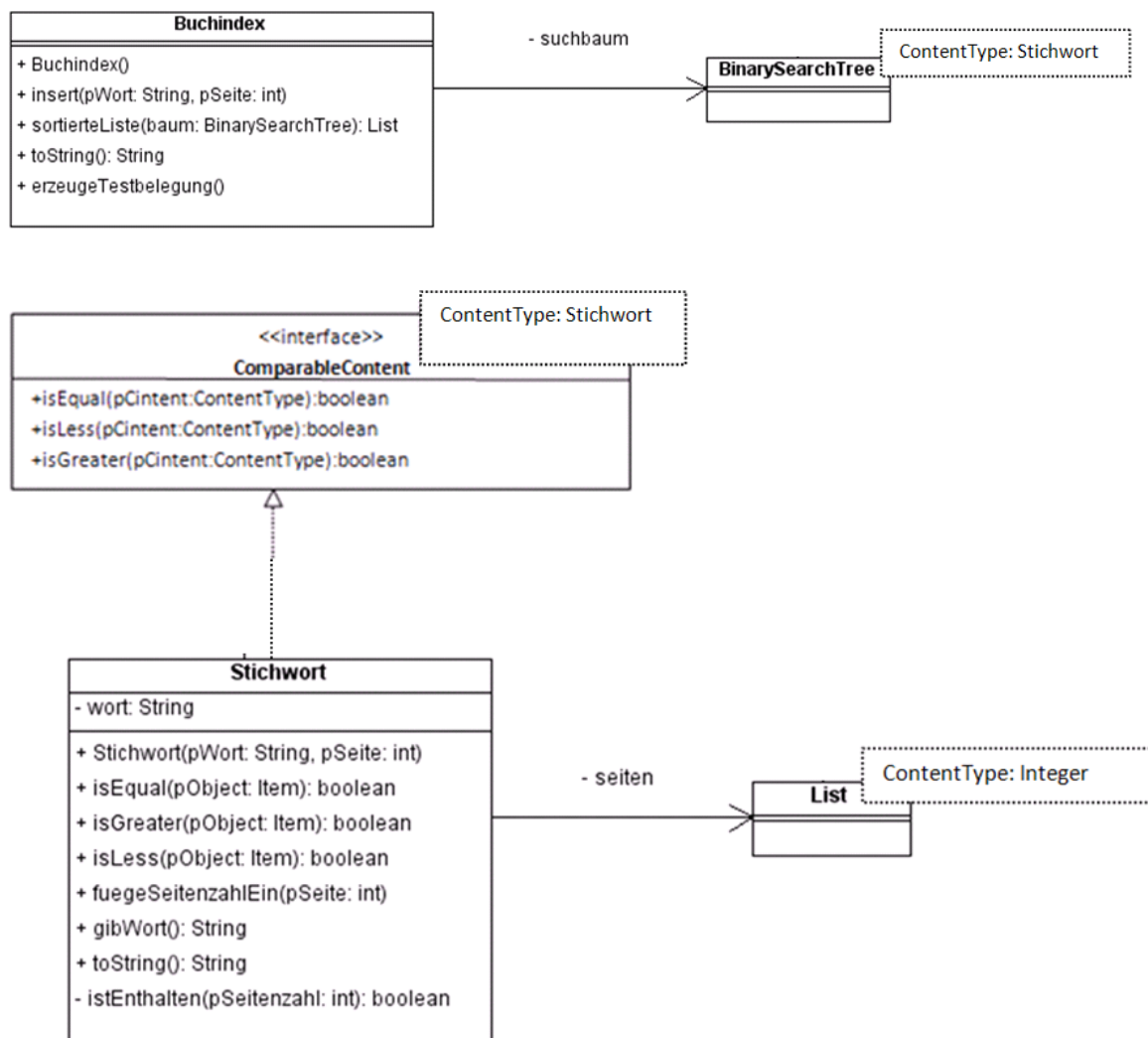
## Entwurfsdiagramm

Buchindex
- stichworte: Datenansammlung<Stichwort>
+ Buchindex ()
+ erzeugeTestbelegung ()
+ fuegeEin (pWort: Text, pSeite: Zahl)
+ sortierteIndexfolge (): Datenansammlung<Stichwort>

Stichwort
- seiten: Datenansammlung<Seitenzahl>
- wort: Text
+ Stichwort (pWort: Text, pSeite: Zahl)
+ fuegeSeitenzahlEin (pSeite: Zahl)
+ gibSeiten (): Datenansammlung<Seitenzahl>
+ gibWort (): Text
- istEnthalten (pSeitenzahl: Zahl): Wahrheitswert

Seitenzahl
- nr: Zahl
+ Seitenzahl (pNr: Zahl)
+ gibSeitenzahl (): Zahl

## Implementationsdiagramm



Bezüglich der Datenstrukturen sind folgende Entscheidungen gefallen:

### Klasse **Buchindex**

Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse **Buchindex** als Suchbaum (Objekt der generischen Klasse **BinarySearchTree**) verwaltet.

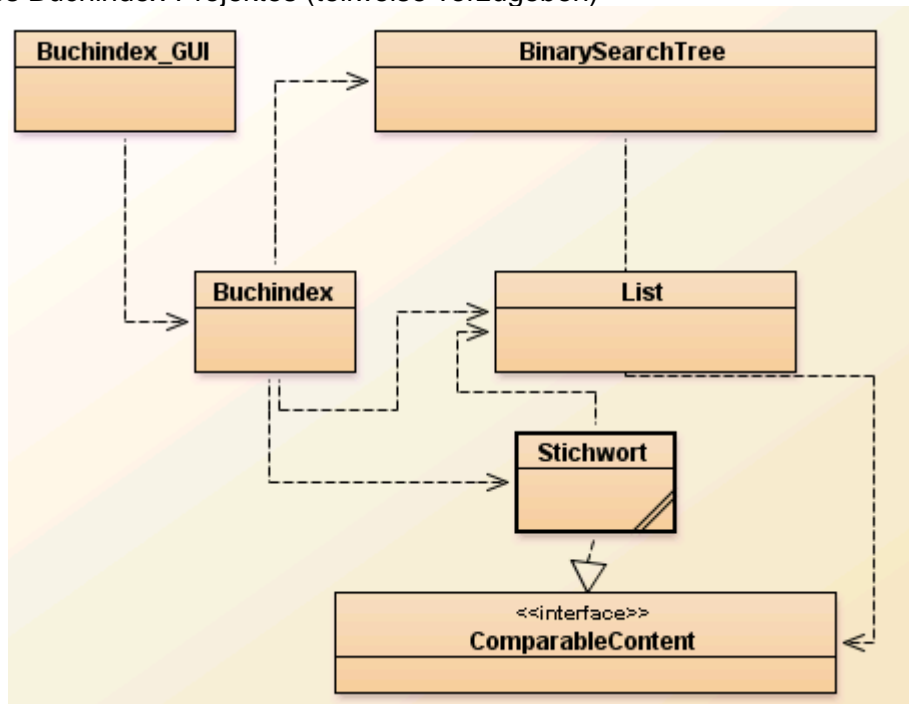
### Klasse **Stichwort**

Da die Stichwörter in einem Objekt der Klasse **BinarySearchTree** verwaltet werden, muss **Stichwort** das Interface **ComparableContent** implementieren.

Die Anzahl Seiten, die zu den Stichwörtern gehören, ist unterschiedlich. Daher werden die Seiten in einer Liste, einem Objekt der generischen Klasse **List**, verwaltet.

Zur Ausgabe können die Seiten in einem Objekt der Klasse **string** dargestellt werden.

Klassen des Buchindex-Projektes (teilweise vorzugeben)



Eine mögliche Benutzungsoberfläche für eine Testumgebung (mit zusätzlichen, schon implementierten Möglichkeiten, die das Testen erleichtern):

