**LK-Q1-V**

1. **Erarbeitung, Implementierung und Verwendung der Datenstruktur binärer Suchbaum im Anwendungskontext**
2. Erarbeitung der Eigenschaften eines binären Suchbaums im Anwendungskontext
3. Erarbeitung der Attribute und Methoden der generischen Klasse BinarySearchTree<ContentType> und des Interfaces ComparableContent
4. Implementierung des Konstruktors und der Methode search der Klasse BinarySearchTree<ContentType>

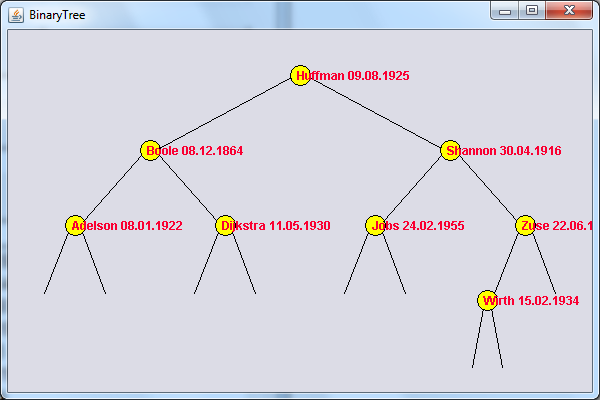
**Der Informatiker-Baum als binärer Suchbaum**

Der Problemkontext Informatiker-Baum wird wieder aufgegriffen. In einem Binärbaum sollen die Namen und die Geburtsdaten von Informatikern abgespeichert werden. Die Knoten im Baum sind nach den Namen lexikographisch geordnet. Hier wird nun die generische Klasse **BinarySearchTree<ContentType>** zur Verwaltung der Informatiker-Daten benutzt.

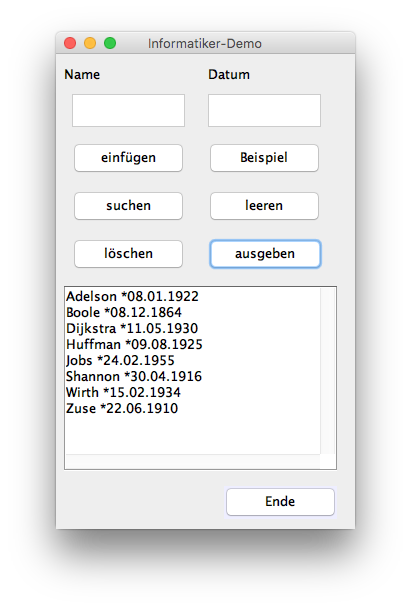
Das Programm soll folgenden Funktionalitäten haben:

* Einfügen der Informatiker-Daten in den Baum
* Suchen nach einem Informatiker über den Schlüssel Name
* Löschen von Informatiker-Daten über den Schlüssel Name
* Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge

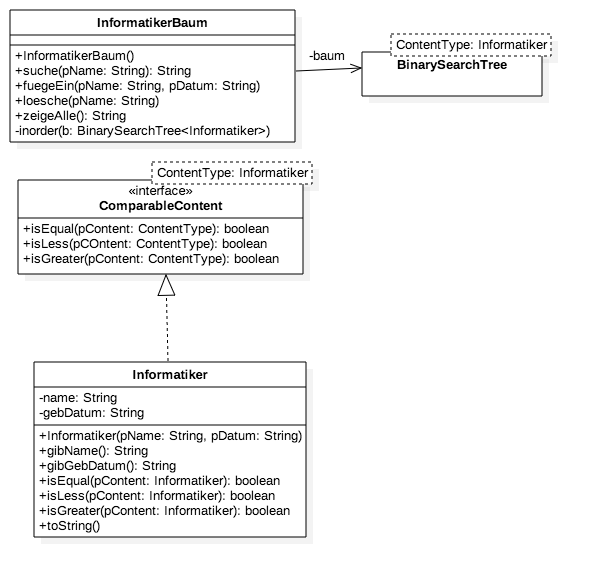
**Beispiel**



**Benutzungsoberfläche**



**Implementationsdiagramm**



**Realisierung des Informatiker-Baums mithilfe der Klasse** B**inarySearchTree<ContentType>**

Mithilfe der generischen Klasse **BinarySearchTree<ContentType>** können beliebig viele Objekte des Typs **ContentType** in einem binären Suchbaum entsprechend einer Ordnungsrelation verwaltet werden.

Ein Objekt der Klasse **BinarySearchTree** stellt entweder einen leeren Baum dar oder verwaltet ein Inhaltsobjekt vom Typ **ContentType** sowie einen linken und einen rechten Teilbaum, die ebenfalls Objekte der Klasse **BinarySearchTree** sind.

Die Klasse der Objekte, die in dem Suchbaum verwaltet werden sollen, muss das generische Interface **ComparableContent** implementieren. Dabei muss durch Überschreiben der drei Vergleichsmethoden **isLess, isEqual, isGreater** (s. Dokumentation des Interfaces) eine eindeutige Ordnungsrelation festgelegt sein.

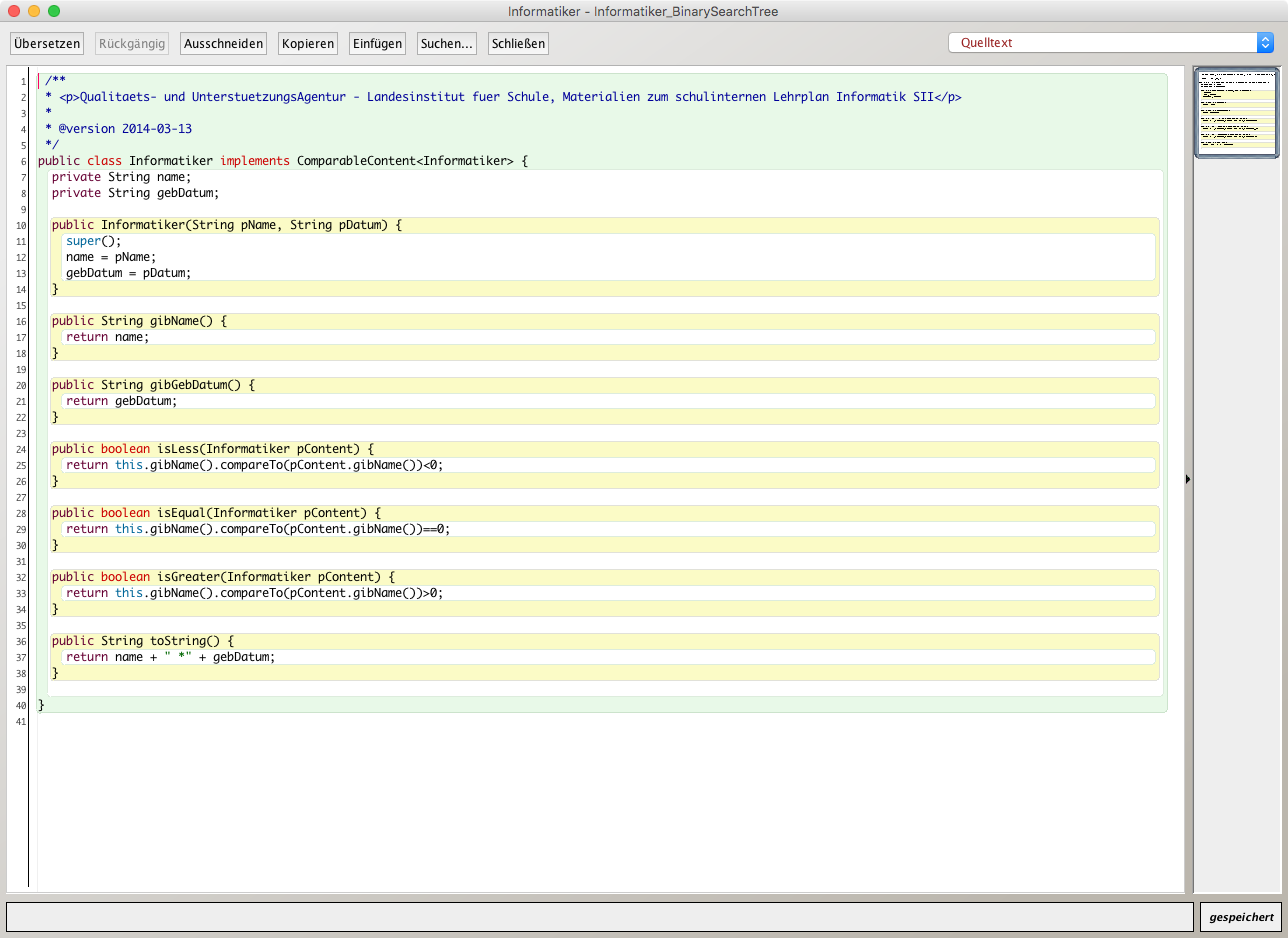
Alle Objekte im linken Teilbaum sind kleiner als das Inhaltsobjekt des Binärbaumes. Alle Objekte im rechten Teilbaum sind größer als das Inhaltsobjekt des Binärbaumes.

Diese Bedingung gilt auch in beiden Teilbäumen.

Die Klasse **BinarySearchTree** ist keine Unterklasse der Klasse **BinaryTree**, so dass deren Methoden nicht zur Verfügung stehen.

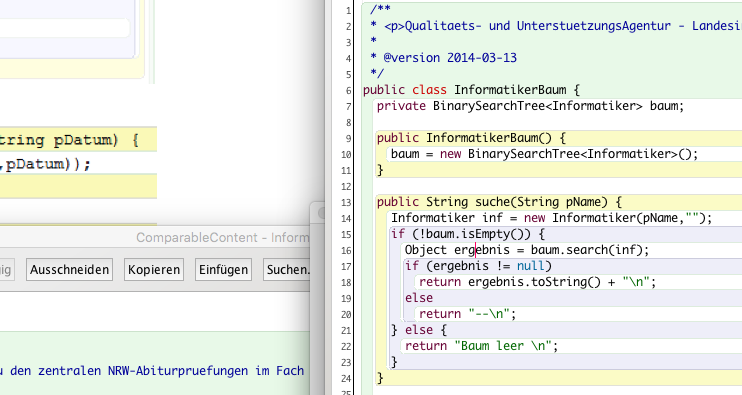
**Die Klasse Informatiker**

Die Klasse **Informatiker** implementiert das Interface **ComparableContent**.

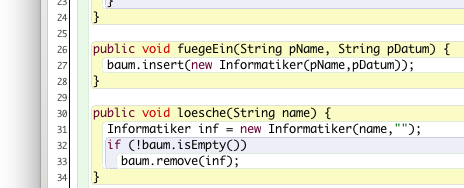


**Methoden der Klasse InformatikerBaum**

Suchen eines Informatikers



Einfügen und Löschen eines Informatikers



Sortierte Ausgabe

