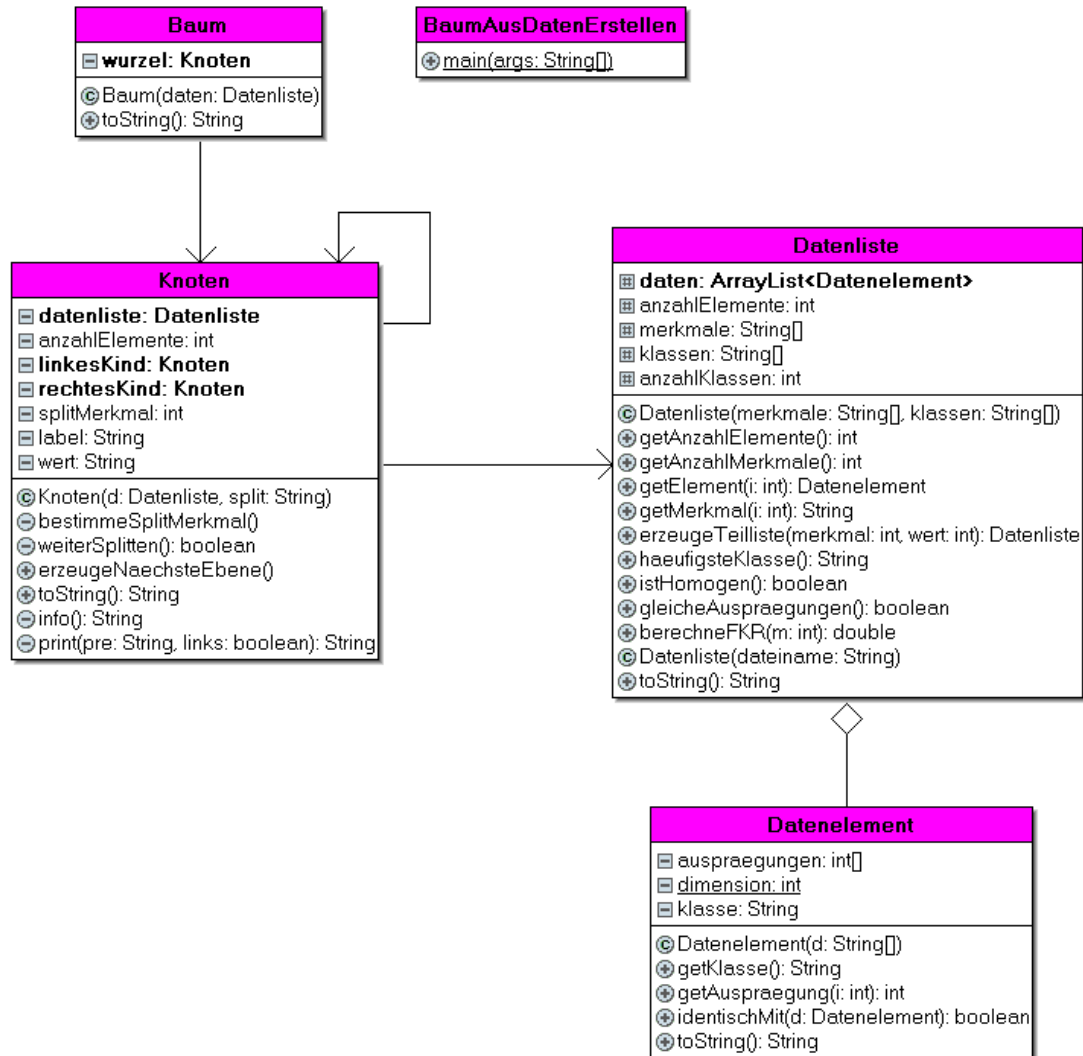


## Implementierungsvariante „nur binär“

### Überblick



- Die Klasse **BaumAusDatenErstellen** enthält die ausführbare main-Methode zur Erstellung und Ausgabe eines Objektes der Klasse **Baum**. Dort wird der Pfad der einzulesenden CSV-Datei, die die Daten enthält, übergeben.
- Der Entscheidungsbaum wird durch verkettete Objekte der Klasse **Knoten** realisiert, wobei jedem Knoten ein Objekt der Klasse **Datenliste** zugeordnet ist. Die Klasse **Baum** kapselt daher nur die Wurzel des Baumes und stößt die rekursive Baumerzeugung an.
- Die Klasse **Knoten** ist im Wesentlichen für die Baumerzeugung zuständig, während die Klasse **Datenliste** diejenigen Methoden enthält, die Berechnungen oder Veränderungen an der Datenliste durchführen.
- Eine Datenliste besteht aus Objekten der Klasse **Datenelement**.

## Beachtenswert

- **Struktur der CSV-Datei:** In der ersten Zeile müssen die in den Daten vorkommenden Klassen durch Komma getrennt angegeben werden, in der zweiten Zeile ebenso die Namen der Merkmale, erst danach folgen ab der dritten Zeile die Daten, deren Merkmalsausprägungen ebenfalls durch Kommas getrennt werden.
- Bei der **grafischen Baumausgabe** wird zunächst der Knoten mit dem dort abgefragten Merkmal ausgegeben, dann in den Zeilen darunter, etwas seitlich eingerückt, die beiden Kinderknoten. Dabei wird zuerst das rechte, dann das linke Kind ausgegeben. Die grafische Ausgabe ähnelt daher einem um 90 Grad gegen den Uhrzeigersinn gedrehten Baum.
- Was **implementieren** die **Schüler**?  
Wir empfehlen, mindestens das Einlesen der Daten aus der CSV-Datei und alle die Ausgabe betreffenden Methoden vorzugeben. Für die Schüler besonders interessant sind die Methoden `bestimmeSplitMerkmal()`, `weiterSplitten()` und `erzeugeNaechsteEbene()` der Klasse `Knoten` sowie die Methode `berechneFKR(int m)` der Klasse `Datenliste`. Zu deren Implementierung ist jedoch ein grundlegendes Verständnis der verwendeten Klassen mit ihren Methoden und Datenstrukturen erforderlich und muss daher im Voraus mit den Schülern besprochen werden.
- **Sonderfall** bei der **Baumerzeugung** (im Artikel nicht thematisiert – bei der Implementierung aber zu beachten): Die Methode `bestimmeSplitMerkmal` der Klasse `Knoten` bestimmt das beste Splitmerkmal, indem für jedes Merkmal die Fehlklassifikationsrate (FKR) mithilfe der Methode `berechneFKR` der Klasse `Datenliste` berechnet wird.  
Hierbei kann die Situation eintreten, dass es keinen Split gibt, der die Anzahl der Fehlklassifikationen (gegenüber dem Vaterknoten) verringert. In einem solchen Fall ist daher auch ein „Split“ mit einem Merkmal, in dem alle Datenelemente übereinstimmen, optimal bezüglich FKR-Minimierung. Dies produziert aber einen mit dem Vaterknoten identischen und einen leeren Kinderknoten, bewirkt also gar keine Aufteilung der Daten. Die Auswahl eines solchen Merkmals muss verhindert werden, damit keine Endlosschleife entsteht. Dies wird realisiert, indem in der Methode `berechneFKR` überprüft wird, ob alle Elemente des Vaterknotens im selben Kinderknoten landen. In diesem Fall wird als FKR ein künstlich hoher Wert zurückgeliefert, um die Wahl dieses Merkmals zu verhindern.
- Aus dem vollständigen Code kann für die Schüler ein **Framework** erzeugt werden, indem bei den zu implementierenden Methoden einfach der Rumpf gelöscht wird.