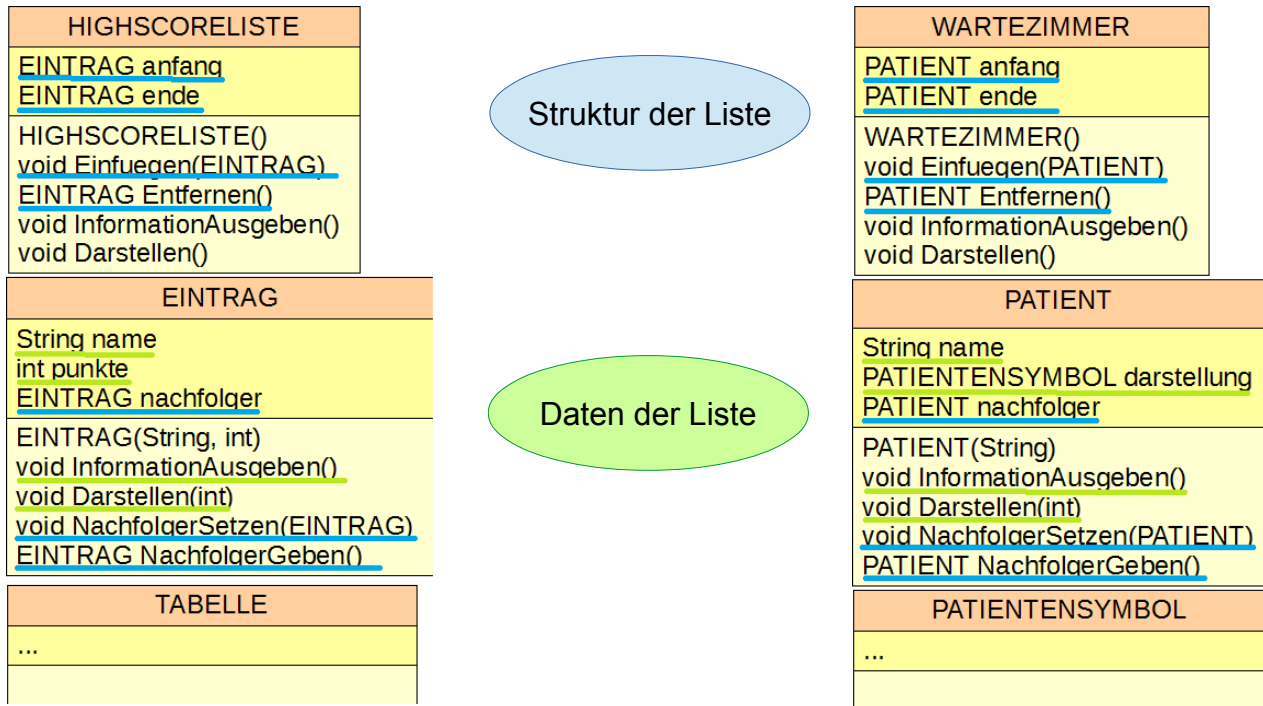


1.3 Trennung von Struktur und Daten

Eine Liste, wie wir sie nun implementiert haben, lässt sich in vielen verschiedenen Sachzusammenhängen einsetzen. Man unterscheidet dabei zwischen einfach und doppelt verketteten Listen. Einfach verkettet bedeutet, dass jeder nur den Nachfolger kennt, wobei in einer doppelt verketteten Liste, jeder Eintrag Vorgänger und Nachfolger kennt.

Als Informatiker wäre es wünschenswert nicht jedes mal einen neue Listenstruktur mit immer den gleichen Attributen und „Einfügen“/„Entfernen“ Methoden implementieren zu müssen. Daher trennen wir zwischen Struktur der Liste und Daten, die in der Liste gespeichert werden:



Um nun die Daten und Struktur voneinander zu trennen, erstellen wir für jeden Teil eigene Klassen, die nur für den jeweiligen Zweck da sind.

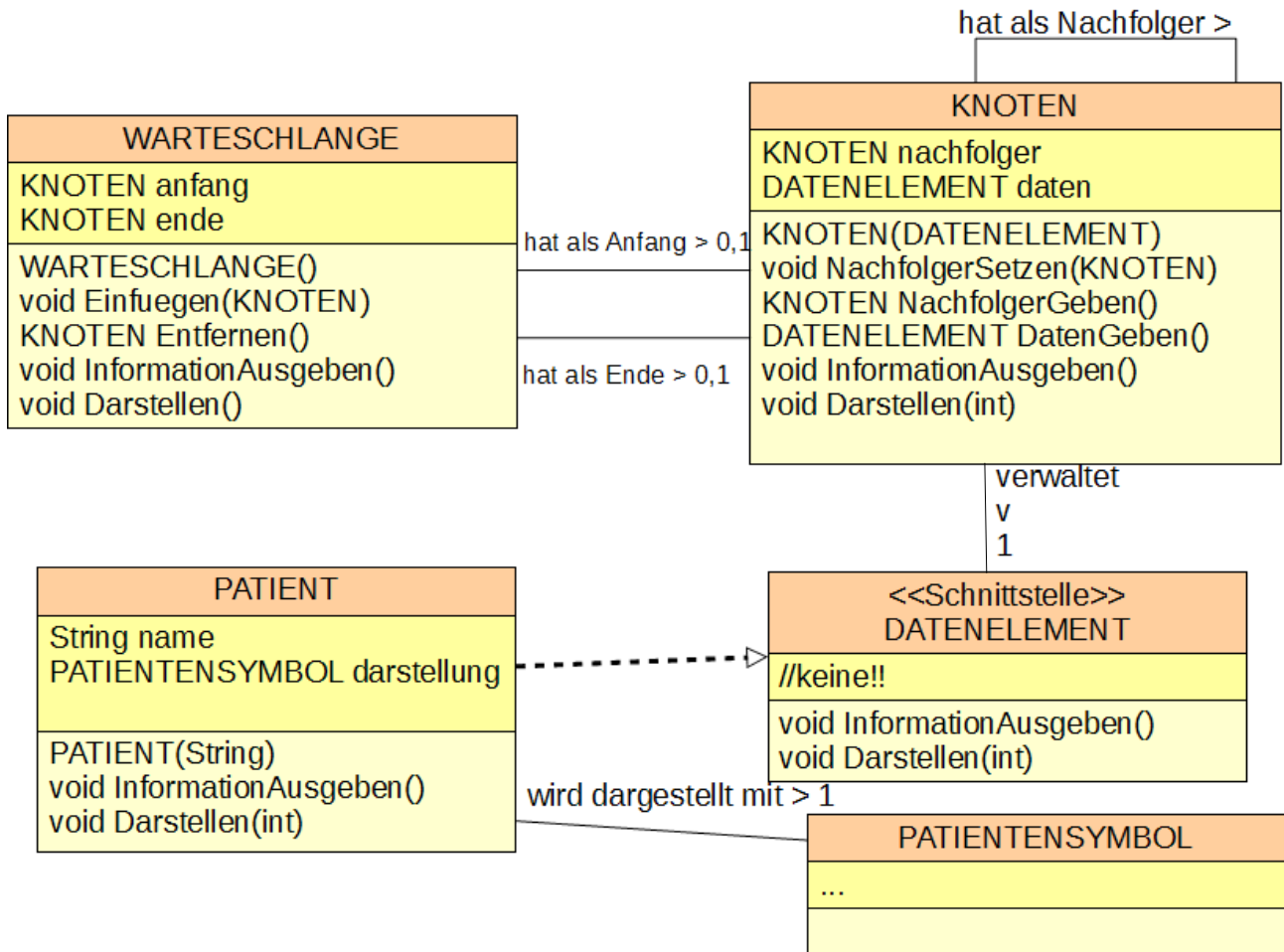
Man stellt fest, dass die Daten keine festgelegten Attribute haben, da alle Daten unterschiedlich sind. Daher verwendet man keine Vererbung mit Klassen sondern die Implementierung einer Schnittstelle (engl. Interface). Schnittstellen sind „Klassen“ die nur aus Methodenrümpfen bestehen. Sie garantieren, dass alle Klassen, die die Schnittstelle implementieren Methoden mit den gegebenen Namen und Rückgabetyt besitzen.

Code:

Definieren der Schnittstelle:	Implementieren der Schnittstelle:
<pre>public interface DATENELEMENT { void Darstellen(int i); ... }</pre>	<pre>public class PATIENT implements DATENELEMENT { //MUSS alle Methoden der Schnittstelle //überschreiben! }</pre>

Im Gegensatz zur Vererbung kann eine Klasse beliebig viele Schnittstellen nutzen.

Es ergibt sich folgendes Klassendiagramm:



Aufgaben:

- 1) Kopiere die Vorlage aus dem Klassenordner (Swing_Arztzimmer_03). Wenn du auf deiner Version weiterarbeiten willst gehe sicher, dass deine Liste einwandfrei funktioniert.
- 2) Erstelle die Klassen KNOTEN und die Schnittstelle DATENELEMENT. Füge die jeweiligen Methoden und Attribute ein.
- 3) Überarbeite die Klassen PATIENT und WARTEZIMMER (wird zu WARTESCHLANGE) so, dass nur noch die jeweiligen Bestandteile der Liste in der Klasse verbleiben. Ergänze dabei die Klasse KNOTEN gegebenenfalls um die Dinge, die raus müssen (falls nicht schon geschehen). (Hinweis: Überlege, welche Teile der Darstellen Methoden in PATIENT und WARTEZIMMER in welche Klasse WARTESCHLANGE oder KNOTEN gehören).
- 4) Überlege anhand geeigneter Darstellungen (Diagramme) wie die Methoden der neuen WARTESCHLANGE umgesetzt werden müssen. Baue dann die neuen Methodenrumpfe ein.
- 5) Weitere mögliche Erweiterungen:
 - Einfügen/Entfernen an beliebiger Stelle
 - Ausgeben der Länge der Warteschlange
 - Suchen nach Element in der Schlange (Tipp: Neue Methode, die vergleicht mit Element)
 - Sortieren/Sortiert Einfügen (Tipp: Listenelemente müssen eine Hierarchie haben!)