

**Allgemeines Prinzip und rekursive Struktur einer einfach verketteten Liste.  
Graphische Veranschaulichung: Methode Einfügen**

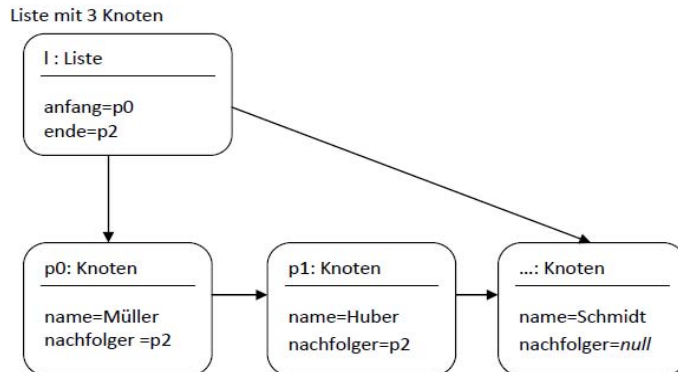


Abb. 6: Ausgangslage: Liste mit drei Knoten

zunächst muss ein neuer Knoten erzeugt werden: `Knoten k=new Knoten("Meier");`

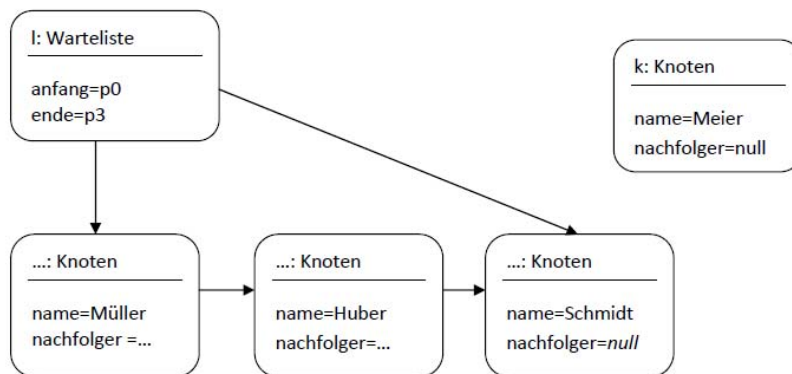


Abb. 7: Ein neuer Knoten wurde erzeugt, ist aber noch nicht in die Liste eingefügt. Beachte, dass der letzte Knoten der Liste keine Referenz auf einen Nachfolger besitzt.

Der neue Knoten wird solange nicht vom *Garbage Collector* gelöscht, solange noch die Referenz `k` vorhanden ist, also solange die Methode `add(String)` ausgeführt wird.

**Zweiter Schritt: Nachfolger setzen**

Um kein Objekt zu verlieren, wird nun als erstes der Nachfolger des bisherigen Endes gesetzt.

Der Name des Objektes ist nicht direkt verfügbar. Durch die Referenz `ende` haben wir aber Zugriff auf das Objekt.

Die Anweisung lautet daher `ende.setNachfolger(k);`

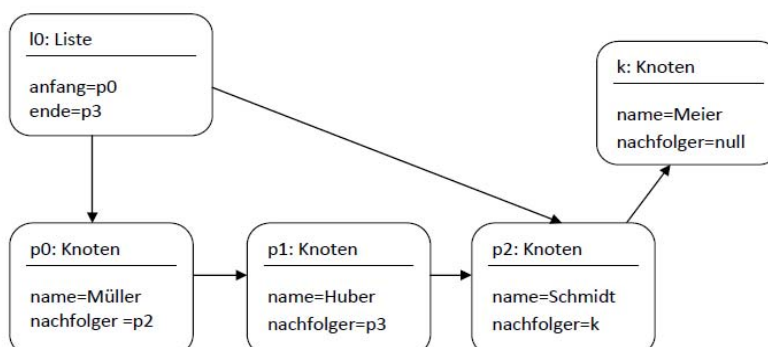


Abb. 8: Beim ursprünglich letzten Element der Liste wird die Referenz des Nachfolgers auf den neuen Knoten gesetzt.

### Dritter Schritt: Neues Ende festlegen

Nun wird noch die Referenz `ende` des Objektes `Liste` auf das neue Ende „umgebogen“ mit der Anweisung `ende=k`; Da `ende` und `k` Attribut bzw. eine lokale Variable der Klasse `Liste` sind, kann die Wertzuweisung direkt vorgenommen werden und bedarf nicht des Aufrufes einer verändernden Methode, wie das in der Anweisung `ende.setNachfolger(k)` im zweiten Schritt notwendig ist, weil hier von der Klasse `Liste` auf das Attribut der Klasse `Knoten` zugegriffen wird.

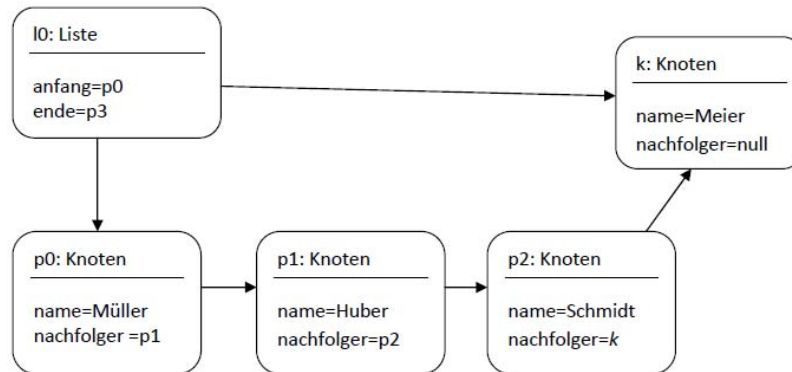


Abb. 9: Nachdem auch die Referenz `ende` auf den neuen Knoten gesetzt ist, ist die Liste wieder

**Rekursive Struktur:** Alle Knoten sind Objekte der gleichen Klasse und verfügen damit über dieselben Methoden, z. B. die Methode `setNachfolger(...)`. Beim Suchen nach einem Knoten oder bei der Ermittlung der Länge der Liste werden die Methoden rekursiv aufgerufen. Ein Objekt der Klasse `Knoten` verweist auf ein Objekt derselben Klasse.

