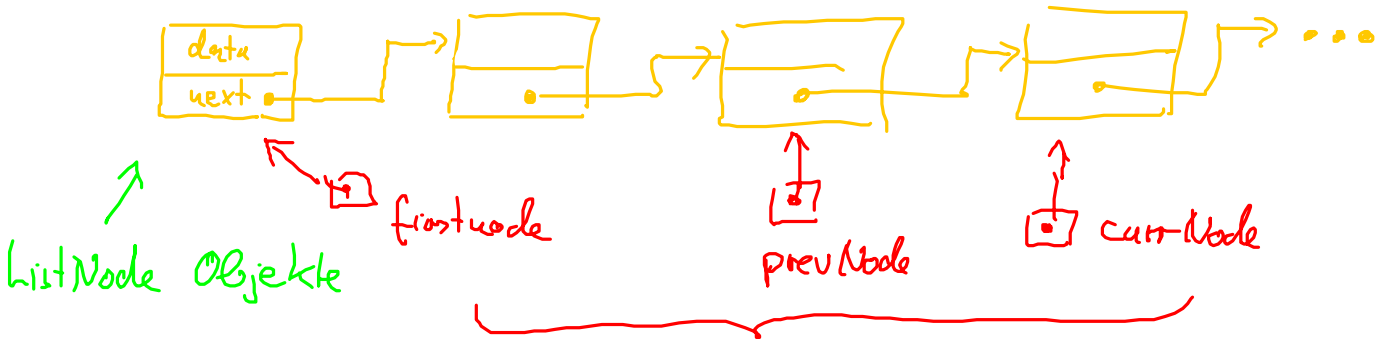
schriftliche Rücksprache

Di 14.12.04 12^{00} - 14^{00} Uhr

Listen als Datenstrukturen

Klasse LinkedList selber gestalten

Idee:



private Datenfelder der Klasse LinkedList zur Implementation (implementationsabhängig)

machen nur die Operationen public

bereits implementiert:

- Konstruktor
- isEmpty()
- reset()
- endOfList()
- advance()

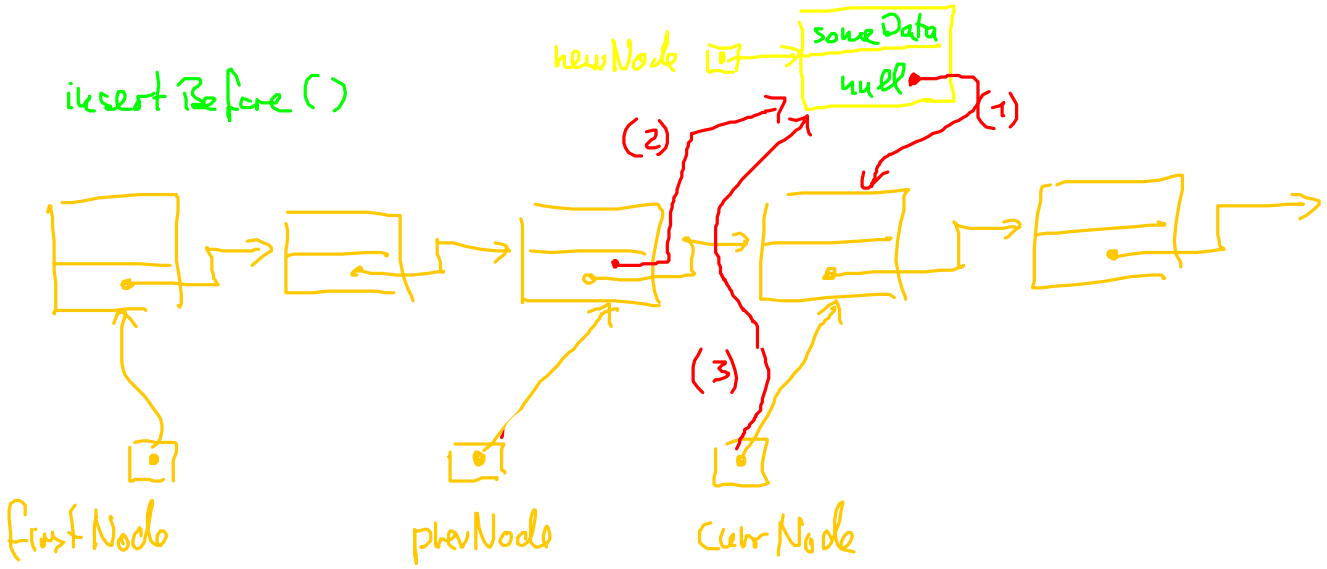
---

geht currentData() gibt Daten des momentanen Listenelementes

zurück

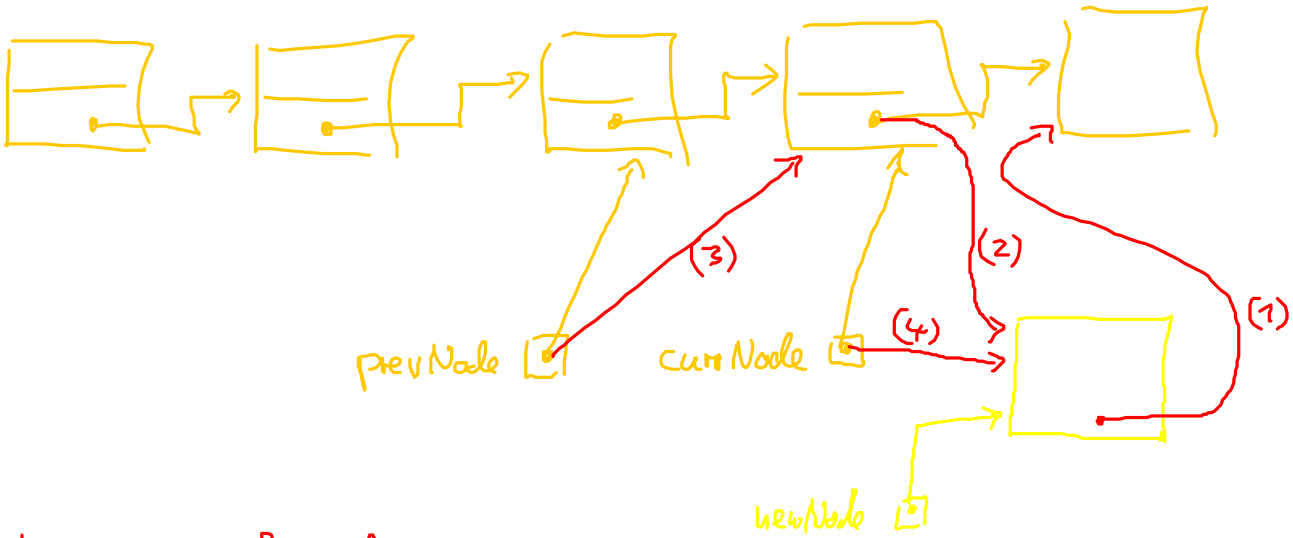
insertBefore()	fügt neuen Knoten vor aktuellem Listenelem. (ein)
insertAfter()	fügt ... hinter ----- .
delete()	löscht akt. Listenelement

Java code




```
// Erzeuge neuen Knoten mit Konstruktor der Klasse ListNode

ListNode newNode = new ListNode( someData );

in Rot: die zu verändernden Referenzen

newNode.setNext( currNode );      (1)

prevNode.setNext( newNode );      (2)

currNode = newNode; // Alternative
                    currNode = prevNode.getNext();
```

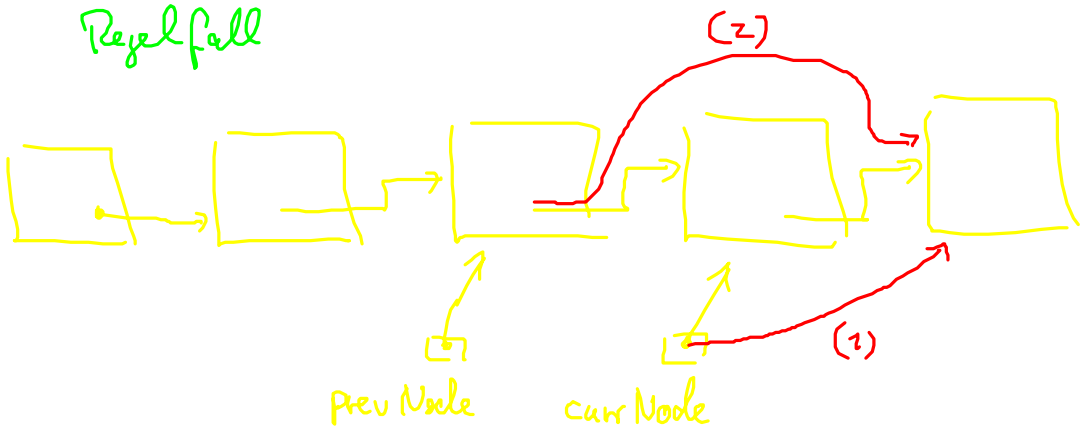
insertAfter() Standardfall



rot = gearderk Referenzen

```
newNode.setNext( currNode.getNext() );      (1)

currNode.setNext( newNode );                (2)

prevNode = currNode;                        (3)

currNode = newNode;                         (4)
```

delete() Regelfall




```
currNode = currNode.getNext();

prevNode.setNext( currNode );
```

damit ist Listenelement aus der Liste ausgehängt

es existiert physikalisch im Speicher und weiter (kann von uns nicht mehr angesprochen werden). Es existiert so lange weiter bis die automatische Java-Garbage-Collection den Speicher wieder zur Verfügung stellt

---

Ein Test für Listenklasse:

- Einlesen eines Strings in eine Liste von Zeichen
- Ausgabe der Liste auf dem Bildschirm
- Löschen des Anfangs der Liste bis zu einem vorgegebenen Zeichen
- Ausgabe der gekürzten Liste

Liste enthält allgemeine Objekte

Zeichen sind char in Java

↑ eingebauter Typ ⇒ keine Objekte

daher Wrapperklasse Character von char verwenden

Umgekehrt beim Datenzugriff müssen die allgemeinen Objekt gecasted werden auf Character Objekte und aus denen der char Wert rausgeholt werden

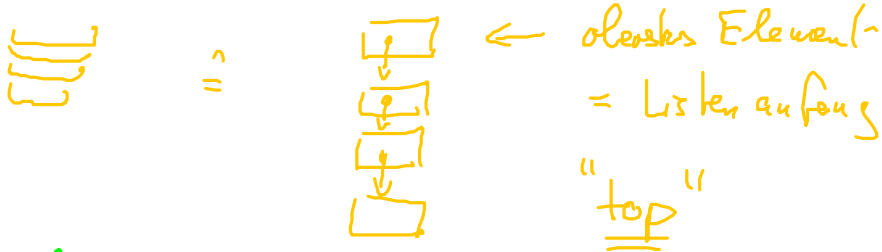
```
Character Ch;
Ch = (Character) stringList.currentData();
char ch = Ch.charValue();
```

Java Applet ListDemo

5.7 Stacks

spezielle Listen, bei denen Einfügen und Löschen nur am Anfang der Liste erlaubt ist

Vorstellung: Stapel (= stack) von Tablets



Operationen heißen dann:

top()	Daten des obersten Elements zurückgeben
pop()	oberstes Element löschen
push()	neues oberstes Element einfügen
isEmpty()	Test auf leer

Klasse dafür schreiben

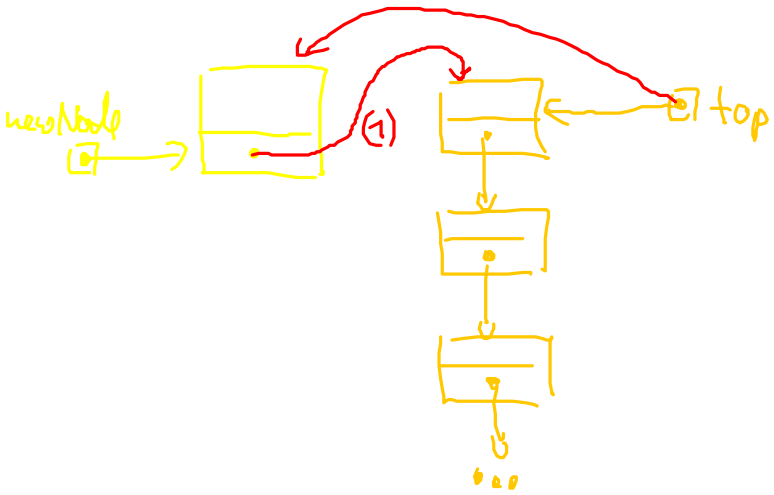
Anwendungen: Erkennung korrekter Klammerausdrücke

RPN-Rechner

Abarbeitung von Methodenaufrufen in Prog. Sprachen

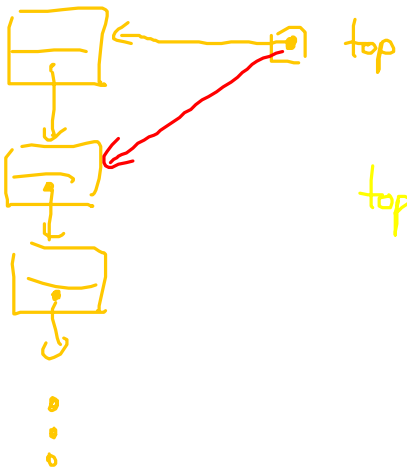
Java Klasse ansehen

push() Standardfall




```
newNode.setNext(top);   (1)
                        (2)
top = newNode;
```

pop() Standardfall




```
top = top.getNext();
```