

Hinweise zum Miniprojekt 3

Auf den folgenden Seiten sind die Hinweise zu Miniprojekt 3 – wie in der Vorlesung vorgestellt – zusammengefasst.

1. Lebensmittelliste.java und Lebensmittel.java sowie LTest.java zur Demonstration der FIFO-Liste
2. Schematische Darstellungen der FIFO, LIFO und doppelt verketteten Liste

```
class Lebensmittel
{ String Name;
  Lebensmittel next;
  Lebensmittel(String Str)
  {
    Name = Str;
  }
}

class Lebensmittelliste
{
  Lebensmittel Kopf, Fuss;

  void einfuegen(Lebensmittel lbm)
  {
    // pruefen ob der Parameter nicht null ist
    if(lbm== null)
      {System.out.println("kein Lebensmittel - Objekt uebergeben");
        return;
      }
    else
      {
        // erster Fall Liste ist leer
        if (Kopf == null)
          {
            Kopf = Fuss = lbm;
            lbm.next=null;
          }
        else
          {// Liste nicht leer
            // Fifo: am Ende einfuegen
            Fuss.next= lbm; // Schritt 1
            Fuss = lbm;    // Schritt 2
          }
        }
      }
}

Lebensmittel suche(String dasLbm)
{ // Durchlaufe die Liste mit dem Verweis hilf bis gefunden oder
  // das Ende der Liste (hilfe == null) erreicht ist ....

  Lebensmittel hilf = Kopf;
  while (hilfe != null)
  {
    if (hilfe.Name.compareTo(dasLbm) ==0)
      {
        return hilf;
      }
    else hilf = hilf.next;
  }
}
```

```
    }
    return null;
}

Lebensmittel entnehme()
{ // FIFO: Das erste Element, das in die Liste eingefuegt wurde ist ueber
  // Kopf direkt erreichbar!

    Lebensmittel hilf = null;

    // Betrachte die folgenden Faelle:
    // Liste leer
    // Liste nicht leer und genau ein Element ist vorhanden (->Fuss muss
    // noch manipuliert werden)
    // Liste nicht leer und mehr als ein Element

    if (Kopf == null)
        { // Liste leer -> liefere null, da nichts entnommen werden kann
          return null;
        }
    else
        { // Liste ist nicht leer
          if (Kopf == Fuss)
              { // Liste enthaelt nur ein Element und wenn es entnommen
                // wird ist die Liste leer
                hilf = Kopf; hilf.next=null;
                Kopf = Fuss = null;
                return hilf;
              }
          else
              {
                // Liste enthaelt mehr als ein Element
                hilf = Kopf; //Schritt 1
                Kopf = Kopf.next; //Schritt 2
                hilf.next = null; //Schritt 3
                return hilf;
              }
            }
        }
    }
}
```

```
class LTest
{
    public static void main(String [] args)
    {
        Lebensmittelliste LLager = new Lebensmittelliste();

        String [] Eingabe = { "Gouda","Gorgonzola",
                               "Creme Fraiche", "Milch", "Sahne", "Joghurt",
                               "Quark", "alles Kaese"};

        String hilf;

        Lebensmittel lbm=null;

        for (int i=0; i< Eingabe.length;i++)
            LLager.einfuegen(new Lebensmittel(Eingabe[i]));

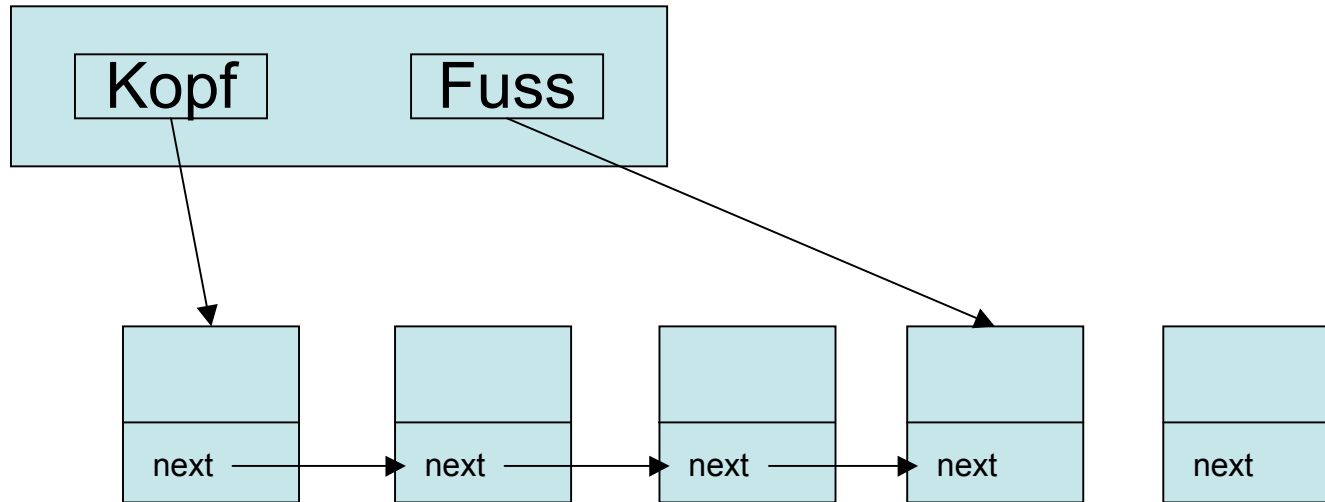
        for (int j=0 ; j<args.length; j++)

        {
            lbm = LLager.suche(args[j]);
            if (lbm == null)
                { System.out.println(args[j] + " nicht vorhanden");
                }
            else System.out.println(lbm.Name + " vorhanden");
        }

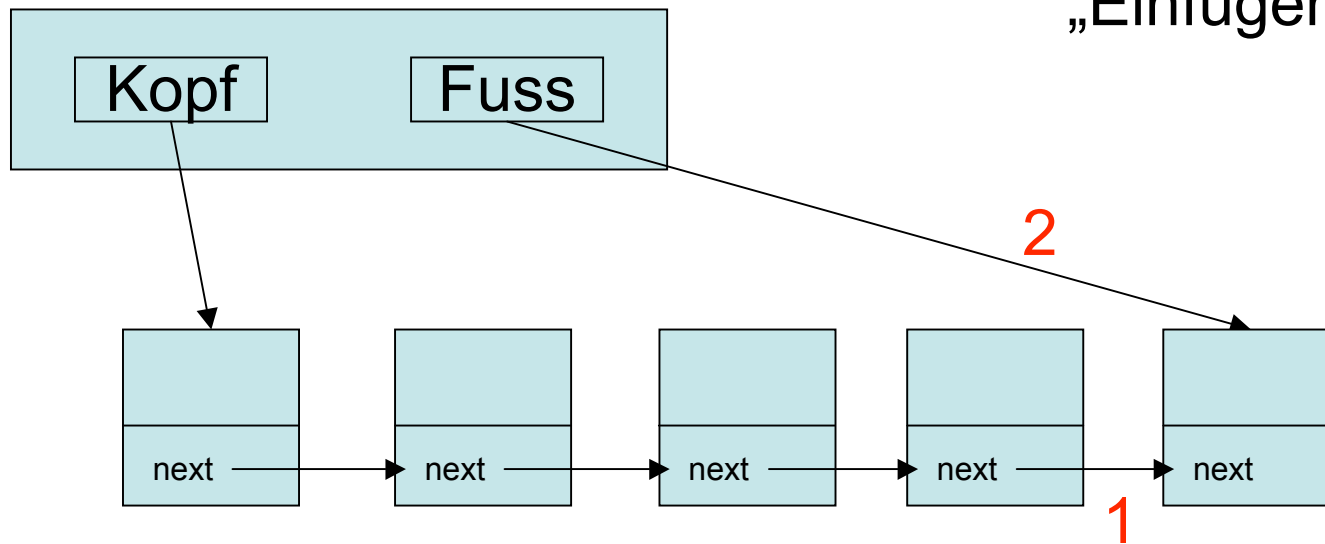
        System.out.println("-----");

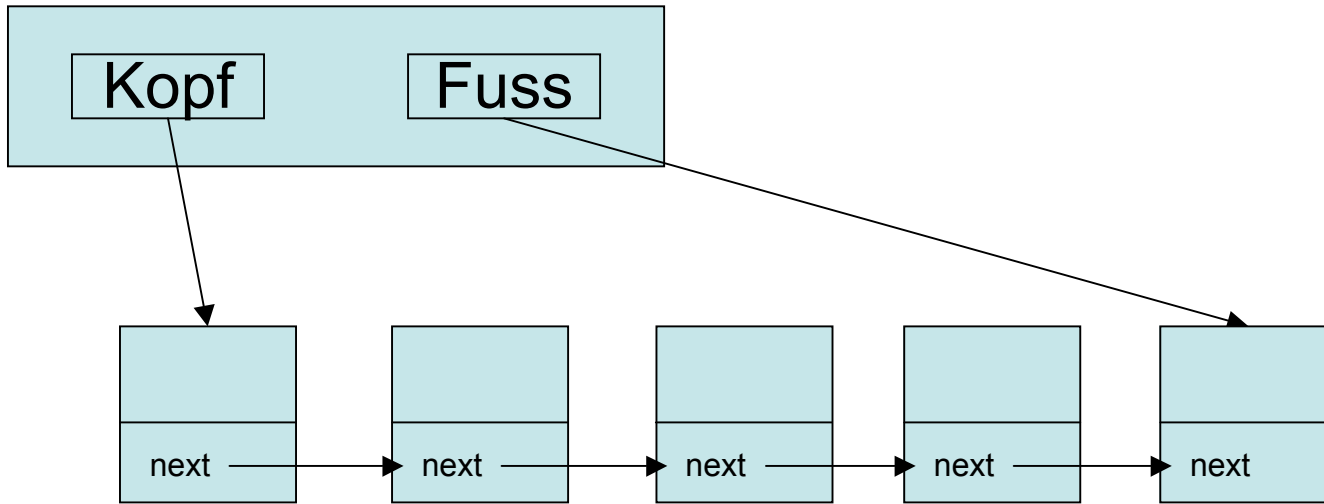
        while (LLager.Kopf !=null)
        { lbm = LLager.Kopf;
          while (lbm!=null)
          {
              System.out.println(lbm.Name);
              lbm =lbm.next;
          }
          System.out.println("-----");
          LLager. entnehme();
        }

    }
}
```

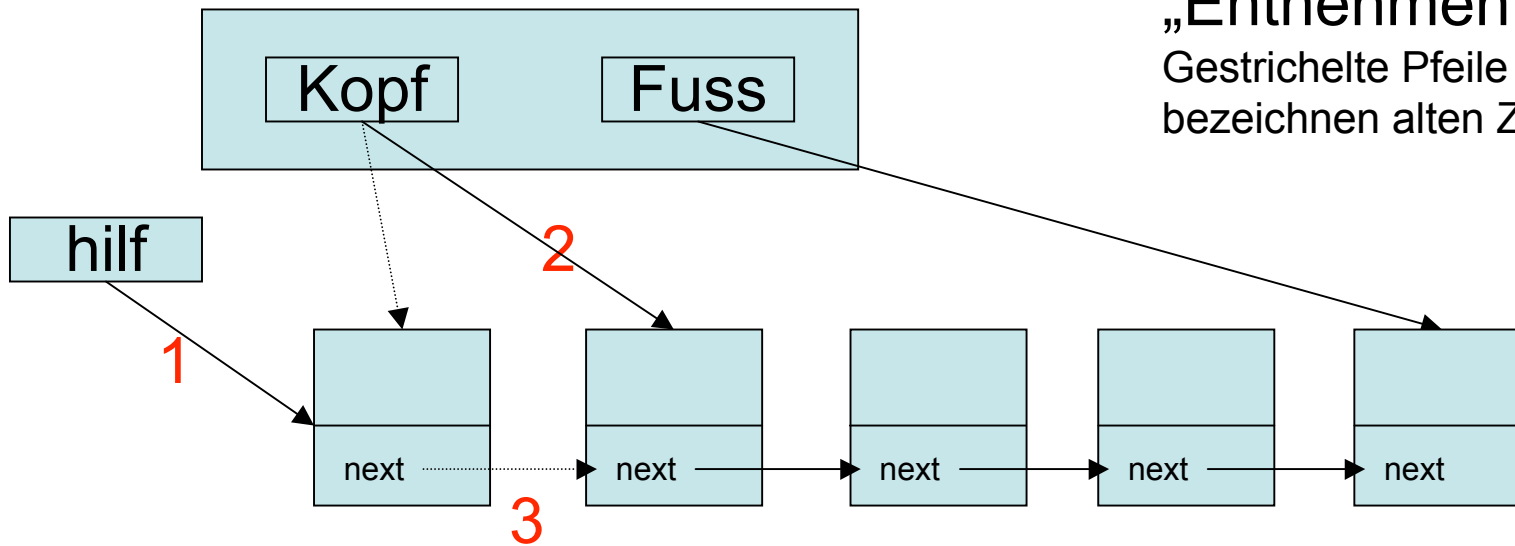


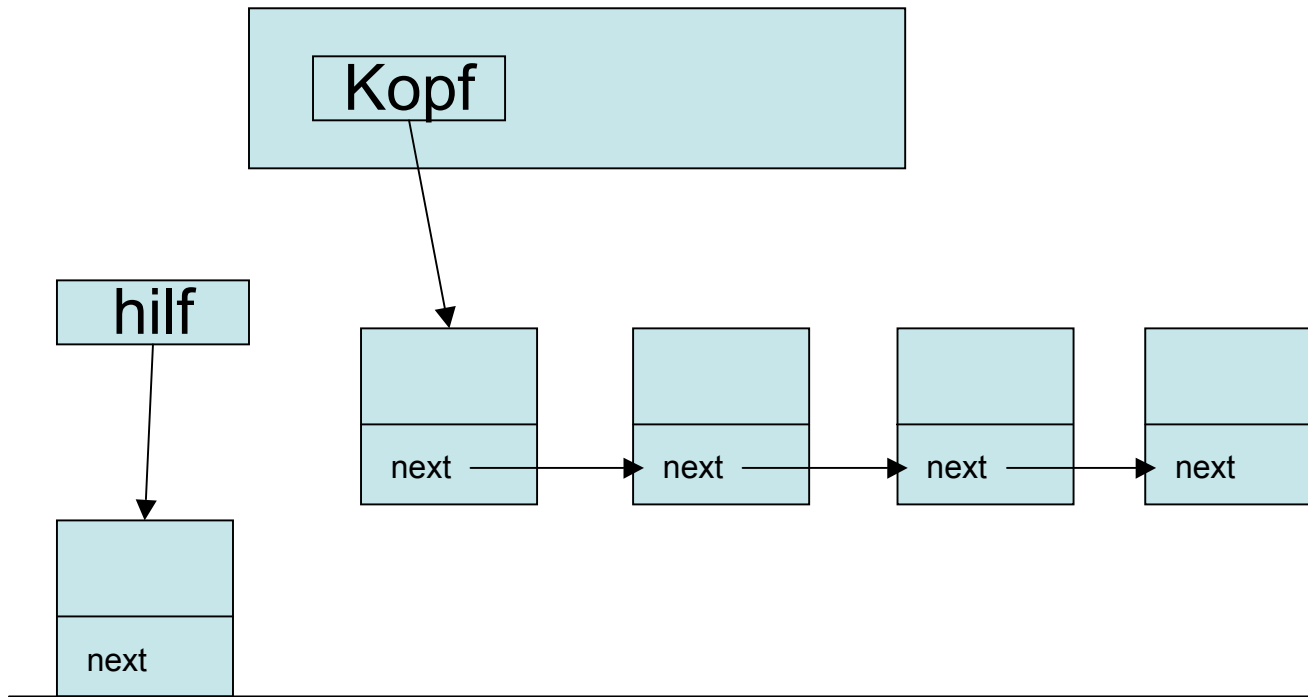
„Einfügen nach FiFo“



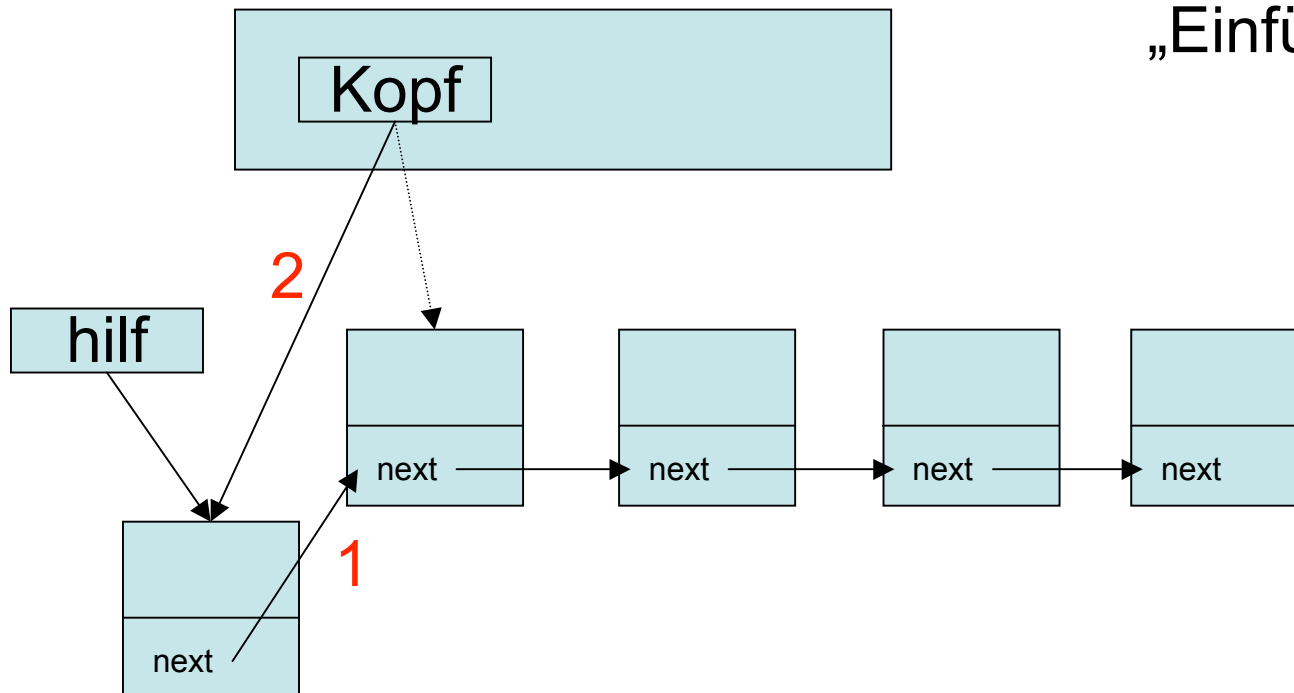


„Entnehmen FIFO“
Gestrichelte Pfeile
bezeichnen alten Zustand





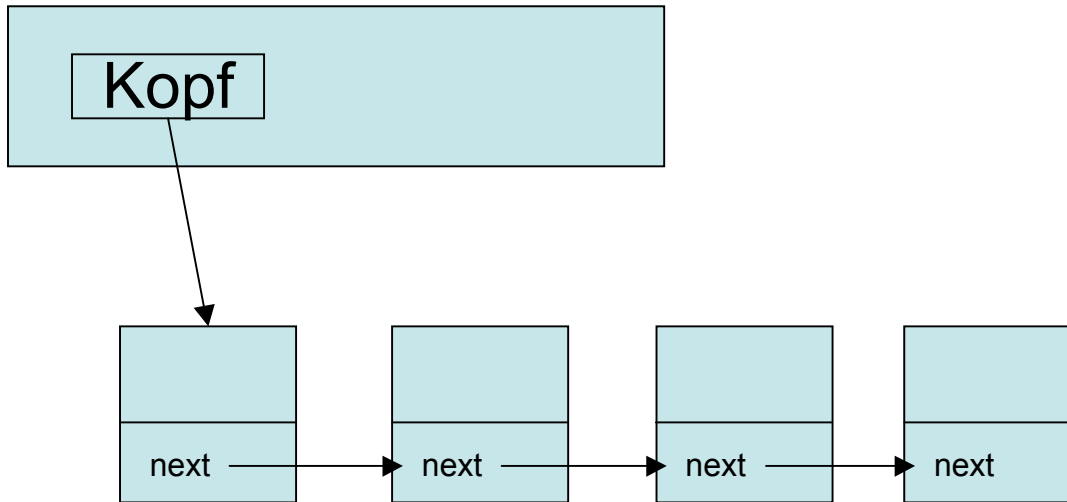
„Einfügen nach LiFo“



```

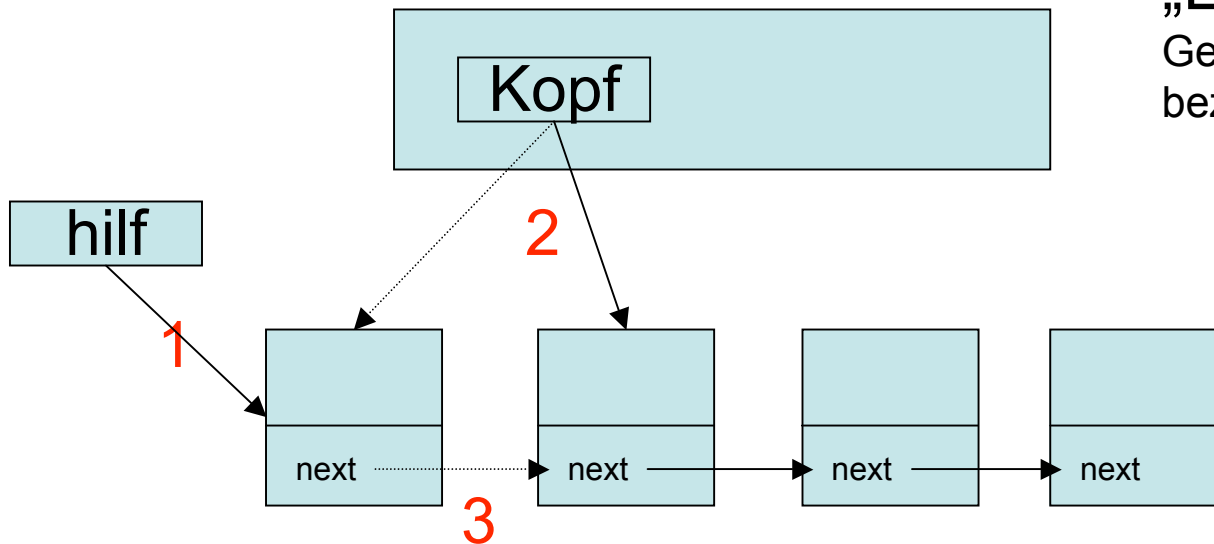
.
.
hilf.next = Kopf; // 1
Kopf = hilf;     // 2
.
.

```



„Entnehmen LIFO“

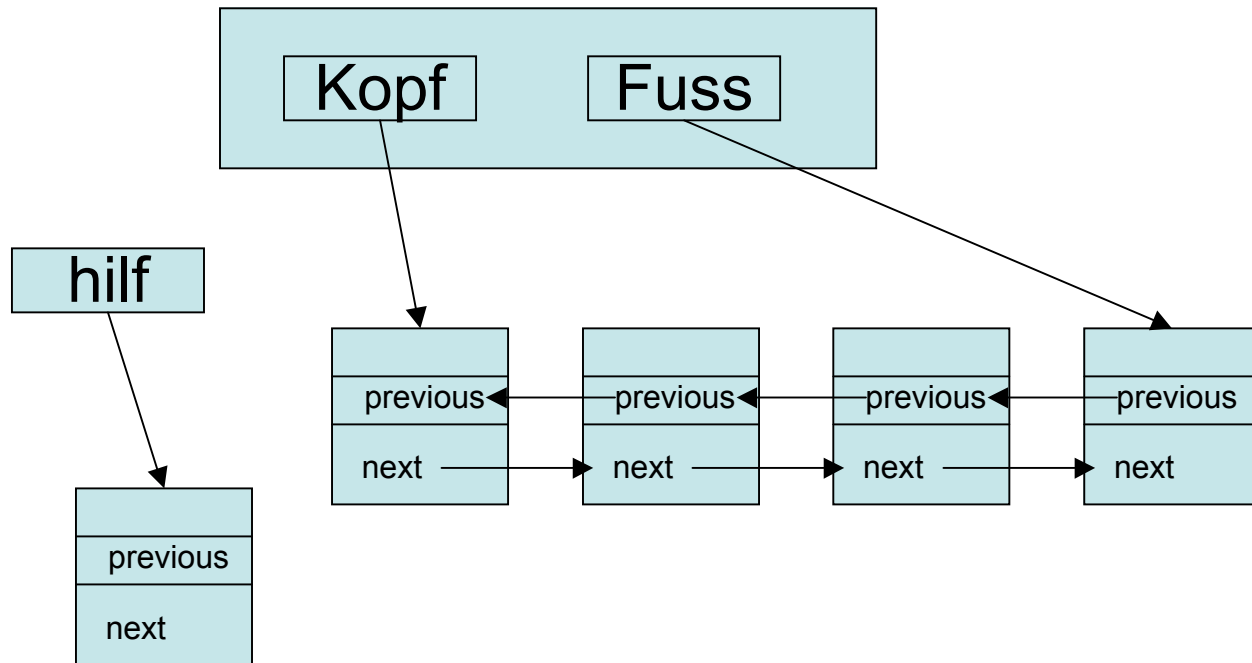
Gestrichelte Pfeile
bezeichnen alten Zustand



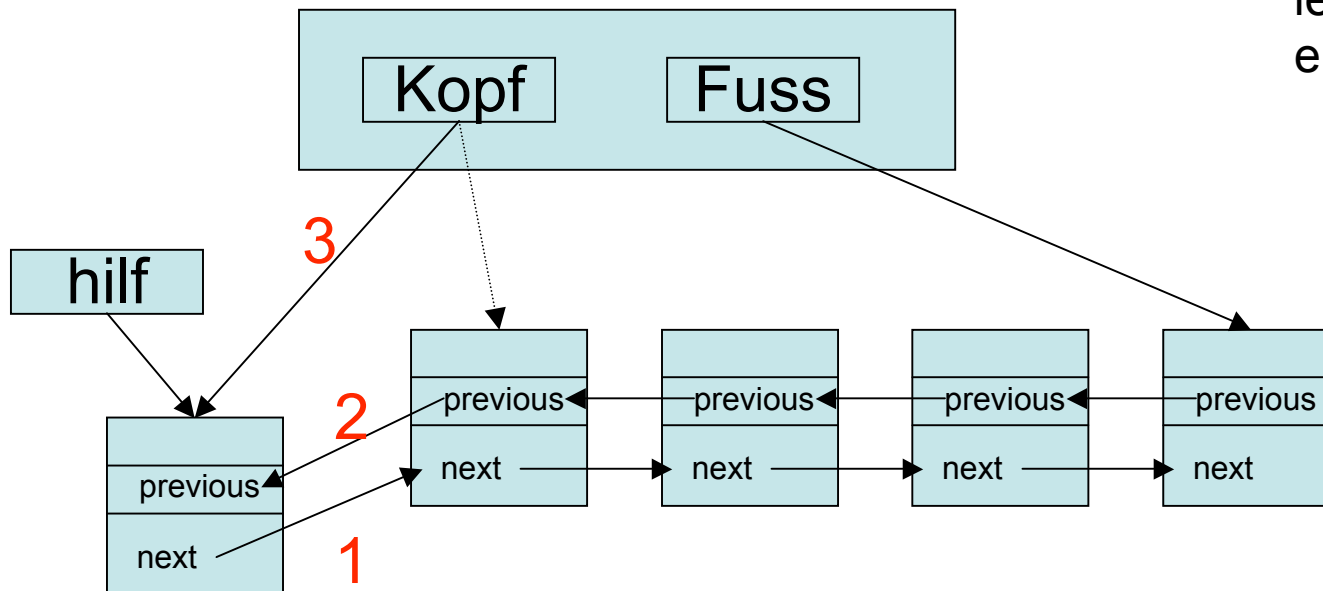
```

.
hilf = Kopf;      // 1
Kopf = Kopf.next; // 2
hilf.next = null; // 3
return hilf;
.

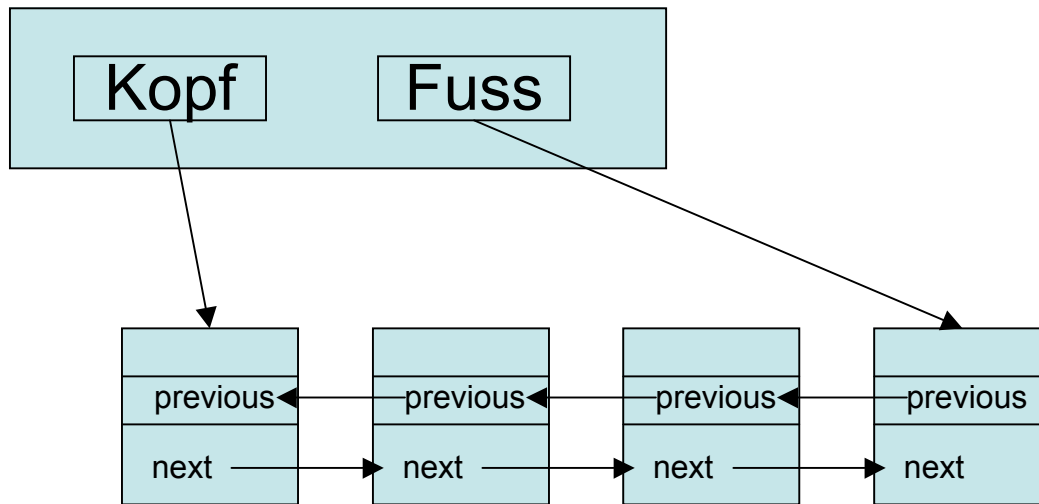
```



„Einfügen am Anfang
 Beachte die Spezialfälle
 leere Liste und Liste mit
 einem Element

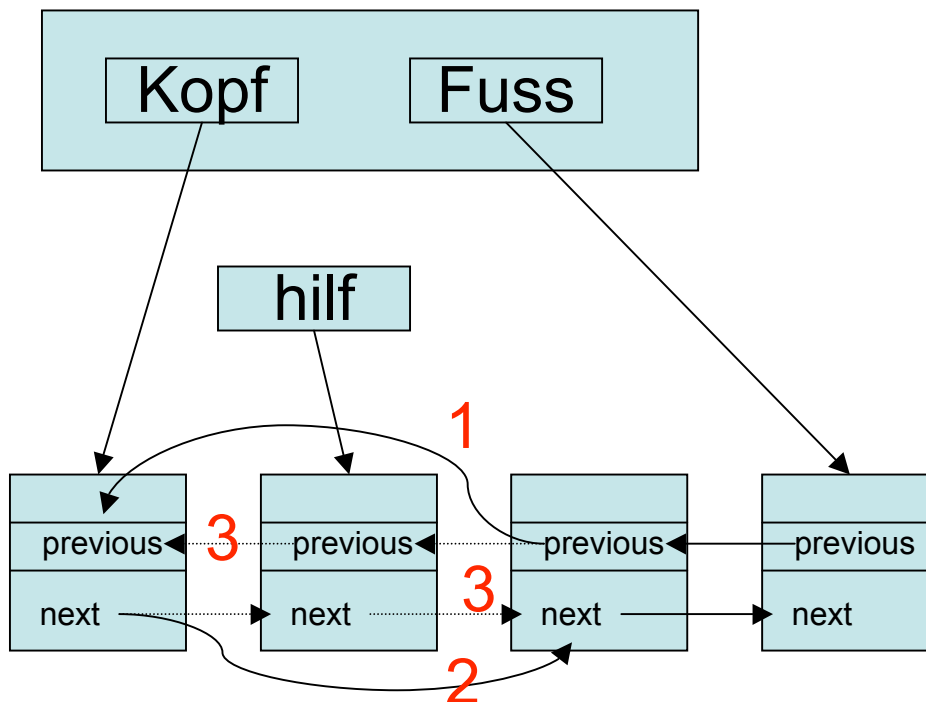


```
// einfuegen in nicht-
// leere Liste
hilf.next = Kopf;      // 1
Kopf.previous = hilf; // 2
Kopf = hilf;          // 3
.
.
```



Entnehmen nach Nummer

1. Durchzählen bis gewünschtes Element gefunden
2. Element „ausklinken“
3. Kopf und Fuss updaten, falls letztes oder erstes Element entnommen wurde



```

if (hilf.next != null)
  { hilf.next.previous = hilf.previous;} // 1
else
  { // letztes Element -> Fuss updaten
    Fuss = hilf.previous;
  }

if (hilf.previous != null)
  {
    hilf.previous.next = hilf.next; // 2
  }
else
  { // erstes Element -> Kopf updaten
    Kopf = hilf.next;
  }
}

hilf.next = hilf.previous = null; // 3

return hilf;

```