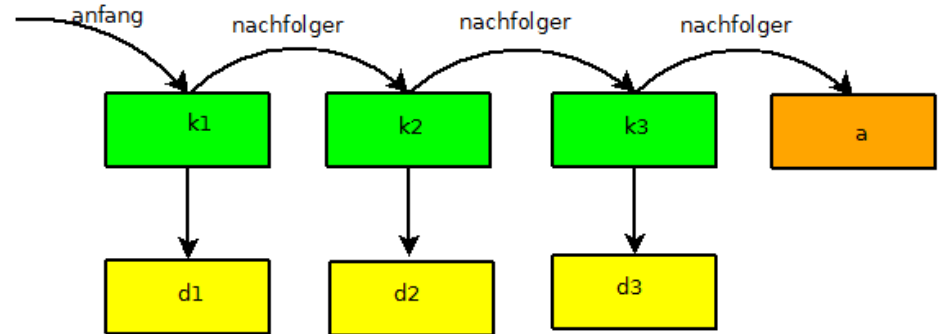
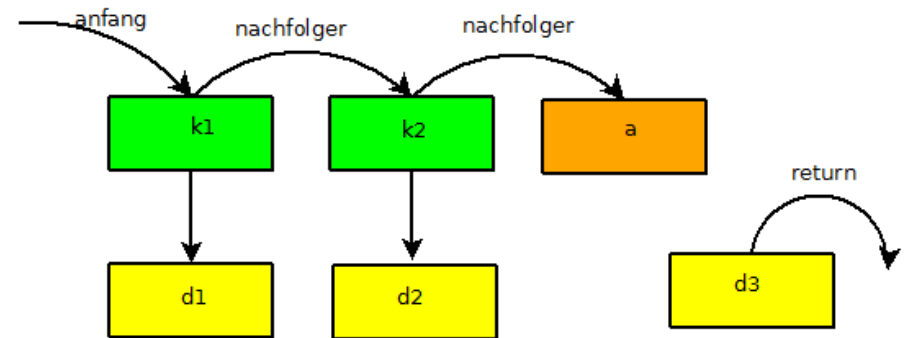


# Die Methode public Datenelement endeEntfernen ()



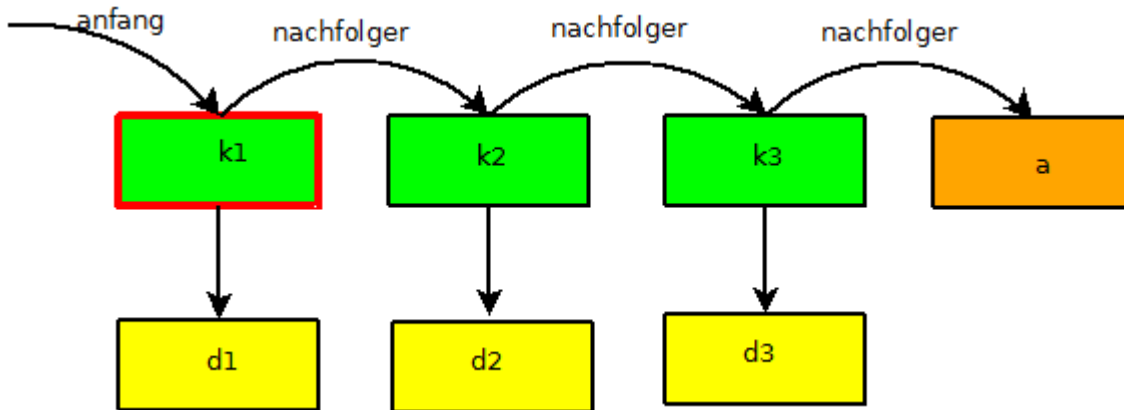
```
//Klasse Liste
public Datenelement endeEntfernen(){
    Datenelement d;
    d = anfang.getEnde(null);
    anfang = anfang.endeEntfernen(d);
    return d;
}
```



```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

```
//Klasse Abschluss
public Listenelement endeEntfernen(Datenelement d){
    return this ;
}
```

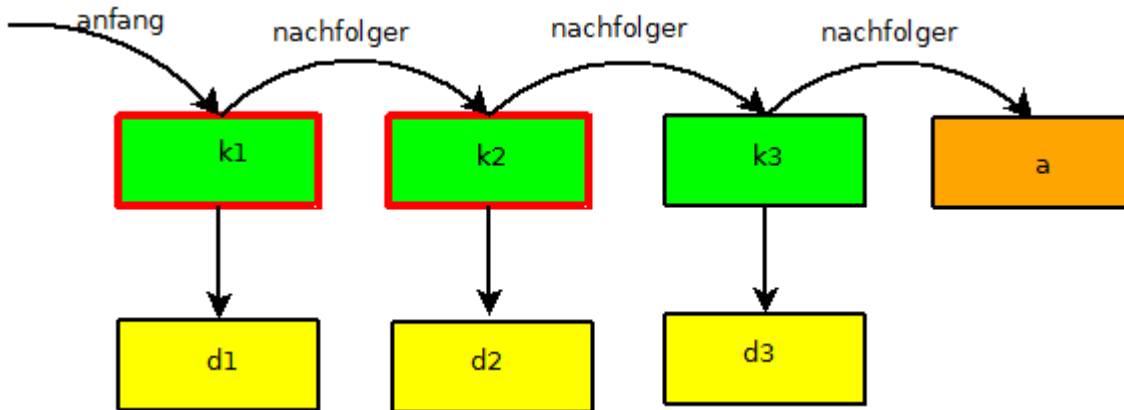
## Die Methode public Datenelement endeEntfernen ()



k1.endeEntfernen(d3)  
nachfolger = k2.endeEntfernen(d3);  
(rekursiver Aufruf; wird noch nicht ausgewertet;  
k1 bleibt aktiv und wartet auf Antwort;  
durch roten Rahmen gekennzeichnet)

```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

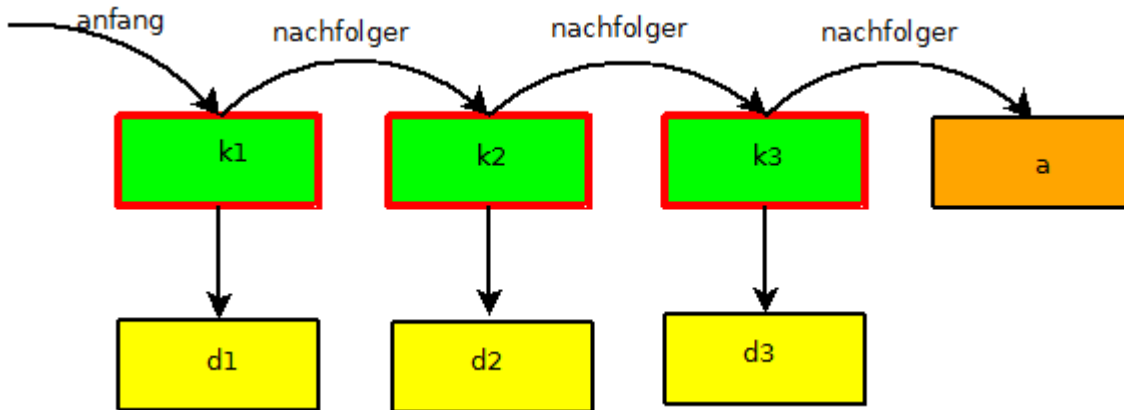
## Die Methode public Datenelement endeEntfernen ()



k2.endeEntfernen(d3)  
nachfolger = k3.endeEntfernen(d3);  
(rekursiver Aufruf; wird noch nicht ausgewertet;  
k2 bleibt aktiv und wartet auf Antwort)

```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

## Die Methode public Datenelement endeEntfernen ()

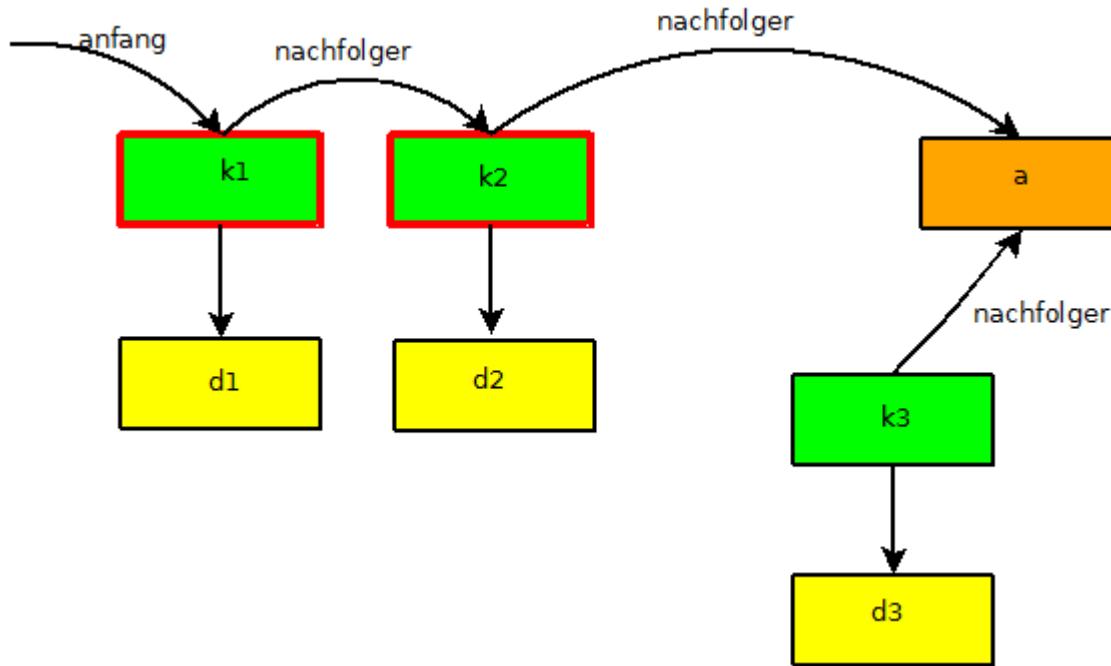


k3.endeEntfernen(d3)  
return nachfolger (also a)

(Rekursionsende)

```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

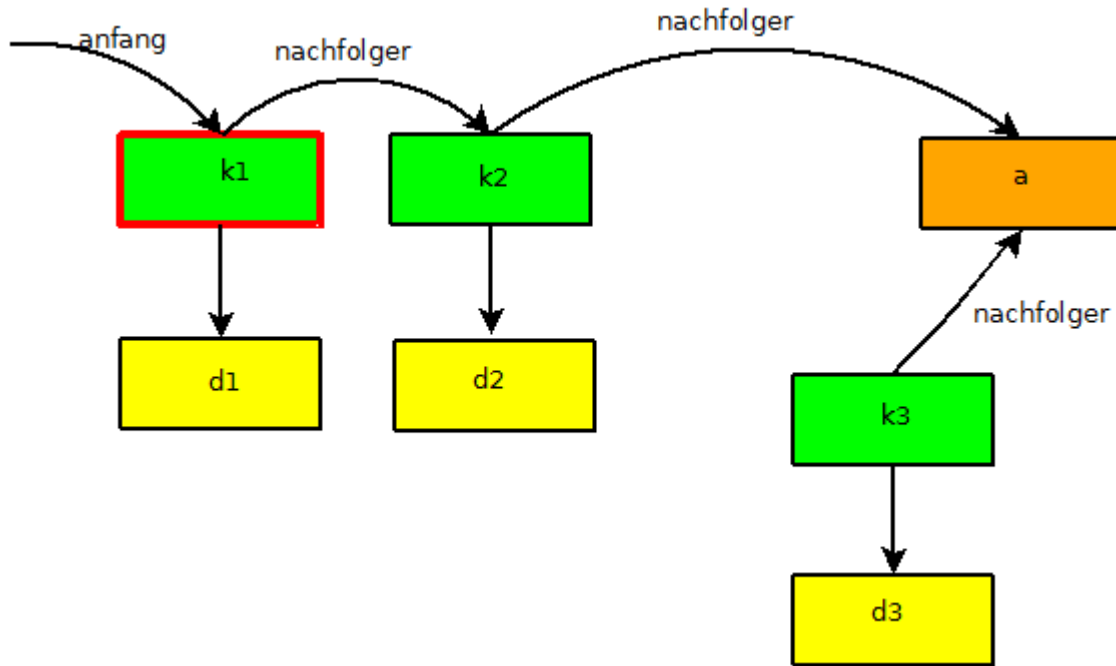
## Die Methode public Datenelement endeEntfernen ()



Darauf hat k2 gewartet,  
neuer Nachfolger von k2 ist a;  
return this (also k2)

```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

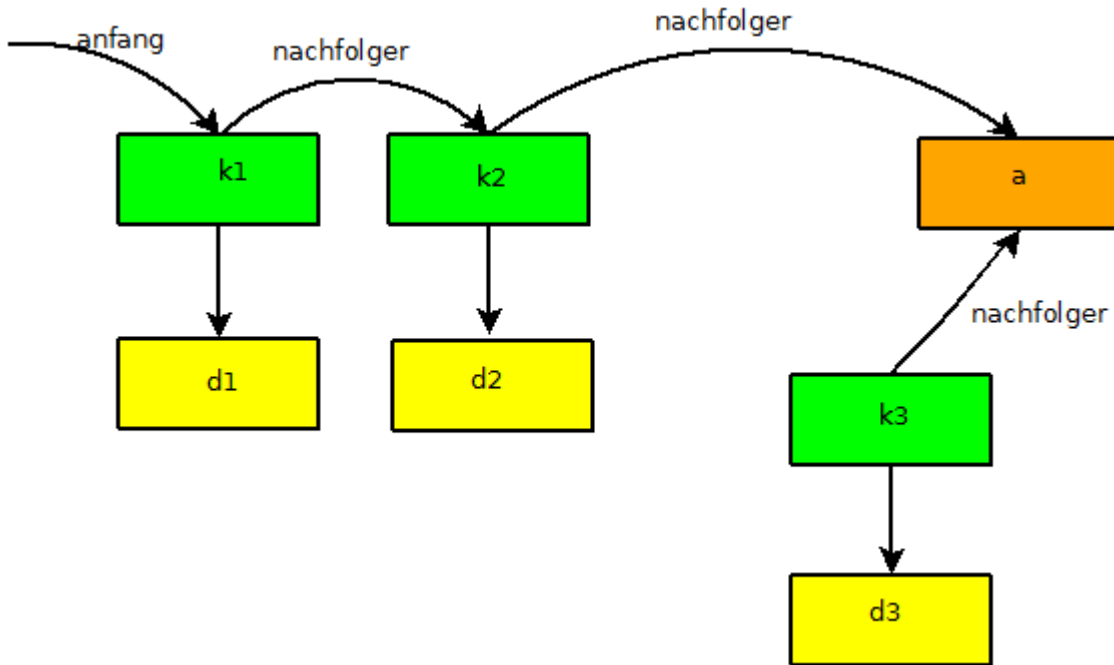
## Die Methode public Datenelement endeEntfernen ()



Darauf hat k1 gewartet,  
neuer und alter Nachfolger von k1 ist k2;  
return this (also k1)

```
//Klasse Knoten
public Listenelement endeEntfernen(Datenelement d){
    if(inhalt==d){
        return nachfolger;
    }
    else{
        nachfolger=nachfolger.endeEntfernen(d);
        return this;
    }
}
```

## Die Methode public Datenelement endeEntfernen ()



Auswertung des Aufrufs in Liste,  
neuer und alter Anfang der Liste ist k1;  
return d3

```
//Klasse Liste  
public Datenelement endeEntfernen(){  
    Datenelement d;  
    d = anfang.getEnde(null);  
    anfang = anfang.endeEntfernen(d);  
    return d;  
}
```