



# Rekursive Methoden: Längenbestimmung mit Methode `gibLaenge()`



# Methode gibLänge(): Implementierung

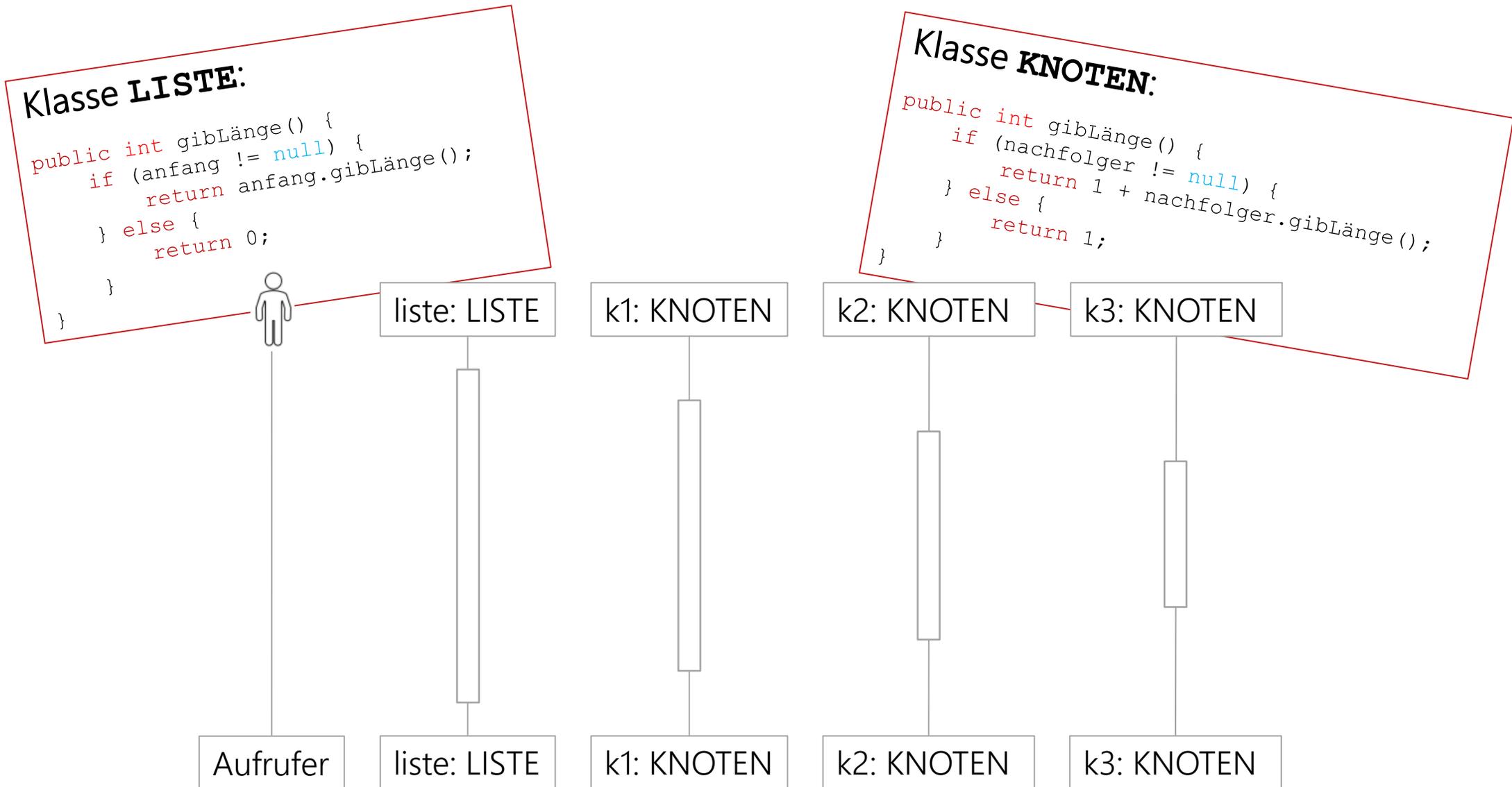
## Klasse LISTE:

```
public int gibLänge() {
    if (anfang != null) {
        return anfang.gibLänge();
    } else {
        return 0;
    }
}
```

## Klasse KNOTEN:

```
public int gibLänge() {
    if (nachfolger != null) {
        return 1 + nachfolger.gibLänge();
    } else {
        return 1;
    }
}
```

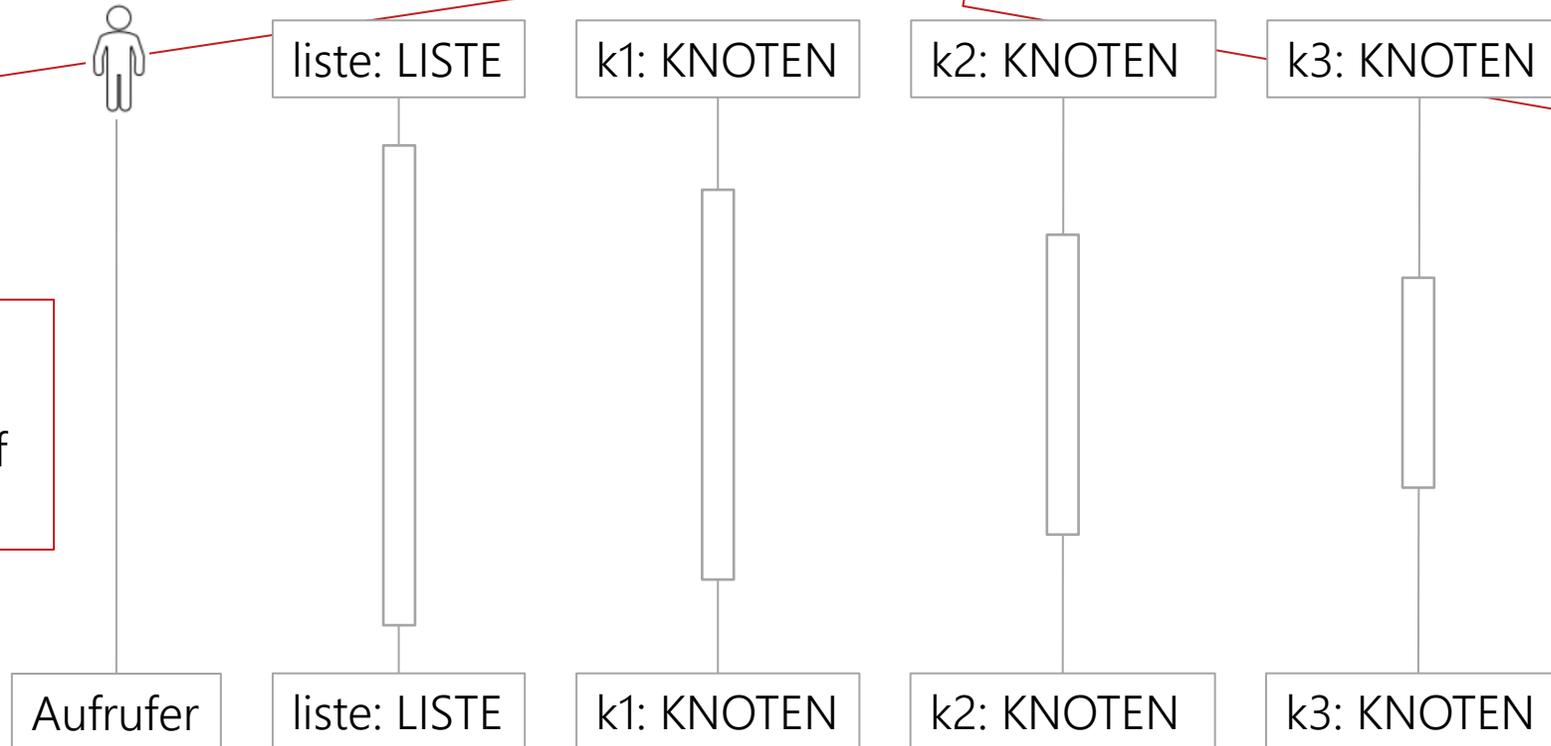
# Methode gibLänge () : Sequenzdiagramm



# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

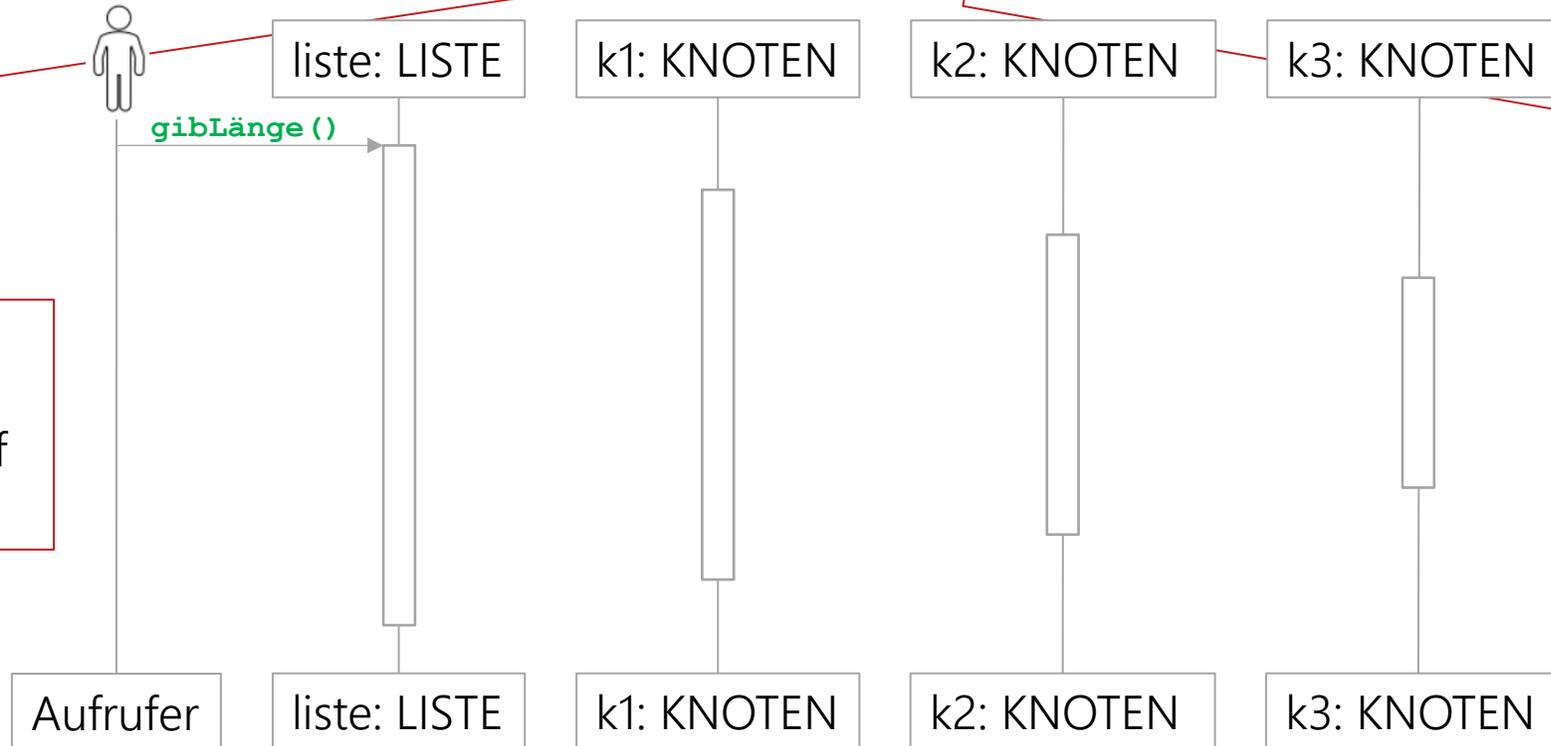


Der Aufrufer ruft die Methode **gibLänge ()** auf **liste** auf.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge () {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```



Der Aufrufer ruft die Methode **gibLänge ()** auf **liste** auf.

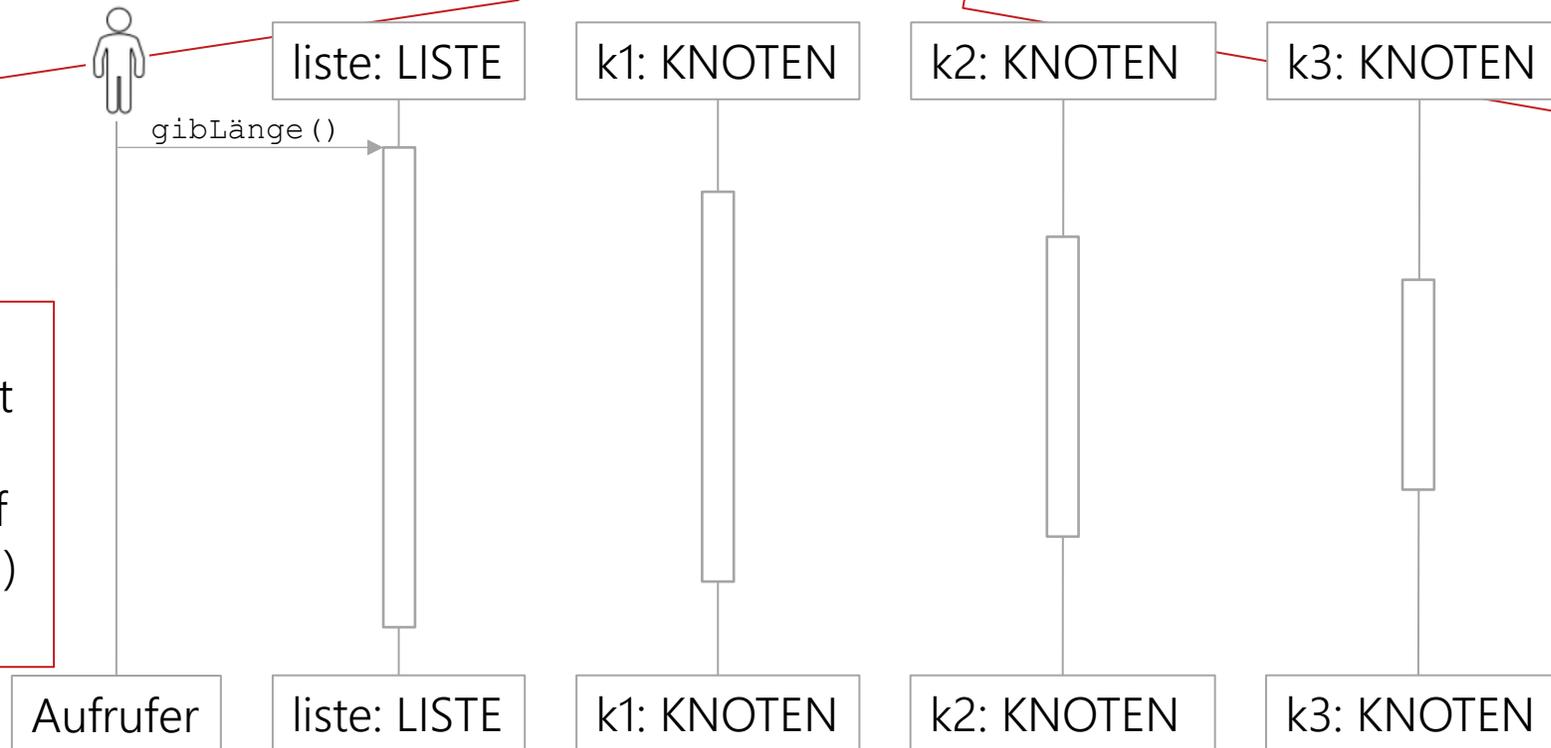
# Methode gibLänge () : Sequenzdiagramm

## Klasse LISTE:

```
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

## Klasse KNOTEN:

```
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```



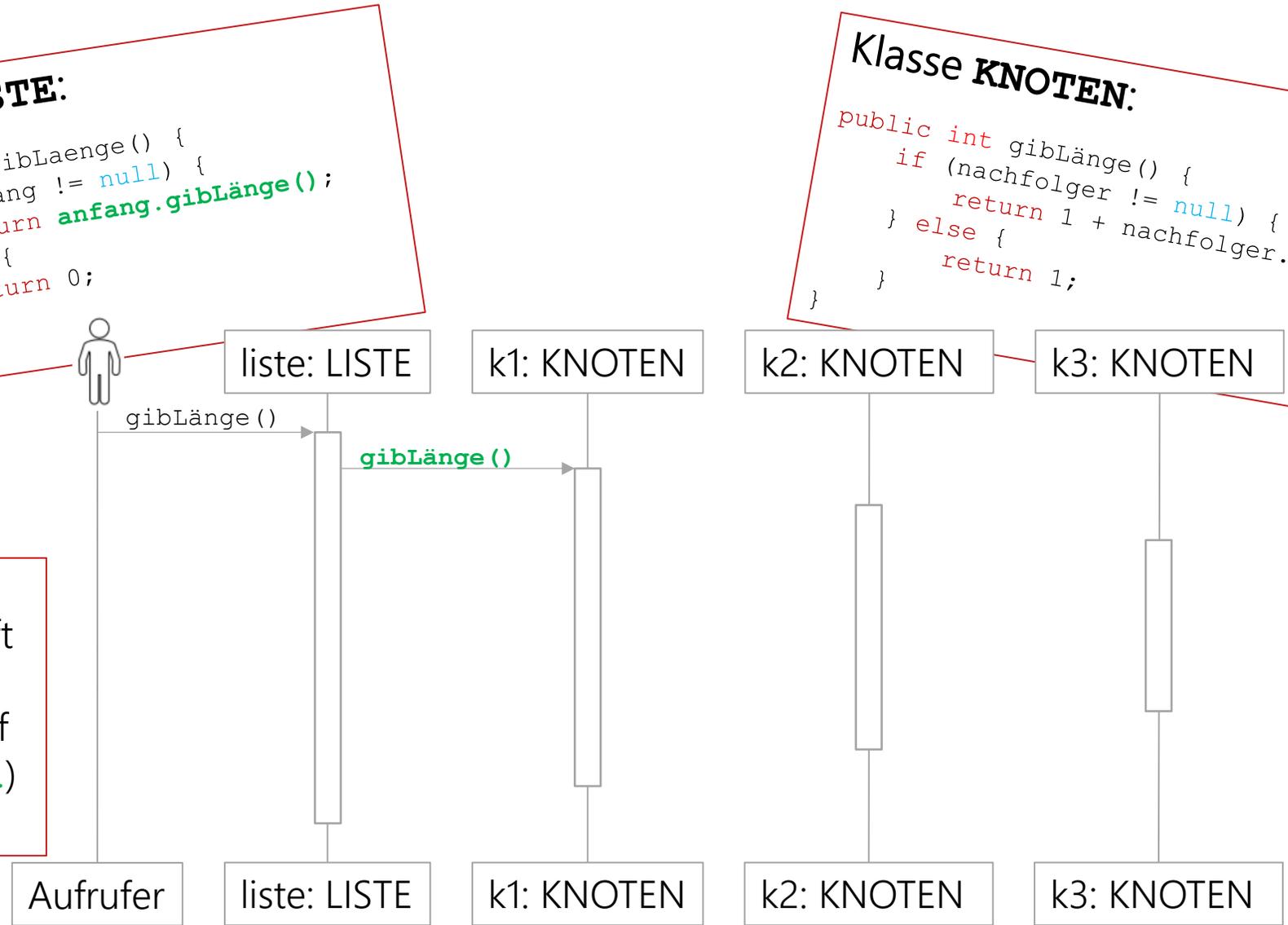
**liste** hat einen **anfang**. Also ruft sie die Methode **gibLänge()** auf **anfang** (hier: **k1**) auf.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLaenge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

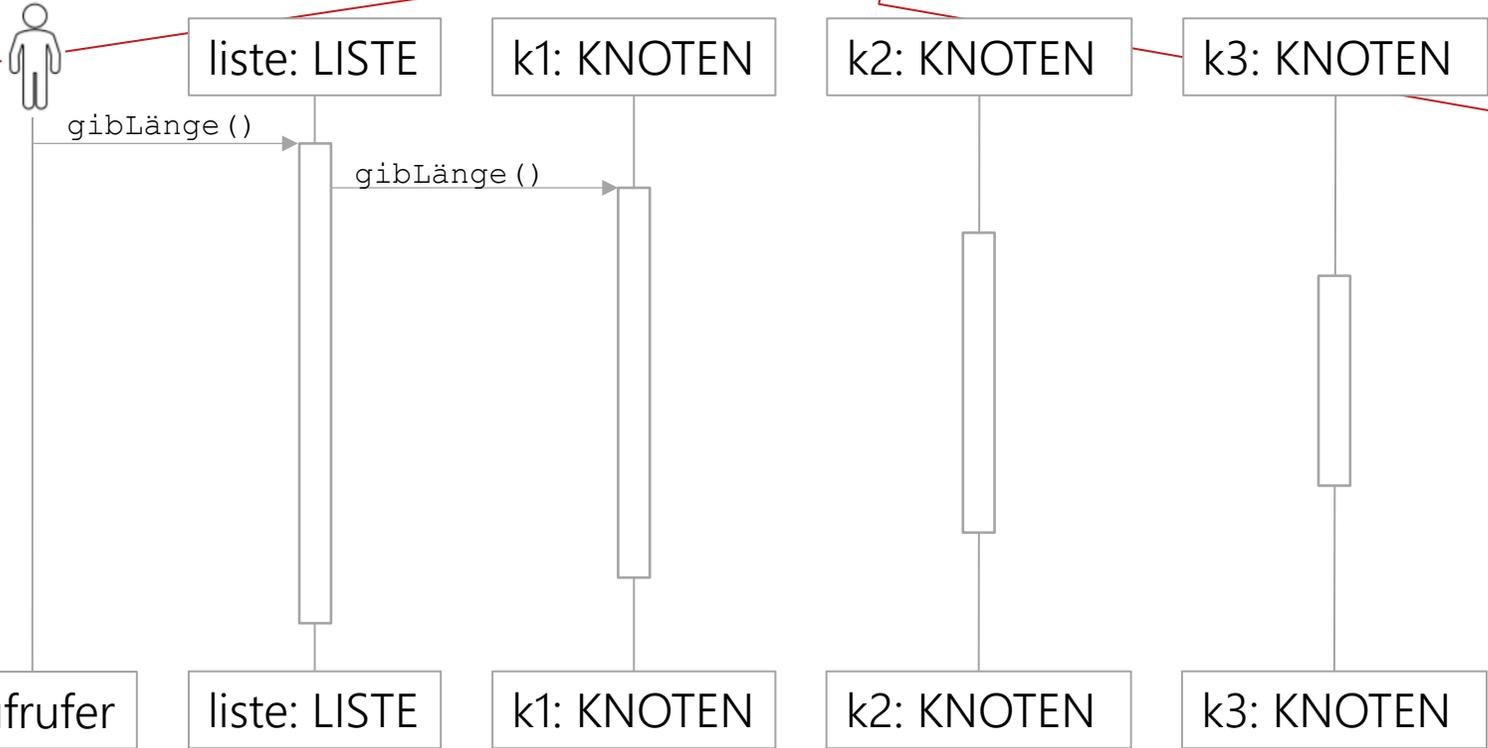
**liste** hat einen **anfang**. Also ruft sie die Methode **gibLänge ()** auf **anfang** (hier: **k1**) auf.



# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

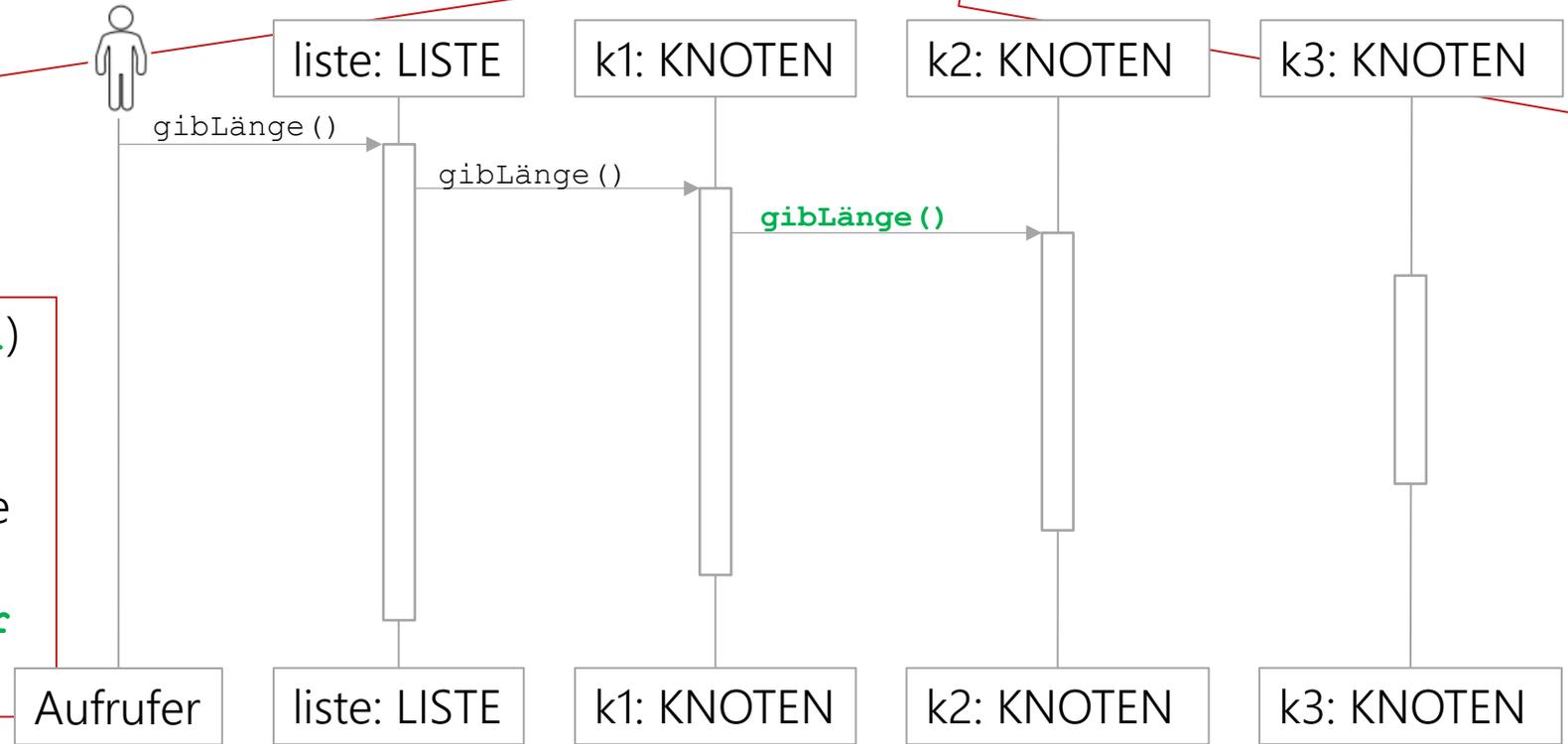


**anfang** (hier: **k1**) hat einen Nachfolger. Also wird die Methode **gibLänge()** auf **nachfolger** (**k2**) aufgerufen.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

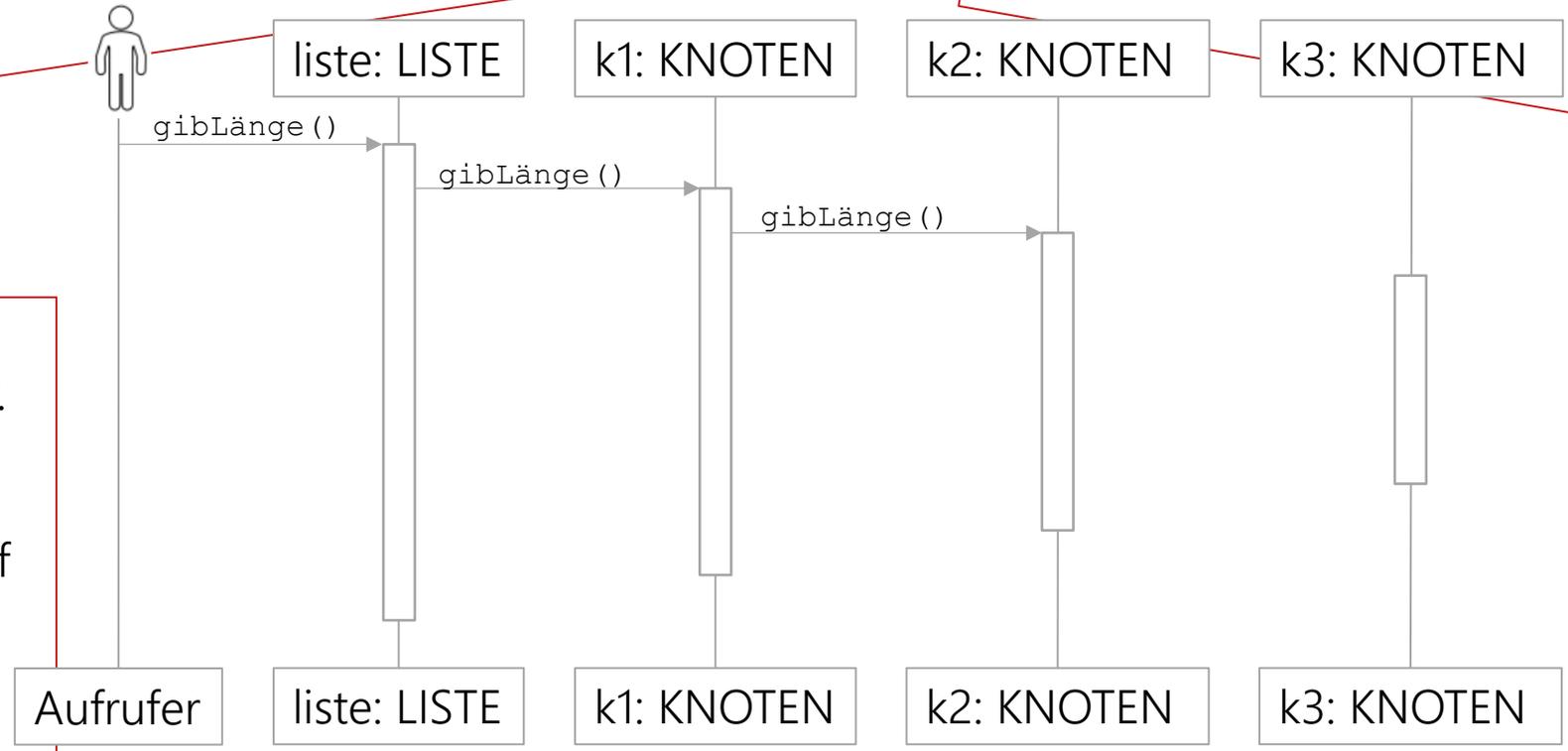


**anfang** (hier: **k1**) hat einen Nachfolger. Also wird die Methode **gibLänge ()** auf **nachfolger** (**k2**) aufgerufen.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

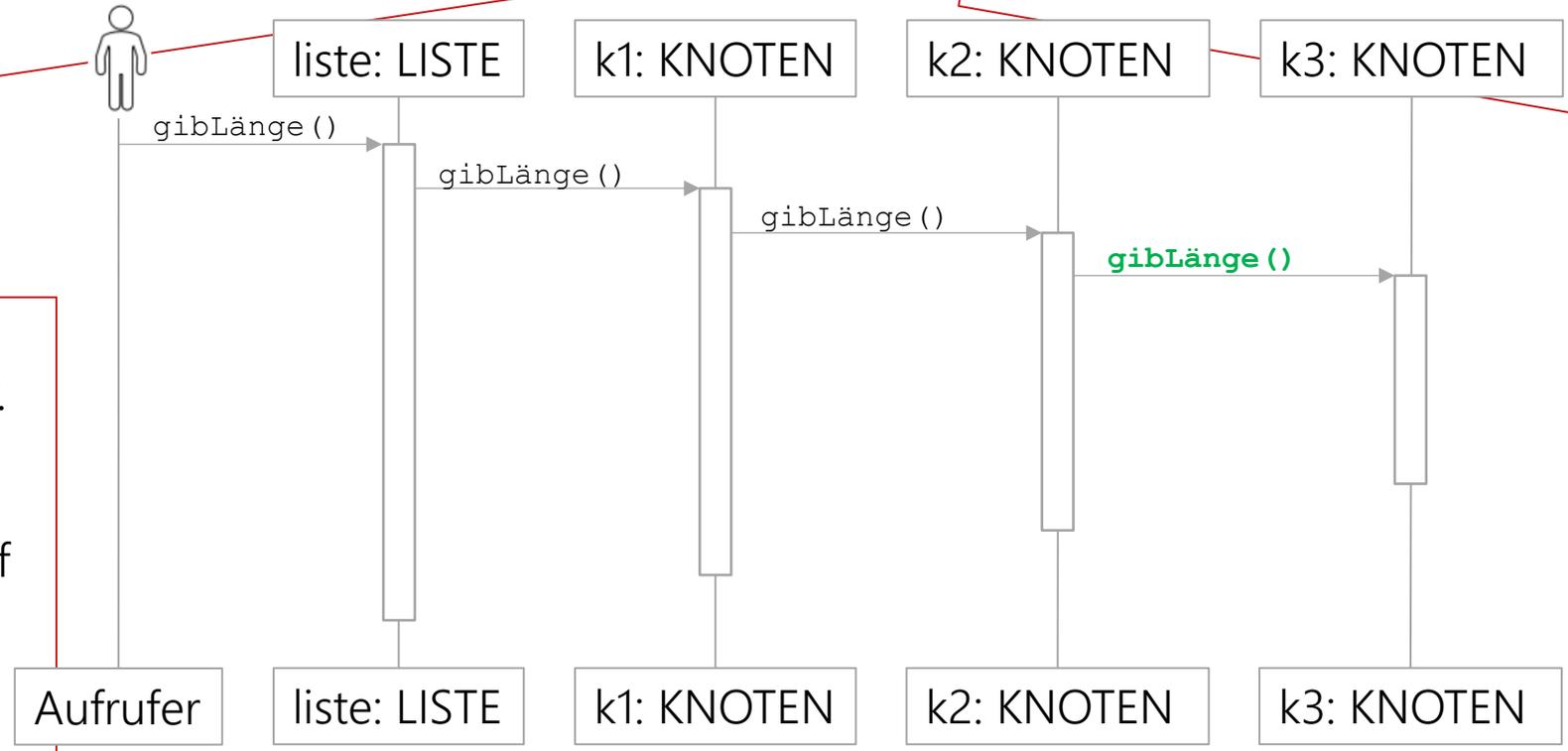


**k2** hat ebenfalls einen Nachfolger. Also wird die Methode **gibLänge()** auf seinem **nachfolger (k3)** aufgerufen.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

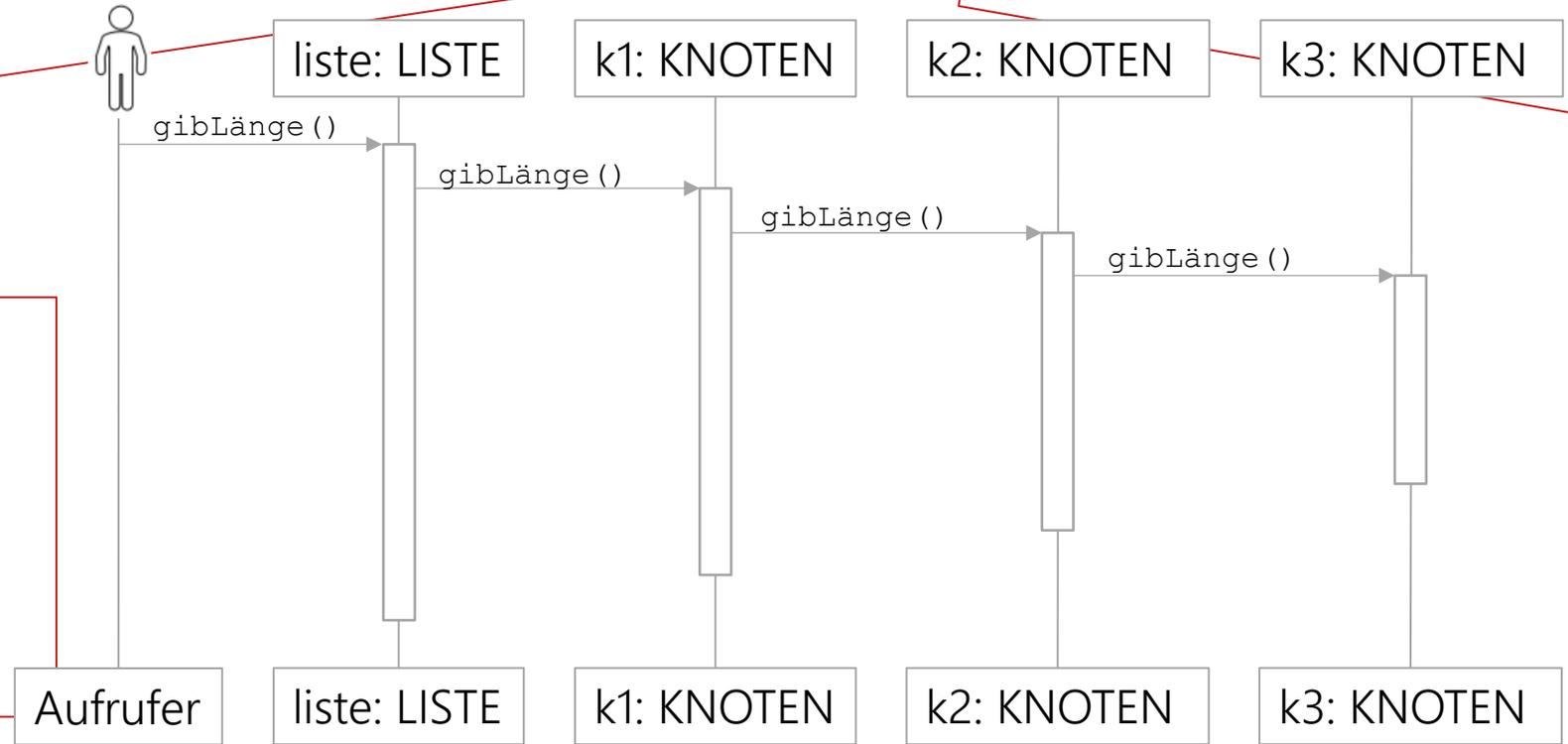


**k2** hat ebenfalls einen Nachfolger. Also wird die Methode **gibLänge ()** auf seinem **nachfolger (k3)** aufgerufen.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

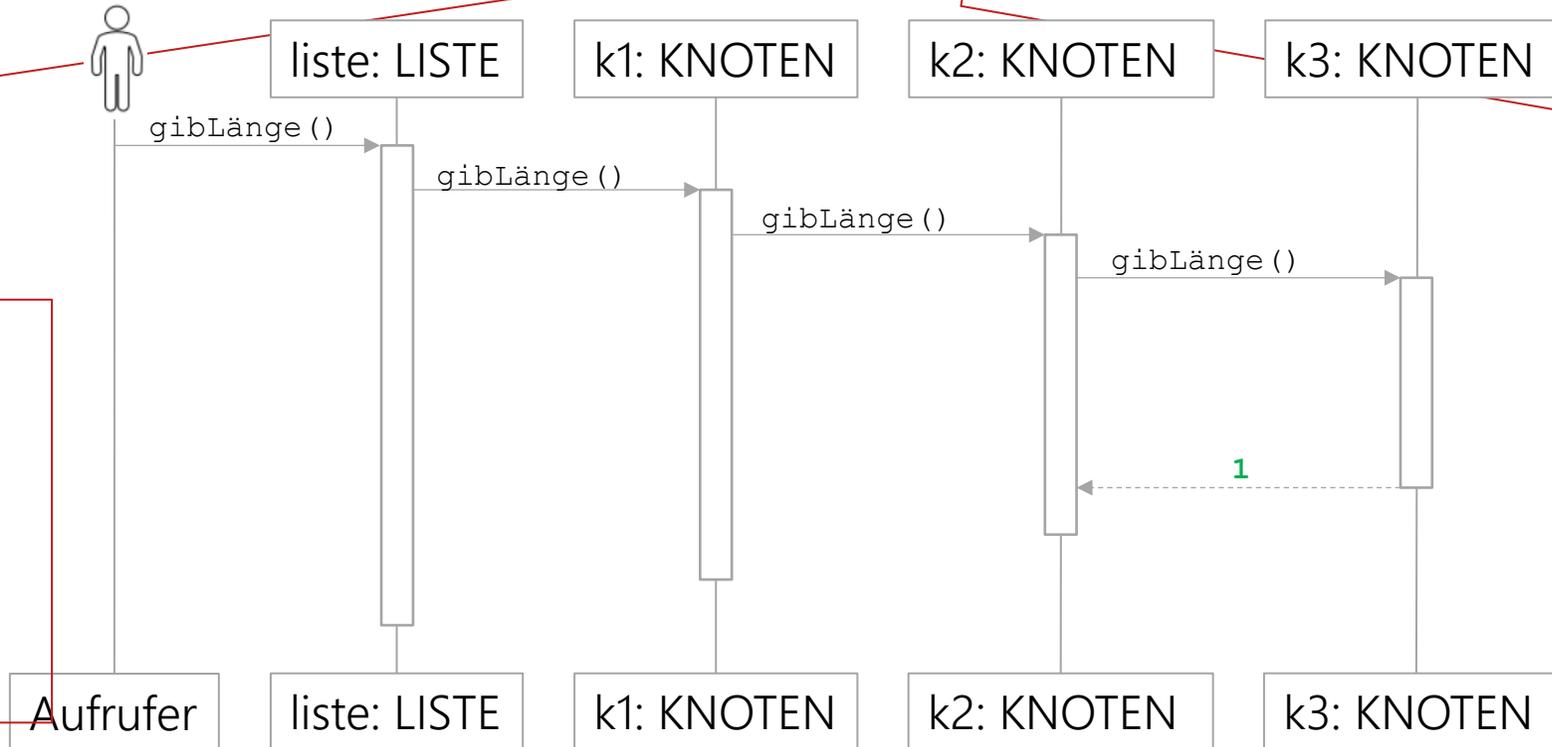


**k3** hat keinen Nachfolger. Die Methode **gibLänge ()** gibt **1** zurück. (Abstieg abgeschlossen.)

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

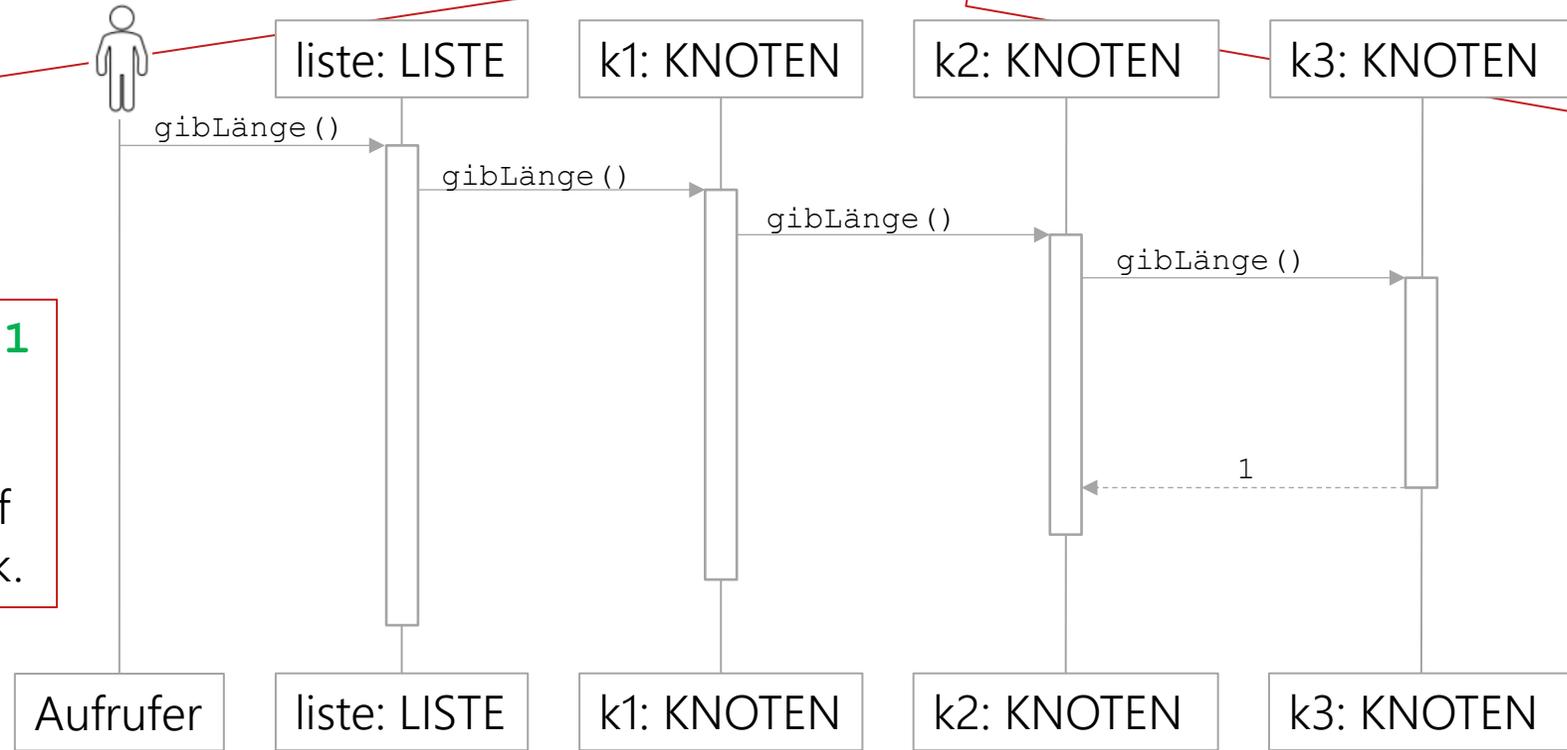


**k3** hat keinen Nachfolger. Die Methode **gibLänge ()** gibt **1** zurück. (Abstieg abgeschlossen.)

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

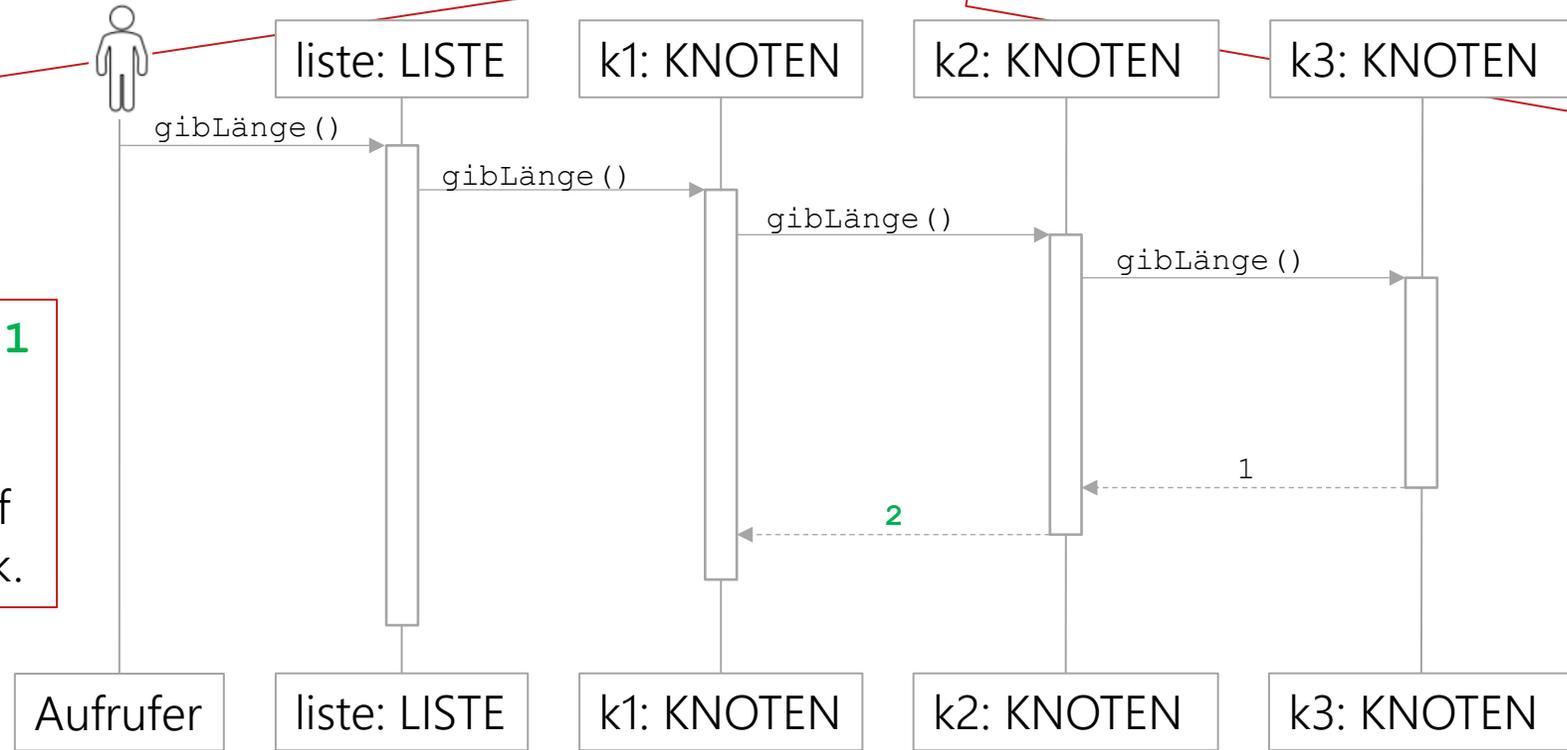


**k2** bekommt die **1** von seinem Nachfolger. Er addiert **1** drauf und gibt **2** zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

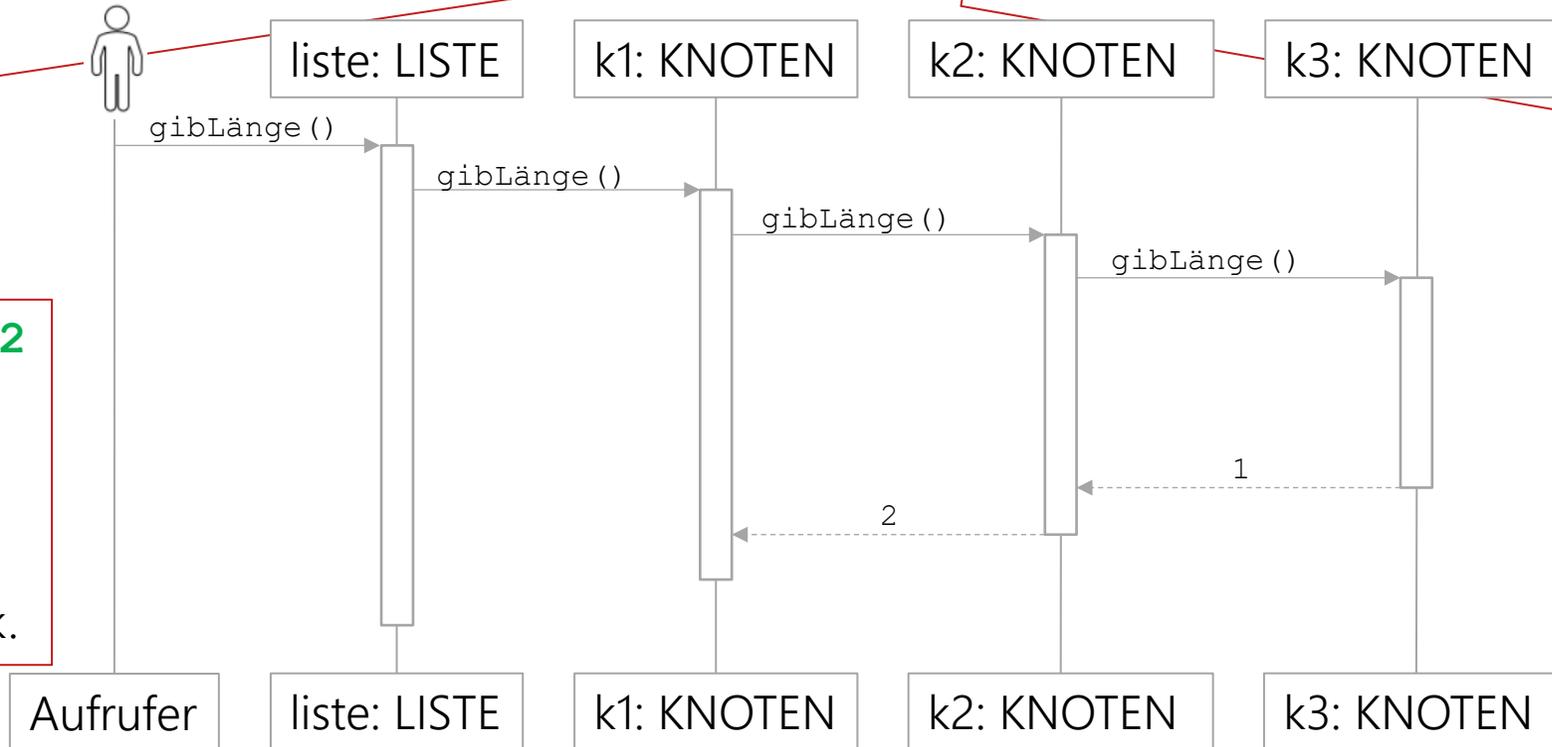


**k2** bekommt die **1** von seinem Nachfolger. Er addiert **1** drauf und gibt **2** zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

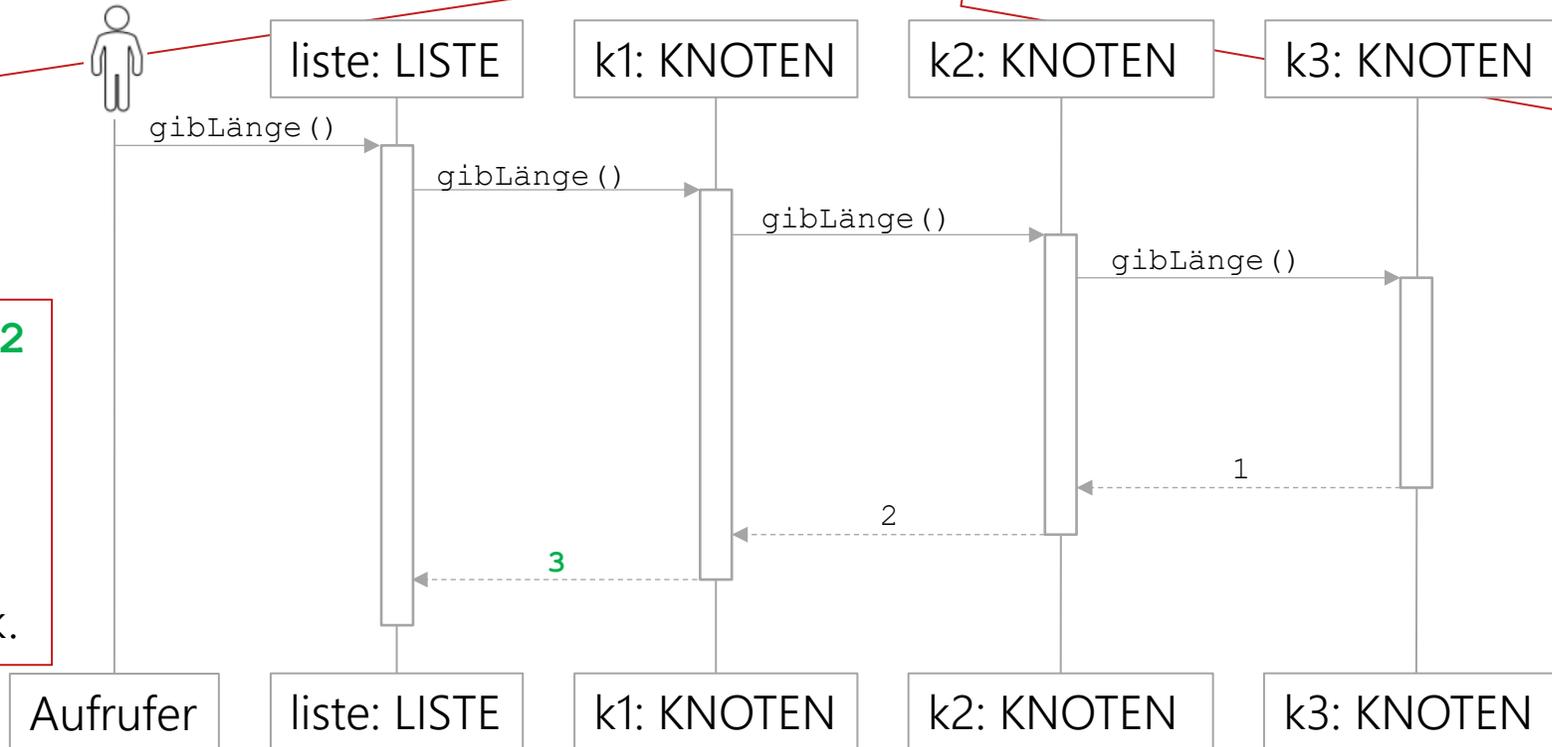


**k1** bekommt die **2** von seinem Nachfolger. Er addiert ebenfalls **1** drauf und gibt **3** zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

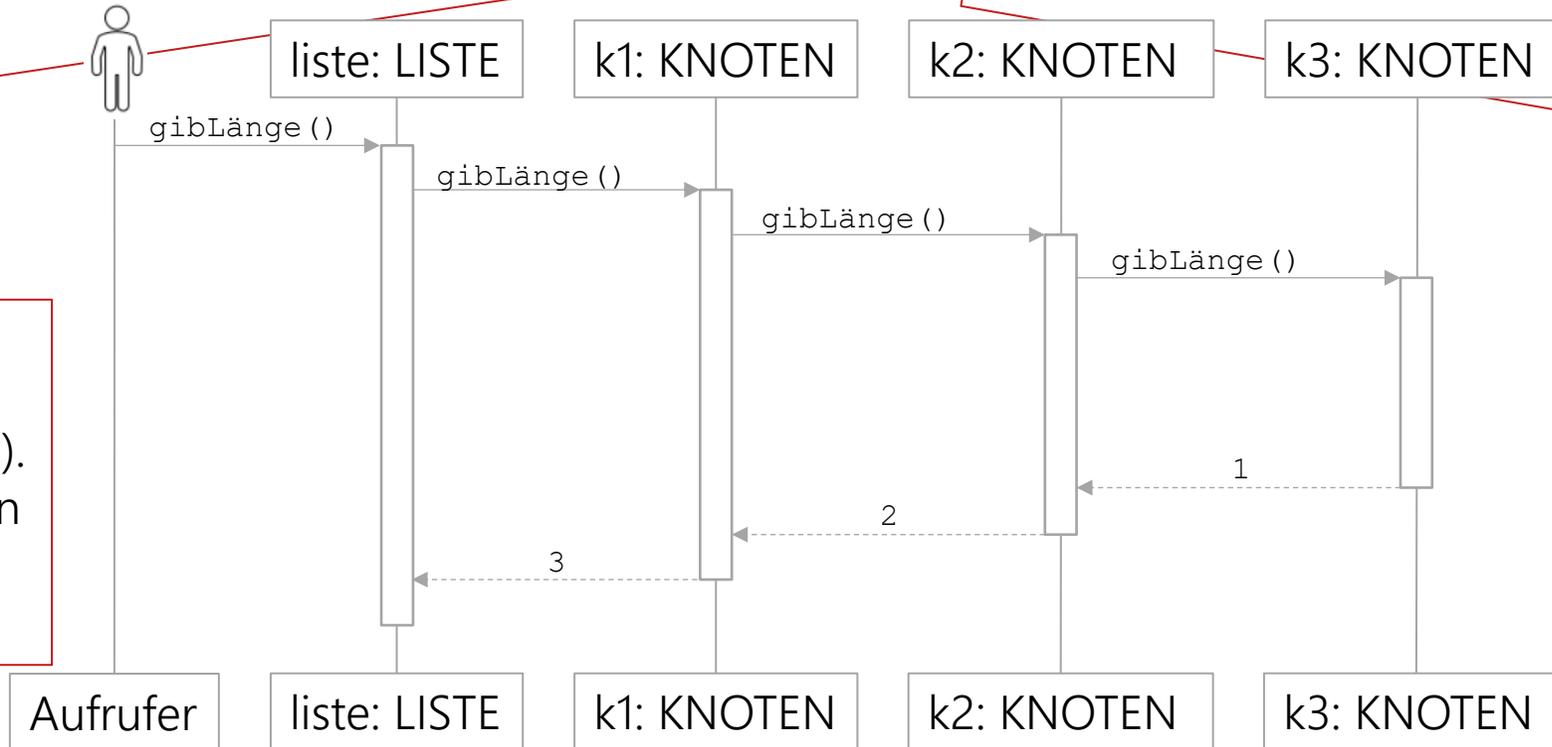


**k1** bekommt die **2** von seinem Nachfolger. Er addiert ebenfalls **1** drauf und gibt **3** zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

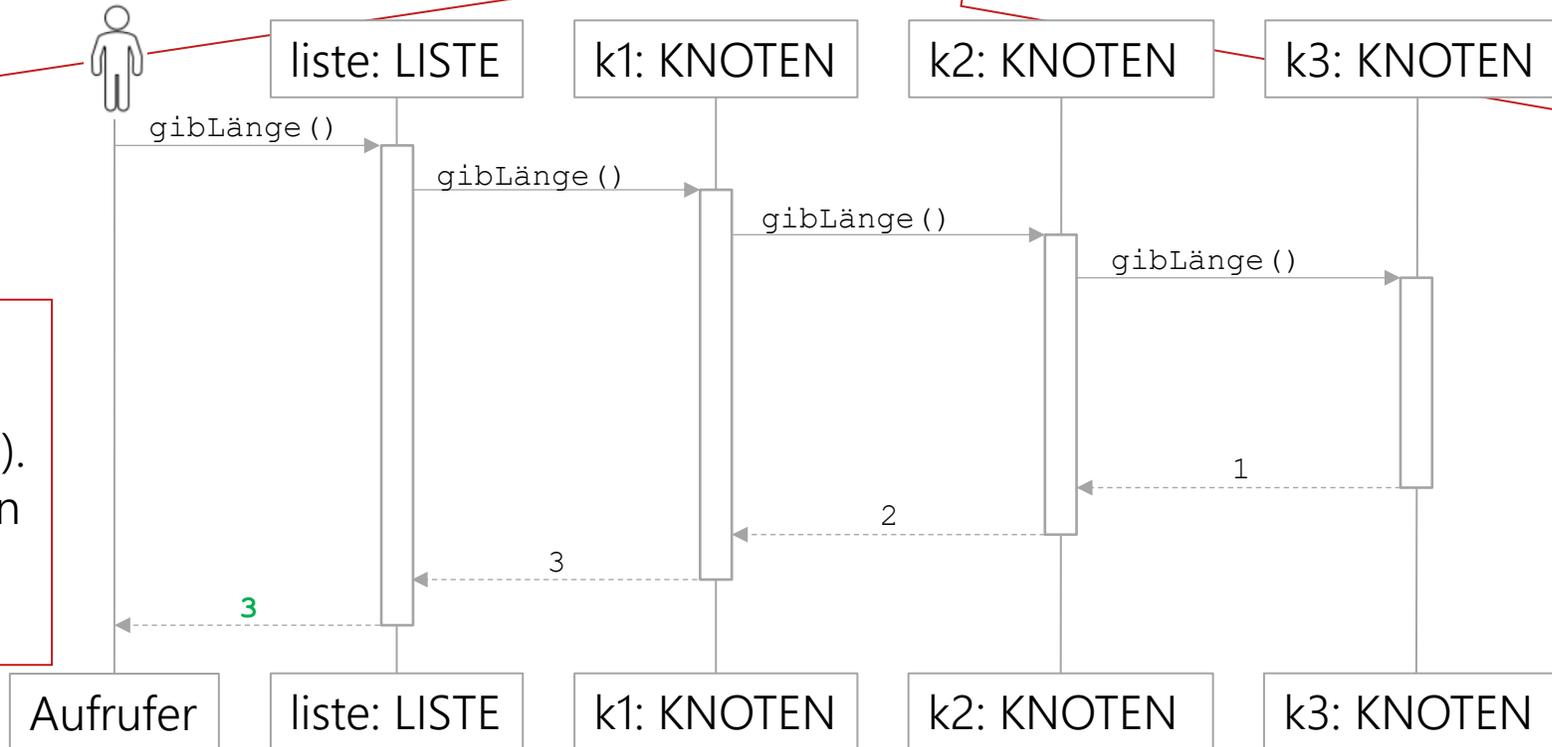


**liste** bekommt die **3** von ihrem **anfang** (hier: **k1**). Sie gibt diese **3** an den Aufrufer zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```

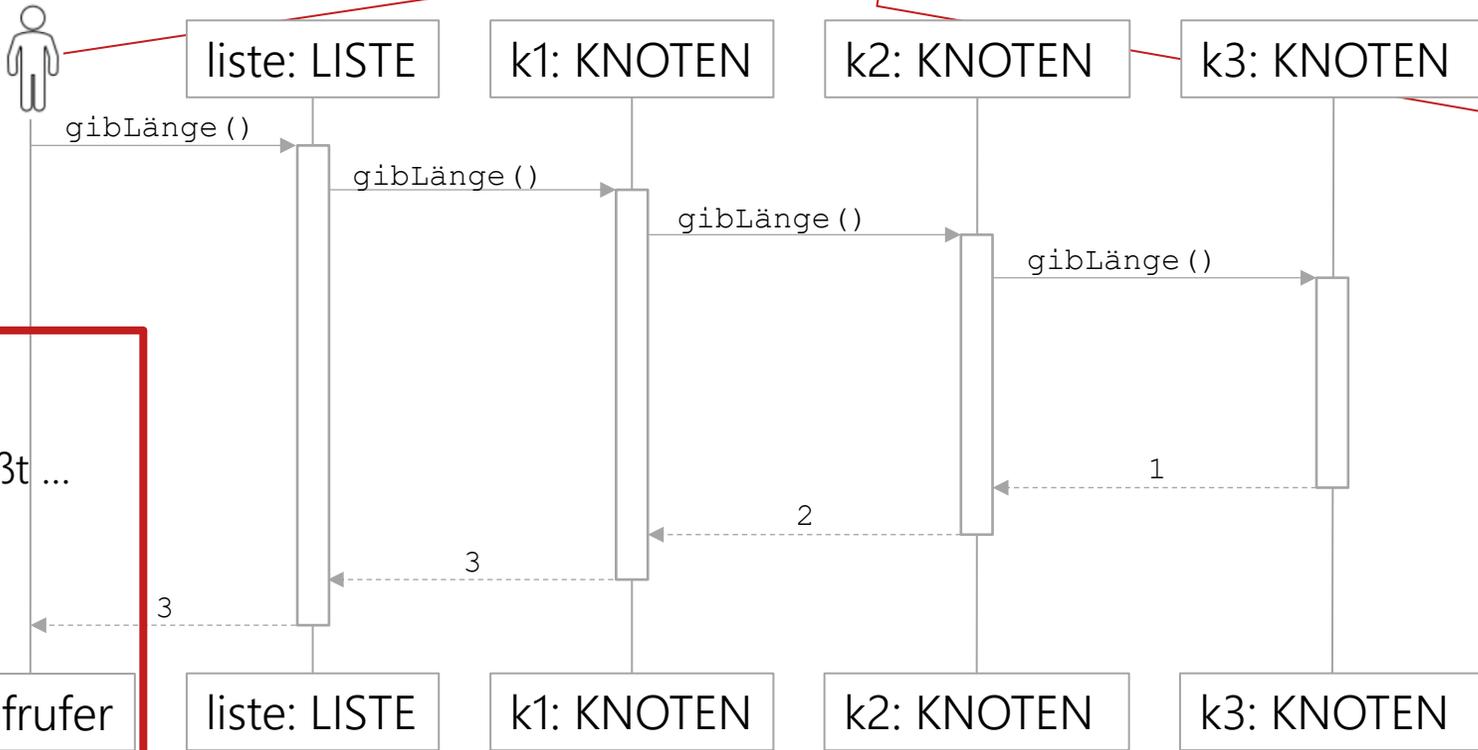


**liste** bekommt die **3** von ihrem **anfang** (hier: **k1**). Sie gibt diese **3** an den Aufrufer zurück.

# Methode gibLänge () : Sequenzdiagramm

```
Klasse LISTE:  
public int gibLänge() {  
    if (anfang != null) {  
        return anfang.gibLänge();  
    } else {  
        return 0;  
    }  
}
```

```
Klasse KNOTEN:  
public int gibLänge() {  
    if (nachfolger != null) {  
        return 1 + nachfolger.gibLänge();  
    } else {  
        return 1;  
    }  
}
```



Sehr einfach, wenn man verstanden hat, „wer was worauf“ aufruft – das heißt ...

- welches Objekt ruft ... (wer)
- welche Methode ... (was)
- auf welchem Objekt auf? (worauf)