

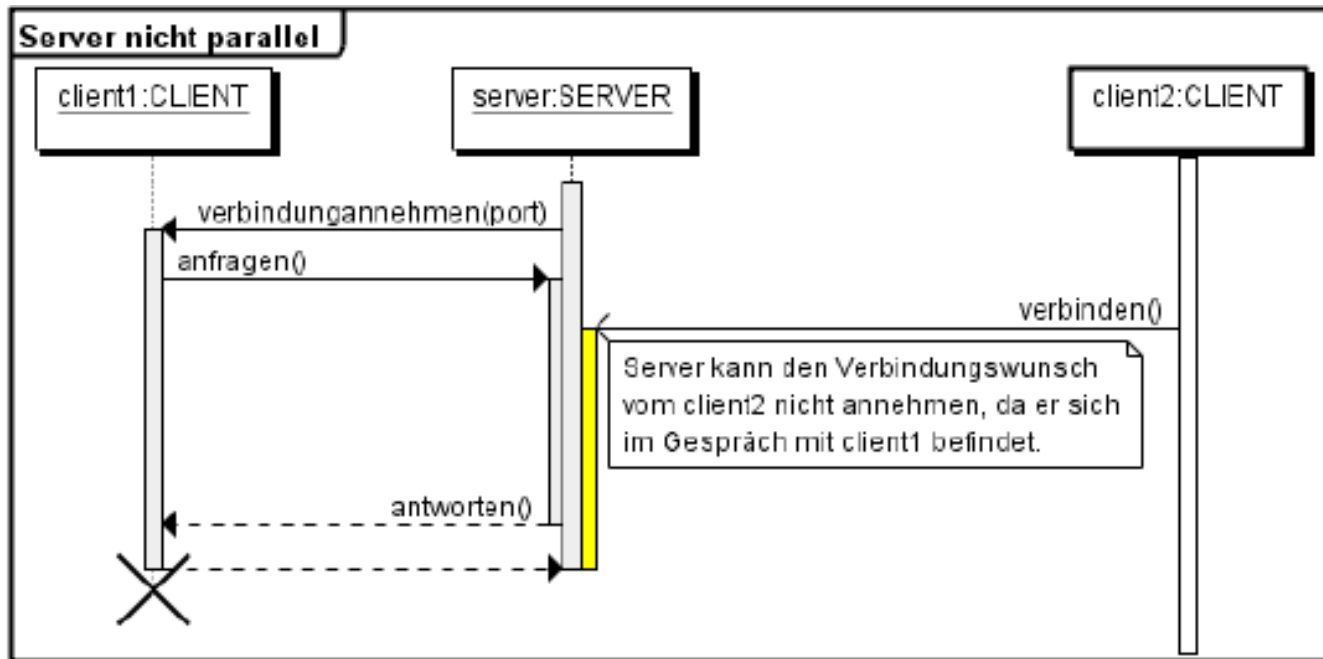
## 2. Kommunikation und Synchronisation von Prozessen

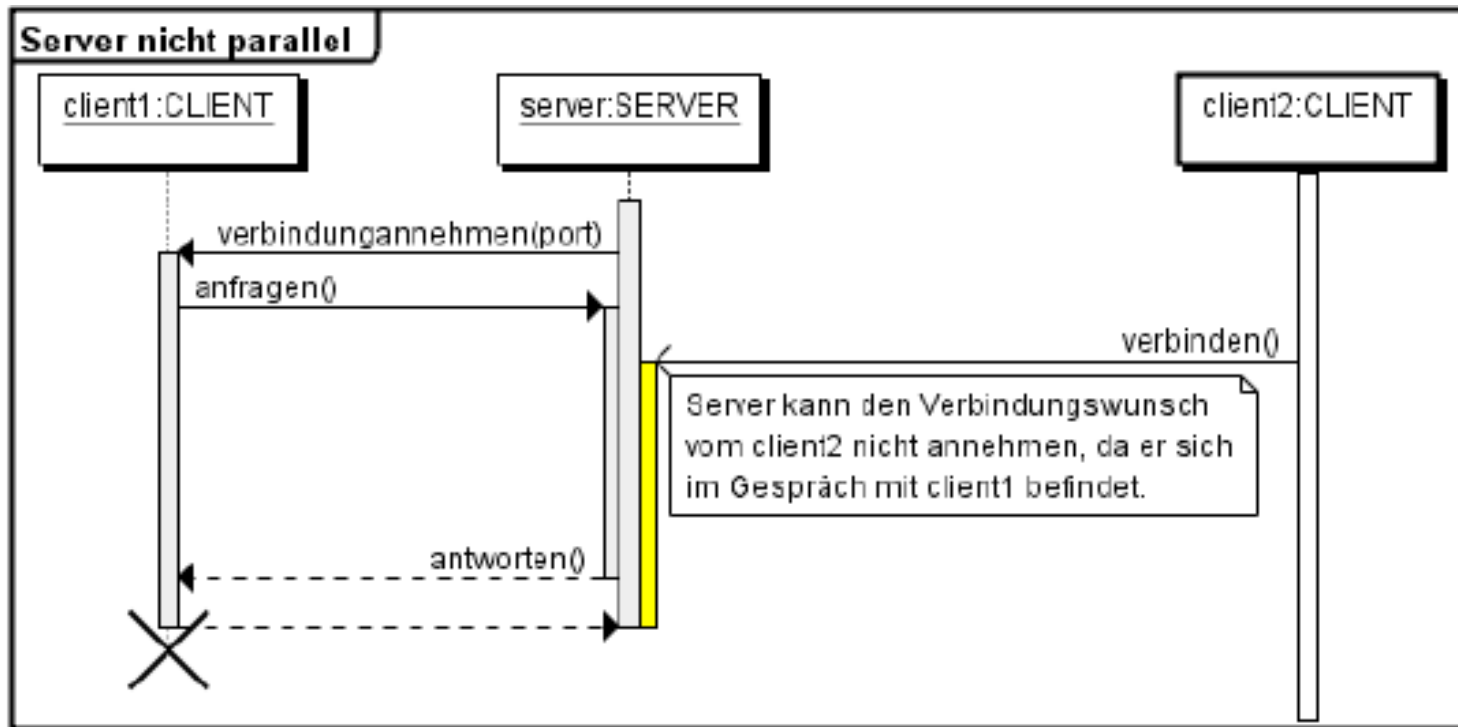
### 2.3 Modellierung von Prozessen

Beispiel:

Die Server-Client-Kommunikation (Bsp. "p03\_wetterauskunft") lässt nur einen Client zu. Teste dies im Programm.

Die Situation lässt sich anhand eines Sequenzdiagramms gut darstellen:





Die eigentliche Arbeit erledigt der Server innerhalb der Wiederholung im Konstruktor und ist währenddessen nicht für eine neue Verbindung bereit. Möchte man mehrere Clients zulassen, muss die zu wiederholende Sequenz ausgelagert werden.

Um das Problem zu lösen, müssen mehrere Prozesse gleichzeitig ablaufen. In diesem Fall spricht man von **parallelen Prozessen**.

Greifen die Prozesse dabei auf gemeinsame Ressourcen zu, können sie sich auch gegenseitig beeinflussen. In diesem Fall spricht man von **nebenläufigen Prozessen**.

Man unterscheidet dabei zwischen **schwergewichtigen Prozessen** (ganze Programme laufen parallel) und **leichtgewichtigen Prozessen** (Programmteile laufen parallel).

Leichtgewichtige Prozesse können in Java mit der Klasse **Thread** realisiert werden.

Dazu überschreibt man die Methode `run()` einer eigenen Klasse, die von `Thread` abgeleitet ist.

Im Beispiel ist dies die Klasse `CLIENTPROZESS`.

### Beispielprogramm für den Einsatz von Threads:

```
public class Minus extends Thread{
    public void run(){
        for (int i = 1; i<=10000; i++){
            System.out.print("-" + i);
        }
    }
}

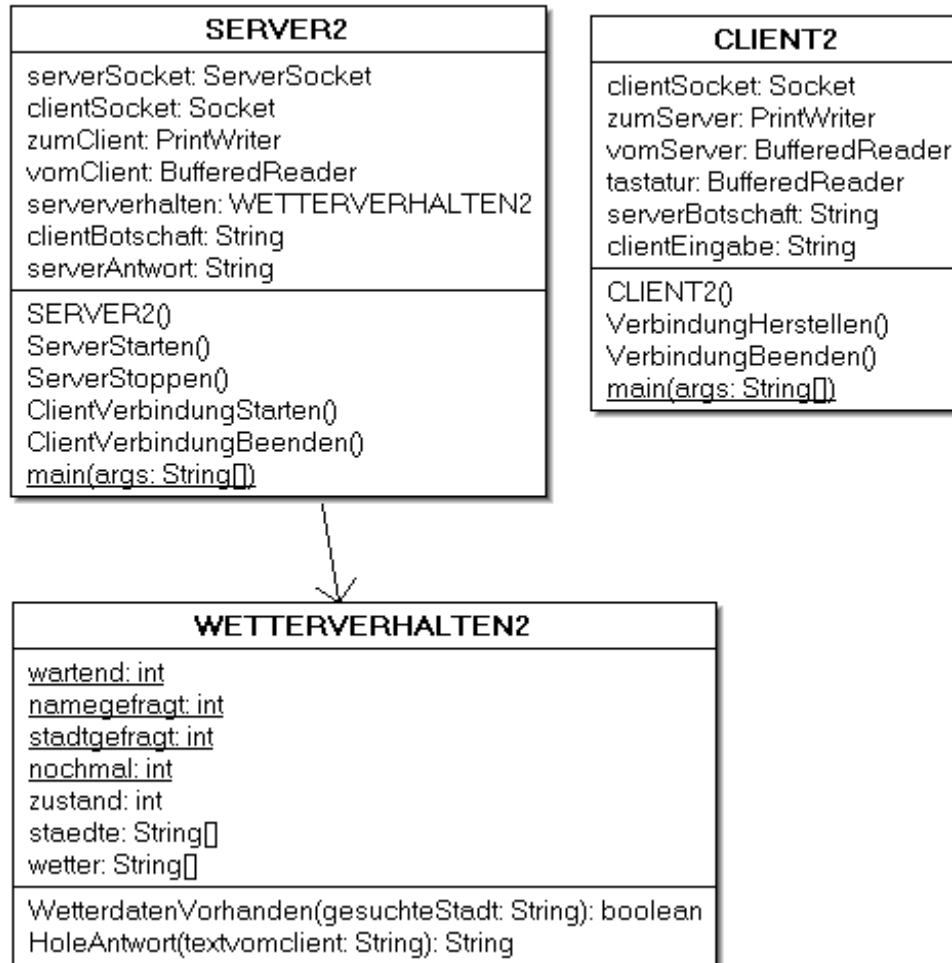
public class Plus extends Thread{
    public void run(){
        for (int i = 1; i<=10000; i++){
            System.out.print("+" + i);
        }
    }
}

public class Test{ public static void main(String[] args){
    Plus p = new Plus();
    p.start();
    Minus m = new Minus();
    m.start(); }
}
```

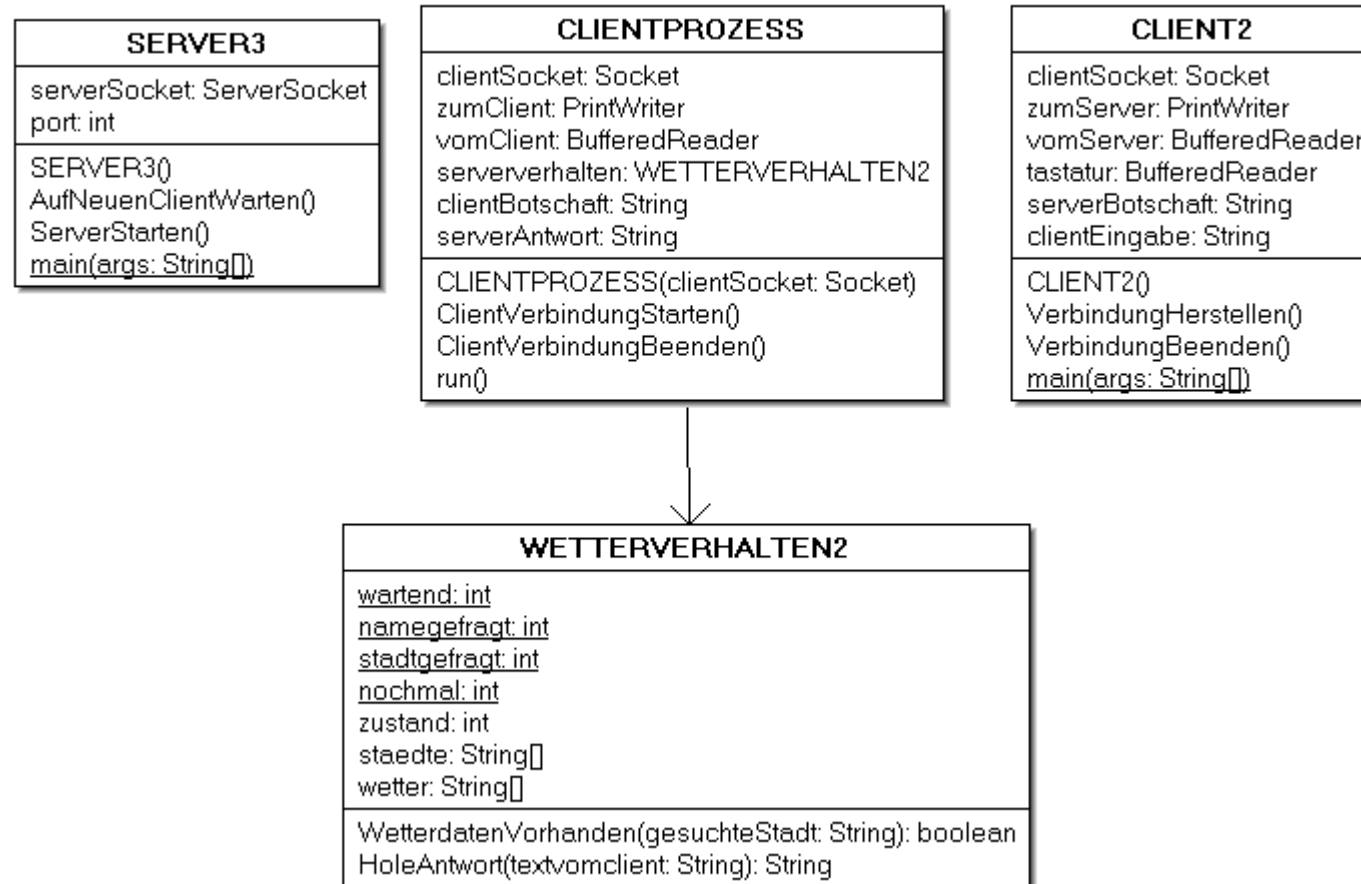
### Umbau der Client-Server-Modellierung

erster Ansatz	Servercode nach Umbau
Starten des Servers  Warten auf eine Clientverbindung und Herstellen der Verbindung  wiederhole Client-Botschaft lesen; Antwort ermitteln und senden; bis Server-Endesignal gesetzt  Clientverbindung beenden Server stoppen	Starten des Servers    wiederhole immer Warten auf eine Clientverbindung und Herstellen der Verbindung Auslagern der Verbindung in Thread endewiederhole

### Client-Server vor dem Umbau



### Client-Server nach dem Umbau



Umsetzung: BlueJ-Projekt “p05\_mehrclientsparallel”

## Übung

Buch, Seite 75/76

Aufgabe 1 (Roboter tanzen)

Aufgabe 2 (Roboter bewachen eine Grenze)