



Informatik in Jgst. 10

Klasse Taktgeber

Automatische Abläufe

Rückblick:

Animation eines Kreises:

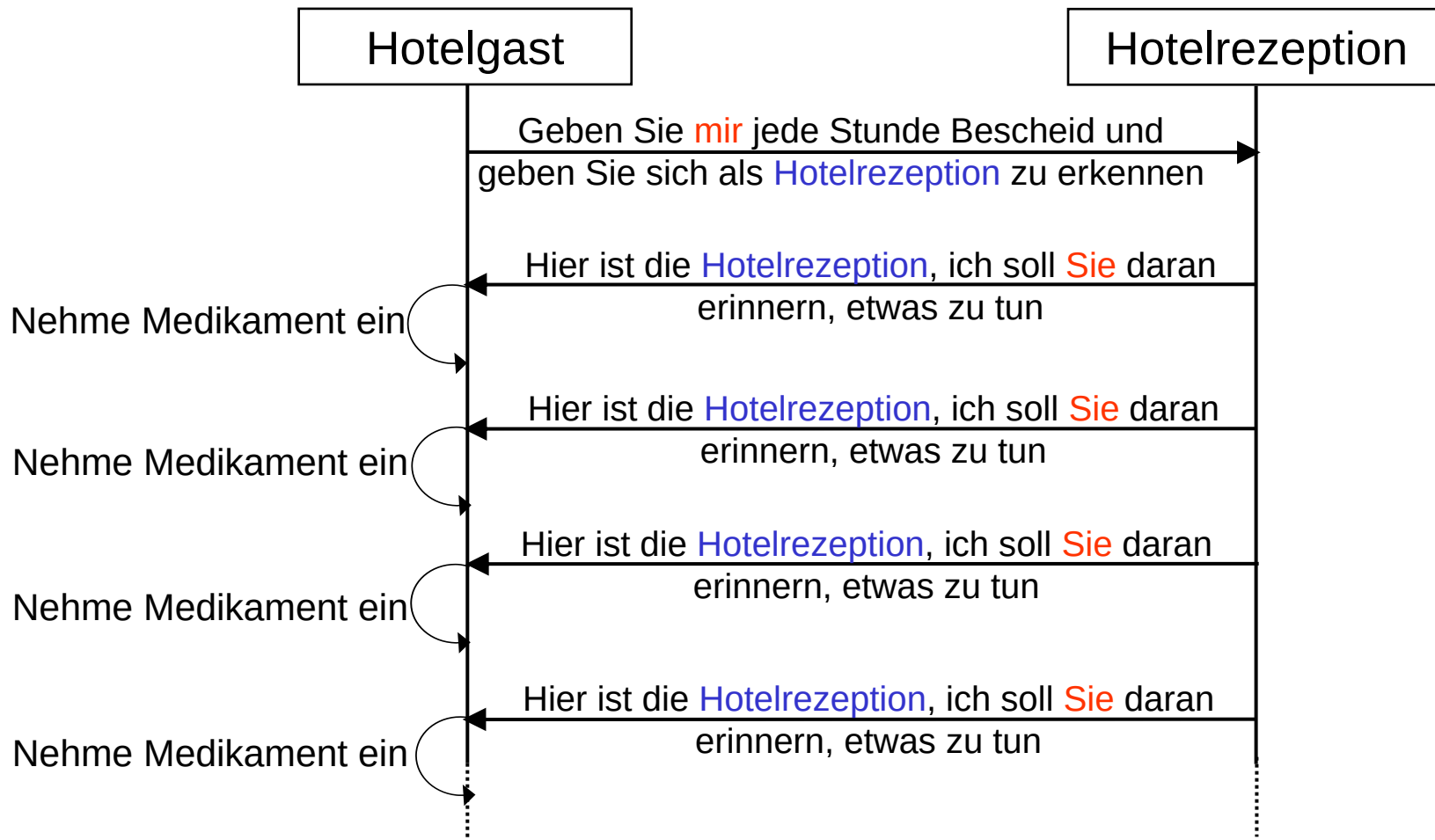
```
...  
public void animiereKreis()  
{  
    for(int t=0; t<1000; t++)  
    {  
        x = x+dx;  
        y = y+dy;  
        kreis.setzePosition(x,y);  
        //warte 100ms  
        StaticTools.warte(100);  
    }  
}  
...
```

Probleme der for-Schleife?

1. **Blockiert das System solange bis for-Schleife abgearbeitet ist**
2. **Zeitliche Steuerung schwierig**

Automatische Abläufe

Kommunikation am realen Beispiel:



Automatische Abläufe

Allgemeines(1):

- Klasse Taktgeber in JGUIToolbox beinhaltet Java-Klasse Timer

Taktgeber
[-] t: Timer [#] delay: int [#] startDelay: int [#] timerSignal: boolean [#] anzahl: int [#] begrenzteAnzahl: boolean [-] linkObj: ITuWas [-] id: int
© Taktgeber() © Taktgeber(ITuWas, int) [-] tuWas(): void + setzeLink(ITuWas, int): void + wurdeSignalisiert(): boolean + warteBisTaktsignal(): void + setzeZeitZwischenAktionen(int): void + setzeAnfangszeitverzögerung(int): void + mehrfach(int): void + einmal(): void + einmal(int): void + endlos(): void + stop(): void + laufend(): boolean

Automatische Abläufe

Methoden des Taktgebers:

UML-Klassendiagramm



Methoden

ITuWas: Objekt, dass mit Taktgeber verknüpft werden soll

int: Zahl, die den Taktgeber identifiziert (wichtig bei mehreren Taktgebern)

int: Zeit in Millisekunden, die zwischen den Signalen gewartet wird

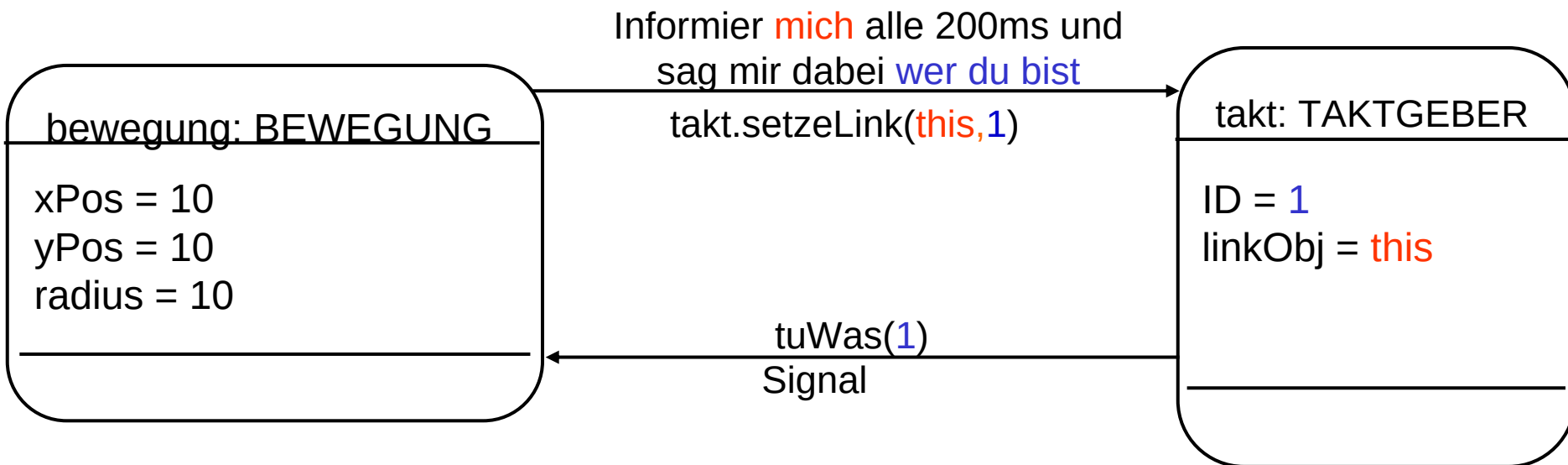
int: Anzahl wie oft signalisiert werden soll

int: Verzögerung in Millisekunden bis einmal signalisiert wird

Automatische Abläufe

Allgemeines(2):

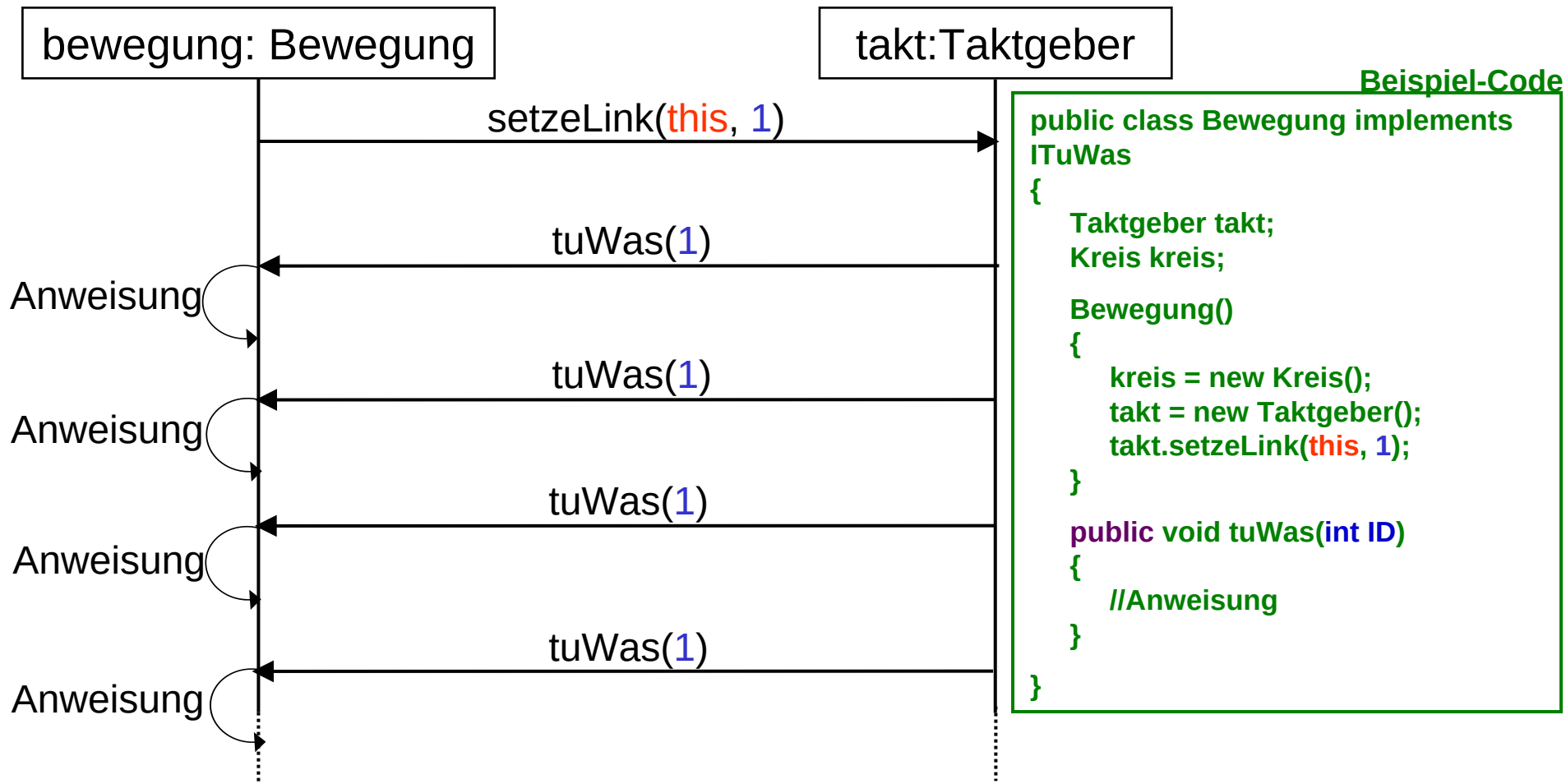
- Ein Taktgeber sendet Signal an verknüpftes Objekt für den Aufruf einer Methode (tuWas(int ID)) des Objekts



- Diese Methode unterbricht dabei nur kurz das System

Automatische Abläufe

Kommunikation anhand des Beispiel-Codes:



Automatische Abläufe

Beispiel-Code

```
public class Bewegung implements ITuWas
{
    Taktgeber takt;
    Kreis kreis;

    Bewegung()
    {
        kreis = new Kreis();
        takt = new Taktgeber();
        takt.setzeLink(this, 1);
    }

    public void tuWas(int ID)
    {
        //Anweisung
    }
}
```

Sicherstellung das die Klasse Bewegung die Methode `public void tuWas(int ID)` besitzt

Instanzieren eines Objekts der Klasse Taktgeber

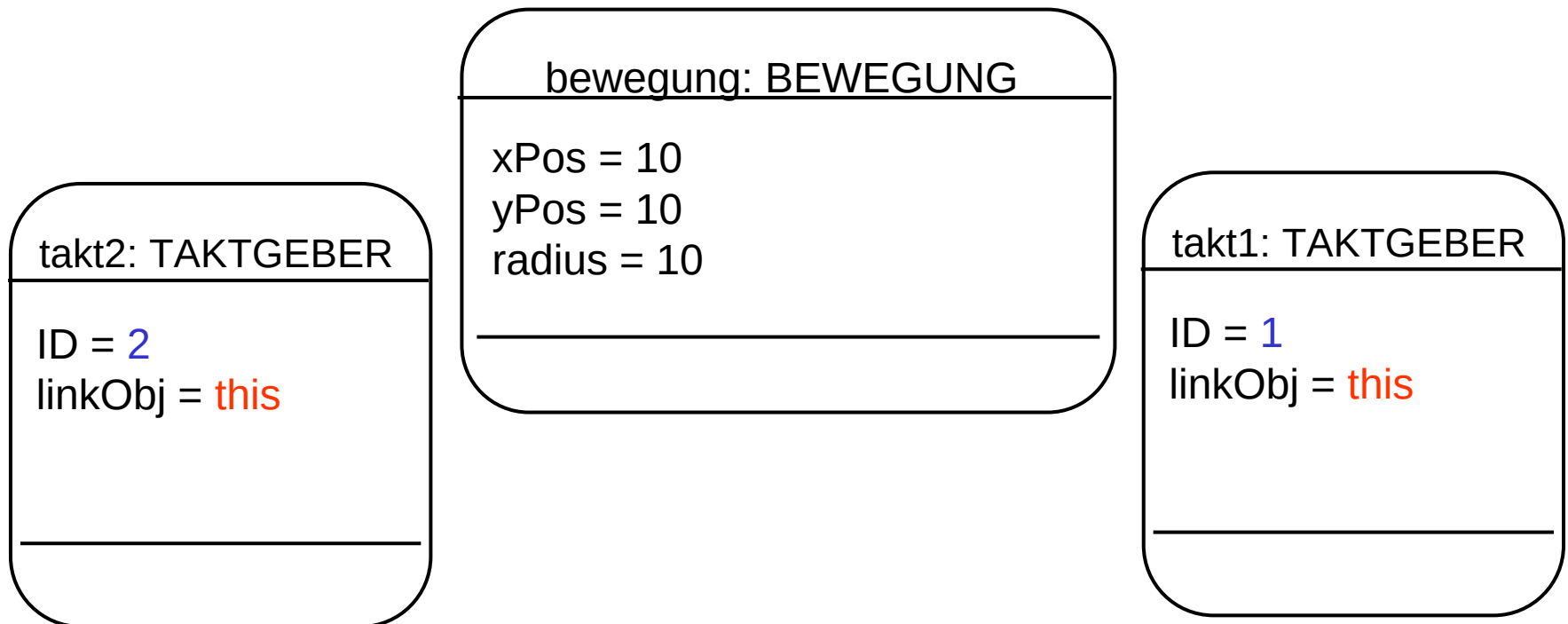
Taktgeber-Objekt mit dieser Klasse verknüpfen und ID vergeben

Methode `public void tuWas(int ID)`, führt Anweisung in bestimmten Zeitabständen aus

Automatische Abläufe

Mehrere Taktgeber

- Ein Objekt einer Klasse kann mehrere Taktgeber besitzen und kann jeden Taktgeber identifizieren (mehrere zeitlich gesteuerte Prozesse möglich)



Automatische Abläufe

Mehrere Taktgeber - Beispiel-Code

```
public class Bewegung implements ITuWas {  
    ....  
    public void tuWas(int ID){  
        if (ID == 1) {          //Nachricht von Taktgeber 1: Bewegung  
            this.bewege();  
        }else if (ID == 2) { //Nachricht von Taktgeber 2: Farbe ändern  
            this.aendereFarbe();  
        }  
        ...  
    public void bewege() { ... }  
    public void aendereFarbe() { ... }  
}
```

Automatische Abläufe

Zusammenfassung:

Vorteile

1. tuWas(int ID) unterbricht nur kurz das System
2. Zeitliche Steuerung effektiver
3. Mehrere zeitliche Abläufe möglich

Kommunikation

1. Objekt sagt Taktgeber, gib mir Signal und sag mir wer du bist (takt.setzeLink(**this**, 1))
2. Taktgeber sendet Signal an verknüpftes Objekt
3. Aufruf von tuWas(int ID) des Objekts

Programm-Code

1. public class Bewegung implements ITuWas {...}
2. takt = new Taktgeber();
3. takt.setzeLink(**this**, 1);
4. **public** void tuWas(int ID){...}