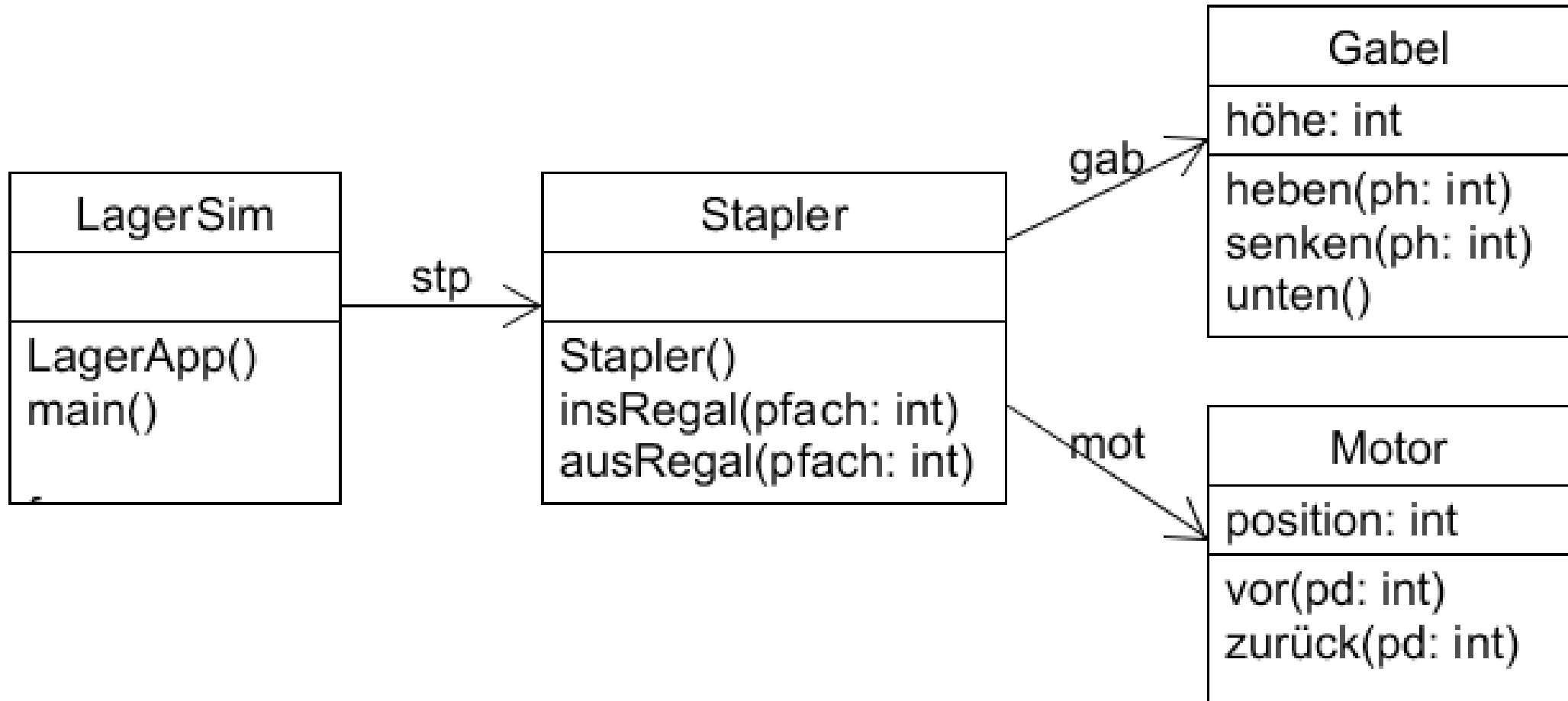


OOP mit Java

Sequenzdiagramm

Kommunikation zw. Objekten

Beispiel: Lager mit Gabelstapler



Beispiel: Lager mit Gabelstapler

```
class LagerSim
{
    Stapler stp;
    ...

    void main()
    {
        stp.insRegal(2);
        stp.ausRegal(3);
    }
}
```

Beispiel: Lager mit Gabelstapler

```
class LagerSim
{
    Stapler stp;
    ...

    void main()
    {
        stp.insRegal(2);
        stp.ausRegal(3);
    }
}
```

Hauptobjekt
„sendet“
dem Objekt stp
einen Befehl




Beispiel: Lager mit Gabelstapler

```
class Stapler
{
    Gabel gab;
    Motor mot;
    ...
    void insRegal (int pfach)
    {
        gab.unten ();
        mot.vor (100) ;
        gab.heben (pfach*100+25) ;
        mot.vor (100) ;
        ...
    }
}
```

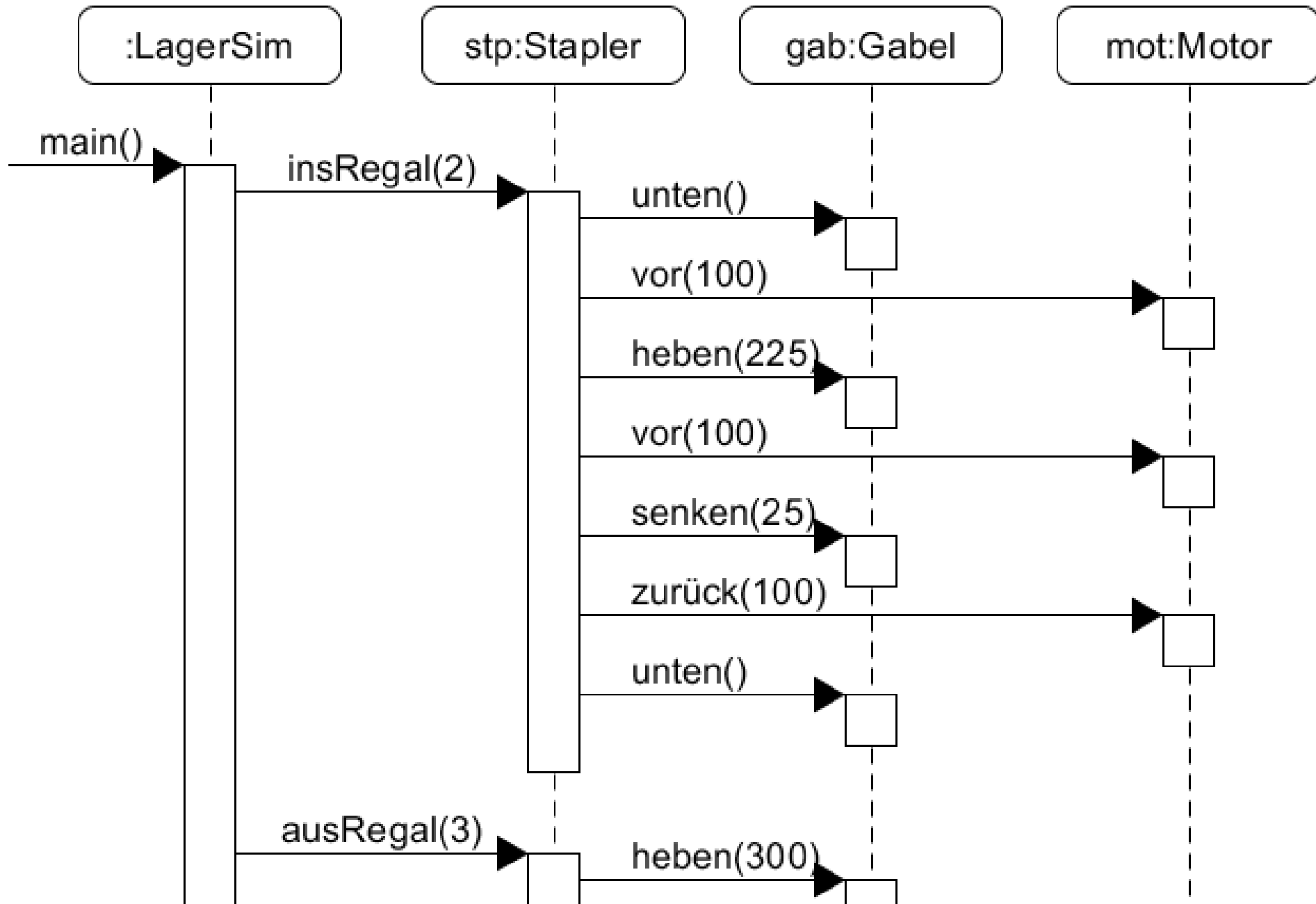
Beispiel: Lager mit Gabelstapler

```
class Stapler
{
    Gabel gab;
    Motor mot;
    ...
    void insRegal(int pfach)
    {
        gab.unten ();
        mot.vor (100) ;
        gab.heben (pfach*100+25) ;
        mot.vor (100) ;
        ...
    }
}
```

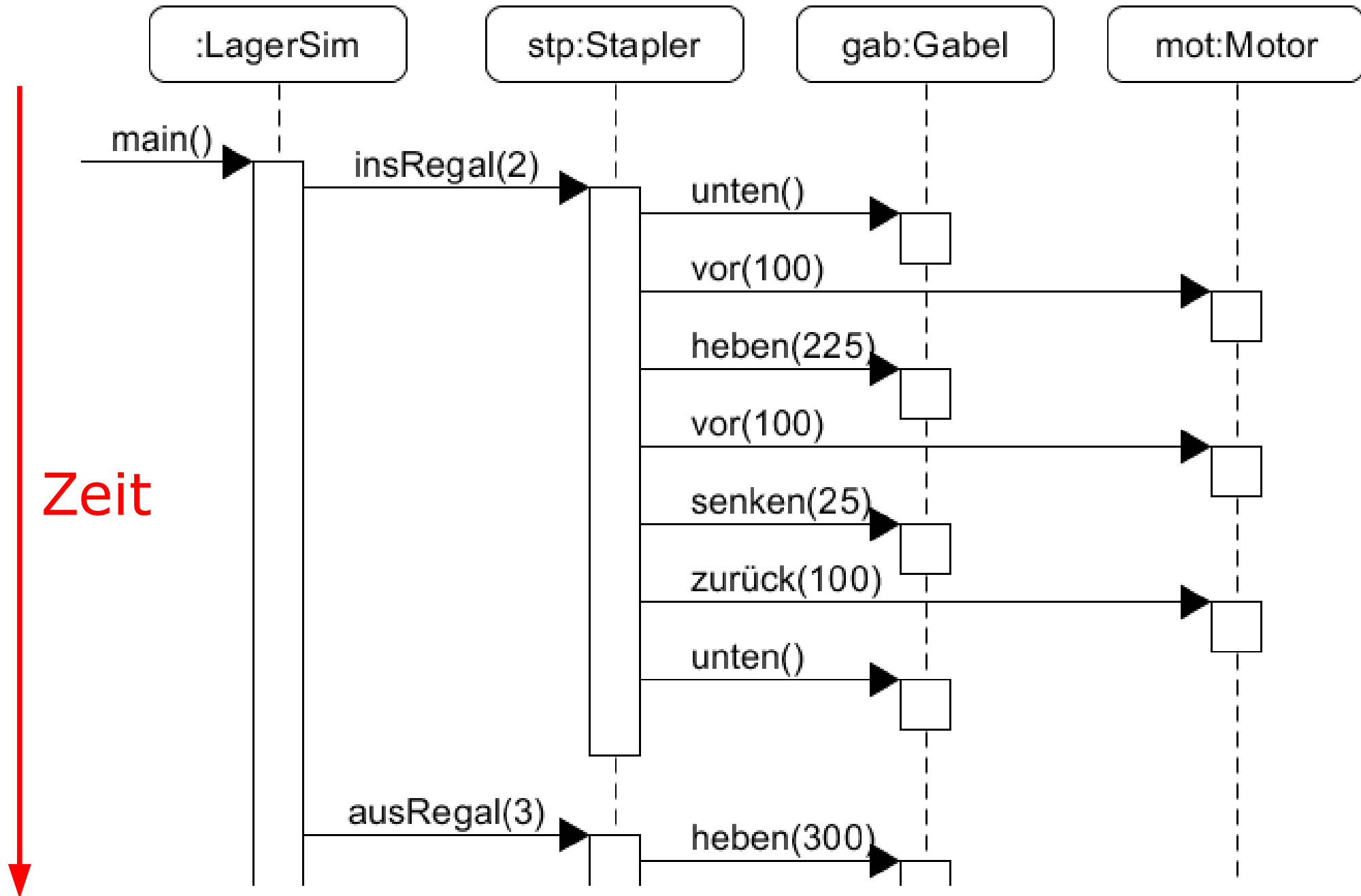
Stapler-Objekt
„sendet“
den Objekten
gab und mot
Befehle



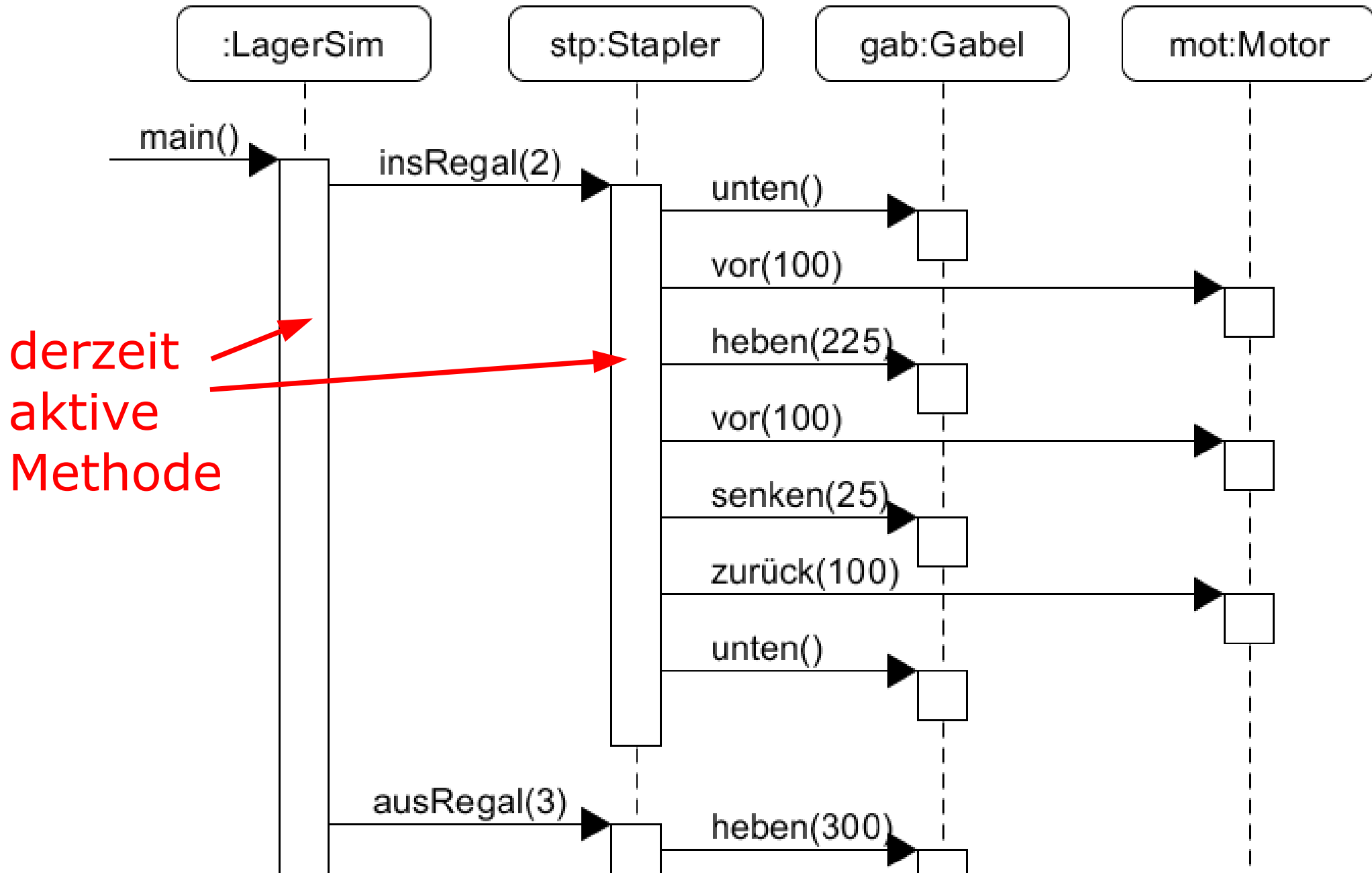
Sequenzdiagramm



Sequenzdiagramm



Sequenzdiagramm



Methoden bleiben „aktiv“

main() ruft stp.insRegal() auf,
stp.insRegal() ruft gab.senken() auf

Während gab.senken() ausgeführt wird,
sind stp.insRegal() und main() noch aktiv

Nach Ende von gab.senken() kehrt die
Ausführung zurück zu stp.insRegal().

Nach Ende von stp.insRegal() zurück zu
main(), die weitere Methoden aufruft.

Zweck von Sequenzdiagrammen

Sequenzdiagramme geben einen Überblick,

welches Objekt
zu welcher Zeit
mit welcher Methode

gerade aktiv ist, und

welche Methode welche andere aufruft.

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
Freigegeben unter CC BY-NC-SA 4.0, März 2021

