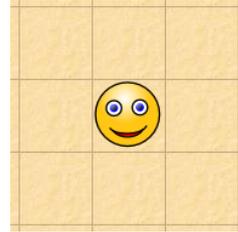


OOP: TASTATUR-STEUERUNG

Wir haben bereits gesehen, wie man den Smiley mit der Tastatur in eine Richtung steuern kann. Das ging so:

```
public void act()
{
    if(Greenfoot.isKeyDown("right"))
    {
        this.setLocation(this.getX() + 1, this.getY());
        Greenfoot.delay(10);
    }
}
```



Ändern Sie die act-Methode in der Smiley-Klasse jetzt so ab, dass Sie das Smiley-Objekt mit den Pfeiltasten auch in die anderen drei Richtungen steuern können!

OOP: EIGENE METHODEN

Wenn Sie es geschafft haben, ein Smiley-Objekt mit der Tastatur in alle Richtungen zu bewegen, fällt Ihnen vielleicht auf, dass die act-Methode irgendwann ziemlich unübersichtlich werden wird. Wenn man auf eine Pfeiltaste drückt, können ja noch ganz andere Dinge passieren (z.B. Sound abspielen), der Programm-Code wird also immer länger.

Um den Überblick zu behalten, was in der act-Methode passiert, ist es sinnvoll, **eigene Methoden** zu schreiben. Unser Programm soll später in etwa so aussehen:

```
public void act()
{
    if(Greenfoot.isKeyDown("down"))
    {
        this.moveDown();
    }
}
.....
```

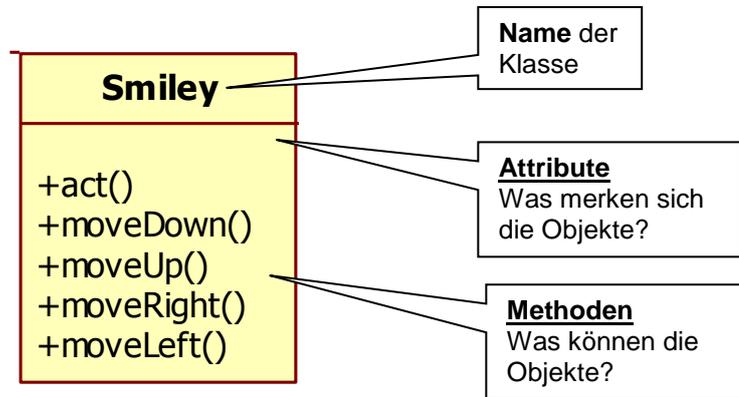
Damit das funktioniert, müssen wir eine eigene Methode mit dem Namen *moveDown* erstellen:

```
public void moveDown()
{
    // Bewegt den Smiley ein Feld nach unten und wartet dann 10 'Millisekunden'
    this.setLocation(this.getX(), this.getY() + 1);
    Greenfoot.delay(10);
}
```



Wenn man innerhalb einer Klasse eine Methode aufruft, die sich auf ein Objekt dieser Klasse bezieht, kann man **this** übrigens auch weglassen! Statt `this.moveDown()`; würde auch `moveDown()`; funktionieren.

Um sich notieren zu können, was man eigentlich programmieren will, gibt es in der Objektorientierten Programmierung eine eigene grafische Sprache mit dem Namen UML (Unified Modelling Language). Damit sieht unsere Smiley-Klasse in der UML so aus:



Lösung:

```

public class Smiley extends Actor
{
    public void act()
    {
        if(Greenfoot.isKeyDown("down"))
        {
            this.moveDown();
        }

        if(Greenfoot.isKeyDown("up"))
        {
            this.moveUp();
        }

        if(Greenfoot.isKeyDown("right"))
        {
            this.moveRight();
        }

        if(Greenfoot.isKeyDown("left"))
        {
            this.moveLeft();
        }
    }

    public void moveDown()
    {
        // Bewegt den Smiley ein Feld nach UNTEN und wartet dann 10 'Millisekunden'
        setLocation(getX(), getY() + 1);
        Greenfoot.delay(10);
    }

    public void moveUp()
    {
        // Bewegt den Smiley ein Feld nach OBEN und wartet dann 10 'Millisekunden'
        setLocation(getX(), getY() - 1);
        Greenfoot.delay(10);
    }

    public void moveRight()
    {
        // Bewegt den Smiley ein Feld nach RECHTS und wartet dann 10 'Millisekunden'
        setLocation(getX() + 1, getY());
        Greenfoot.delay(10);
    }

    public void moveLeft()
    {
        // Bewegt den Smiley ein Feld nach LINKS und wartet dann 10 'Millisekunden'
        setLocation(getX() - 1, getY());
        Greenfoot.delay(10);
    }
}

```