HOFFROGGE

From good to great!

Gutes kann noch besser werden,

Sortiments- und Platzierungsoptimierung in einer neuen Dimension.

https://github.com/HInformatikAG

Fragen ?¿

it-ag@hoffrogge.com

Lehreinheit 5

Ziele

- Interfaces

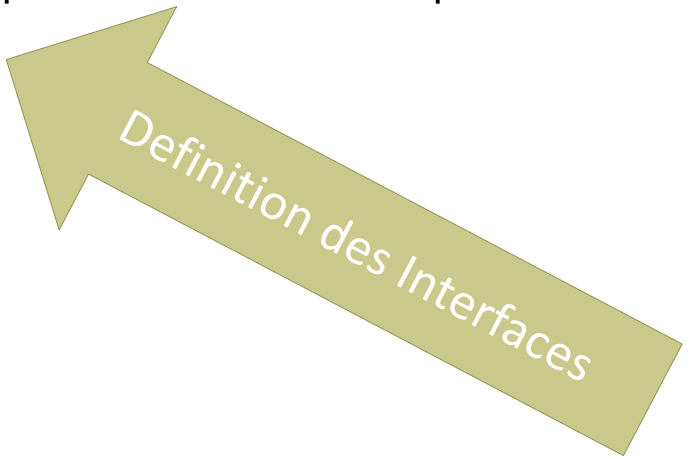
Interfaces

Muster für Klassen

- Ein Interface ist eine besondere Art einer Klasse
- Das Interface definiert, welche Aufgaben (Methoden) eine Klasse ausführen kann
  - Aber nicht, wie die Klasse diese Methoden ausführt
- Jede Klasse, die dieses Muster übernimmt, implementiert das Interface
- Vorteil: Viele verschiedene Klassen, die aber alle die gleichen Methoden haben
  - Sie können natürlich noch andere Methoden haben, aber sie haben auf jeden Fall die Methoden des Interfaces

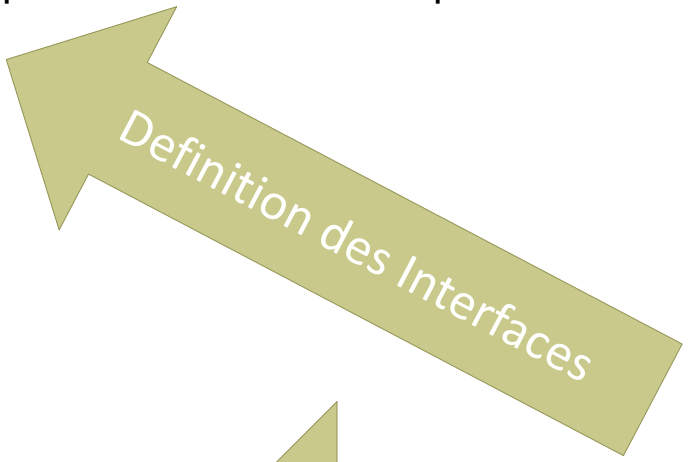
Interfaces

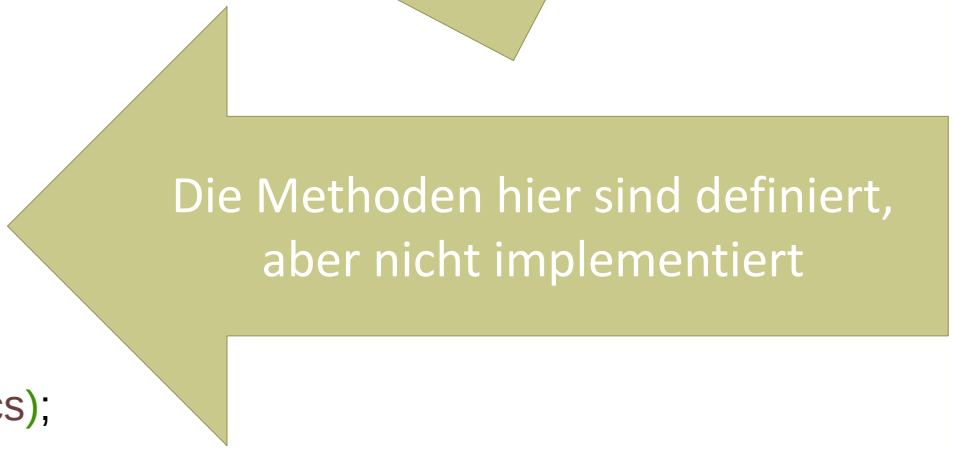
```
public interface TetrominoSpielstein extends Comparable<TetrominoSpielstein> {

    int getX();

    void setX(int x);

    int getY();

    void setY(int y);

    void bewegeNachUnten();

    void bewegeNachRechts();

    void bewegeNachLinks();

    void zeichnen(Graphics graphics);

}
```



Interfaces

```
public interface TetrominoSpielstein extends Comparable<TetrominoSpielstein> {

    int getX();

    void setX(int x);

    int getY();

    void setY(int y);

    void bewegeNachUnten();

    void bewegeNachRechts();

    void bewegeNachLinks();

    void zeichnen(Graphics graphics);

}
```





Interfaces


```
public class DummySpielstein implements TetrominoSpielstein {

    private int x;
    private int y;

    public DummySpielstein() {
        this.x = 120;
        this.y = 100;
    }

    @Override
    public void zeichnen(Graphics graphics) {

        graphics.setColor(getFuellFarbe());

        graphics.drawString("Ich bin nur ein Platzhalter ;)", x, y);
    }
}
```

Interfaces

Implementiert das Interface

```
public class DummySpielstein implements TetrominoSpielstein {

    private int x;
    private int y;

    public DummySpielstein() {
        this.x = 120;
        this.y = 100;
    }

    @Override
    public void zeichnen(Graphics graphics) {

        graphics.setColor(getFuellFarbe());

        graphics.drawString("Ich bin nur ein Platzhalter ;)", x, y);
    }
}
```

Methode aus dem Interface

Interfaces

- Klasse TetrominoBlock implementiert das Interface TetrominoSpielstein
- Klasse TetrominoLanger implementiert das Interface TetrominoSpielstein
- Klasse TetrominoT implementiert das Interface TetrominoSpielstein
- Klasse TetrominoL implementiert das Interface TetrominoSpielstein
- Klasse TetrominoZ implementiert das Interface TetrominoSpielstein
- Klasse TetrominoLUmgedreht implementiert das Interface TetrominoSpielstein
- Klasse TetrominoZUmgedreht implementiert das Interface TetrominoSpielstein

Interfaces

- Alle Klassen haben die Methoden aus dem Interface
- Man kann also alle Klassen nach dem X-Wert fragen (getX())
- Oder sie zeichnen (zeichnen(Graphics graphics))
- Die Klasse Spielfeld kann TetrominoSpielsteine zeichnen
- Dabei ist es egal, welcher TetrisSpielstein es ist, solange das Interface implementiert wird