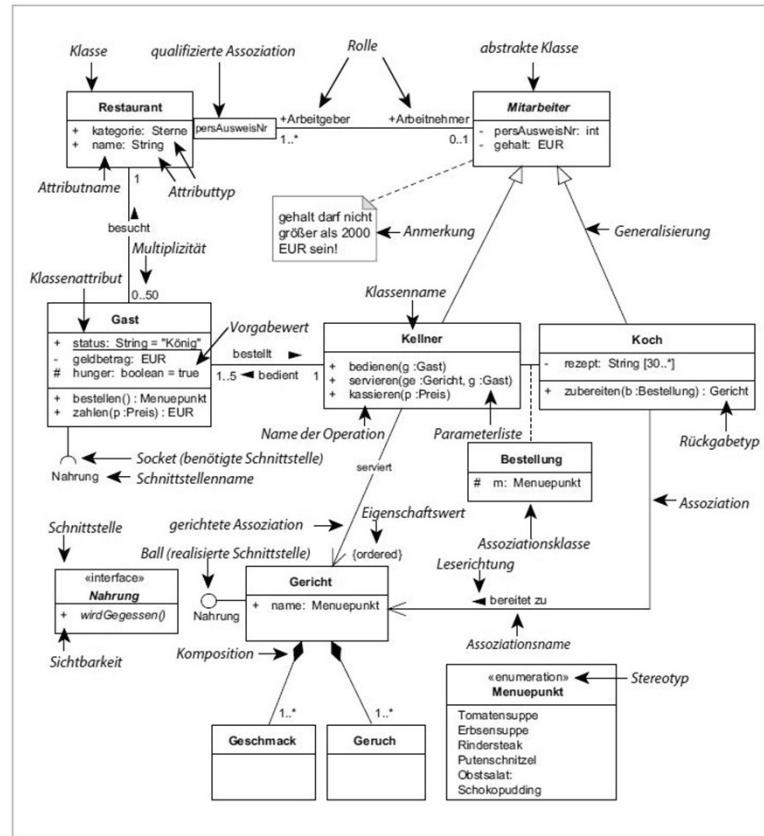


KLASSENDIAGRAMM

MARKUS SZYNSKA

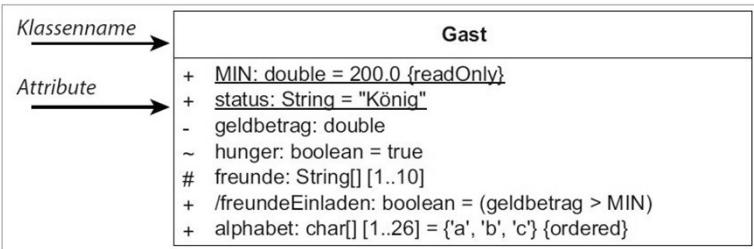
1

ÜBERSICHT



MARKUS SZYNSKA

KLASSE



```
class Gast
{
    public static final double MIN = 200.0;
    public static String status = "König";
    private double geldbetrag;
    boolean hunger;
    protected String[] freunde;
    public char[] alphabet;

    B public boolean freundeEinladen() {
        return geldbetrag > MIN;
    };

    C public Gast() {
        freunde = new String[10];
        alphabet = new char[26];
    }

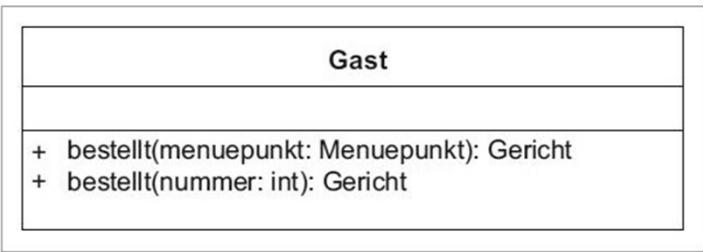
    D hunger = true;
    freunde[0] = new String();
    for (int i = 0; i < 26; i++)
        alphabet[i] = (char) ('a' + i);

    }
}
```

OPERATIONEN(METHODEN) EINER KLASSE

Klassename	Gast
Attribute	<ul style="list-style-type: none">+ MIN: double = 200.0 {readOnly}+ status: String = "König"- geldbetrag: double~ hunger: boolean = true# freunde: String[] [1..10]+ /freundeEinladen: boolean = (geldbetrag > MIN)+ alphabet: char[] [1..26] = {'a', 'b', 'c'} {ordered}
Operationen	<ul style="list-style-type: none">+ setGeldbetrag(geldbetrag: double): void+ getGeldbetrag(): double+ getMINAsString():String+ setFreund(in freund: String): void+ getBesterFreund(out freund: String): void+ getFreunde(): String[]+ getAlphabet(start: int, end:int): String {ordered}+ bestellt(menupunkt: Menuepunkt): Gericht+ bestellt(nummer: int): Gericht+ zahlt(d: double): double

OPERATIONEN(METHODEN) EINER KLASSE



```
public Gericht bestellt(Menuepunkt menuepunkt)
{
    Koch koch = new Koch();
    Gericht gericht = koch.kocht(menuepunkt);
    return gericht;
}

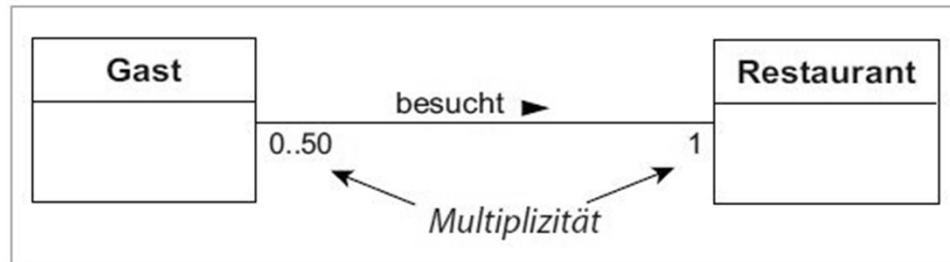
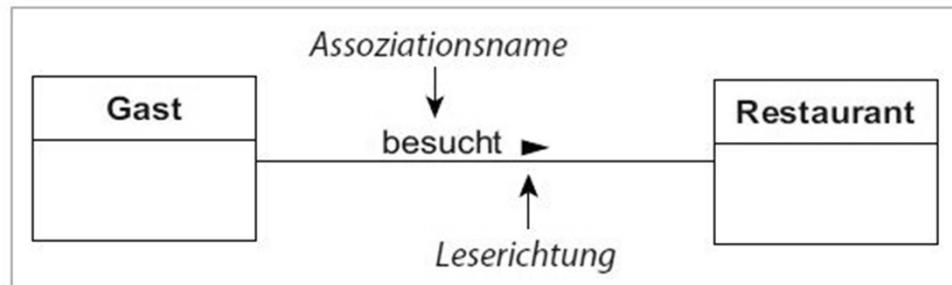
public Gericht bestellt(int numero)
{
    Menuepunkt menuepunkt = new Menuepunkt(numero);
    Koch koch = new Koch();
    Gericht gericht = koch.kocht(menuepunkt);
    return gericht;
}
```



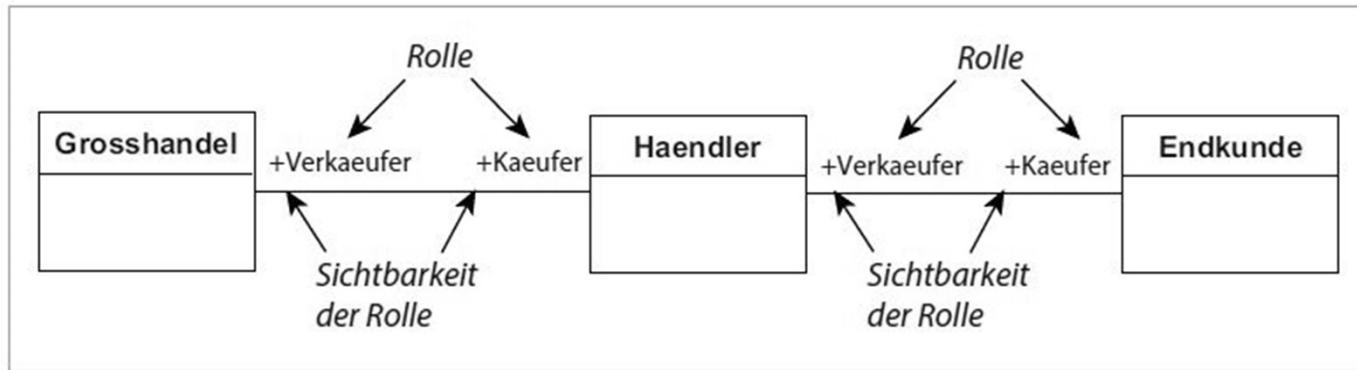
ASSOZIATIONEN

MARKUS SZYNSKA

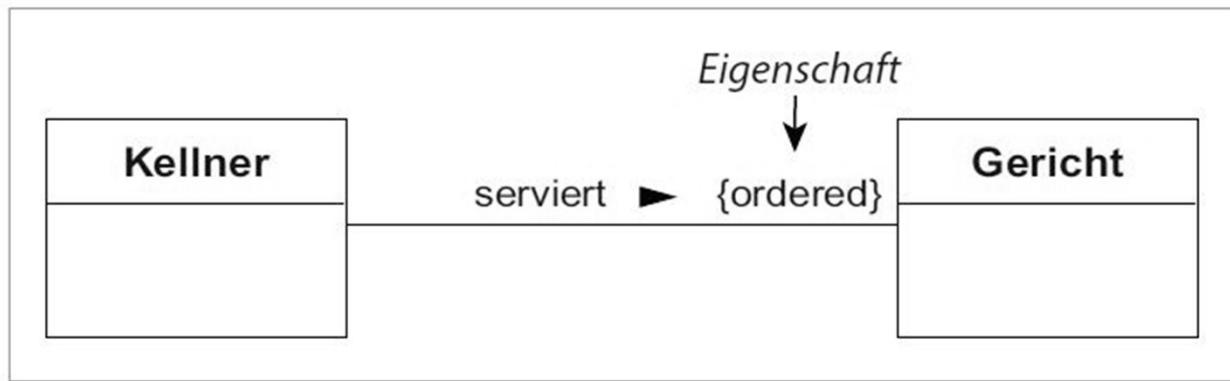
ASSOZIATIONEN MIT ASSOZIATIONSNAME UND MULTIPLIZITÄT



ASSOTIATIONEN MIT ROLLENANGABEN

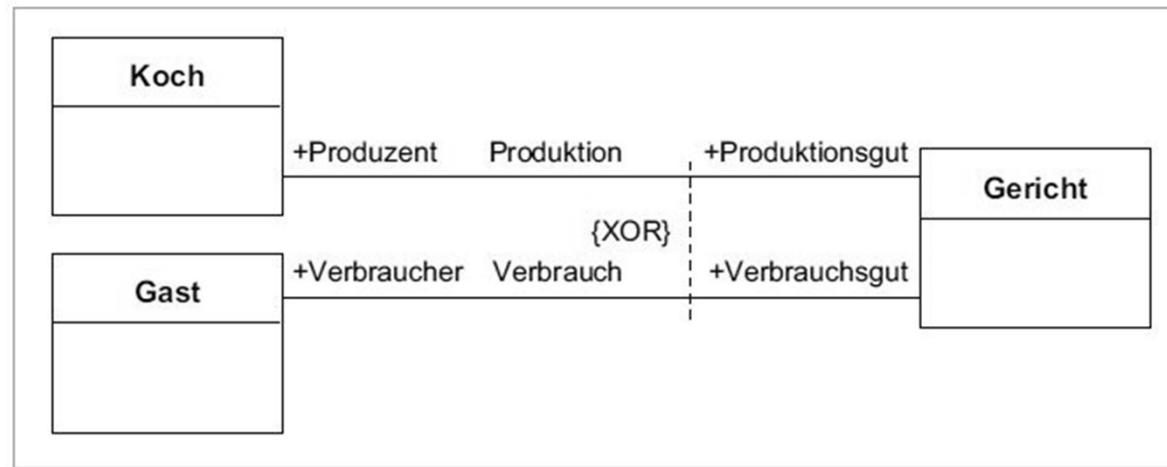


ASSOTIATIONEN MIT EIGENSCHAFT



MARKUS SZYNSKA

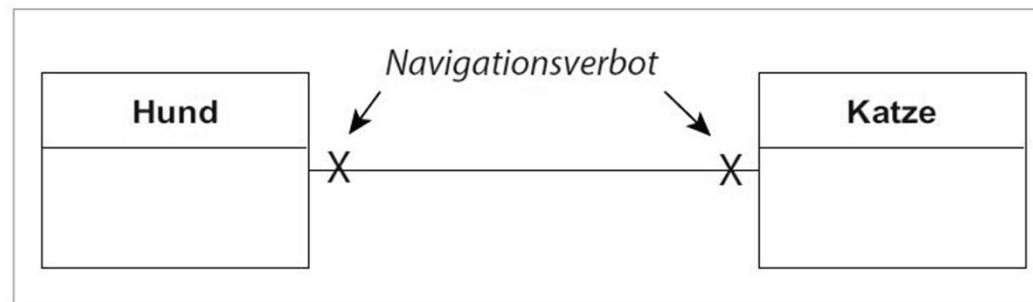
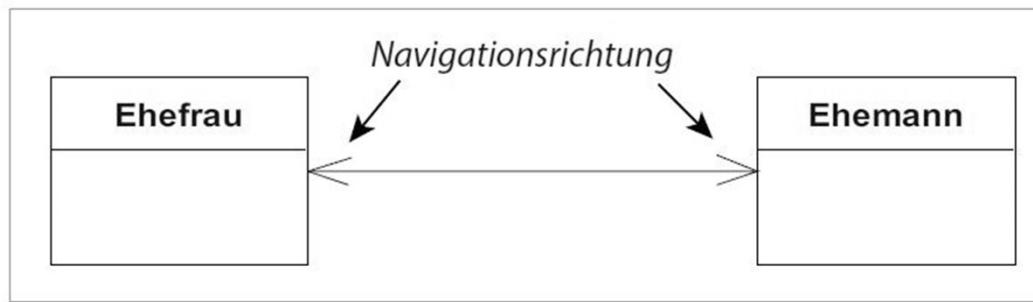
ASSOTIATIONEN MIT EINSCHRÄNKUNG



MARKUS SZYNSKA

10

ASSOTIATIONEN MIT BIDIREKTIONALER NAVIGIERBARKEIT



ASSOCIATIONS MIT UNSPEZIFIZIERTER UND UNIDIREKTIONALER NAVIGIERBARKEIT



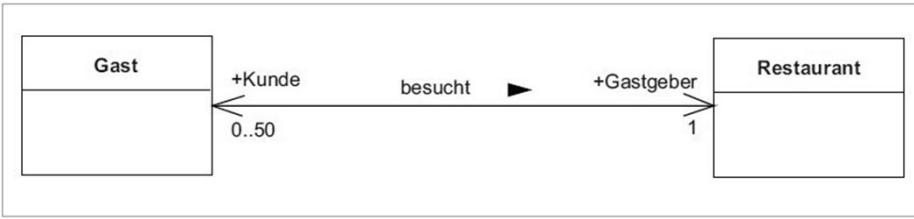
MARKUS SZYNSKA

12

ASSOTIATIONEN MIT TEILWEISER SPEZIFIKATION DER NAVIGIERBARKEIT UND BESITZANZEIGE

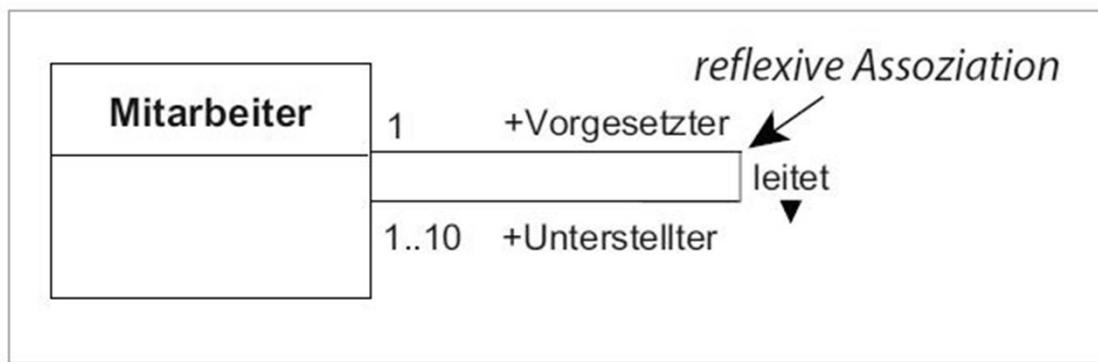


BEISPIEL ASSIOZIATIONSDEFINITION



```
class Gast
{
A    public Restaurant gastgeber;
B    public Gast(Restaurant r)
{
    gastgeber = r;
}
class Restaurant
{
C    public Gast[] kunde;
public Restaurant()
{
D    kunde = new Gast[50];
}
```

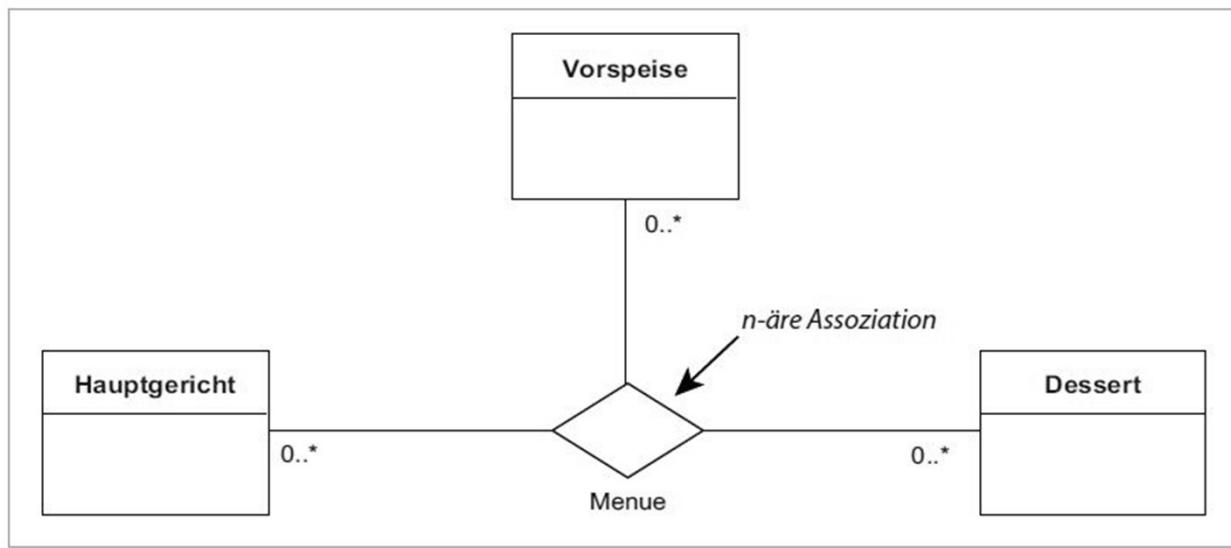
REFLEXIVE ASSOZIATION



MARKUS SZYNSKA

15

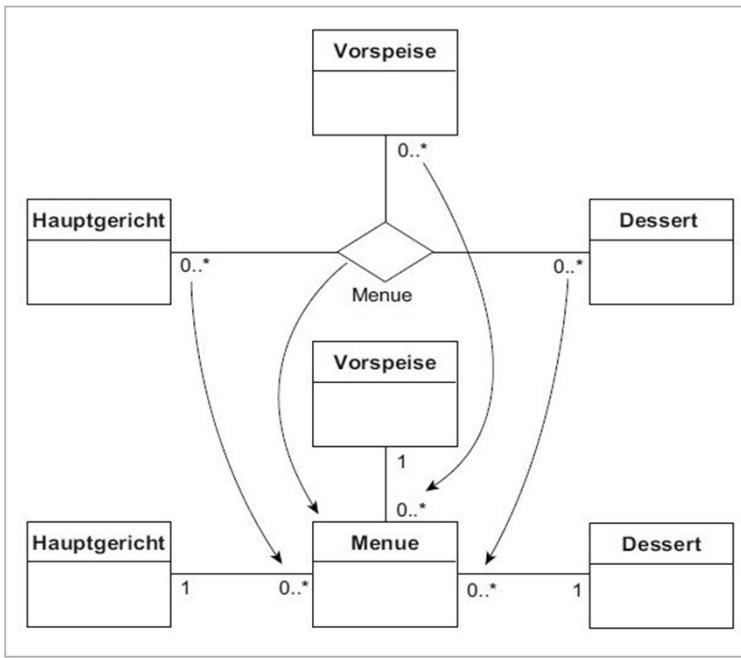
N-ÄRE ASSOZIATION



MARKUS SZYNSKA

16

N-ÄRE ASSOZIATION – UMWANDLUNG IN BINÄRE ASSOZIATION



MARKUS SZYNSKA

17

N-ÄRE ASSOZIATION – UMWANDLUNG IN BINÄRE ASSOZIATION BEISPIEL JAVA

```
A class Menue
{
    B public Vorspeise vorspeise;
    public Hauptgericht hauptgericht;
    public Dessert dessert;

    C public Menue(Vorspeise v, Hauptgericht h, Dessert d)
    {
        vorspeise = v;
        hauptgericht = h;
        dessert = d;
    }
}
```

```
class Vorspeise
{
    A public ArrayList menue;

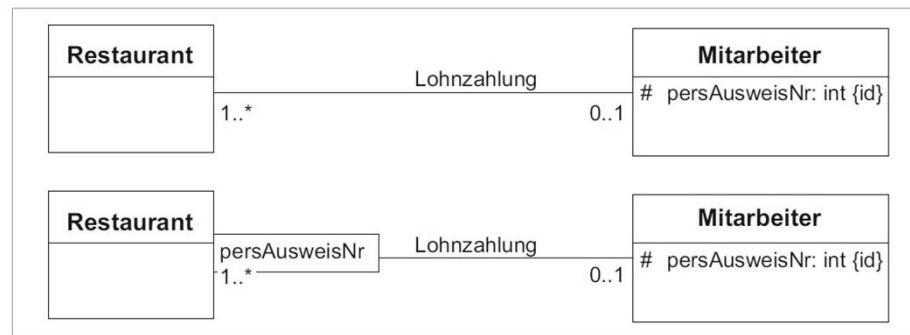
}

class Hauptgericht
{
    A public ArrayList menue;

}

class Dessert
{
    A public ArrayList menue;
}
```

QUALIFIZIERTE ASSOZIATION



QUALIFIZIERTE ASSOZIATION IN JAVA

```
class Mitarbeiter
{
    A protected int persAusweisNr;

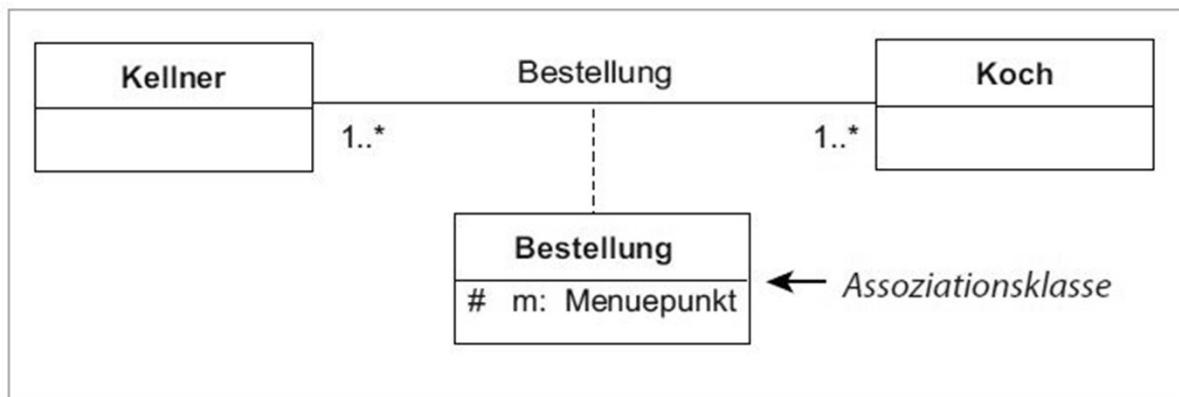
}

class Restaurant
{
    B protected ArrayList mitarbeiter;
}
```

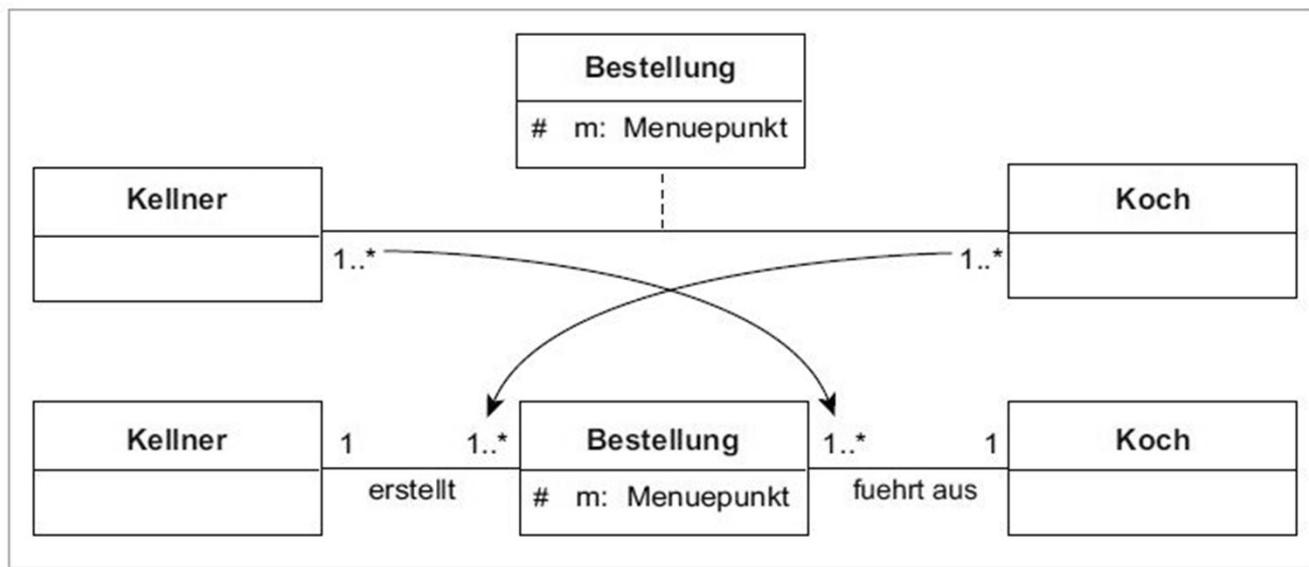
MARKUS SZYNSKA

20

ASSOZIATIONSKLASSE



ASSOZIATIONSKLASSE IN JAVA



MARKUS SZYNSKA

22

ASSOZIATIONSKLASSE IN JAVA

```
class Kellner
{
    A  public ArrayList bestellung;

}

class Koch
{
    B  public ArrayList bestellung;

}

class Bestellung
{
    C  protected Menuepunkt m;
        public Kellner ke;
        public Koch ko;
}
}
```

MARKUS SZYNSKA

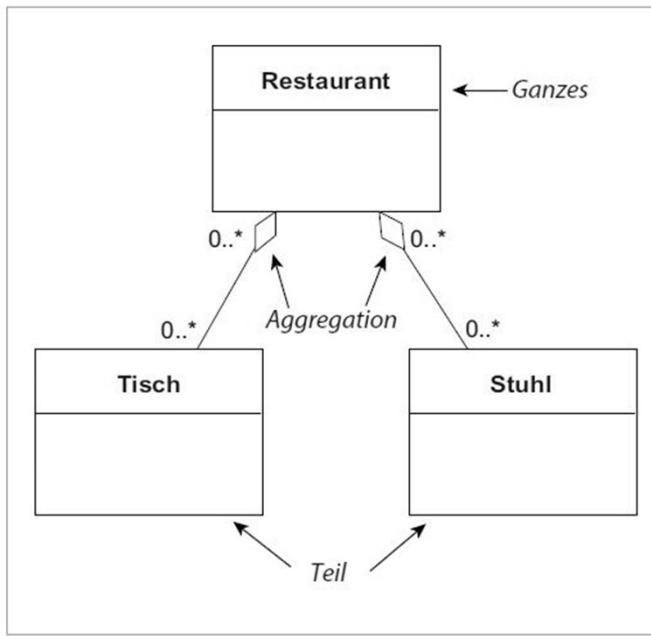
23

AGGREGATIONEN

MARKUS SZYNSKA

24

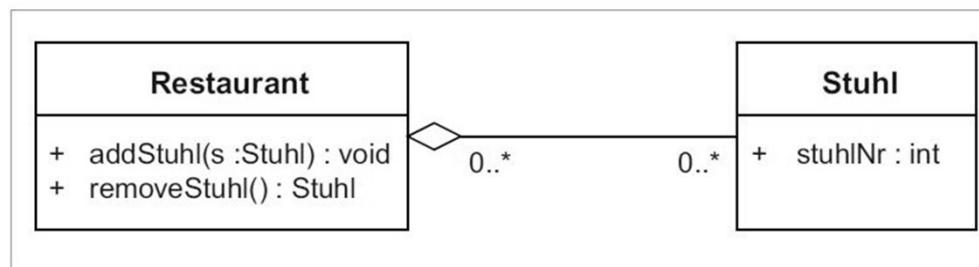
AGGREGATION



MARKUS SZYNSKA

25

AGGREGATION



```
class Stuhl
{
    public int stuhlNr;
    public Stuhl(int nr)
    {
        stuhlNr = nr;
    }
}
```

AGGREGATION

```
class Restaurant
{
    A public ArrayList stuhl;

    B public Restaurant()
    {
        stuhl = new ArrayList();
    }

    C public void addStuhl(Stuhl s)
    {
        stuhl.add(stuhl.size(), s);
    }

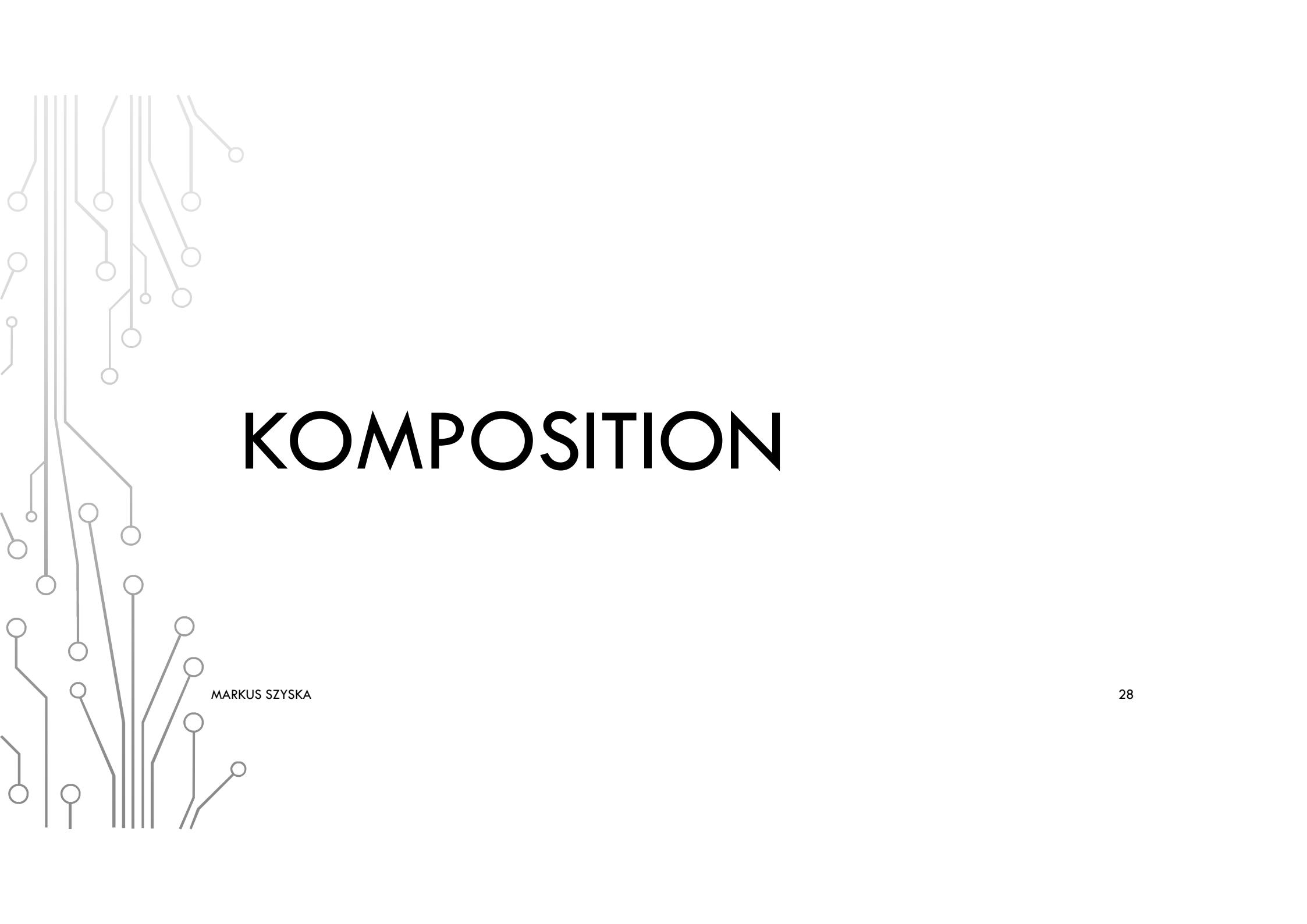
    D public Stuhl removeStuhl()
    {
        return (Stuhl)stuhl.remove(stuhl.size()-1);
    }
}
```

```
public static void main(String[] args)
{
    E Restaurant r = new Restaurant();
    Stuhl[] s = new Stuhl[10];
    for(int i = 0; i < 10; i++)
        s[i] = new Stuhl(i);

    F for(int i = 0; i < 10; i++)
        r.addStuhl(s[i]);

    G for(int i = 0; i < 10; i++)
        s[i] = r.removeStuhl();

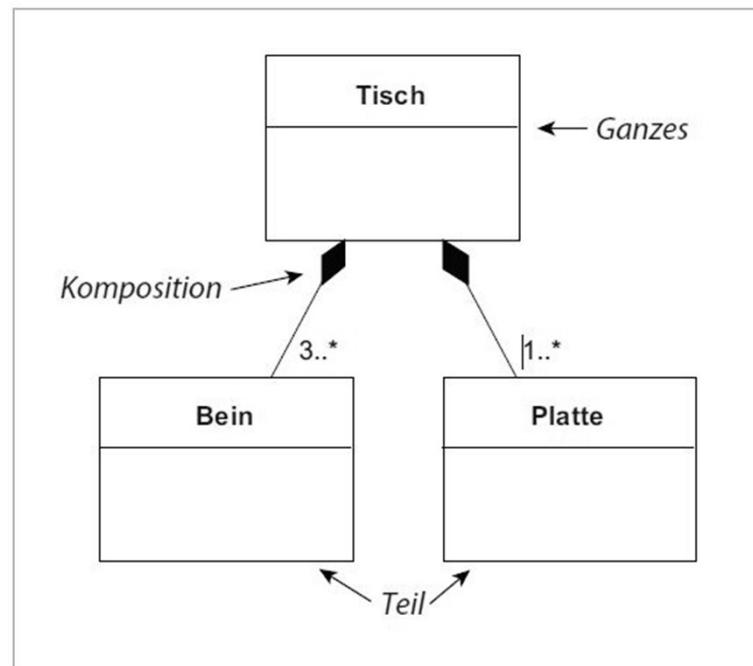
}
}
```



KOMPOSITION

MARKUS SZYNSKA

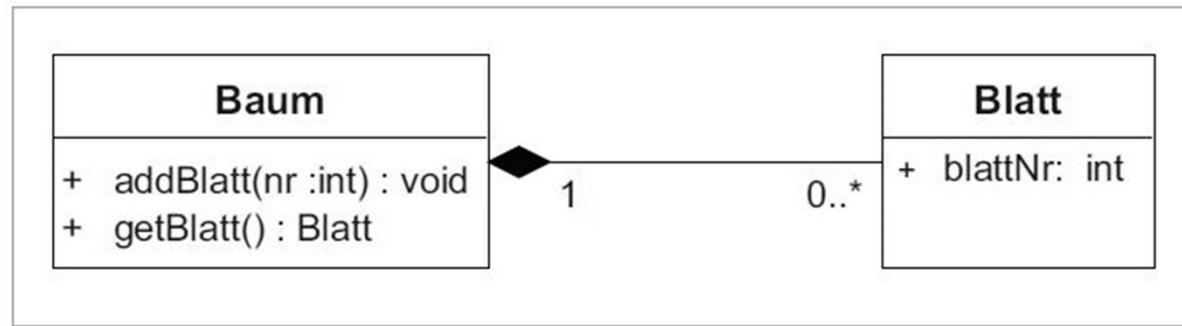
KOMPOSITION



MARKUS SZYNSKA

29

KOMPOSITION



MARKUS SZYNSKA

30

KOMPOSITION

```
class Blatt
{
    public int blattNr;
    public Blatt(int nr)
    {
        blattNr = nr;
    }
}
```

MARKUS SZYNSKA

31

KOMPOSITION

```
class Baum
{
A private ArrayList bl;

B public Baum()
{
    bl = new ArrayList();
}

C public void addBlatt(int nr)
{
    bl.add(new Blatt(nr));
}

D public Blatt getBlatt()
{
    Blatt neuesBlatt = new Blatt(0);
    neuesBlatt.blattNr=((Blatt)bl.get(bl.size()-1)).blattNr;
    return neuesBlatt;
}
```

MARKUS SZYNSKA

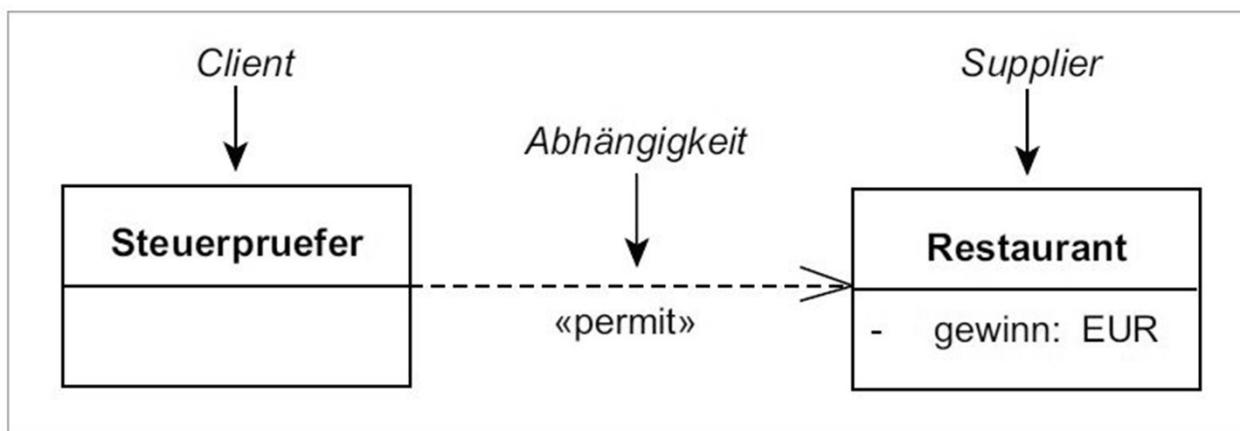
32



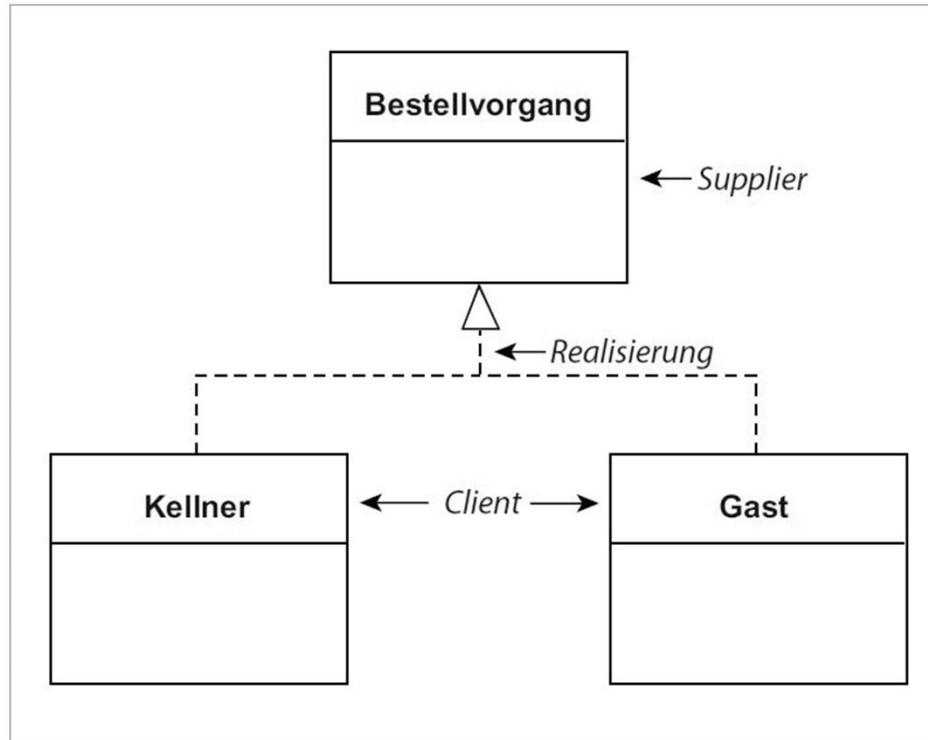
ABHÄNGIGKEIT

MARKUS SZYNSKA

ABHÄNGIGKEIT



ABHÄNGIGKEIT



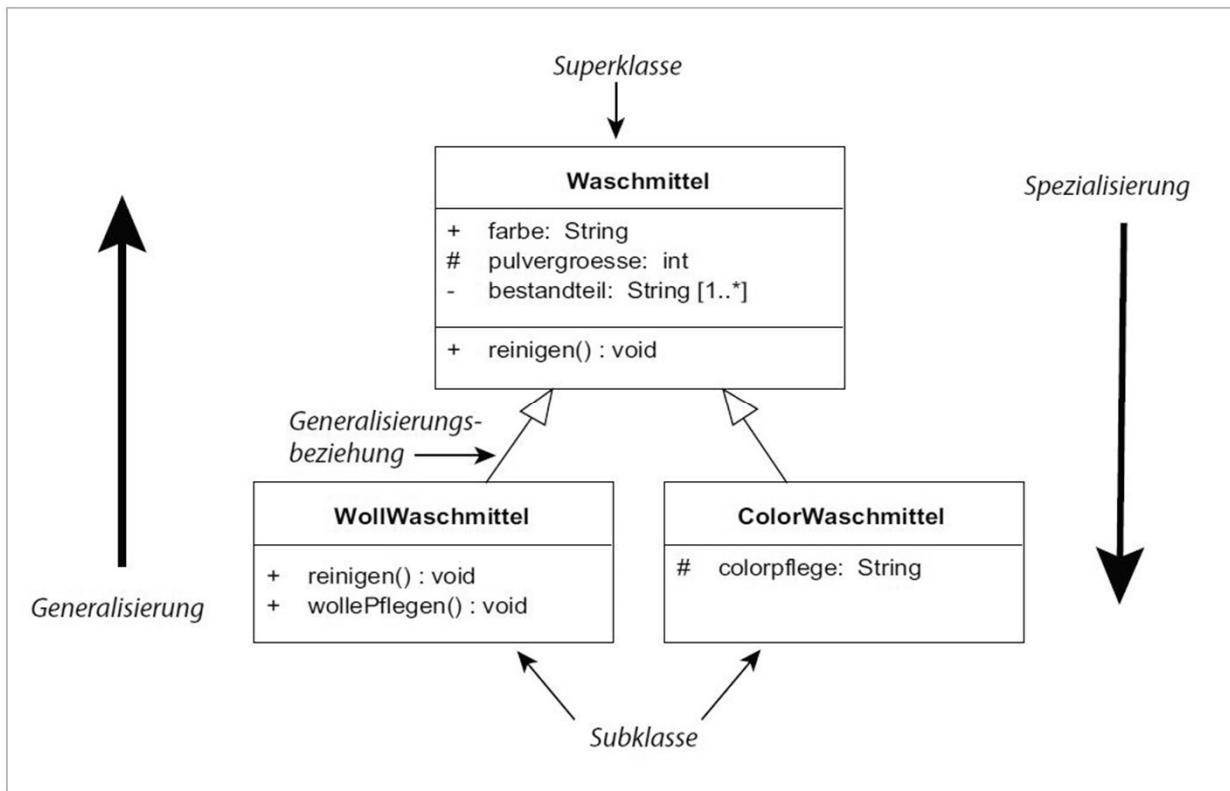
MARKUS SZYNSKA

35

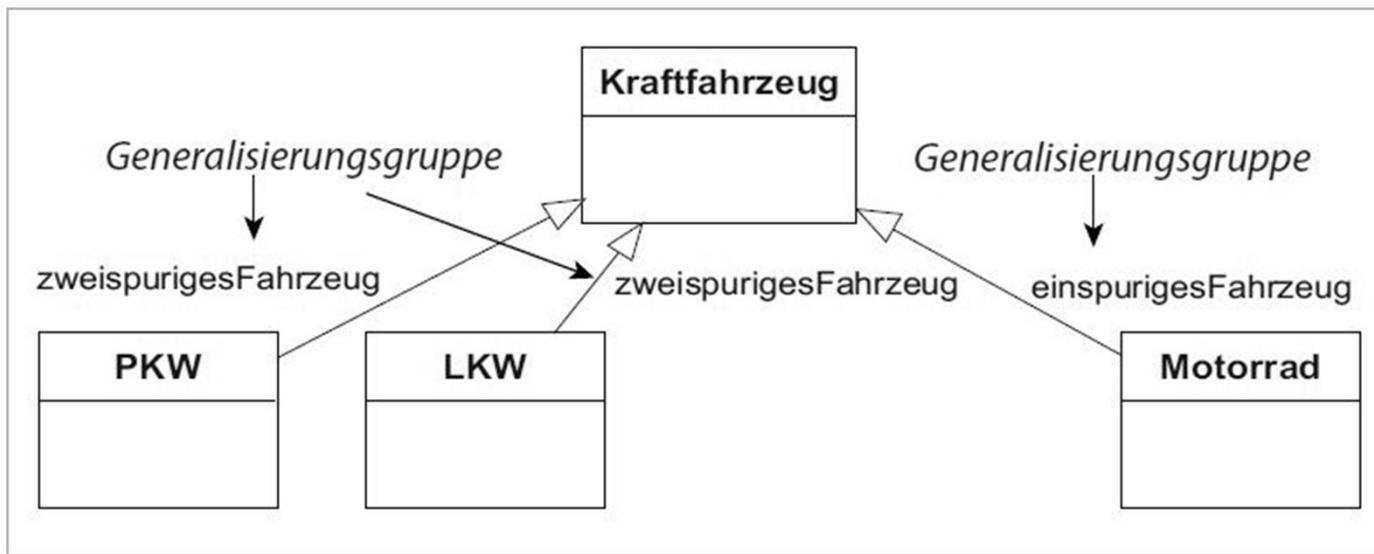
GENERALISIERUNG

MARKUS SZYNSKA

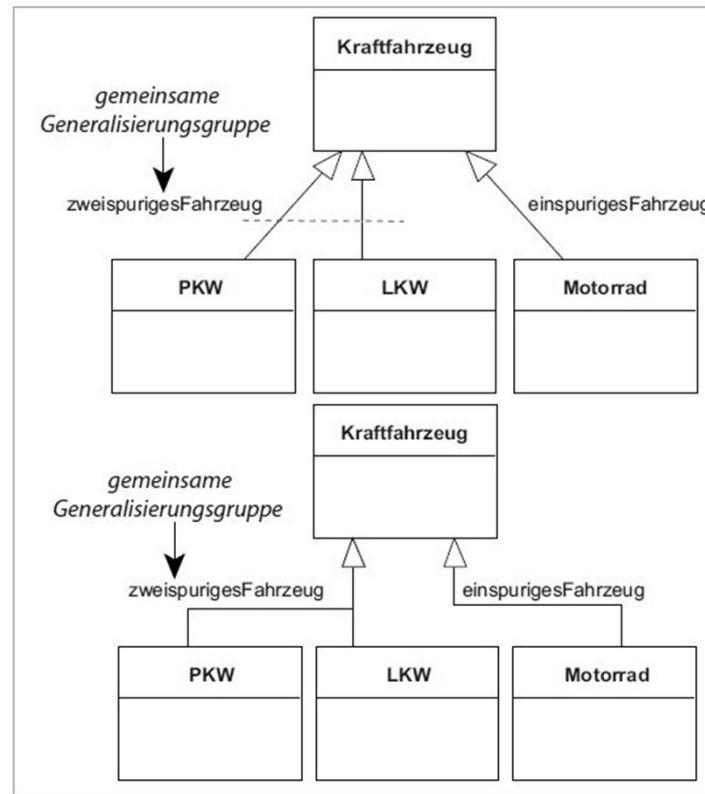
GENERALISIERUNG



GRUPPIEREN VON GENERALISIERUNGEN



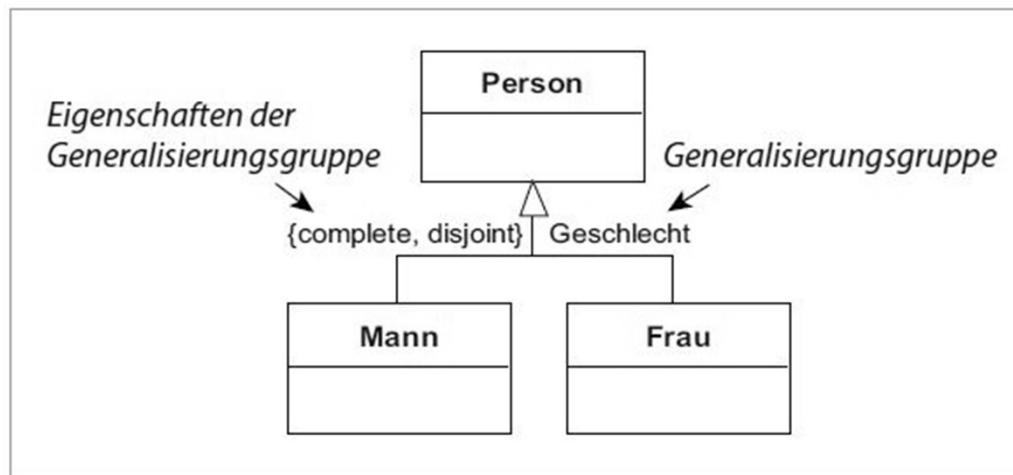
GEMEINSAME GENERALISIERUNGSGRUPPEN



MARKUS SZYNSKA

39

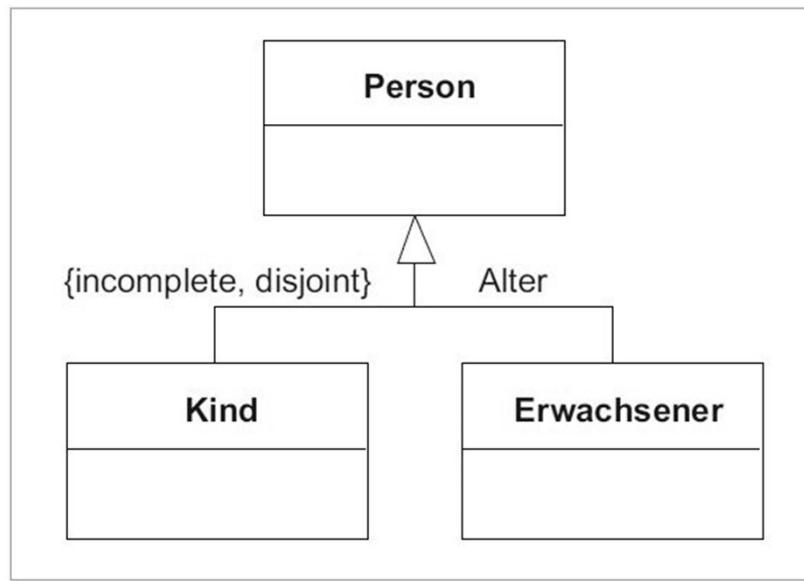
{complete, disjoint}-Eigenschaft



MARKUS SZYNSKA

40

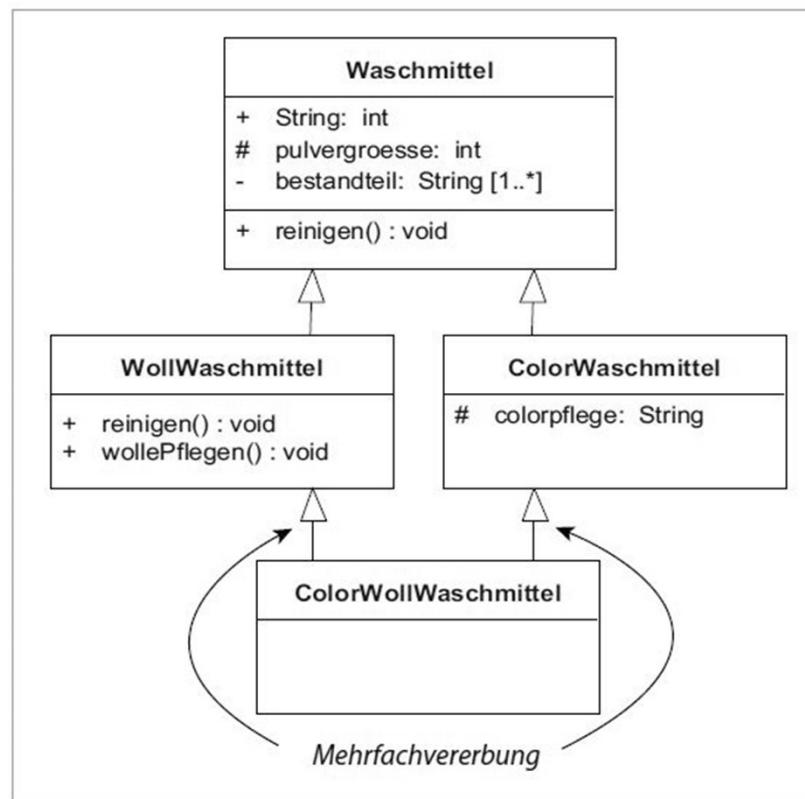
{incomplete, disjoint}-Eigenschaft



MARKUS SZYNSKA

41

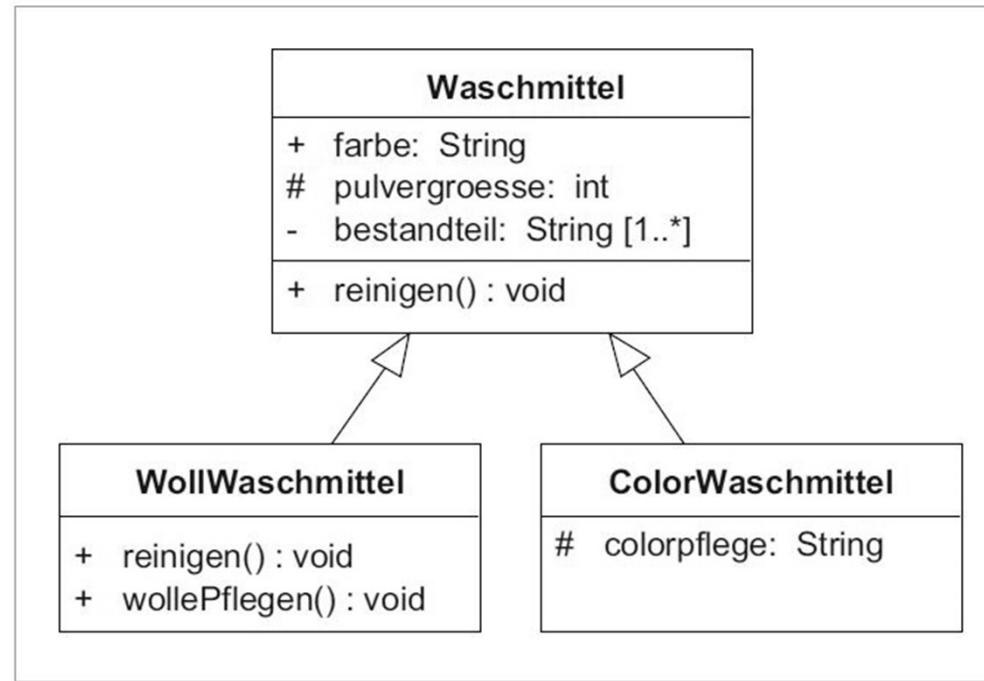
Mehrfachvererbung



MARKUS SZYNSKA

42

Generalisierung Java



MARKUS SZYNSKA

43

Stereotyp

Stereotyp

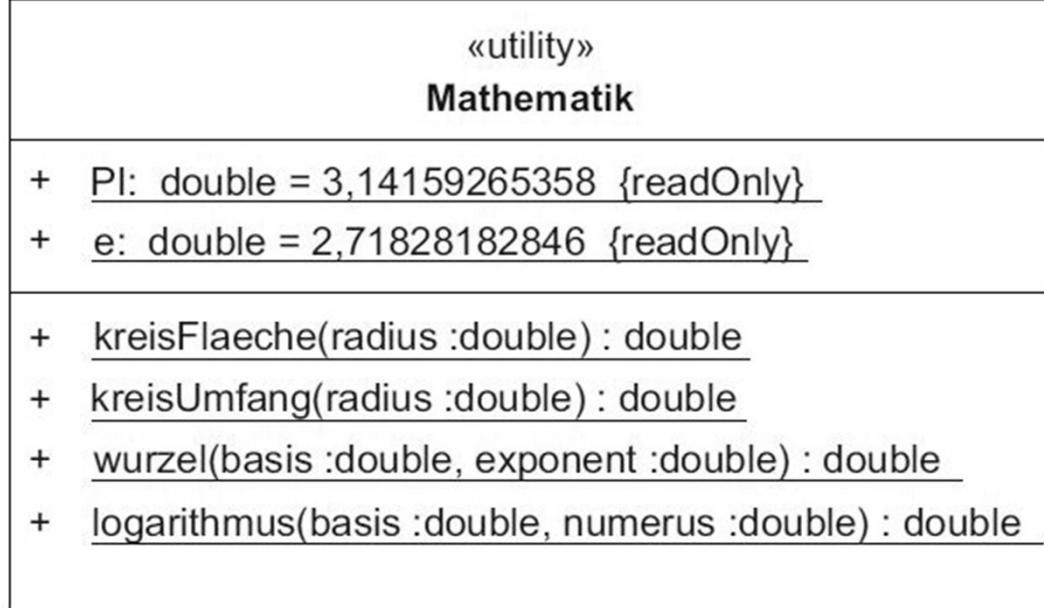
«enumeration»
Menuepunkt

Tomatensuppe
Erbsensuppe
Rindersteak
Putenschnitzel
Obstsalat
Schokopudding

MARKUS SZYNSKA

44

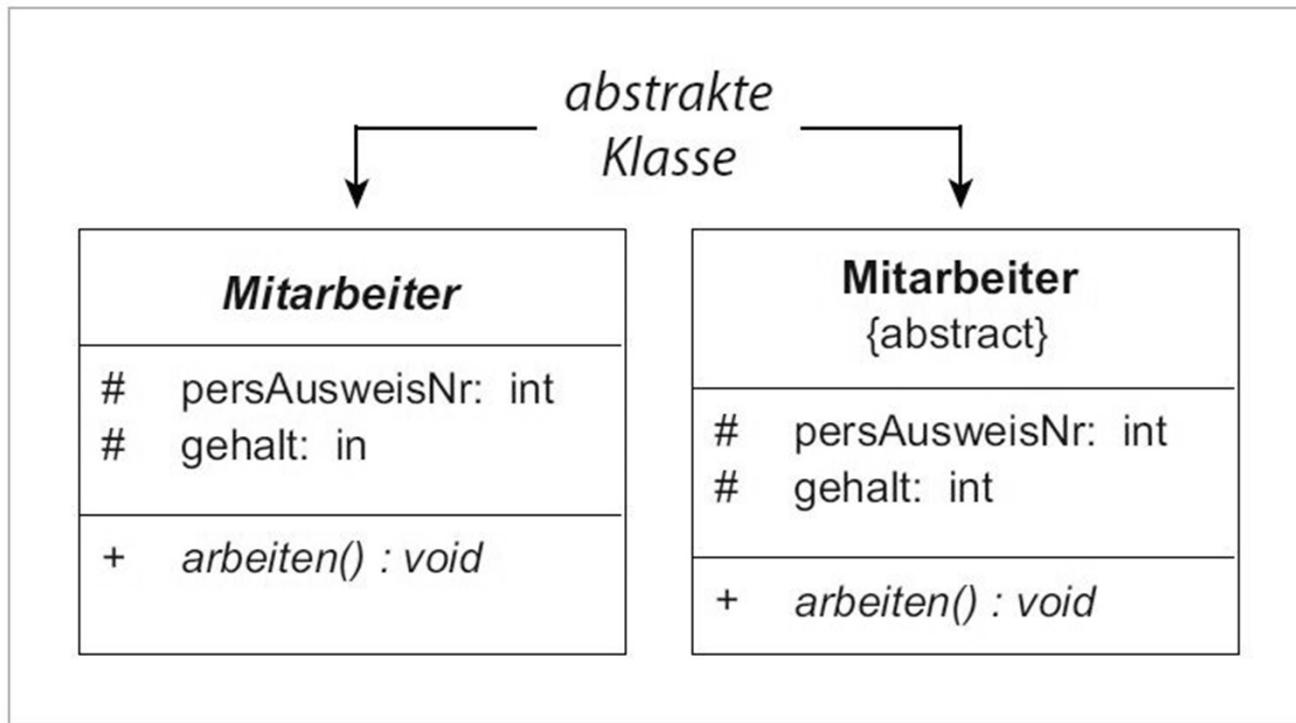
Stereotyp Beispiel



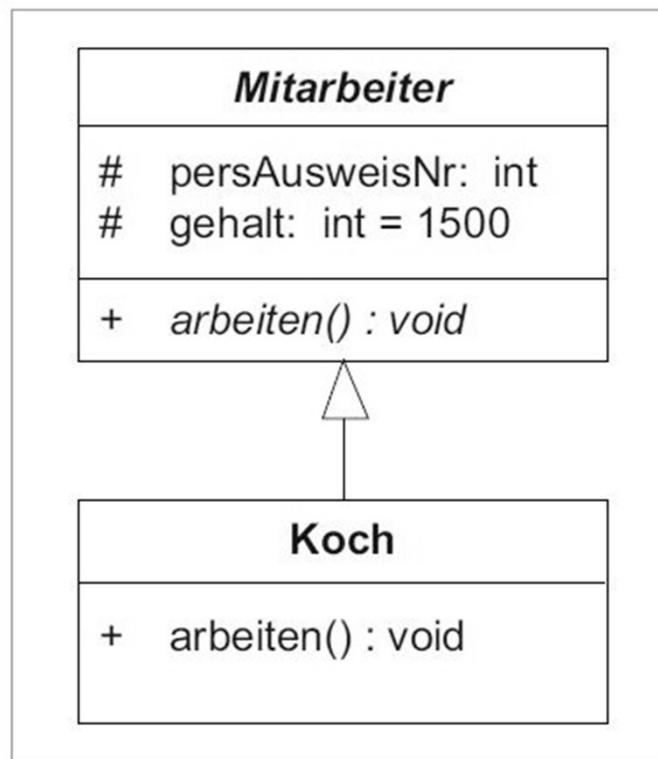
ABSTRAKTE KLASSEN

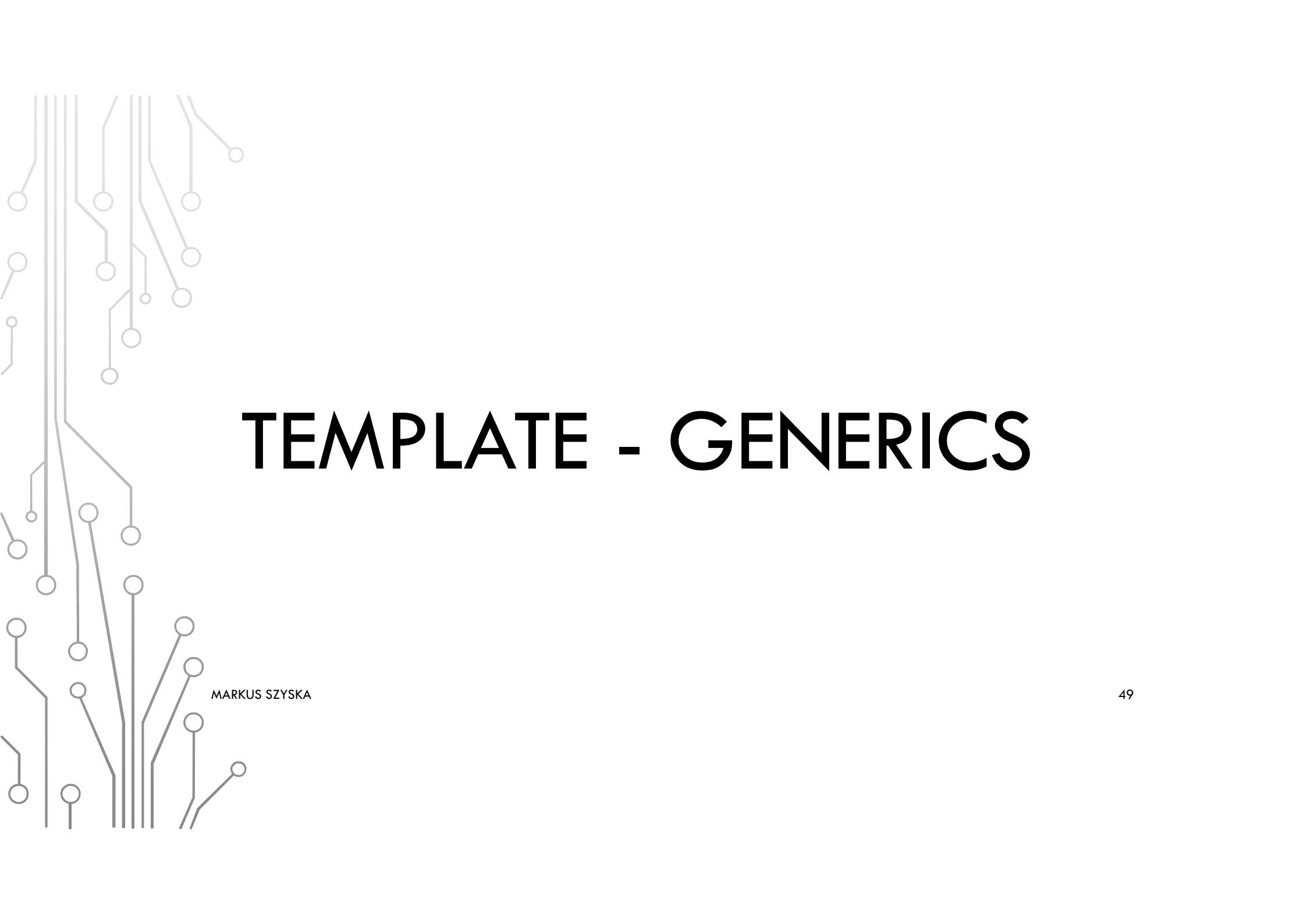
MARKUS SZYNSKA

Abstrakte Klassen



Abstrakte Klasse in Java

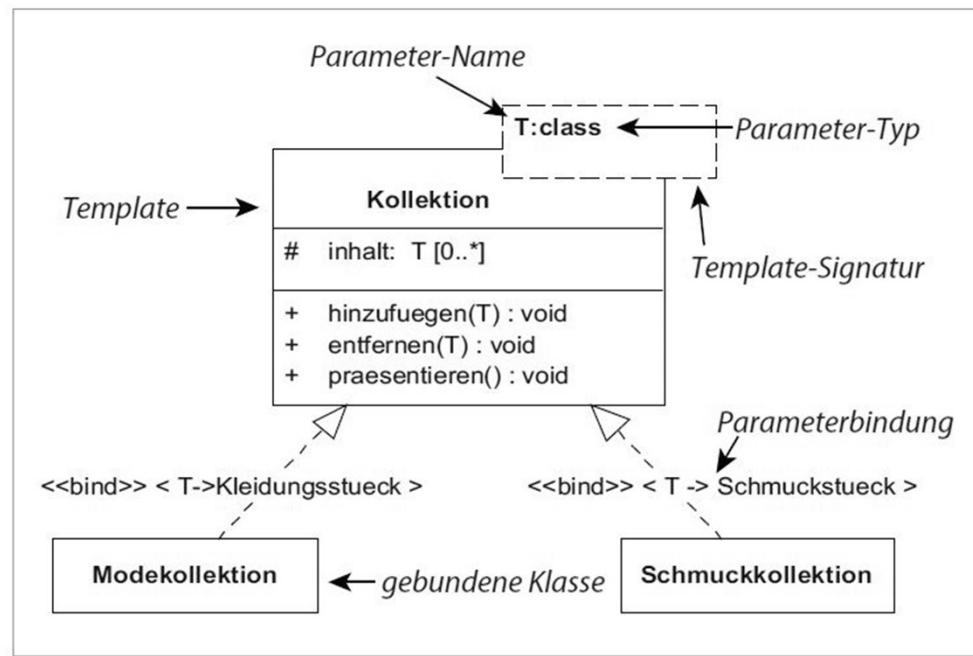




TEMPLATE - GENERICS

MARKUS SZYNSKA

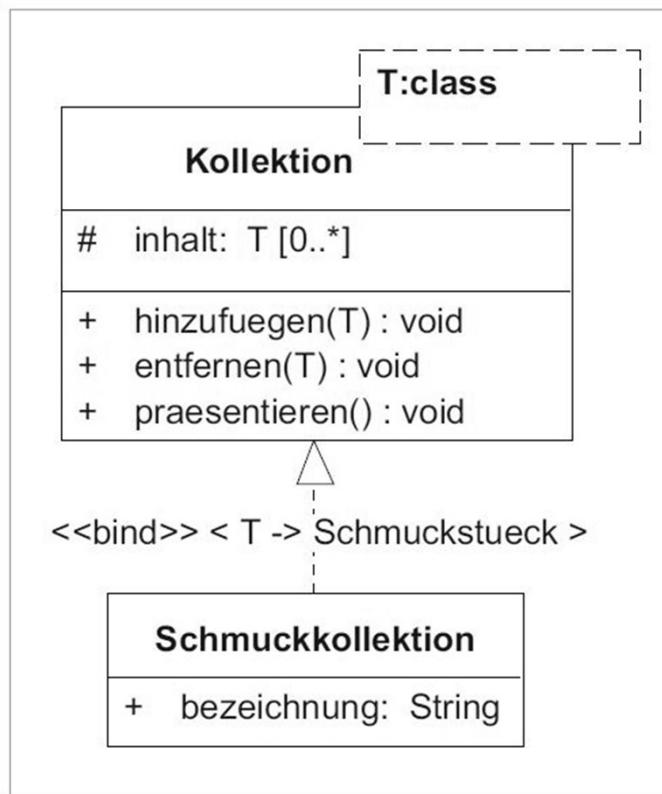
Template - Generics



MARKUS SZYNSKA

50

Template – Generics in Java



Template – Generics in Java

```
A class Kollektion<T>
{
    B protected ArrayList<T> inhalt;

    C public Kollektion()
    {
        inhalt = new ArrayList<T>();
    }

    D public void hinzufuegen(T elem)
    {
        inhalt.add(elem);
    }

    E public void entfernen(T elem)
    {
        inhalt.remove(elem);
    }

    F public void praesentieren()
    {
        for (T temp : inhalt)
            System.out.println(temp);
    }
}
```

MARKUS SZYNSKA

52

Template – Generics in Java

```
class Schmuckstueck
{
    public String bezeichnung;
    public Schmuckstueck(String bez)
    {
        bezeichnung = bez;
    }

A  public String toString()
{
    return bezeichnung;
}

}
```

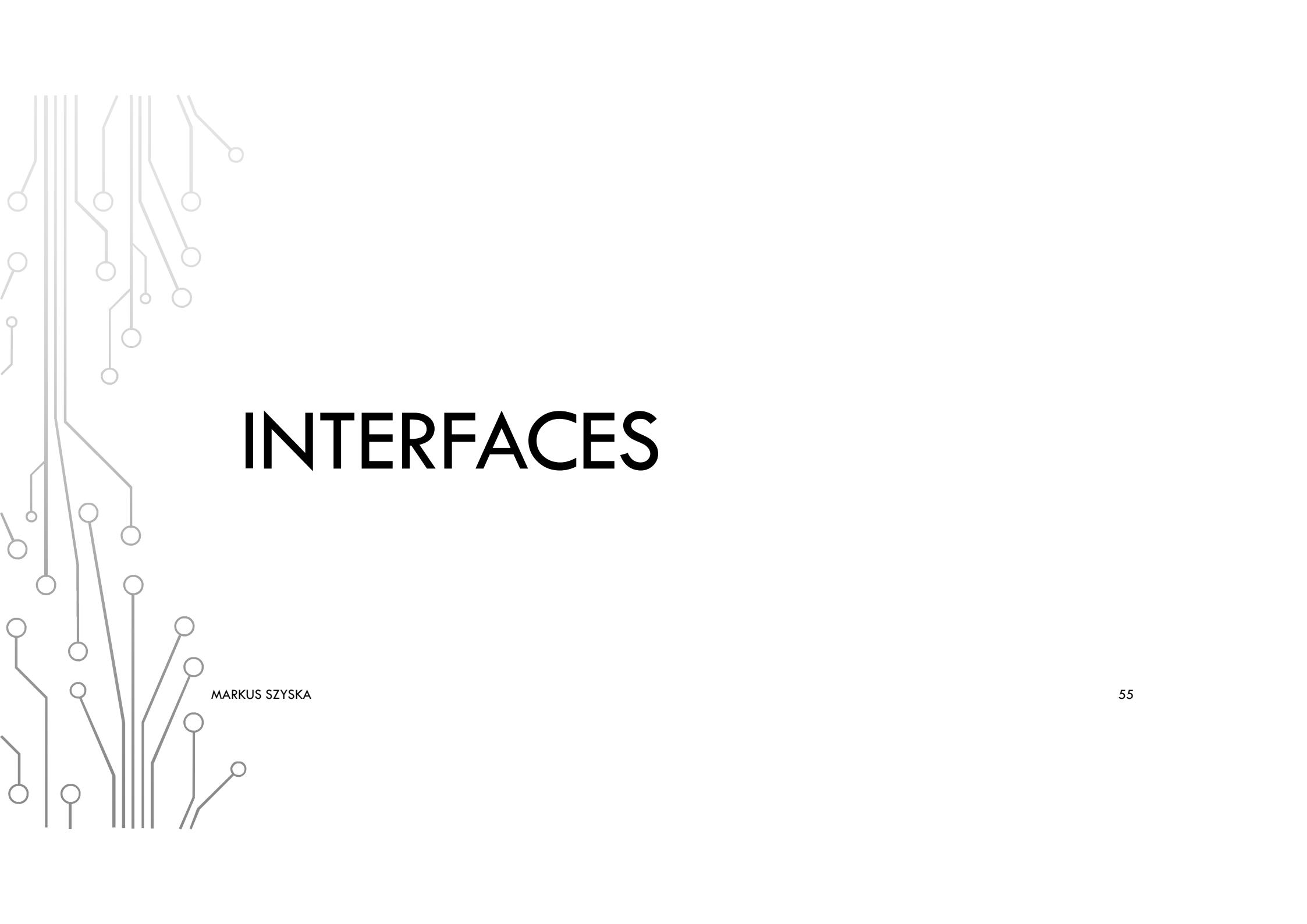
Template – Generics in Java

```
public static void main(String[] args)
{
    A Schmuckstueck s1 = new Schmuckstueck("Ohrring");
    Schmuckstueck s2 = new Schmuckstueck("Halskette");
    Schmuckstueck s3 = new Schmuckstueck("Diamant-Ring");

    B Kollektion<Schmuckstueck> schmuckKollektion =
        new Kollektion<Schmuckstueck>();

    C schmuckKollektion.hinzufuegen(s1);
    schmuckKollektion.hinzufuegen(s2);
    schmuckKollektion.hinzufuegen(s3);

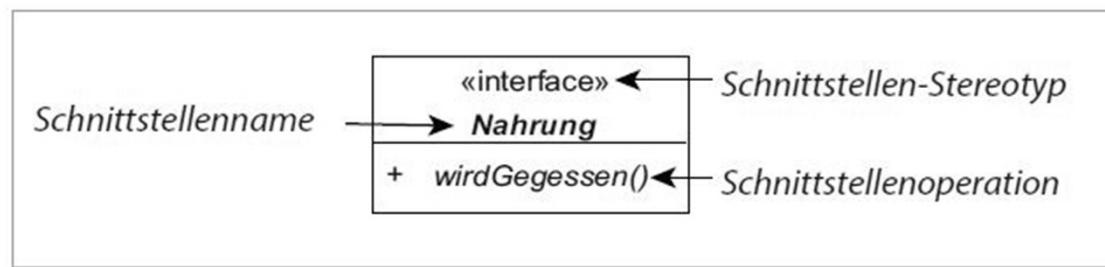
    D schmuckKollektion.praesentieren();
}
```



INTERFACES

MARKUS SZYNSKA

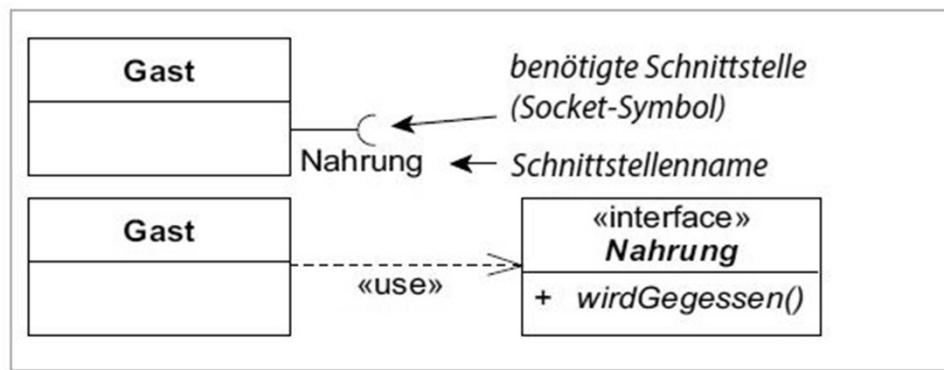
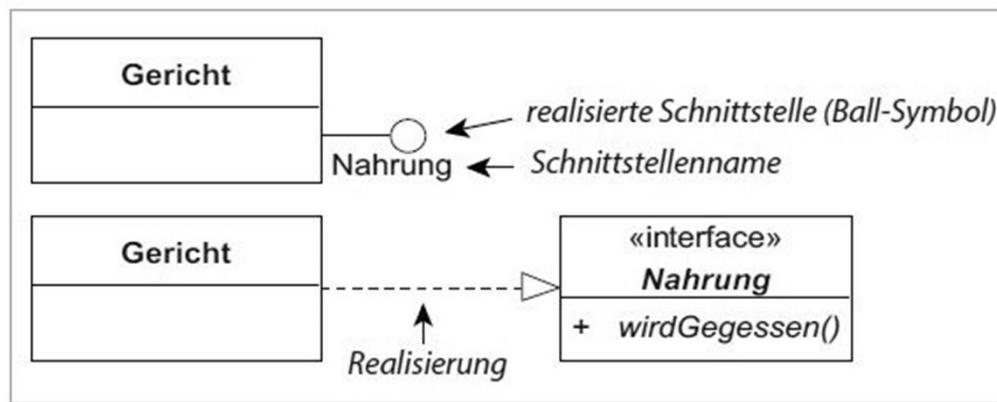
Interfaces



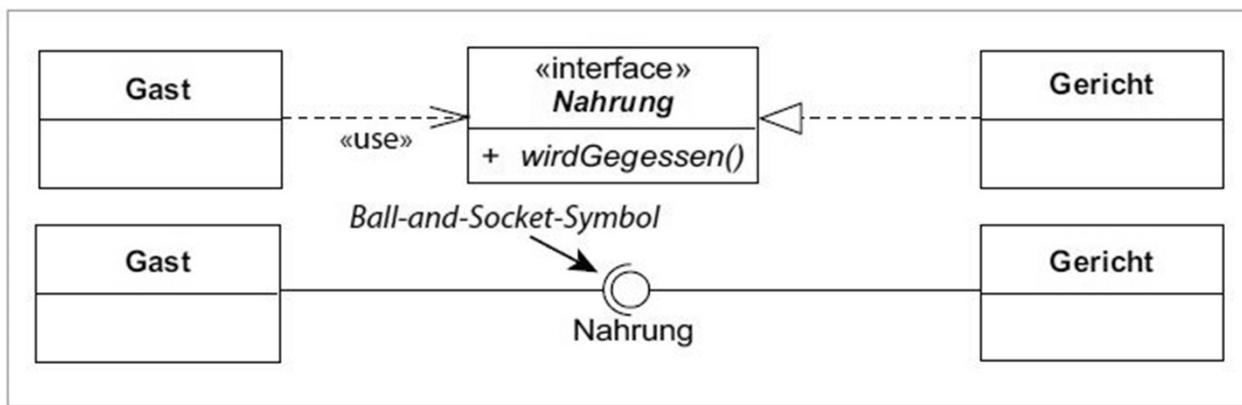
MARKUS SZYNSKA

56

Interfaces



Interfaces



Interfaces



Interfaces Beispiel Java

```
A public interface Nahrung
{
    B   public void wirdGegessen();
}
```

```
A class Gericht implements Nahrung
{
    public String name;
    public Gericht(String n)
    {
        name = n;
    }

    B   public void wirdGegessen()
    {
        System.out.println("Gericht "+name+" wird gegessen");
    }
}
```

Interfaces Beispiel Java

```
class Gast
{
    A public Nahrung mittagessen;

    public Gast()
    {

        B   mittagessen = new Gericht("Wiener-Schnitzel");

    }
}
```

```
public static void main(String[] args)
{
    A   Gast gast = new Gast();
        gast.mittagessen.wirdGegessen();

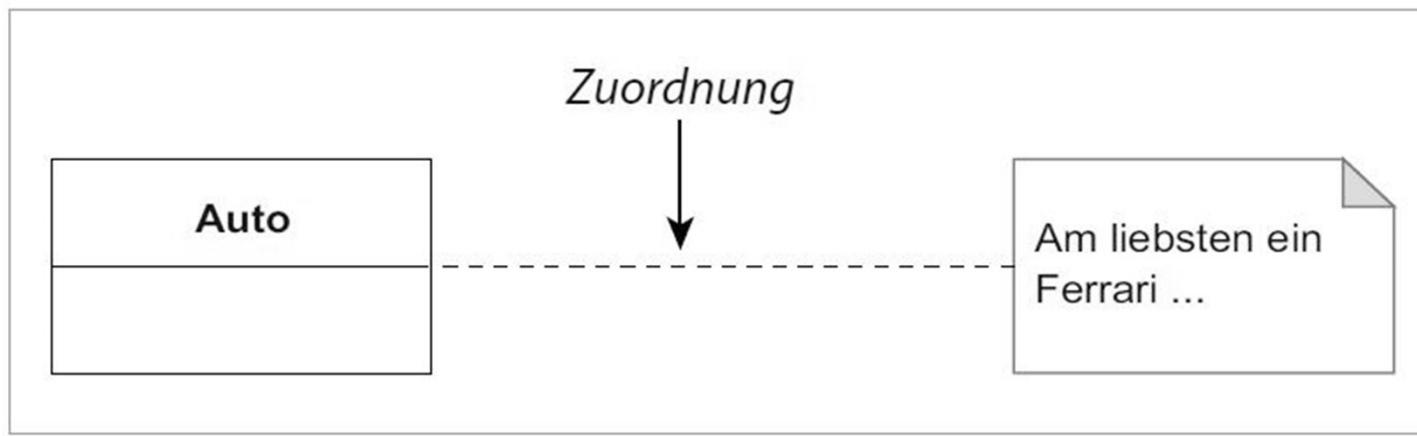
}
```



ANMERKUNGEN

MARKUS SZYNSKA

Anmerkungen



MARKUS SZYNSKA

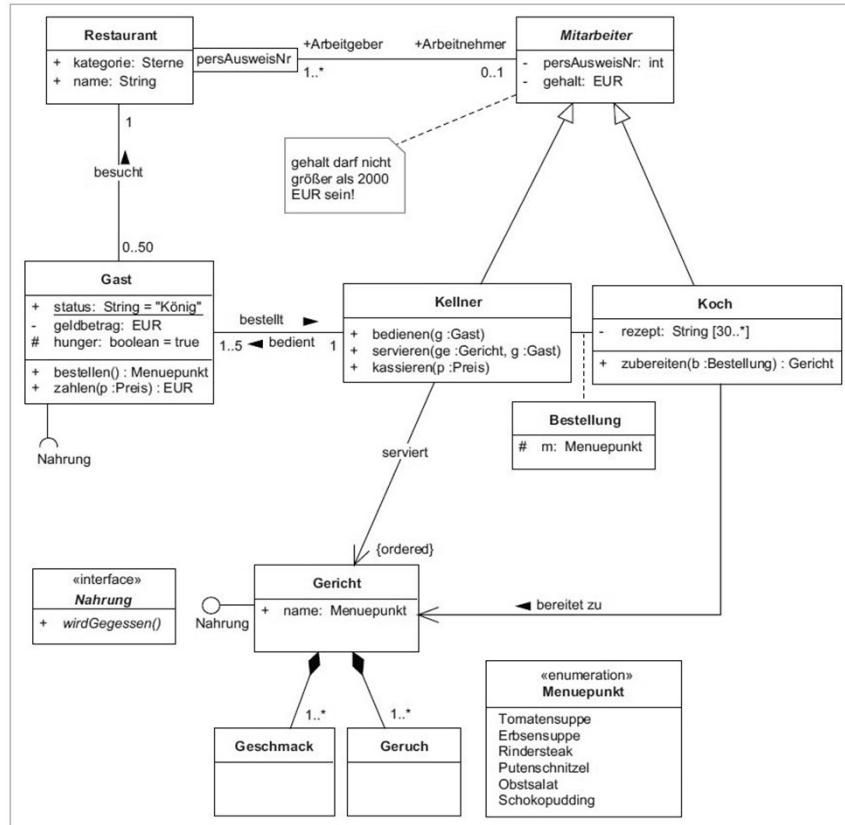
63



LESEN EINES KLASSENDIAGRAMMS

MARKUS SZYSCA

Lesen eines Klassendiagramms



MARKUS SZYNSKA

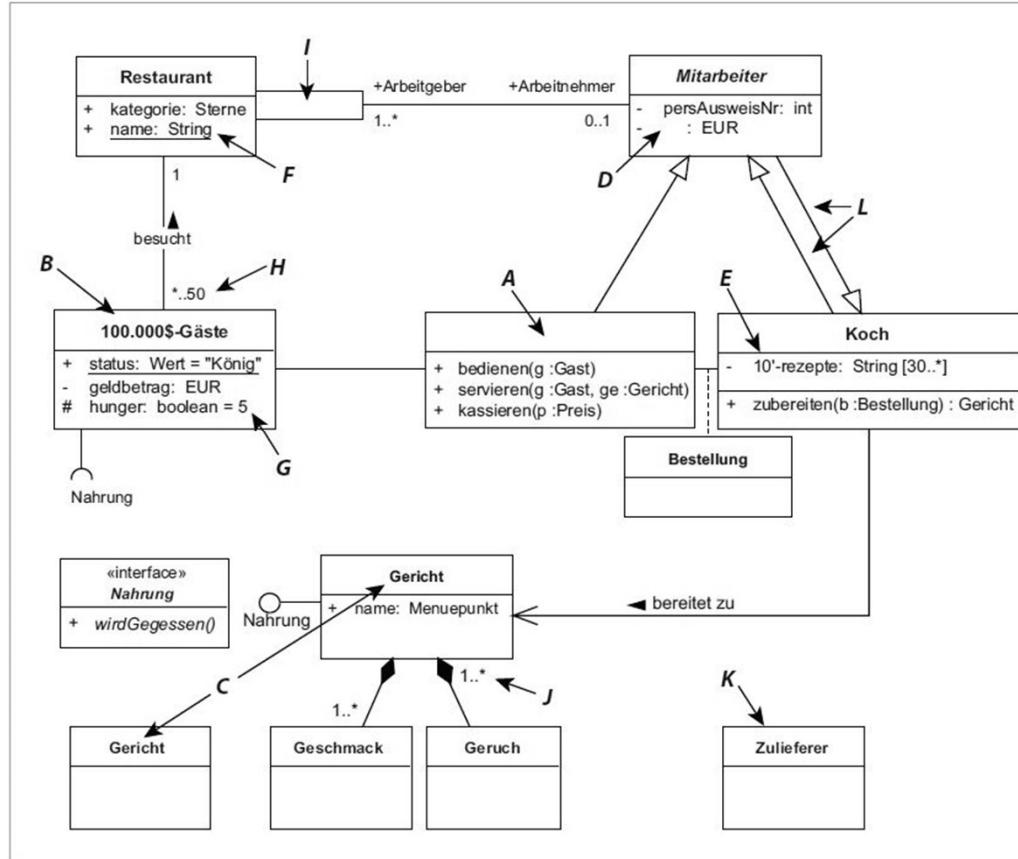
65



IRRUNGEN UND WIRRUNGEN

MARKUS SZYNSKA

Irrungen und Wirrungen



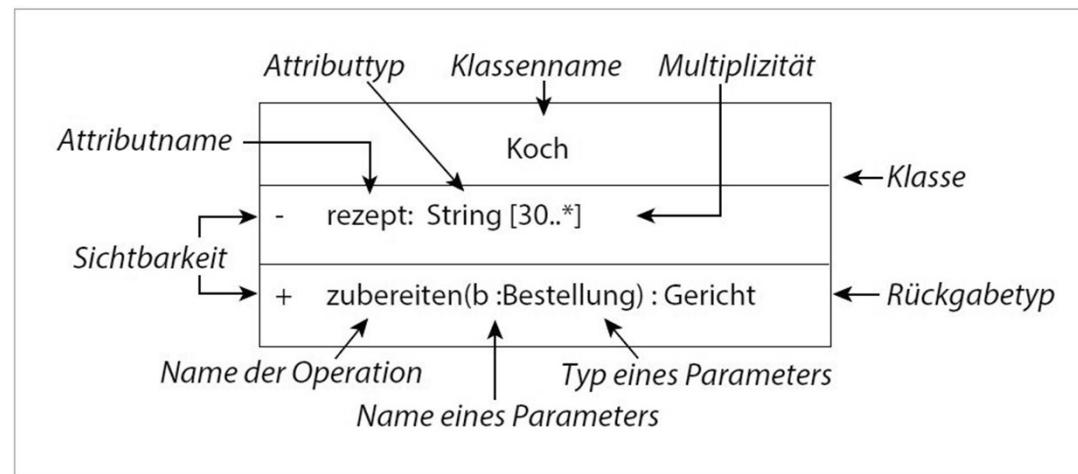
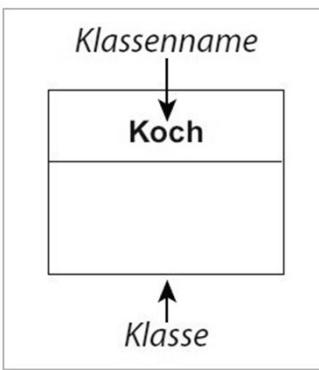
MARKUS SZYSKA



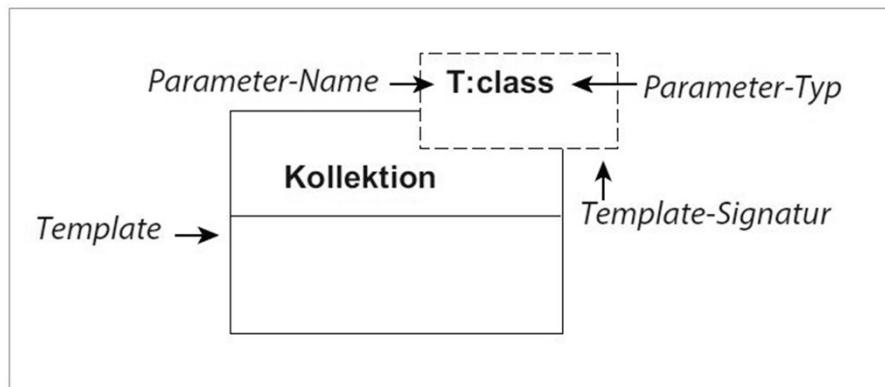
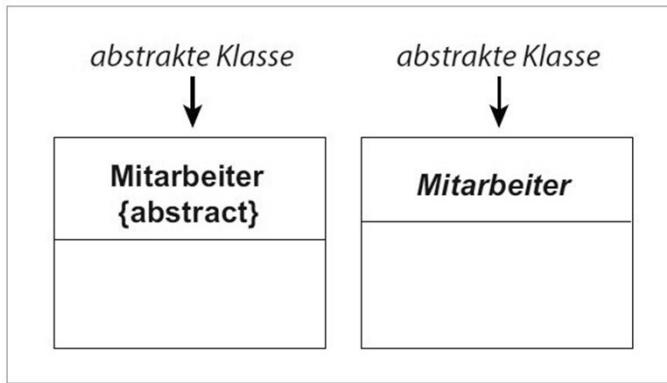
ZUSAMMENFASSUNG

MARKUS SZYNSKA

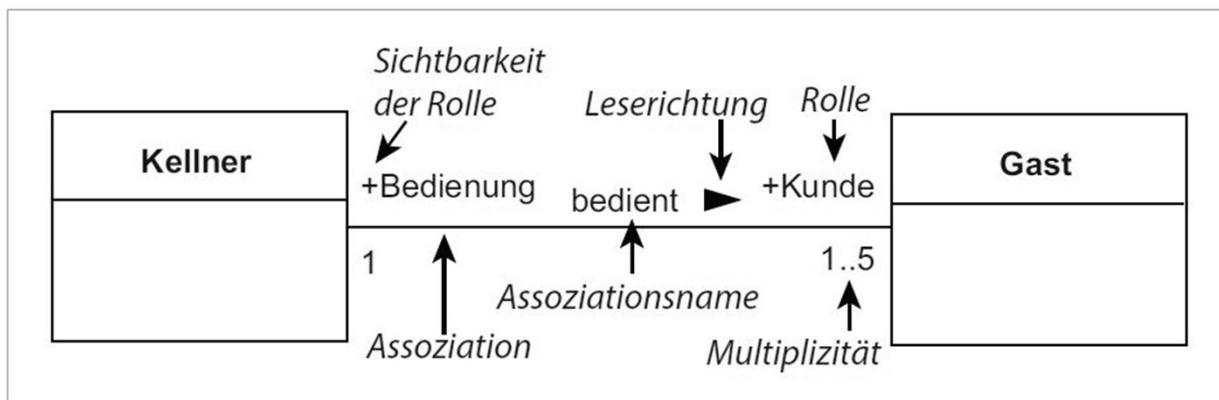
Zusammenfassung



Zusammenfassung

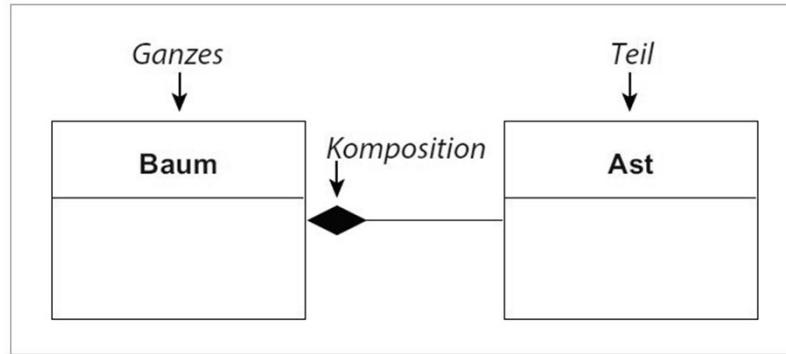
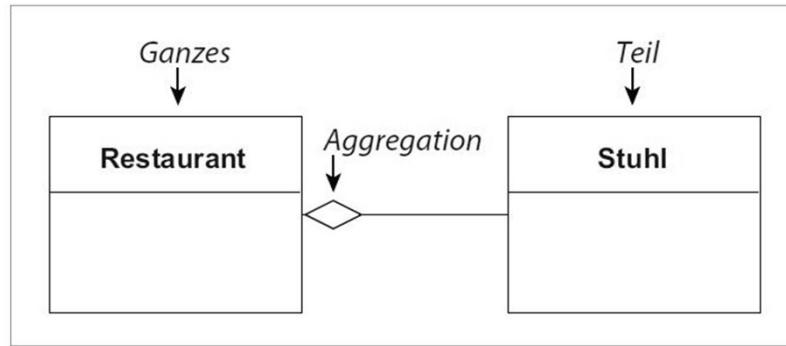


Zusammenfassung

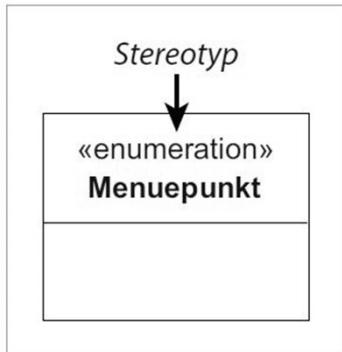
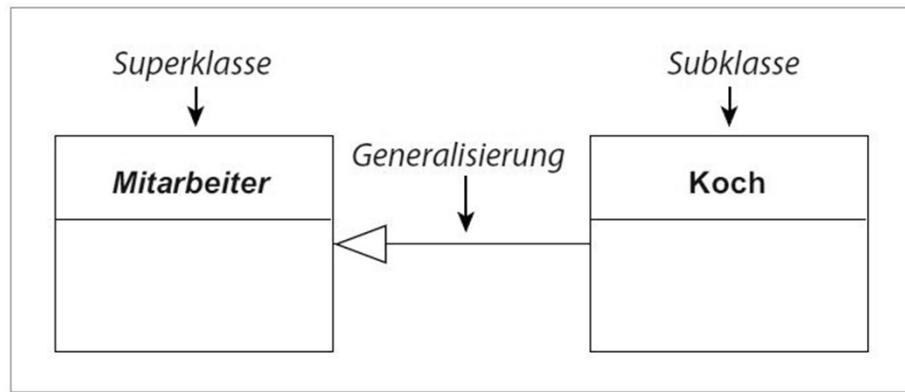


MARKUS SZYNSKA

Zusammenfassung



Zusammenfassung



Zusammenfassung

