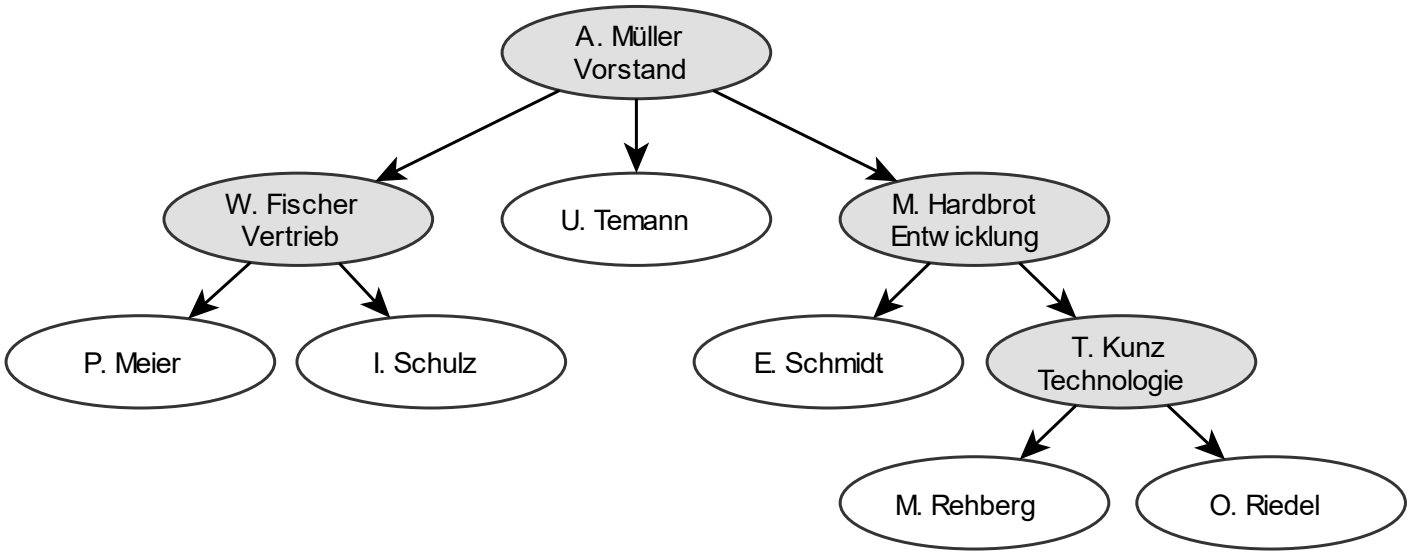
TBZ
Technische Berufsschule Zürich

# Modul 326
# Composite Design Pattern

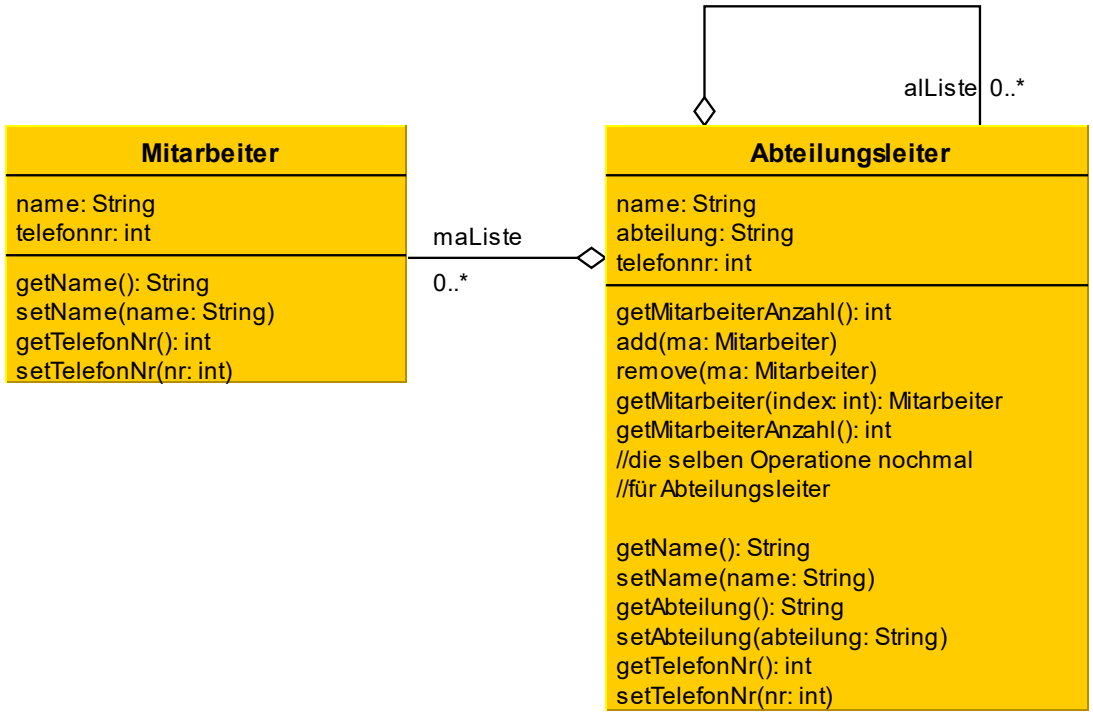
D. A. Waldvogel

12.03.2018

# Abteilungshierarchie



# Erster naiver Entwurf



# was ist hier schlecht ?

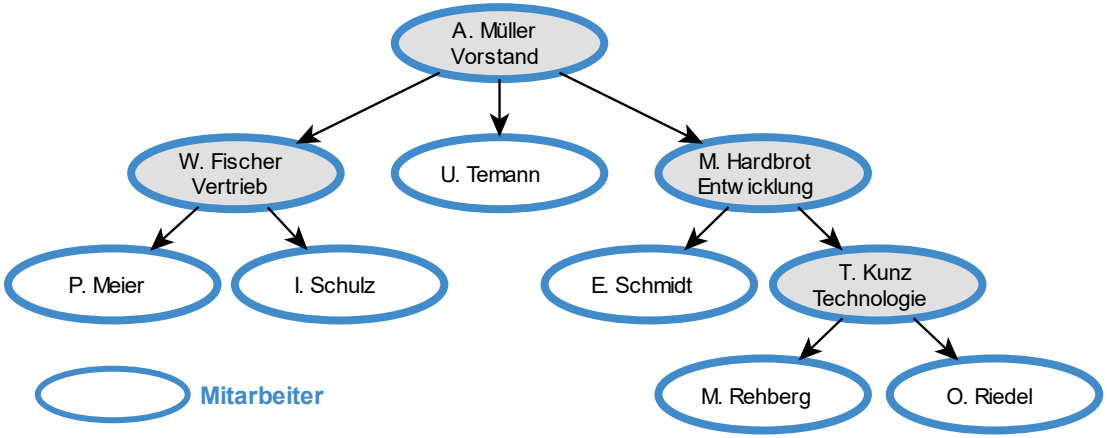
Technische Berufsschule Zürich

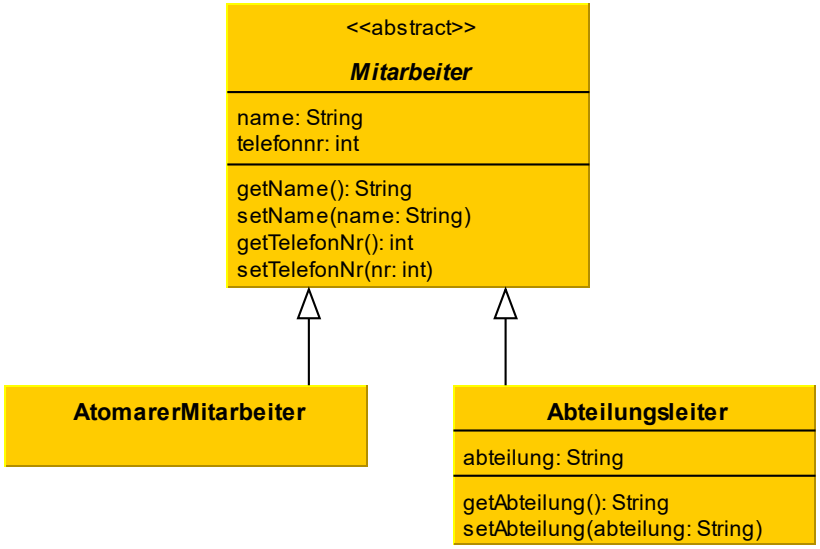
```
private void print(Abteilungsleiter leiter) {
    for(int i= 0; i < leiter.getMitarbeiterAnzahl(), i++){
        Mitarbeiter ma = leiter.getMitarbeiter(i);
        System.out.println(ma.getName()+ ". Abteilung: "+ma.getAbteilung());
    }

    for(int i= 0; i < leiter.getAbteilungsleiterAnzahl(), i++){
        Abteilungsleiter al = leiter.getAbteilungsleiter(i);
        System.out.println(al.getName()+" ist Leiter von "+al.getAbteilung());
        print(al);//Rekursiver Aufruf
    }
}
```

# Antwort ?

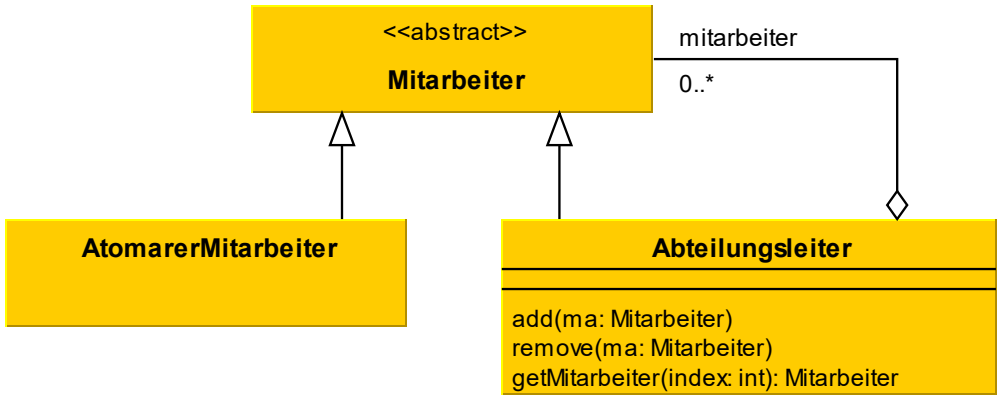
- extrem fehleranfällig
- unschön
- unflexibel

# Vererbung


A. Müller
Vorstand
W. Fischer
Vertrieb
U. Temann
M. Hardbrot
Entw icklung
P. Meier
I. Schulz
E. Schmidt
T. Kunz
Technologie
M. Rehberg
O. Riedel
Mitarbeiter

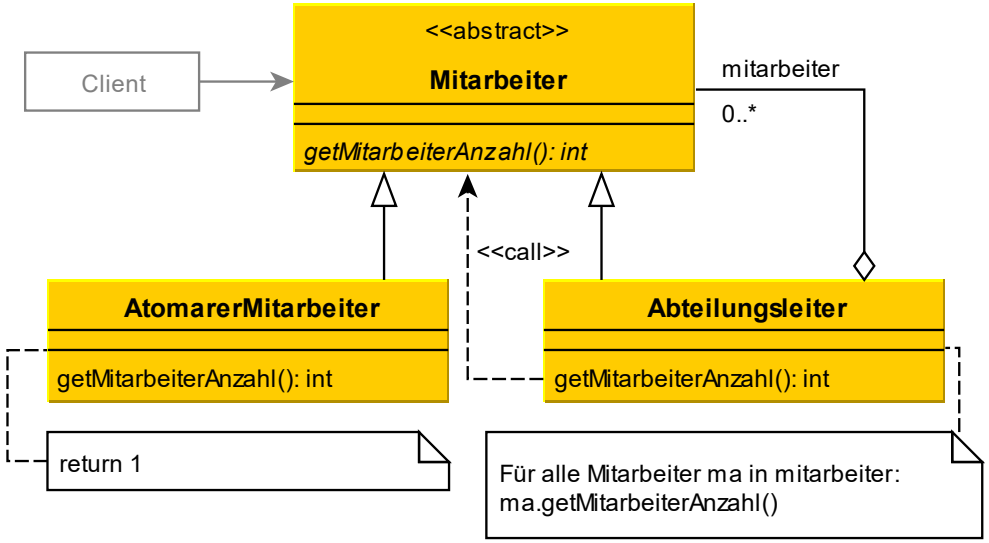

<<abstract>>
Mitarbeiter
name: String
telefonnr: int
getName(): String
setName(name: String)
getTelefonNr(): int
setTelefonNr(nr: int)
AtomarerMitarbeiter
Abteilungsleiter
abteilung: String
getAbteilung(): String
setAbteilung(abteilung: String)

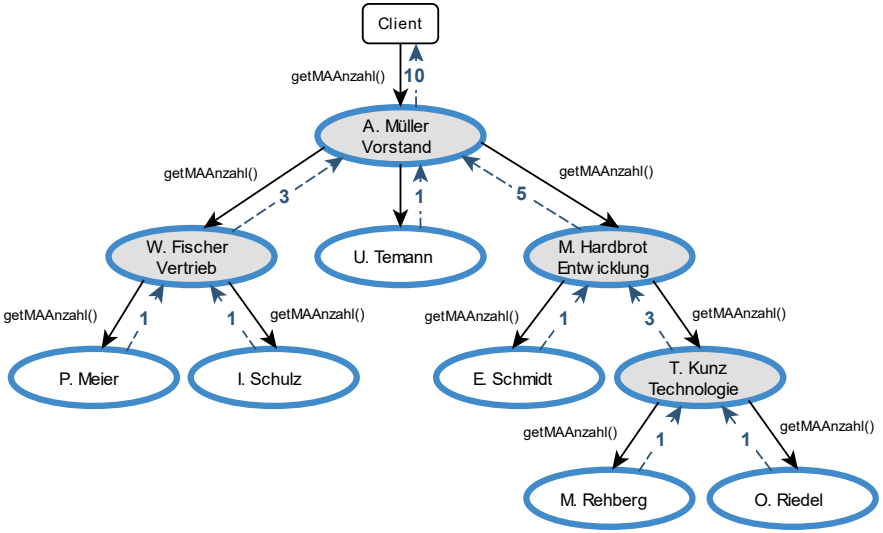
# Realisierung




```
class Abteilungsleiter extends Mitarbeiter{

    private List<Mitarbeiter> mitarbeiter = new ArrayList<Mitarbeiter>();

    public void add(Mitarbeiter ma){
        mitarbeiter.add(ma);
    }

    public void remove(Mitarbeiter ma){
        mitarbeiter.remove(ma);
    }

    public Mitarbeiter getMitarbeiter(int index){
        return mitarbeiter.get(index);
    }
}
```

# Realisierung



# getMitarbeiterAnzahl()

Technische Berufsschule Zürich

```
abstract class Mitarbeiter{
    public abstract int getMitarbeiterAnzahl();
}

class AtomarerMitarbeiter extends Mitarbeiter{
    public int getMitarbeiterAnzahl() {
        return 1;
    }
}

class Abteilungsleiter extends Mitarbeiter{

    private List<Mitarbeiter> mitarbeiter = new ArrayList<Mitarbeiter>();

    public int getMitarbeiterAnzahl() {
        int summe = 1; //1 für sich selbst
        for (Mitarbeiter ma : mitarbeiter) {
            summe += ma.getMitarbeiterAnzahl();
        }
        return summe;
    }

    //Verwaltungsmethoden...
}
```

# Client

- Mitarbeiterzahl mit einem Aufruf berechnen

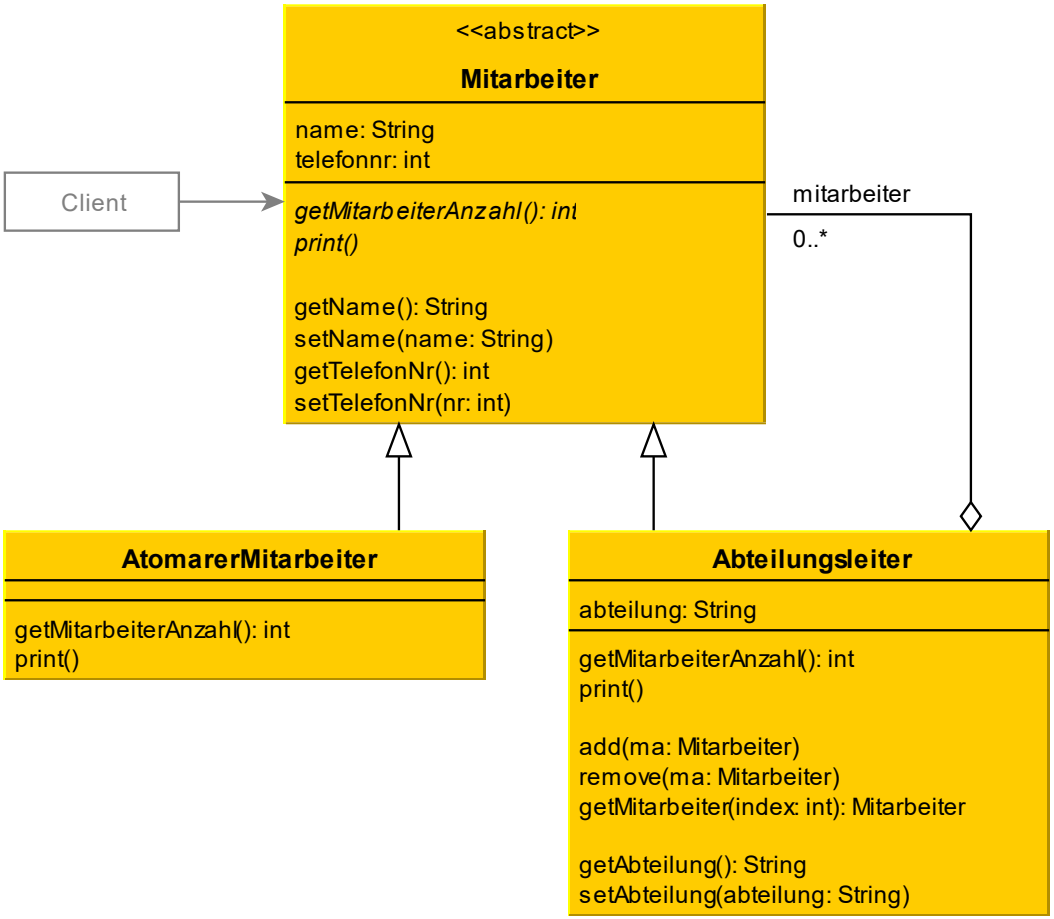
```
public static void main(String[] args) {
    Mitarbeiter vorstand = new Abteilungsleiter();
    //Mitarbeiterhierarchie unter vorstand aufbauen (add())

    //Mitarbeiterhierarchie nutzen
    System.out.println(vorstand.getMitarbeiterAnzahl());
}
```

# Auftrag

TBZ Technische Berufsschule Zürich

- Erstellen Sie folgende Klassen
  - Mitarbeiter (abstract)
  - AtomarerMitarbeiter abgeleitet von Mitarbeiter
  - Abteilungsleiter abgeleitet von Mitarbeiter
  - Testclient
- Bauen Sie die Hierarchie gemäss Folie 2

# Auftrag


<<abstract>>
Mitarbeiter
name: String
telefonnr: int
getMitarbeiterAnzahl(): int
print()
getName(): String
setName(name: String)
getTelefonNr(): int
setTelefonNr(nr: int)
Client
mitarbeiter
0..*
AtomarerMitarbeiter
getMitarbeiterAnzahl(): int
print()
Abteilungsleiter
abteilung: String
getMitarbeiterAnzahl(): int
print()
add(ma: Mitarbeiter)
remove(ma: Mitarbeiter)
getMitarbeiter(index: int): Mitarbeiter
getAbteilung(): String
setAbteilung(abteilung: String)

# Resultat

```
Abteilungsleiter A. Müller (Vorstand). Tel: 4
      Abteilungsleiter W. Fischer (Vertrieb). Tel: 1
            P. Meier. Tel: 123
            I. Schulz. Tel: 112
      U. Temann. Tel: 442
      Abteilungsleiter M. Hardbrot (Entwicklung). Tel: 3
            M. Rehberg. Tel: 323
            Abteilungsleiter T. Kunz (Technologie). Tel: 2
                  M. Rehberg. Tel: 223
                  O. Riedel. Tel: 212
```

# Lösung

Technische Berufsschule Zürich

```
public static void main(String[] args) {
    /*
     * Firmenhierarchie einmalig aufbauen
     */
    //Vertrieb
    Abteilungsleiter al1 = new Abteilungsleiter("W. Fischer", "Vertrieb", 001);
    al1.add(new AtomarerMitarbeiter("P. Meier", 123));
    al1.add(new AtomarerMitarbeiter("I. Schulz", 112));
    //Technologie
    Abteilungsleiter al2 = new Abteilungsleiter("T. Kunz", "Technologie", 002);
    al2.add(new AtomarerMitarbeiter("M. Rehberg", 223));
    al2.add(new AtomarerMitarbeiter("O. Riedel", 212));
    //Entwicklung
    Abteilungsleiter al3 = new Abteilungsleiter("M. Hardbrot", "Entwicklung", 003);
    al3.add(new AtomarerMitarbeiter("M. Rehberg", 323));
    al3.add(al2);
    //Vorstand
    Abteilungsleiter vorstand = new Abteilungsleiter("A. Müller", "Vorstand", 004);
    vorstand.add(al1);
    vorstand.add(new AtomarerMitarbeiter("U. Temann", 442));
    vorstand.add(al3);

    /*
     * Firmenhierarchie ausdrucken
     */
    vorstand.print("");
}
```

# Fragen?

# Quelle

- https://www.philipphauer.de/study/se/design-pattern/composite.php