

Composite

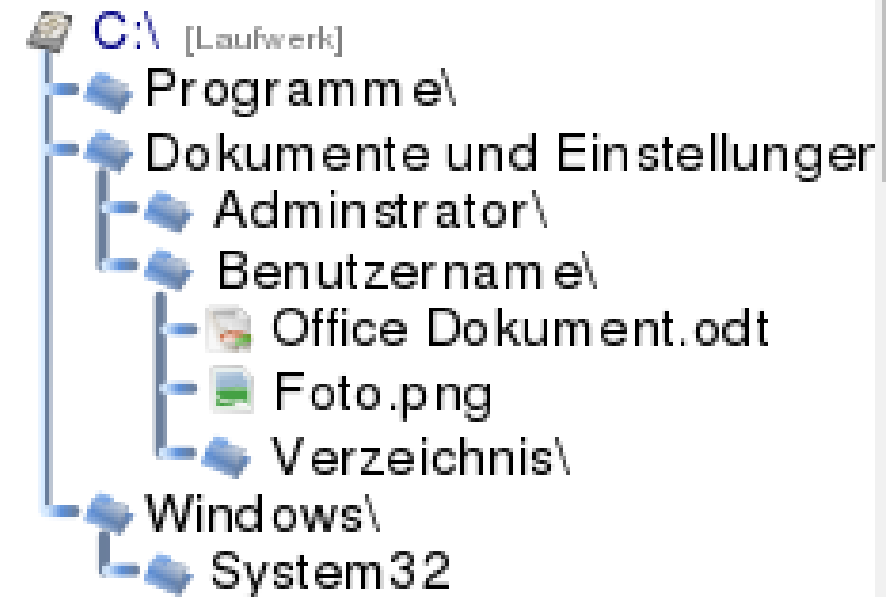
Bastian Schwaighofer

Allgemeines

- Strukturmuster

→ Erleichtert Softwareentwurf durch Beziehungen

- Gern Verwendet für Teil-Ganzes- Hierarchien
 - Objekte werden in Baumstruktur gespeichert

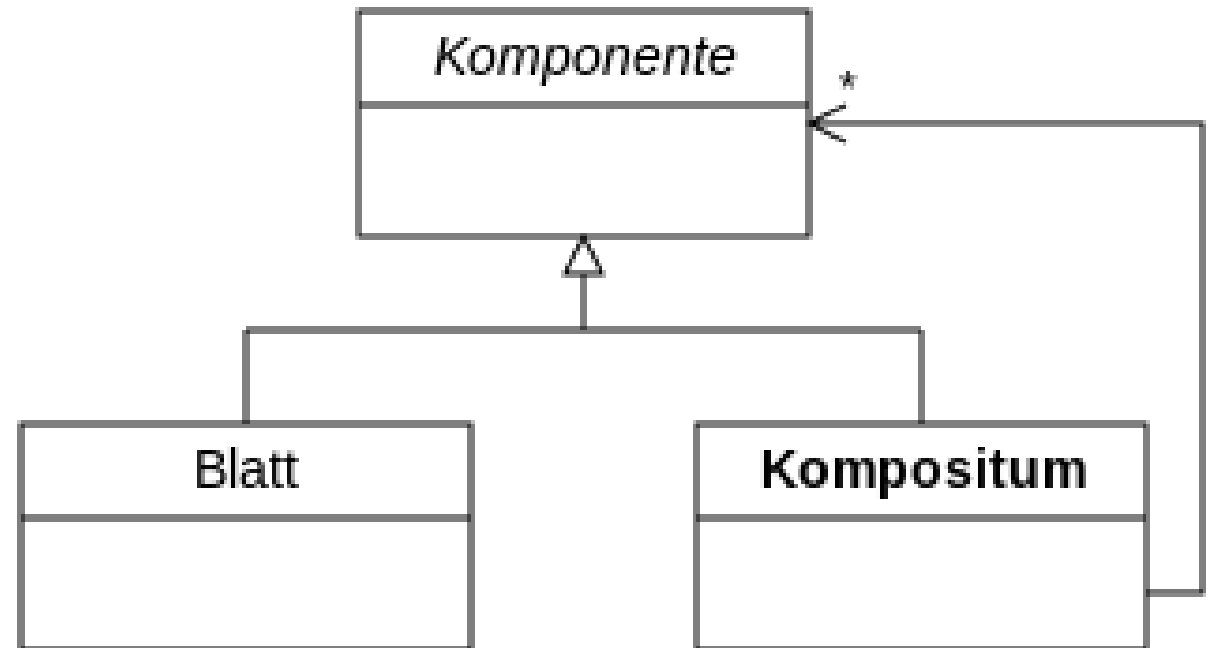


E:\ [Laufwerk]

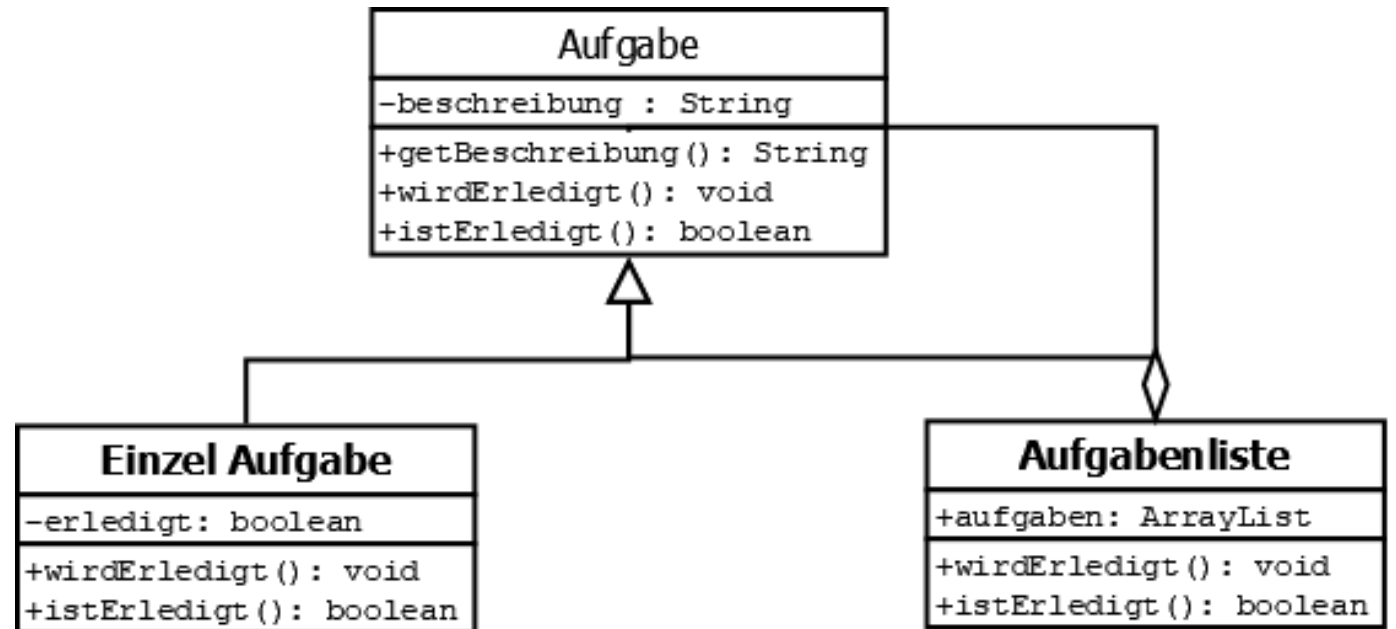


Unterteilung

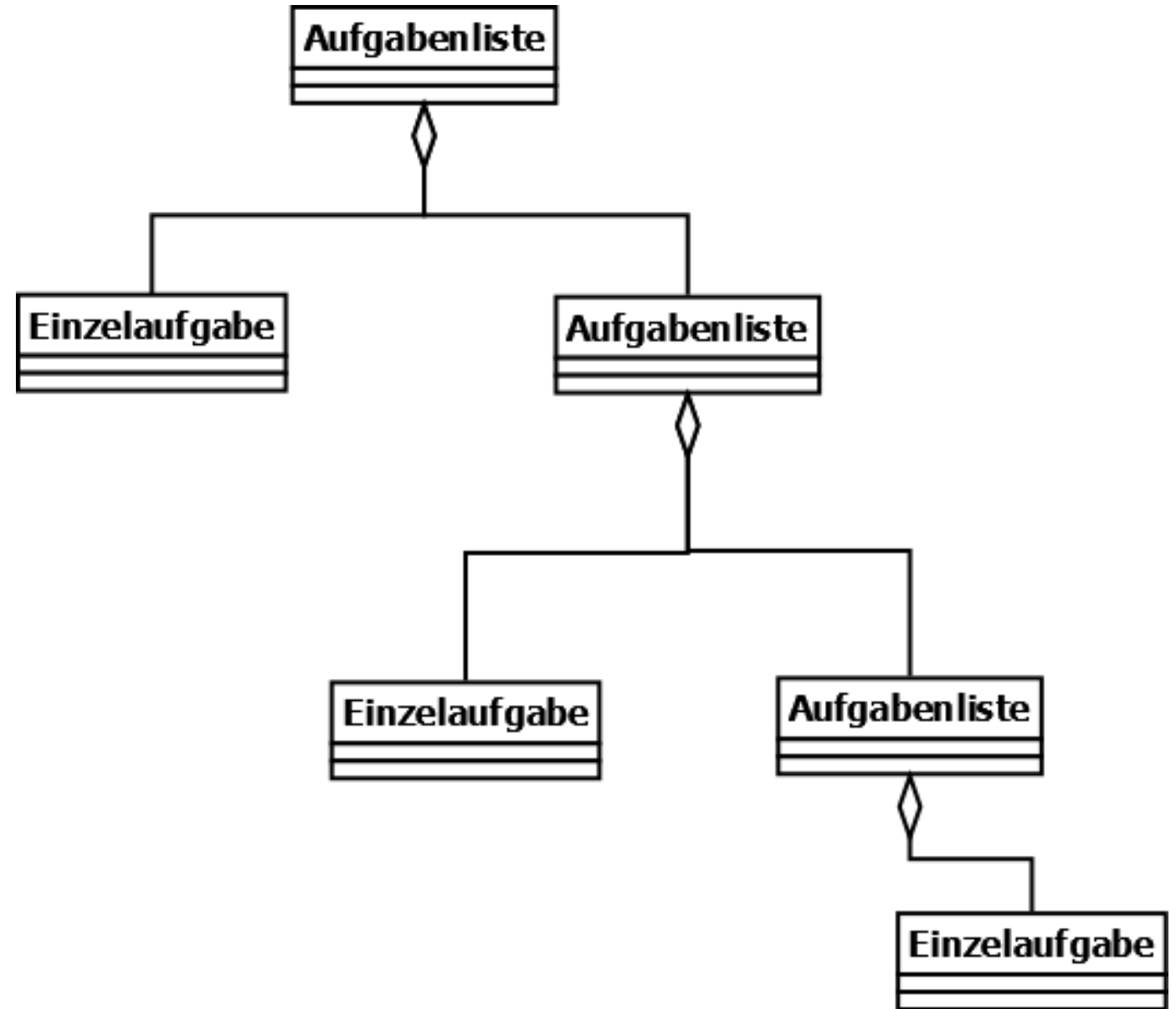
- Komponente
 - Abstrakte Basis
- Kompositum
 - Enthält Blätter oder Komposita
- Blatt
 - Einzelobjekt



Aufgabenmanager



Aufbau



SOLID

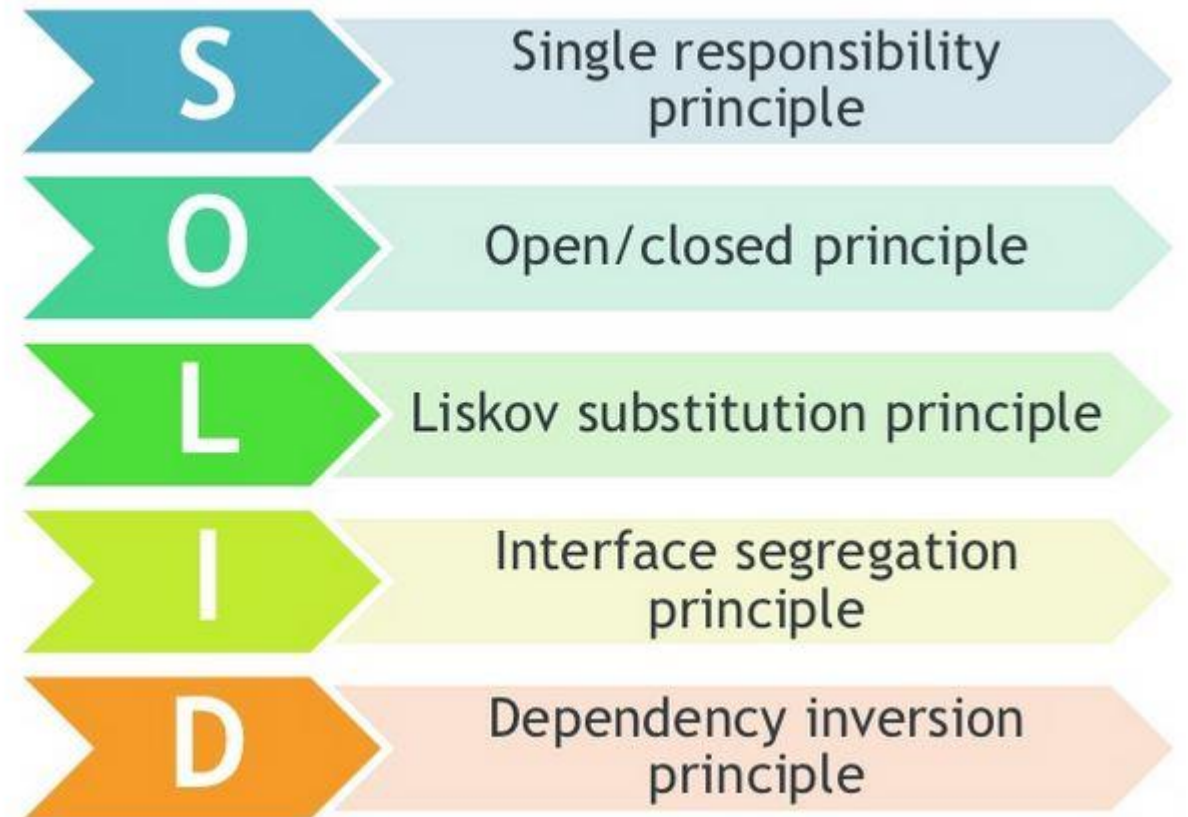
S : Aufgabe kennt nur eigenen Status (erledigt / nicht erledigt)

O: Leicht erweiterbar (z.B.: Terminierte Aufgabe)

L: nein da Add nur bei Liste

I: Aufgabe wird von allen benutzt

D: Abhängigkeit von Abstrakter Aufgabe



Code Beispiel

```
public AufgabenListe(String beschreibung) {  
    super(beschreibung);  
    aufgaben = new ArrayList<>();  
}  
@Override  
public void wirdErledigt()  
{  
    for(Aufgabe aufgabe: aufgaben)  
    {  
        aufgabe.wirdErledigt();  
    }  
}  
@Override  
public boolean istErledigt()  
{  
    for(Aufgabe aufgabe: aufgaben)  
    {  
        if (!aufgabe.istErledigt()) {  
            return false;  
        }  
    }  
    return true;  
}
```

Anwendung in Java

AWT: java.awt.Component

