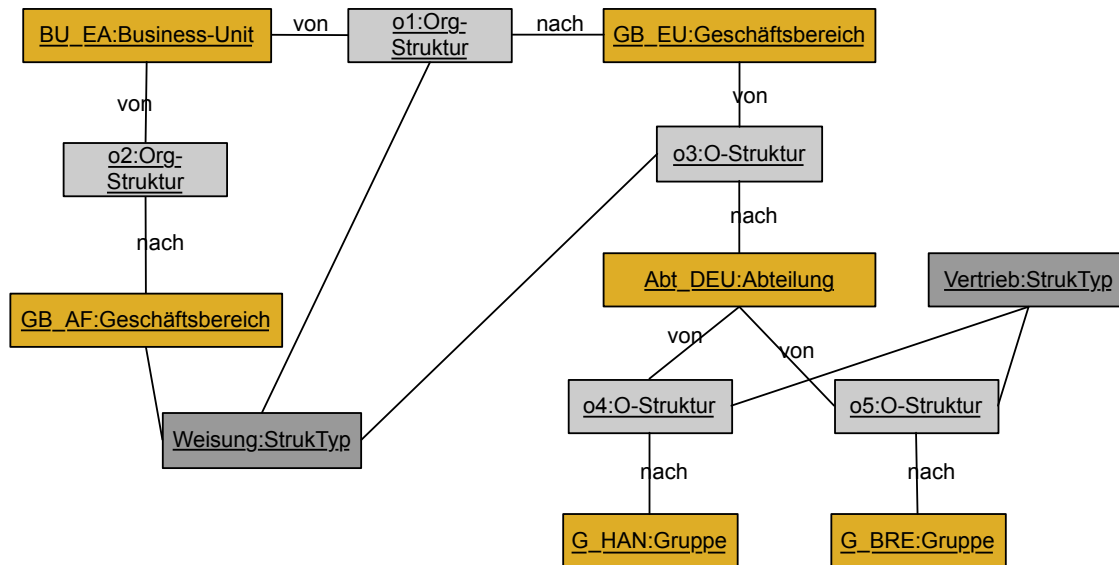


## Software Engineering 2 Übungsblatt 1

### - Analyse-Pattern -

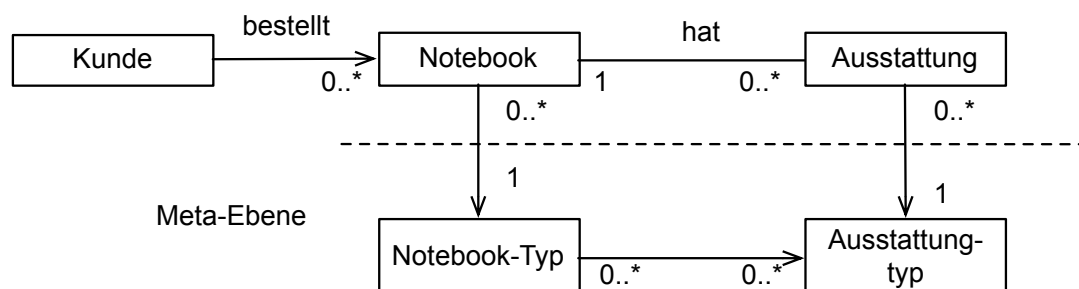
#### Aufgabe 1: Organisationshierarchie-Pattern

In Kapitel 2.3 wurde das Organisationshierarchie-Pattern anhand der Modellierung der Organisationsstruktur eines Unternehmens eingeführt. ...



#### Aufgabe 2: Wissens-Ebene-Pattern (Pflichtaufgabe)

Entwerfen Sie ein Klassendiagramm für ...



### Aufgabe 3: Implementierung des Wissens-Ebene-Pattern (Pflichtaufgabe)

Erweitern Sie Ihr Klassenmodell aus Aufgabe 2 um die notwendigen Methoden zum Pflegen der Beziehungen. Programmieren Sie die Klassen aus und schreiben Sie ein Hauptprogramm, das Ihre Klassen testet und ....

```
import java.util.ArrayList;

public class Notebook {

    private ArrayList<Equipment> vEquipment;
    private N_Type n_type;
    private String name;

    public Notebook( String name, N_Type n ) {
        vEquipment = new ArrayList<Equipment>();
        this.name = name;
        this.n_type = n;
    }

    public String toString() {
        return "\n--- Notebook ["+getName()+"] Ausstattung: "+vEquipment;
    }

    public String getName(){ return name; }
    public void setName(String name){ this.name = name; }

    public N_Type getN_Type(){ return n_type; }

    public void assign(Equipment equip) {
        if( this.getN_Type().isallowed( equip.getE_Type() ) ) {
            vEquipment.add( equip );
            System.out.println();
        }
    }
}
```

```
import java.util.ArrayList;

public class N_Type {

    private String name;
    private ArrayList<E_Type> vE_Types;

    public N_Type( String name ) {
        vE_Types = new ArrayList<E_Type>();
        this.name = name;
    }

    // Methode um die Ausstattungs-Typen zu liefern
    public ArrayList getE_Types() {
        return vE_Types;
    }

    // Methode um die Ausstattungs-Typen zuzuweisen
    public void addE_Type(E_Type e_type) {
        vE_Types.add( e_type );
    }

    // Pruefung, ob Ausstattungstype erlaubt ist
    public boolean isallowed(E_Type e_type) {
        return vE_Types.contains(e_type);
    }
}
```

```
public class Test {  
  
    public static void main( String[] args ) {  
        // Equipment-Typen  
        E_Type eT1 = new E_Type("Apple_BatterieTyp_A");  
        E_Type eT2 = new E_Type("Apple_BatterieTyp_B");  
        E_Type eT3 = new E_Type("Apple_BatterieTyp_C");  
        E_Type eT4 = new E_Type("IBM_BatterieTyp");  
        // Equipment  
        Equipment e1 = new Equipment("A_Bat_1", eT1);  
        Equipment e2 = new Equipment("A_Bat_2", eT2);  
        Equipment e4 = new Equipment("Lenovo_Bat_1", eT4);  
  
        // Notebook-Typen  
        N_Type nT1 = new N_Type("MacBookPro");  
        // Notebook  
        Notebook nb1 = new Notebook("NB1", nT1);  
  
        // Ausstattungstypen den Notebooktypen zuordnen  
        nT1.addE_Type( eT1 );  
        nT1.addE_Type( eT2 );  
        nT1.addE_Type( eT3 );  
  
        // Ausstattungen einem Notebook zuweisen  
        nb1.assign(e1);  
        nb1.assign(e4);  
        nb1.assign(e2);  
    }  
}
```