

## Was bedeutet Information Expert?

**In einfachen Wörtern => gib jede Klasse ihre genauere Aufgabe!**

**Information Expert** ist ein **Entwurfsmuster** bzw. ein **Prinzip** in der objektorientierten Softwareentwicklung, das im Rahmen von **GRASP (General Responsibility Assignment Software Patterns)** eingeführt wurde. Dieses Prinzip hilft dabei, die **Verantwortlichkeiten** (also Methoden und Funktionen) auf die richtigen Klassen in einem System zu verteilen. Es besagt, dass diejenige Klasse oder dasjenige Objekt, das über die **notwendigen Informationen** verfügt, auch die Verantwortung für die **Operation** oder Methode haben sollte, die diese Informationen verarbeitet.

## Was bedeutet Information Expert?

Das **Information-Expert-Prinzip** besagt: **Weise die Verantwortung für eine Aufgabe der Klasse zu, die die benötigten Informationen hat, um die Aufgabe erfolgreich durchzuführen.** Mit anderen Worten: Diejenige Klasse, die über die **notigen Daten** verfügt, sollte auch für die **Verarbeitung dieser Daten** verantwortlich sein.

### Beispiel:

Stell dir vor, du hast ein **Bestellsystem**, in dem es eine Klasse **Bestellung** gibt, die mehrere **Artikel** enthält. Nun soll der **Gesamtpreis** der Bestellung berechnet werden.

Gemäß dem Information-Expert-Prinzip wird die Methode zur Berechnung des Gesamtpreises (z.B. `berechneGesamtpreis()`) der Klasse **Bestellung** zugewiesen, weil die Klasse **Bestellung** die Informationen über alle enthaltenen Artikel hat.

```
class Artikel {
    private double preis;

    public double getPreis() {
        return preis;
    }
}

class Bestellung {
    private List<Artikel> artikelListe;

    public double berechneGesamtpreis() {
        double gesamtpreis = 0;
        for (Artikel artikel : artikelListe) {
            gesamtpreis += artikel.getPreis();
        }
        return gesamtpreis;
    }
}
```

- **Bestellung** ist der **Information Expert** für die Berechnung des Gesamtpreises, weil sie die Liste der Artikel enthält und somit die notwendigen Informationen hat.

## Warum ist das Information-Expert-Prinzip wichtig?

1. **Verantwortlichkeiten sinnvoll verteilen:** Das Prinzip hilft dabei, die Aufgaben in einem System auf die Klassen zu verteilen, die am besten geeignet sind, diese Aufgaben zu erfüllen. Dadurch wird das System besser strukturiert.
2. **Hohe Kohäsion und niedrige Kopplung:** Wenn jede Klasse die Verantwortung für ihre eigenen Daten und Aufgaben hat, führt dies oft zu **hoher Kohäsion** (weil die Klasse fokussiert bleibt) und **niedriger Kopplung** (weil Klassen nur ihre eigenen Daten bearbeiten und nicht die anderer Klassen).
3. **Verständlicher und wartbarer Code:** Wenn die Klassen genau die Aufgaben erfüllen, die zu ihren Daten passen, wird der Code **logischer** und **leichter zu verstehen**. Es ist einfacher, Methoden zu finden und zu ändern, weil klar ist, wo welche Funktionalität zu finden ist.

## Was passiert, wenn man das Prinzip nicht befolgt?

Wenn das Information-Expert-Prinzip ignoriert wird, kann es zu Problemen kommen wie:

- **Unnötige Abhängigkeiten:** Klassen greifen auf Daten anderer Klassen zu, was die **Kopplung** erhöht.
- **Verwirrung:** Die Zuständigkeiten der Klassen werden unscharf, und es wird schwerer, zu verstehen, welche Klasse welche Aufgabe erfüllt.
- **Schwierige Wartung:** Wenn Verantwortlichkeiten unklar verteilt sind, wird es schwieriger, den Code zu erweitern oder zu ändern, ohne Fehler zu verursachen.

## Fazit:

Das **Information-Expert-Prinzip** hilft dabei, die Verantwortlichkeiten in einem System klar und sinnvoll zu verteilen. Es besagt, dass die Klasse, die die nötigen Informationen hat, auch die Verantwortung für die Bearbeitung dieser Informationen tragen sollte. Das führt zu **besserem Design**, mit **hoher Kohäsion**, **niedriger Kopplung** und **leicht wartbarem Code**.