

The background of the slide is a complex, abstract network diagram. It consists of numerous nodes of varying sizes, some solid black, some solid blue, and some white with black outlines. These nodes are interconnected by a web of thin, light gray lines. The overall composition is dynamic and suggests a digital or data-driven environment. A large, solid black rectangular box is positioned in the lower right quadrant, serving as a container for the title and author information.

DATENKAPSELUNG

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

DATENKAPSELUNG

- Datenkapselung ist ein Prinzip der Objektorientierung
- Kombination von Daten (Feldern) und Methoden, die auf diese Daten zugreifen, in einer Klasse
- Verwendung von Zugriffsmodifizierern zur Kontrolle der Sichtbarkeit
- Hilft bei der Datenintegrität und verbessert die Codequalität
- Änderungen in der internen Implementierung haben keinen Einfluss auf andere Teile des Programms

SICHTBARKEIT

- **public**
 - Von anderen Objekten lesbar und beschreibbar
 - Methoden können von anderen Objekten aufgerufen werden
- **private**
 - Zugriff nur innerhalb der Klasse selbst
 - Attribute und Methoden können nicht von Objekten anderer Typen zugegriffen werden
- **Standard-Sichtbarkeit**
 - Wenn keine Sichtbarkeitsmodifizierer angegeben sind, ist die Standard-Sichtbarkeit private

SICHTBARKEIT

- **protected**
 - Zugriff nur innerhalb der Klasse selbst und in abgeleiteten Klassen möglich
 - Methoden und Attribute können von Unterklassen verwendet werden
- **internal**
 - Zugriff innerhalb desselben Assemblies möglich
 - Von Objekten innerhalb des Projekts (des Assemblies) lesbar und beschreibbar
- **protected internal**
 - Zugriff innerhalb des gleichen Assemblies und von abgeleiteten Klassen möglich
 - Kombination aus **protected** und **internal**
- **private protected**
 - Zugriff nur innerhalb der Klasse und von abgeleiteten Klassen im gleichen Assembly
 - Kombination aus **private** und **protected**

SICHTBARKEIT

- **Sichtbarkeiten** kontrollieren den Zugriff auf Attribute und Methoden einer Klasse
- Verhindert den direkten Zugriff und Veränderung von Daten durch externe Objekte
- Klare Trennung zwischen öffentlichen und privaten Schnittstellen

DATENKAPSELUNG

- Private Variablen innerhalb einer Klasse oder Struktur beginnen mit einem **Unterstrich** und werden **klein** geschrieben.

```
public class Person
{
    private string _name;
    private int _age;
}
```

GETTER UND SETTER METHODEN

- **Getter:** Methode, die den **aktuellen Wert** eines privaten Attributs zurückgibt
- **Setter:** Methode, die den **Wert eines privaten Attributs** setzt oder ändert
- Direkter Zugriff auf private Attribute wird verhindert
- Setter ermöglichen das Validieren oder Überprüfen von Werten vor dem Setzen (z.B. kein negatives Alter zulassen)
- Änderungen an internen Daten können kontrolliert durchgeführt werden, ohne den Rest des Codes zu beeinflussen

GETTER UND SETTER METHODEN

```
public class Person
{
    private int _age;

    public void SetAge(int newAge)
    {
        if(newAge >= 0)
        {
            _age = newAge;
        }
    }

    public int GetAge()
    {
        return _age;
    }
}
```