

# ESOP - Information Hiding

Assoc. Prof. Dr. Mathias Lux  
ITEC / AAU

# Geheimnisprinzip



In großen Softwaresystemen muss der globale Namensraum strukturiert und eingeschränkt werden: information hiding

- Unterscheidung zwischen öffentlichen und geheimen Bezeichnern.

# Beispiel



```
public class ShipExample {  
    // actual position of the ship  
    private int positionX, positionY;  
    // maximum number for x and y  
    private int maxX = 320, maxY = 640;
```



```
    public ShipExample () {  
        this.positionX = maxX/2;  
        this.positionY = maxY/2;  
    }  
  
    public void moveShip (int offSetX, int offsetY) {  
        positionX += offSetX;  
        positionY += offsetY;  
        // check for violation of maximum  
        if (positionX > maxX)  
            positionX = maxX;  
        if (positionY > maxY)  
            positionY = maxY;  
    }  
}
```

# Geheimnisprinzip



- Klienten können ausschließlich auf die spezifizierten Operationen zugreifen
- Eine geprüfte Komponente kann durch einen fehlerhaften externen Zugriff nicht zerstört werden

# Geheimnisprinzip



- Bezeichner, die in der Spezifikation eines abstrakten Datentyps vorkommen, sollten öffentlich sein
- Bezeichner, die nur für die Implementierung notwendig sind, sollten für Klienten verborgen bleiben

# Grundsätzlich ...



- Nie mehr öffentlich preisgeben als notwendig!

# Beispiel: Zu Öffentlich



```
Stack armerStack = new Stack();  
armerStack.push(1);  
armerStack.push(2);  
armerStack.push(3);  
armerStack.top = 0; // 2 und 3 werden "gelöscht"  
int drei = armerStack.pop();
```