

# LU05b - Prinzipien 2 (LSP, ISP)

## Prinzipien

- *DRY (Don't Repeat Yourself)*
- *SRP (Single Responsibility Principle)*
- *OCP (Open/Closed Principle)*
- **LSP (Liskov Substitution Principle)**
- **ISP (Interface Segregation Principle)**
- *DIP (Dependency Inversion Principle)*
- *KISS (Keep It Simple, Stupid)*
- *YAGNI (You Ain't Gonna Need It)*
- *SoC (Separation of Concerns)*
- *Law of Demeter (Principle of Least Knowledge)*

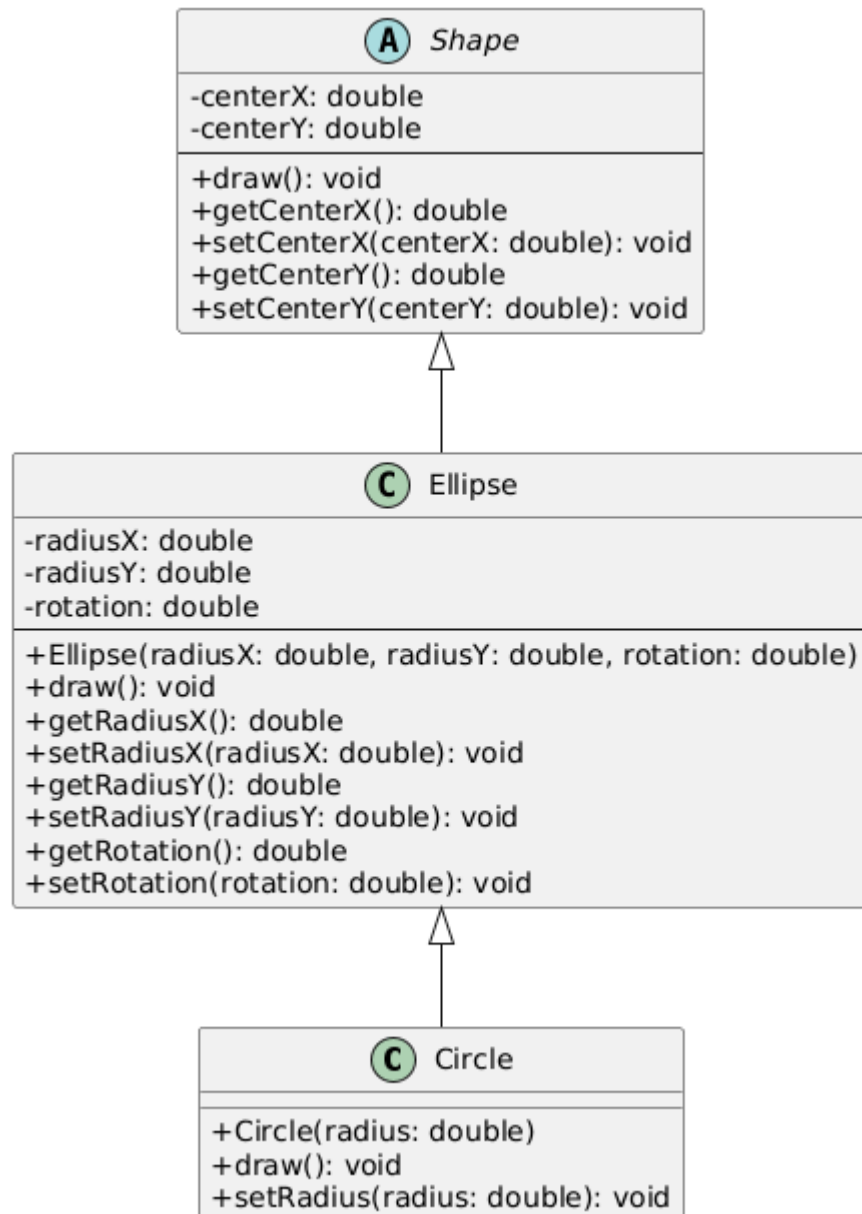
SOLID steht wiederum für SRP, OCP, LSP, ISP, DIP

Wir fokussieren uns dieses Mal auf **LSP** und **ISP**.

## LSP

Das Liskov Substitution Principle besagt, dass ein Programm, das Objekte einer Basisklasse T verwendet, auch mit Objekten der davon abgeleiteten Klasse S korrekt funktionieren muss, ohne dabei das Programm zu verändern.

Folgendes Beispiel verletzt das LSP, weil ein Kreis zwar eine Spezialform einer Ellipse ist, aber in diesem Fall nicht mit den Eigenschaften der Klasse `Ellipse` kompatibel ist. `setRadiusX()` und `setRadiusY()` können nach wie vor auf dem Kreis aufgerufen werden und dadurch unerwünschte

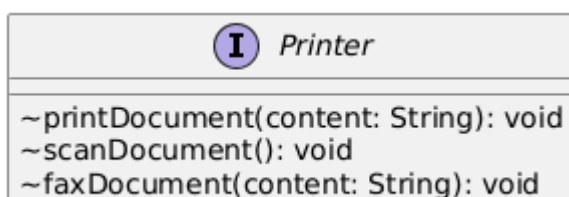


Nebeneffekte hervorrufen.

Besser wäre es die Klasse Circle direkt von Shape erben zu lassen.

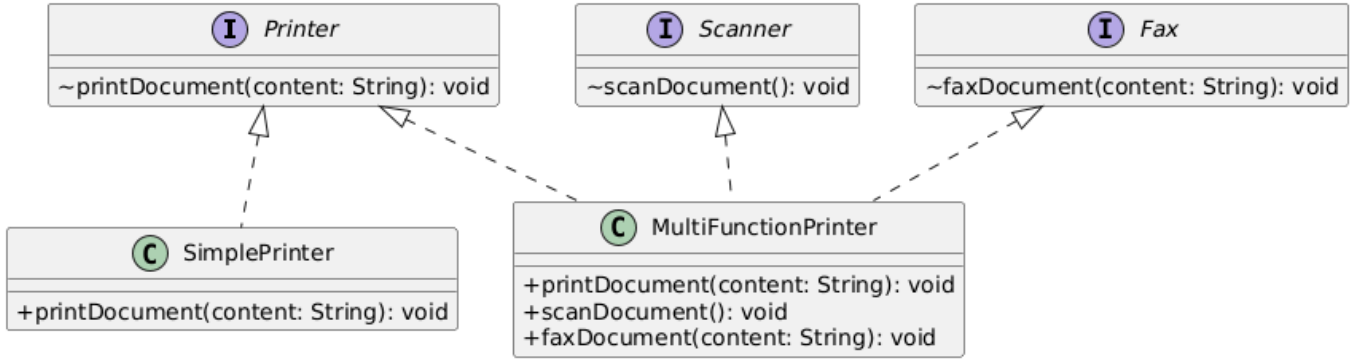
## ISP

Das Interface Segregation Principle besagt, dass Interfaces nur logisch untrennbare Methoden vorgeben sollten. Das heisst in diesem Beispiel, die Methoden printDocument, scanDocument und faxDocument sollten nur in einem gemeinsamen Interface definiert werden, falls sämtliche (zukünftige) Implementierungen alle Methoden zwingend benötigen.



Ist dies nicht der Fall, so sollte man die Interfaces aufteilen. Dadurch können zukünftige Klassen nur

einzelne Interfaces implementieren und sind nicht gezwungen irgendwelche Dummy-Implementierungen zu erstellen.



From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu05/b?rev=1757933633>

Last update: **2025/09/15 12:53**

