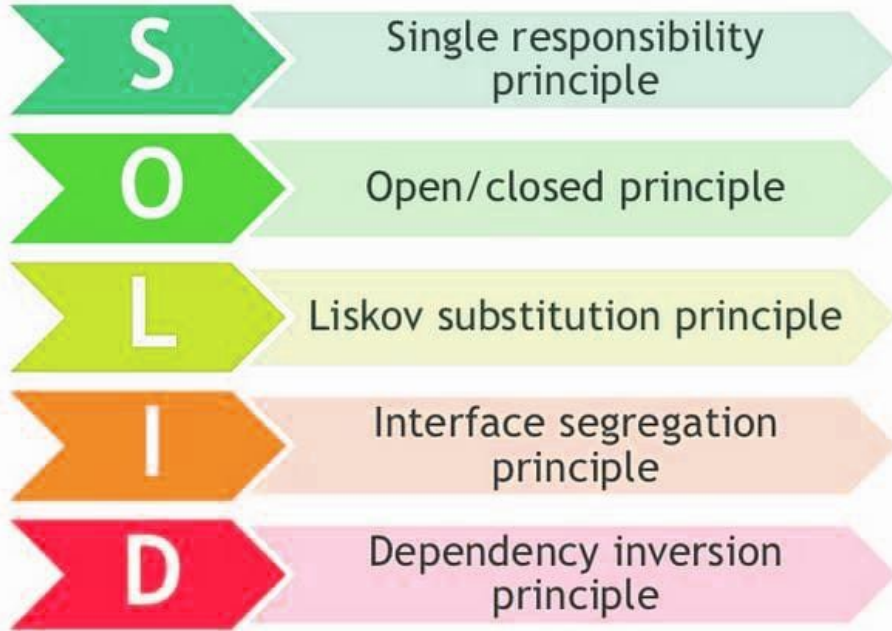


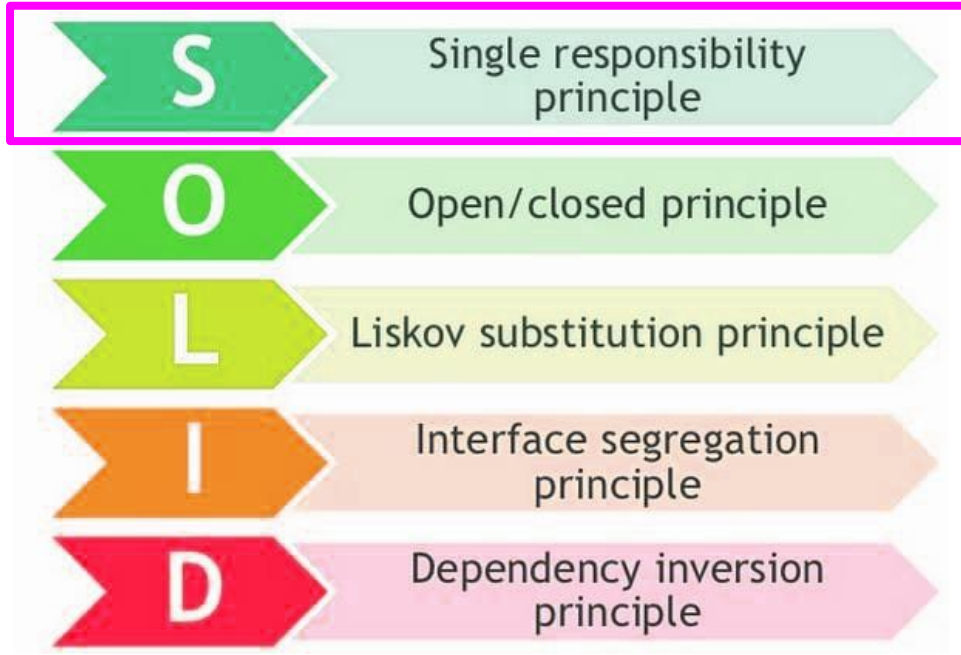
# SOLID principles

---



# SOLID principles

---



*A module, a class, or a method*  
should be responsible for a  
**single functionality** of a  
software system

Uncle Bob



A class should have  
only one reason to  
change!

Uncle Bob



A class should have  
only one **reason** to  
change!

Uncle Bob



A class should have  
only one reason to  
change!

reason  $\longleftrightarrow$  actor

# How to spot SRP violations

---

- Is the class serving more than one actor?
- Do the functionalities of the class serve the same actor?
- Is the class responsible for many functionalities?

# SRP violation problems

---

- Classes are entangled
- God classes



