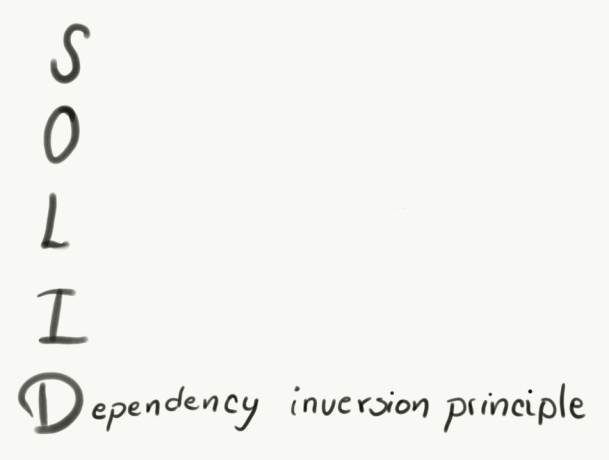
Inversion of Control Containers
&
Dependency Injection Pattern

With “Injector”

Summary

- From SOLID to Inversion Of Control Containers
- Code Example Explanation
- Example that violates DI principle
- Example that respects DI principle
- Example that use IoC Containers
- Questions



High-level modules should not depend on low-level modules. Both should depend on abstractions (e.g. interfaces).

Abstractions should not depend on details. Details (concrete implementations) should depend on abstractions.

From SOLID to Inversion of Control Containers

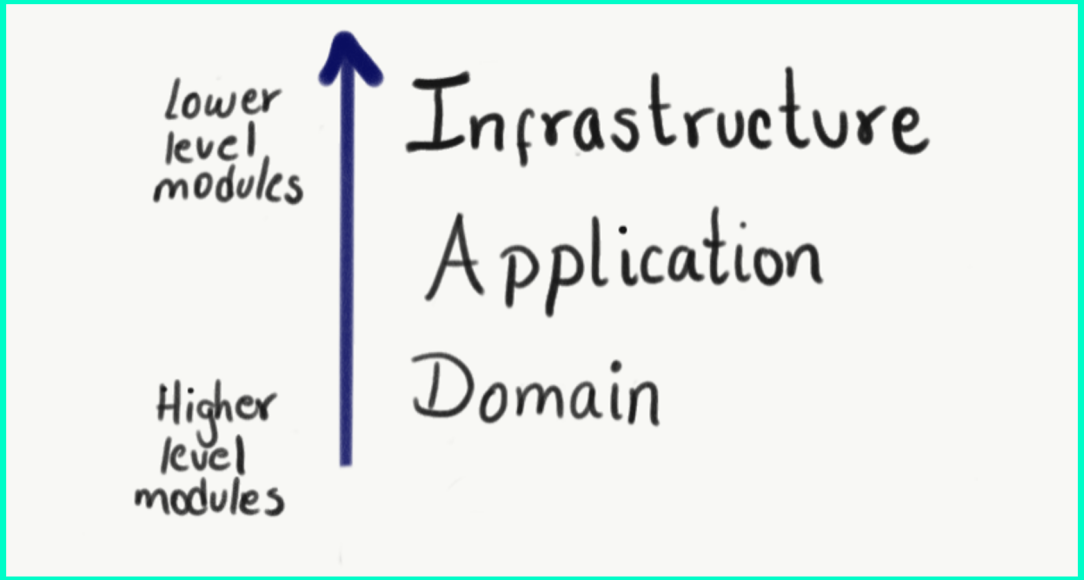
Code Example Design

Architecture

Domain Layer: Contains the state and the behaviour of the domain.

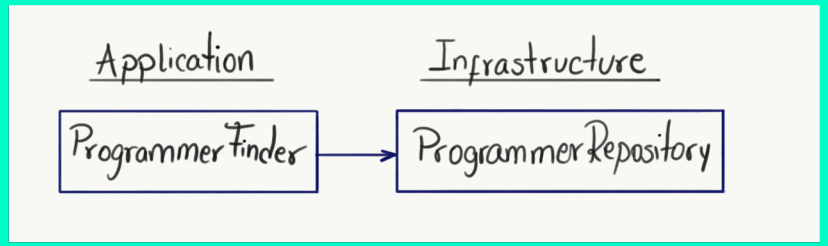
Application Layer: Coordinates application activities and creates and accesses domain objects.

Infrastructure Layer: Supports other layers, contains specific implementations of code for code that has to interact with components outside our system.

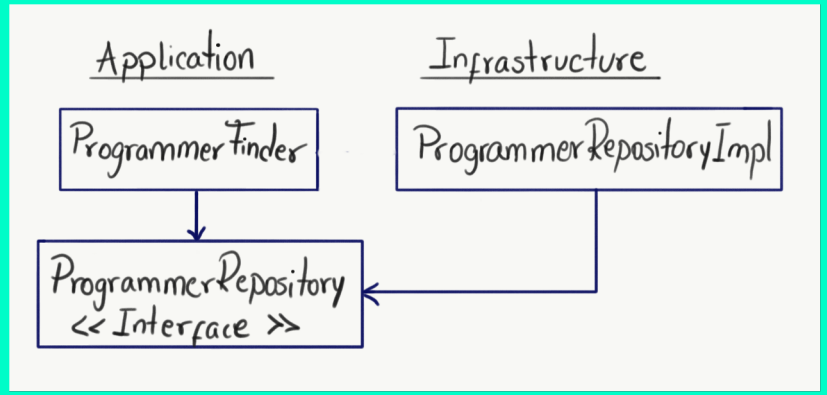

Lower level modules
Higher level modules
Infrastructure
Application
Domain

TALK IS CHEAP,
SHOW ME THE CODE.

Dependency inversion Violation



Dependency inversion Solution



TALK IS CHEAP,
SHOW ME THE CODE.

Injector


TALK IS CHEAP,
SHOW ME THE CODE.