Inversion of Control (IoC) and Dependency Injection (DI)

www.cs.uoi.gr/~zarras/http://www.cs.uoi.gr/~zarras/se.htm

martinfowler.com/articles/injection.html#ComponentsAndServices
www.baeldung.com/inversion-control-and-dependency-injection-in-spring
crosp.net/blog/software-architecture/clean-architecture-part-2-the-clean-architecture/

What is Inversion of Control (IoC)?

What is IoC?

Inversion of Control (IoC) in software engineering is the act of transferring the control of application objects (components, services, …) to a software engineering framework.

The framework will be responsible for creating objects, assembling/associating complex objects and managing the objects lifecycle.

Typically, the part of the framework that does this is called a **container.**

What is IoC?

Frequent terms that are used for the application objects that are managed by the framework is **beans** or **components.**

NOT ALL application objects worth being managed by the framework !!!

Typically, this is useful for **important objects** that constitute the backbone of the application e.g.,

- controllers that accept web requests,
- service objects that realize the core operations of the application,
- data mapper objects that interact with the database management system,
- other objects that should be easily configured when the application starts and/or reconfigured while the application is running ….

What is Dependency Injection (DI)?

What is DI?

Dependency Injection (DI) in software engineering is the act of assembling/associating **composite/aggregate** objects.

The basic idea is to have an **assembler object** that is responsible for setting the object fields of a composite/aggregate object to refer to other constituent objects.

Typically, in frameworks the role of the **assembler** is played by the **container** and the injection is transparent to the application.

Often, the application guides the assembler with guidelines provided in configuration files or more recently with inline annotations.