

Domain Driven Design

Goals

- Create a common language to facilitate communication between tech, bussiness actors, i.e. non-technical stakeholders.
- It is a **design technics** not a development technics.
- Re-create a link between data and functionalities.

Principles

- All complex design for an application has to be based on a domain model.
- Focus first on the business concerns and its logical point of view before to analyze technical constraints.
- DDD comes from the book of Eric Evans : Domain Driven Application (2004).
- Even if POO is more convenient to apply this method, it

Domain Driven Design

Keywords

- **Entity** - it has to have a unique identity (id), reference to the object is unique too. Identity can be a composition between several property values of the object.
- **Value-object** - use for the values of their attributes (properties) so they have not a single identity. They can contain another Value-object.
- **Agregate** - Group objects that are considered as a single entity in the point of view of the data modifications.

Domain Driven Design

Keywords

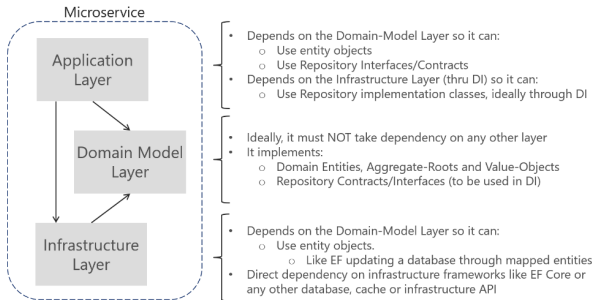
- **Service** - Pieces of algorithms that do not really belong to object as behaviors. Can apply treatments to several `entity` or `value-object`. A service has to respect modularity.
- **Module** - Provide a global view of the model. Has to create a consistent view of the model. Low coupling between different modules reduce complexity of the model.
- **Factory** - Inspired from the design pattern. Contain the logic of the object creation. Can take in charge how to create identity of objects. Not mandatory.
- **Repository** - Store the objects and provide access to them.

Conception with DDD

Architecture with layers

- Recommend to build the whole architecture by composition of several layers: Domain, Infrastructure, Application and User layers.

Dependencies between Layers in a Domain-Driven Design service



Your turn !

E-shop

- You have to write an application to manage basket in shop on internet.
- Design the domain layer containing these concepts :
 - Reference : for each product, unique, properties : id, name, description and price.
 - Basket : contains command lines - a command line concerns only one product, with a quantity and a total price. Basket has a total price i.e. the sum of command lines prices.
- It is possible to add/suppress command lines to basket.
- No possibility to have several command lines for the same product.

Find which one is Entity, Value-of and Agregate ?

Organisation

Directories/Modules

- domain: main concepts.
- application: implementation of services.
- infrastructure: concrete implementation of repository.
- ui: client implementation.