Domain

When I’m working on a software project, the domain is the conceptual area I’m working inside of. For example, if I’m working on an application that has to do with restaurants, the domain is restaurants.

Domain model

The world is a staggeringly complex place. Even relatively simple-seeming things like restaurants involve way more complexity than could be accurately captured in a software system. So instead of coding to a domain, we have to code to a domain model.

For me, a domain model is a separate thing from any particular code or piece of software. If I come up with a domain model for something to do with restaurants, I could express my domain model on a piece of paper if I wanted to (or just inside my head). My domain model is a standalone conceptual entity, regardless of whether I actually end up writing any software based on it or not.

A domain model also doesn’t even need to be consciously expressed in order to exist. In fact, on most software systems I’ve ever worked on, the domain model of the system only exists in the developers’ minds. The domain model isn’t something that someone planned at the beginning, it’s something that each developer synthesizes in his or her mind based on the code that exists in the application and based on what the developer understands about the domain itself.

Object model

The place where my domain model turns into actual code is in the object model. If my domain concepts include restaurant, order, and customer, then my object model will probably include objects like Restaurant, Order and Customer.

Domain object

Any object in my object model that also exist as a concept in my domain model I would call a domain object. In the previous example, Restaurant, Order and Customer would all be domain objects.

Not every object in a system is a domain object. Some objects are value objects. A value object is an object whose identity doesn’t matter. Examples of concepts that would make sense as value objects rather than domain objects are phone number values or money values.

**So far as I understand, the domain object is object that is related to the domain that we are working with. For example:**

**If we are working on an e-commerce website, the domain is actually a shop, and all goods in the shop are domain objects.**