

# DDD常见概念

C-2 创建: 张林伟, 最后修改: 张林伟 今天 19:36

贫血领域对象（Anemic Domain Object）是指仅用作数据载体，而**没有行为和动作**的领域对象。

失忆症：简单的业务系统采用贫血模型和过程化设计是没有问题的，但在业务逻辑复杂了，**业务逻辑、状态会散落到在大量方法中，原本的代码意图会渐渐不明确**，我们将这种情况称为由贫血症引起的失忆症。

**实体（Entity）**：当一个对象由其**标识**(而不是属性)区分时，这种对象称为实体(Entity)。比如当两个对象的标识不同时，即使两个对象的其他属性全都相同，我们也认为他们是两个完全不同的实体。

不应该给实体定义太多的属性或行为，而应该寻找关联，将属性或行为转移到其他关联的实体或值对象上。

**值对象（Value Object）**：当一个对象用于对事物进行描述而**没有唯一标识**时，那么它被称作值对象。因为在领域中并不是任何时候一个事物都需要有一个唯一的标识，也就是说我们并不关心具体是哪个事物，只关心这个事物是什么。对象是**不可变的**，即所有属性都是只读的，所以可以被安全的共享。

**领域服务（Domain Service）**：一些重要的领域行为或操作，它们不太适合建模为实体对象或者值对象，它们本质上只是一些操作，并不是具体的事物，另一方面这些操作往往又会**涉及到多个领域对象的操作**，它们只负责来协调这些领域对象完成操作而已，那么我们可以归类它们为领域服务。它实现了全部业务逻辑并且通过各种校验手段保证业务的正确性。同时呢，它也能避免在应用层出现领域逻辑。

**聚合及聚合根（Aggregate, Aggregate Root）**：聚合是通过定义领域对象之间清晰的所属关系以及边界来实现领域模型的内聚，以此来避免形成错综复杂的、难以维护的对象关系网。聚合定义了一组具有**内聚关系**的相关领域对象的集合，我们可以把聚合看作是一个**修改数据的单元**。

聚合由根实体，值对象和实体组成。

聚合根具有全局的唯一标识，而实体只有在聚合内部有唯一的本地标识，值对象没有唯一标识，不存在这个值对象或那个值对象的说法；聚合根有独立的生命周期，**实体的生命周期从属于其所属的聚合**，实体完全由其所属的聚合根负责管理维护；值对象无生命周期可言，因为只是一个值；

聚合根属于实体对象，它是领域对象中一个高度内聚的核心对象。（聚合根具有全局的唯一标识，而实体只有在聚合内部有唯一的本地标识，值对象没有唯一标识，不存在这个值对象或那个值对象的说法）。

若一个聚合仅有一个实体，那这个实体就是聚合根；但要有多个实体，我们就要思考聚合内哪个对象有独立存在的意义且可以和外部领域直接进行交互。

**聚合根到聚合根**，通过ID关联；

**聚合根到其内部的实体**，直接对象引用；

**聚合根到值对象**，直接对象引用；

**实体对其他对象的引用规则**：1）能引用其所属聚合内的聚合根、实体、值对象；2）能引用外部聚合根，但推荐以ID的方式关联，另外也可以关联某个外部聚合内的实体，但必须是ID关联，否则就出现同一个实体的引用被两个聚合根持有，这是不允许的，一个实体的引用只能被其所属的聚合根持有；

**值对象对其他对象的引用规则**：只需确保值对象是只读的即可，推荐值对象的所有属性都尽量是值对象；

**工厂（Factory）**：DDD中的工厂也是一种封装思想的体现。引入工厂的原因是：有时创建一个领域对象是一件相对比较复杂的事情，而不是简单的new操作。工厂的作用是隐藏创建对象的细节。

**仓储（Repository）**：资源仓储封装了基础设施来提供查询和持久化聚合操作。这样能够让我们始终关注在模型层面，把对象的存储和访问都委托给资源库来完成。它不是数据库的封装，而是领域层与基础设施之间的桥梁。仓储还有一个重要的特征就是分为仓储定义部分和仓储实现部分，**在领域模型中定义仓储的接口，而在基础设施层实现具体的仓储**。

参考资料：

<https://blog.csdn.net/ityouknow/article/details/81572072>

DDD战术模式

DDD分享备份