**POCO**

POJO的名称有多种，pure old java object 、plain ordinary java object 等。按照Martin Fowler的解释是“Plain Old Java Object”，从字面上翻译为“纯洁老式的java对象”，但大家都使用“简单java对象”来称呼它。  
　　POJO的内在含义是指那些没有从任何类继承、也没有实现任何接口，更没有被其它框架侵入的java对象。  
　　POCO的概念是从java的POJO借用而来，而两者的含义是一致的，不同的仅仅是使用的语言不一样。所以POCO的解释就是“Plain Old C# Object”。

**PO**

PO是指持久对象（persistant object持久对象）。持久对象实际上必须对应数据库中的entity，所以和POJO有所区别。比如说POJO是由new创建，由GC回收。但是持久对象是 insert数据库创建，由数据库delete删除的。基本上持久对象   生命周期和数据库密切相关。另外持久对象往往只能存在一个数据库 Connection之中，Connnection关闭以后，持久对象就不存在了，而POJO只要不被GC回收，总是存在的。  
　　由于存在诸多差别，因此持久对象PO(Persistent Object)在代码上肯定和POJO不同，起码PO相对于POJO会增加一些用来管理数据库entity状态的属性和方法。而ORM追求的目标就是要 PO在使用上尽量和POJO一致，对于程序员来说，他们可以把PO当做POJO来用，而感觉不到PO的存在。

POJO仅包含最简单的字段属性，没有多余的东西，它本质上就是一个普通的JavaBean。但是在POJO的基础上，能够扩展出不同的对象。

扩展：

为POJO增加了持久化的方法（Insert、Update、Delete……）之后，POJO就变成了PO。

为POJO增加了数据绑定功能之后，POJO就变成了View Object，即UI Model。

为POJO增加业务逻辑的方法（比如单据审核、转帐……）之后，POJO就变成了Domain Model。

POJO还可以当作DTO使用。

**ORM**

ORM(Object/Relational Mapping) 对象关系映射，主要是把数据库中的关系数据映射称为程序中的对象。

**DTO**

数据传输对象 (Data Transfer Object)。

**Entity Framework**

Entity Framework的全称是ADO.NET Entity Framework，是微软开发的基于ADO.NET的ORM(Object/Relational Mapping)框架