德（迪）米特法则

（一）定义

迪米特法则（Law of Demeter, LoD）又称为最少知识原则（Least Knowledge Principle, LKP），它有多种定义方法，其中几种典型定义如下：

（1）不要和“陌生人”说话。

（2）只与你的直接朋友通信。

（3）每一个软件单位对其他的单位都只有最少的知识，而且局限于那些与本单位密切相关的软件单位。

简单来说，迪米特法则就是指一个软件实体应当尽可能少的与其他实体发生相互作用。这样，当一个模块修改时，就会尽量少的影响其他的模块，扩展会相对容易，这是对软件实体之间通信的限制，它要求限制软件实体之间通信的宽度和深度。

（二）对象的朋友

在迪米特法则中，对于一个对象，其朋友包括以下几类：

（1） 当前对象本身(this)；

（2）以参数形式传入到当前对象方法中的对象；

（3）当前对象的成员对象；

（4）如果当前对象的成员对象是一个集合，那么集合中的元素也都是朋友；

（5）当前对象所创建的对象。

任何一个对象，如果满足上面的条件之一，就是当前对象的“朋友”，否则就是“陌生人”。

1. 特点

（1）最少依赖；

（2）类内部应该高内聚，设置相应的权限，有选择的暴露方法；

（3）类的依赖关系尽量减少，保持简单和独立，降低耦合；

（4）降低了类之间的耦合度，提高了模块的相对独立性；

（5）由于亲合度降低，从而提高了类的可复用率和系统的扩展性；

（6）从依赖者的角度来说，只依赖应该依赖的对象；

（7）从被依赖者的角度说，只暴露应该暴露的方法。

（四）狭义迪米特法则和广义迪米特法则

迪米特法则可分为狭义法则和广义法则。在狭义的迪米特法则中，如果两个类之间不必彼此直接通信，那么这两个类就不应当发生直接的相互作用，如果其中的一个类需要调用另一个类的某一个方法的话，可以通过第三者转发这个调用。

1. 狭义的迪米特法则

可以降低类之间的耦合，但是会在系统中增加大量的小方法并散落在系统的各个角落，它可以使一个系统的局部设计简化，因为每一个局部都不会和远距离的对象有直接的关联，但是也会造成系统的不同模块之间的通信效率降低，使得系统的不同模块之间不容易协调。

1. 广义的迪米特法则

指对对象之间的信息流量、流向以及信息的影响的控制，主要是对信息隐藏的控制。信息的隐藏可以使各个子系统之间脱耦，从而允许它们独立地被开发、优化、使用和修改，同时可以促进软件的复用，由于每一个模块都不依赖于其他模块而存在，因此每一个模块都可以独立地在其他的地方使用。一个系统的规模越大，信息的隐藏就越重要，而信息隐藏的重要性也就越明显。

（五）主要用途

迪米特法则的主要用途在于控制信息的过载：

在类的划分上，应当尽量创建松耦合的类，类之间的耦合度越低，就越有利于复用，一个处在松耦合中的类一旦被修改，不会对关联的类造成太大波及；

在类的结构设计上，每一个类都应当尽量降低其成员变量和成员函数的访问权限；

在对其他类的引用上，一个对象对其他对象的引用应当降到最低。

1. 校园卡管理系统中的应用

在本实践项目中用户只需要知道校园卡申请有没有通过，不需要知道管理员是怎么进行审核的。