C#开发中三层架构BLL，DAL还有IBLL和IDAL接口，请问为什么要定义接口？有什么用？

这是为了让模块之间解耦，例如某个业务用接口表示了，那么它的具体实现都是基于这个接口的；在使用中，直接操作接口类型即可，而实际实现接口的对象是根据业务传递进来的。（可以看做接口是所有实现的父类，这样只要传递进来的对象是接口的实现，即可通过接口定义的方法对对象进行操作，至于这个对象是谁，不需要关心。）

具体举例来说：汪星人会“汪汪”叫，而喵星人会“喵喵”叫，业务是操纵这些萌物叫，那么可以定义一个萌物接口，里面有一个“叫”方法，汪星人和喵星人都实现了“萌物”，而具体是怎么叫的是由汪星人和喵星人来实现的。业务层调用“叫”的时候只需要让这边传进来一只萌物，调用萌物的“叫”方法即可，至于这只萌物到底是什么，不需要关心，这样就解开了两边的耦合，将来有了新的萌物，只要实现接口然后给业务层传进来就行了，这边叫的业务不需要改变任何代码，因为它得到的是“萌物”的实现。以后所有业务编程都面向接口，即制定了这个接口的规则，那么业务只管去对这个接口操作就行了，至于它（接口）的实现，我不关心，我只管我这边做好就行了，而做具体的人根据接口规范做好实现就行了，两边没有任何关联，关联只是这个制定好的“约定”。这样就解开了两边的耦合。

您看到的这些接口就是制定的这个“约定”，只要操作接口就能根据约定进行操作而不用关心具体操作的是谁。接口这个名字起得很形象，比如USB存储，不管我的设备是怎么实现的（磁盘、flash、eMMC、光驱等）只要我实现了USB接口，我都能通过这个接口存取数据，至于设备是怎么存取数据的？是什么原理？我不关心，以后有了更先进的存储方法，只要我实现了USB接口，以前的设备就能用，这样就解开了两边的耦合；而这个比喻在软件工程中原理是一样的。