



代码重用，也称作代码复用，通常是用已经编写好的代码，再次实现新的功能的方法。当然还有一种方式是发现两个或者多个功能的代码实现是一致的或者很相似，就把通用的代码抽取（abstract）出来，从而达到代码重用的目的。

代码重用的目的在于提高效率，这里指的效率不仅是开发的效率，还有维护的效率，所以我们可以看到代码重用有以上两种形式。虽然说复制粘贴一段代码也算是某种程度上的“重用”，但是由于这种方式并不能提高维护效率，反而增加了维护的成本，所以并不是真正的代码重用。

代码重用需要重用的代码明确定义好输入和输出，自身实现模块化，达到高内聚、低耦合，实现数据封装。这样才能使重用的成本更小，而不是每次重用反而要写很多代码才能使用起来。

代码重用的形式有：程序库 / 类库，设计模式和应用框架。

(1) 程序库 / 类库

程序库 / 类库是最常见的代码复用形式。因为在代码中，例如文件的读 / 写，数据库的操作和用户鉴权的操作都是通用的方法，一般都会被封装在程序库 / 类库中，以便所有开发人员都可以方便地使用。使用程序库 / 类库中的一大优点就是测试人员可以对它所提供的操作经过充分测试，保证其质量，但是缺点是开发人员无法对程序库 / 类库提供的操作的具体实现进行调整，同时程序库 / 类库也要求开发人员花时间去学习它的用法。

(2) 设计模式

设计模式对同类问题提供了通用的解决方法，所以设计模式也算是广义上的重用。由于设计模式所提供的是概念上的解决方法，对于具体问题往往还需要进一步实现代码。利用抽象类或接口往往可以在特定的设计模式中达到代码的重用。

(3) 应用框架

开发人员还可以通过第三方的应用程序或者框架来实现对一个程序的大范围复用，这可以有效地提高生产效率。但缺点在于第三方的应用程序或者框架往往只在某个特定的应用范围内才可以适用。