Delphi开发中几种代码复用方式及其比较 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/141/2021_2022_Delphi_E5_BC_80_E5_c29_141420.htm 我们在软件开发中，经常会碰到多处代码相同或相似的地方，如何处理这种情况，不仅对个人，而且对于一个团队，是一个重要的问题。代码复用的作用也不仅是使得代码编写简单、减少工作量。本文只讨论在一个只使用delphi作为开发工具的团队如何有效的进行代码复用，如何在开发进度、软件维护之间取得平衡。这里以作者领导的一次开发作为例子。 最原始的办法，复制、粘贴，不同的地方稍作修改。恐怕每个人都曾经使用过这种方法。这无疑在写程序时是最快的方法。不过，其缺点也是明显的，别的不说，这种方式阻碍了人的思考，功能确实完成了，但是错过了自己提升自己的机会。表面看是节约了时间，实际是浪费了自己。我当年使用这种方法时，总觉得自己更象一个泥瓦匠。 谁都会对这种方式说"不"吗？问题在于我在实践中发现，更多的程序员倾向于这样编程。我不得不强制执行一些看起来是微不足道甚至是不完全正确的的细节标准，例如：所有的SQL语句必须在DataModal中。大多数程序员反对这一条，反对的理由有很多种，有些不乏道理。不过更多的是程序员感觉这样编程变得繁琐，感到受到了约束。他们反问在Form中直接写sql语句多好，又容易看懂，不需要看程序时跳来跳去的。我只好打开他们曾经完成的系统，使用Find in File 功能，搜索结果表明同样的sql语句出现了若干次。如果不有效的解决这个问题，我实在难以想象如何向程序员灌输如同业务逻辑和界面逻辑分开的概念。好在有些程序员开始

考虑这个问题，他向我提出这样的方案：把所有的sql语句写在单独unit中，供所有的unit调用。虽然我知道，这样做只是为了复用而复用，但是我仍然鼓励这种做法，因为能复用总比不能复用强，半块面包总强于没有面包。 使用函数。我发现这是大多数程序员喜欢使用的方法。从面向过程的程序来说，函数调用几乎是唯一的代码复用方法了。在开发组讨论的过程中，程序员们也最喜欢使用这种方式作为系统之间的接口。这次开发中终于把权限、流程和打印系统单独分裂出来，由单独的程序员开发，提供给其他系统调用。如何调用？程序员首先想到的就是函数调用。确实函数调用也解决了不少的问题。 随着程序不断的开发，系统的复杂度也在不断增加，这时候以函数作为接口的方式也显得越来越复杂。以一个简单的例子而言，权限系统处理操作员管理、系统登陆验证等功能，所以系统都需要当前登陆操作员的信息。开始是想的比较简单，只要知道登陆操作员的id，那么其他相关信息都可以通过权限系统不同的引出函数来查询，但考虑到操作员信息的通用性，而且为其提供若干个引出函数就显得过于繁琐。我们设计了一个操作员类，每个系统都有其接口声明，而具体实现在权限系统中。在系统中增加一个全局操作员对象，这个对象在系统登陆时创建，记录了登陆操作员的所有信息。其他系统只要访问这个全局对象就能得到登陆操作员的信息，包括其所属部门，登陆时间等等。这样做的好处有：（1）隐藏了实现的细节，不仅隐藏了数据库设计，也隐藏了代码实现的细节。例如后来由于种种原因，操作员的表结构发生了变化，而其他系统对此可以一无所知。后来权限系统也重写了读取权限部分，同样，其他系统也一无

所知，只是感觉速度变快了。（2）模块之间是松耦合。（3）提高程序员的水平。 最后一点恐怕有人怀疑，我要更多的解释。在讨论如何得到当前登陆操作员信息时，开始甚至有程序员提出这样的方法：设置全局变量操作员编号变量，让其他系统利用这个id直接读数据库中得到信息的。 代码复用还有应该重要的方面是界面的复用。一次开发中必然会有部门界面使用了相同的元素。这里的代码复用主要有三个手段，框架、控件、窗口继承。 还是举一个简单的例子，这次开发中的部门是按照树型结构来描述的，很多个界面上需要出现这个部门树，比如资料管理界面、比如人员管理界面，界面的左边都是这颗部门树。在不同的界面中又有不同的处理，相应不同的事件。这方面，Delphi为我们提供了一个好的方法：Tframe。我更倾向于认为Frame就是一组控件及相关代码的集合，虽然在设计时它和Form比较相似。在部门树这个Frame中，只有一个Treeview，关键在于它已经具备了按照树型显示部门能力，而且，更进一步，你可以在其上面进行拖拽来完成部门调整的功能。当然对于不需要这个功能的窗口，可以简单的设置一个属性来禁止这个功能。 框架被程序员了解后得到了大量的使用。大家感到确实能非常方便的解决很多问题。而且，对于框架的修改会被所以使用它的窗口有效，不用因为要修改某个共同的地方找遍工程中的每个窗口了。 在Delphi推出Tframe之前的版本，可以使用窗口来代替它。其中的区别不在本文中讨论了。 控件无疑是Delphi最令人心动的了，在眼花缭乱的无数控件面前往往会迷失。控件和框架的作用有相似的地方，不过普遍认为控件更加通用一些，而框架只是用在特定的程序中。如果团队中拥有擅场

于此的高手自然能让团队如虎添翼，没有也不必黯然神伤，因为普通的控件编写也没有想象的那么难。 同样举一个简单的例子。这次开发使用了Ado作为数据联接，adoDataset有sort属性，可以完成本地排序的功能。大家知道许多优秀的Dbgrid控件都能实现点击标题栏自动排序的功能。我们也做了一个增强的dbgrid来实现这个功能，利用ado的特性，非常简单，几句代码就可以了。当然还有其他增强，来适应这个专业系统的其他方面。 