技术短文

LetsGo 安卓应用开发小组

June 5, 2014

修订历史

版 本	日 期	描 述	作 者
初始草案	2014 年 6 月 1 日	第一个草案。介绍单例模式。	王旸

1 引论

LetsGo 是一个关于健身活动的安卓 APP，在这个软件当中，因为我们使用的是安卓软件的框架，所以直接或间接的采用了许多设计模式。在这里我们想要介绍的是单例模式。因为在这个项目中，我们不但应用了这个模式，还自行设计并实现了一个单例模式。因此我们就将这个典型的设计模式拿出来介绍一下，并按照我们自己的理解进行具体的讲解。

2 单例模式

单例模式是一种非常经典的设计模式，在现代的软件工程面向对象程序设计与开发的过程中有许多的应用。接下来我们将对他的背景与实现进行分析。

2.1 单例模式——背景

在软件开发设计的过程中，有时候会遇到一个问题。在不同的地方调用统一个变量的时候，他们其实本质上不是同一个变量。也就是说，其中一个的改变会导致另一个的改变。这个时候其实不一定是好事，也不一定是坏事。有时候单纯是为了实现某种功能的子函数，那当然是不相关比较好。但是有时候，我们需要在全局访问同一个数据的时候，那么应该怎么办呢？在结构化编程的年代，我们采用的方法是定义全局变量，定义 static 变量。但是在面向对象程序设

计的年代，显然这种方法因为其非常差的拓展性与可用性而应该被淘汰。取而代之的是一种叫做单例模式的东西。

2.2 单例模式——原理

单例模式通过在全局定义一个只能有一个实体的类来实现。具体的过程是，将类的构造函数私有化，也就是说，不能在类的外部实例化这个类的成员。然后在类的内部定义一个 singleton 对象，也就是这个类在整个系统中的唯一对象。然后每个外部需要调用的时候，调用这个类的一个 static 函数 getsingleton 函数，然后这个函数就会返回这个实体。这样一来，保证了全局在需要这个实体的时候都访问了同一个。

2.3 单例模式——应用

在软件设计当中，我们有时候会非常需要这样的一个全局的唯一的实体。比如说，在 cocos2D 当中，全局的导演类就必须是一个。又比如说，在很多应用的数据库存储中，为了保障所有数据都是同步的，所以采用了单例模式的数据库管理类。又比如说，在 Android 软件开发框架中，Calandar 这个关于系统时间的类也是单例模式。

3 LetsGo 项目中采用的单例模式

在 LetsGo 项目中，我们对于数据的存储采用了单例模式，以保障在系统的各个组件各个地方能够访问到相同的数据，能够改动的数据也互相同步。具体是这样的。

该跑步软件主要功能是制定计划、进行跑步，然后改写计划状态 (成功、失败或未完成)。但是我们对计划的改写却是在不同的 Activity 中。于是我们对几乎的管理类 RecordListManager 采用了单例模式。

首先，RecordListManager 的构造函数被改为了 private，这样以来，外界就无法私自的实例化一个 RecordListManager 类的实例了。该类也与数据库互相关联。这样一来数据库也不会发生混乱，因为通过全局只有一个的 RecordListManager 来进行管理。

然后，我们在 RecordListManager 中定义了一个 static 并且 public 的 getSingleton 的函数，这样的话外界统一通过这个接口获得这个单例类的实例，免去了各种繁琐的情况出现。

就这样，我们实现了单例这个类的目标，从而有序的管理了我们的 RecordList。

4 总结

对于单例模式，在实践中有很多的应用。我们本次项目也只是利用了它的一个特点。在实践中他应该还有节省资源等优势。面向对象的程序设计与开发有非常多已经被实践所检验的优良设计模式，我们应该好好掌握这些方法，从而加快自己的开发速度，提高我们的开发质量。