实验报告

|  |  |
| --- | --- |
| 一、实验内容  1、某学生会现设有主席、部长和成员三种角色，每个角色都有自己的编号ID，并提供一个display方法来打印自己的ID。现创建一个主席类（chairman），保证整个项目中只能创建一个主席，并通过display方法来验证是否满足该功能。  2、在此基础上，现需要扩充主席为主席团，主席团共设有3个主席。现修改项目使得项目只能创建三个主席，并通过display方法来验证是否满足要求。  请设计对应的类图，并给出实现代码。 | |
| 二、实验过程记录 | |
| 类图 | （1）为了对比单例类Chairman和Member类、Minister类不同的实例创造方式，对Member类和Minister类的display()方法也在Test测试类中进行了调用，因此类图中也包含了Minister类和Member类。     1. 在问题（1）的基础上扩充了主席团，类图中仅展示有关Chairman和ChairmanGroup相关的部分。 |
| 实验过程记 录 | 问题1的要求是保证项目中只能创建一个主席，主要设计的就是在‌Chairman 这个类中的以下方法：（1）private static Chairman instance：持有唯一的Chairman实例；（2）private Chairman(String id)：私有构造方法，防止外部类实例化；（3）public static synchronized Chairman getInstance(String id)：公共的静态方法，用于获取唯一的Chairman实例。使用synchronized关键字以确保线程安全；（4）public void display()：打印主席的 ID；（5）@Override protected Object clone() throws CloneNotSupportedException：防止克隆破坏单例模式。而‌Minister 类和 Member 类分别表示部长和成员，都包含一个id属性和一个display()方法。‌Test 类用于测试单例模式和display()方法,当Test类尝试创建两个主席实例并调用display()方法时，发现chairman1和chairman2的display()方法打印出来的是相同的ID，这验证了它们是同一个实例，即证明了单例模式的正确性。  问题2在此基础上，要求创建给定数目的实例，可以给单例类增加MAX\_INSTANCES这个static整数，调用instance.size()方法判断此时的实例个数是否达到了规定上限，如果没有即可继续增加新实例，然后相应地将这些Chairman实例存储在名为Chairmen的ArrayList中，调用数组遍历的方法就可以实现displayAll()方法，将三个主席的id打印出来。如果要继续创建Chairman实例，就会出现错误提示。 |
| 总结 | 单例模式的目的是保证一个类有且仅有一个实例，并提供一个访问它的全局访问点。单例模式中，主要通过这几个方法来保证类的实例唯一：（1）将构造方法声明为private，防止客户程序利用构造方法创建多个对象；（2）有一个静态私有的成员变量instance，用来保存创建的实例。（3）提供一个可以获得实例的静态方法，通常称为getInstance()方法。该方法返回一个类的实例，同时确保无论客户程序调用该getInstance()方法多少次，得到的始终是同一个对象。  单例模式拥有提供唯一实例的受控访问，节约系统资源，允许可变数目的实例等优点，但也存在缺点，因为没有抽象层而难以扩展，单例类职责过重，在Java自动垃圾回收技术下，实例化而未被使用的对象会被回收，下次使用时重新实例化，会导致对象状态丢失。另外，在实际问题中需要考虑线程安全等问题。 |