19 线程_设计模式

JAVAEE高级

一：单例设计模式

> 1、保证类在内存中只有一个对象
>
> 2/单例写法两种 (饿汉式 / 懒汉式)

1、饿汉式

> 1、饿汉式是空间换时间

```
class Singleton {
    // 1私有构造方法其他类不能访问该构造方法了
    private Singleton() {
    }
    // 2创建本类对象
    private static Singleton s = new Singleton();
    // 3对外提供公共的访问方法
    public static Singleton getInstance() { // 获取实例
        return s;
    }
}
```

```
    public static void main(String[] args) {
        Singleton s1 = Singleton.getInstance();
        Singleton s2 = Singleton.getInstance();
        System.out.println(s1 == s2);
    }
```

2、懒汉式

> 1、懒汉式是时间换空间
>
> 2、懒汉式单例的延迟加载模式

```
class Singleton1 {
    // 1私有构造方法其他类不能访问该构造方法了
    private Singleton1() {
    }
    // 2声明一个引用
    private static Singleton1 s;
    // 3对外提供公共的访问方法
    public static Singleton1 getInstance() { // 获取实例
        if (s == null) {
            // 线程1等待线程2等待
            s = new Singleton1();
        }
        return s;
    }
}
```

3、三种格式

```
class Singleton2 {
    // 1私有构造方法其他类不能访问该构造方法了
    private Singleton2() {
    }
    // 2声明一个引用
    public static final Singleton2 s = new Singleton2();
}
```

```
    public static void main(String[] args) {
        Singleton2 s1 = Singleton2.s;
        Singleton2 s2 = Singleton2.s;
        System.out.println(s1 == s2);
    }
```

二：Runtime类设计模式

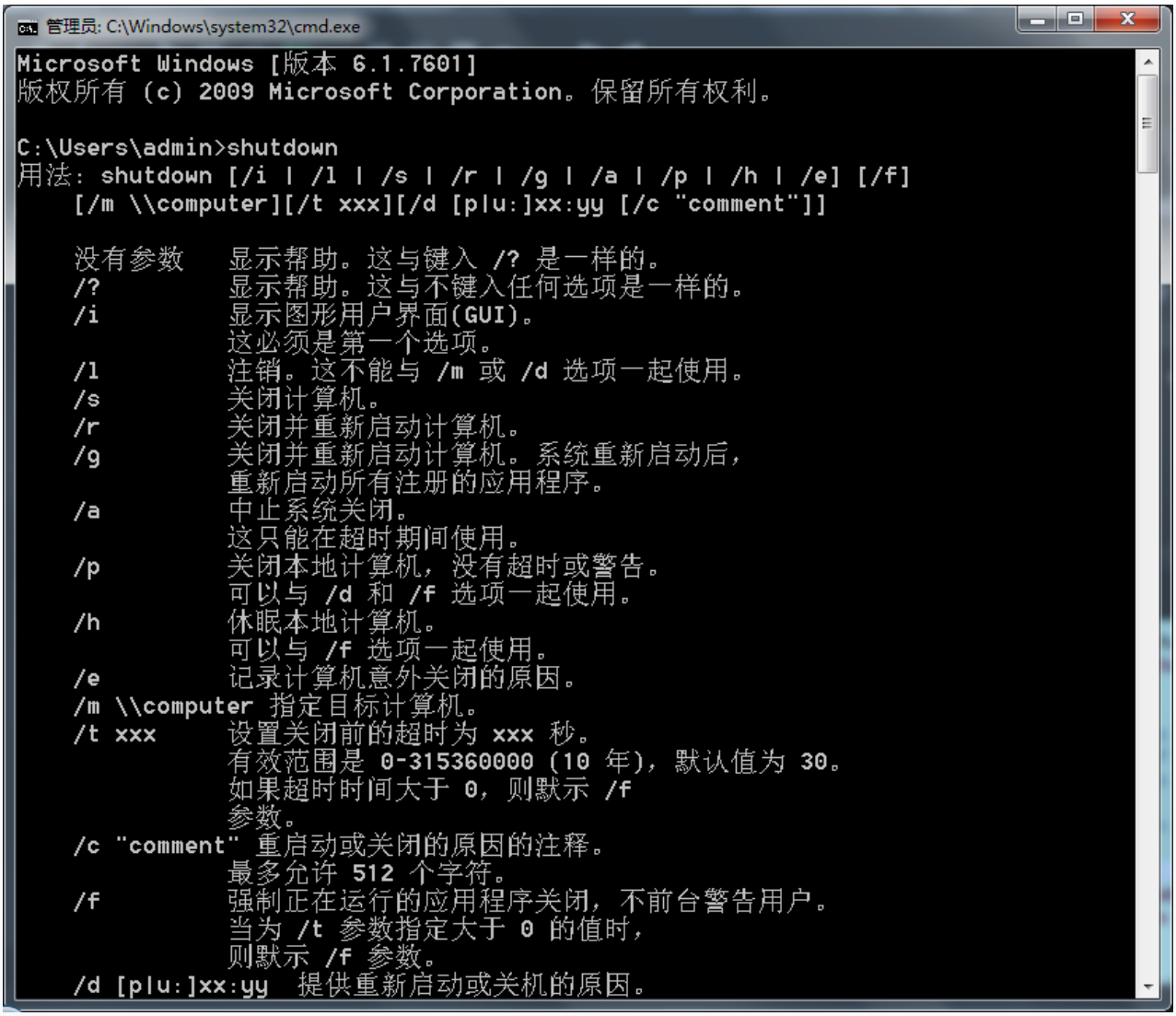
1、Runtime类是一个单例类饿汉模式

```
1.      public static void main(String[] args) throws IOException {
2.              Runtime r = Runtime.getRuntime();            //获取运行时对象
3.              r.exec("shutdown -s -t 300");//300秒后关机
4.              r.exec("shutdown -a");//取消关机
5.      }
```

源码解析

```
6  public class Runtime {
7      private static Runtime currentRuntime = new Runtime();
8
9      /**
0       * Returns the runtime object associated with the current Java applic
1       * Most of the methods of class <code>Runtime</code> are instance
2       * methods and must be invoked with respect to the current runtime ob
3       *
4       * @return  the <code>Runtime</code> object associated with the curre
5       *          Java application.
6       */
7      public static Runtime getRuntime() {
8          return currentRuntime;
9      }
0
1      /** Don't let anyone else instantiate this class */
2      private Runtime() {}
```

2、shutdown 的描述



三：Timer定时器

```
class MyTimerTask extends TimerTask {

    @Override
    public void run() {
        System.out.println("起床背英语单词");
    }
}
```

```
public static void main(String[] args) throws InterruptedException {
        Timer t = new Timer();
        //在指定时间安排指定任务
        //第一个参数:任务;第二个参数:执行的时间;第三个参数是:重复执行
```

```
5.          t.schedule(new MyTimerTask(), new Date(119, 6, 1, 14, 22, 50),3
    000);
6.          while(true) {
7.              Thread.sleep(1000);
8.              System.out.println(new Date());
9.          }
10.     }
```