

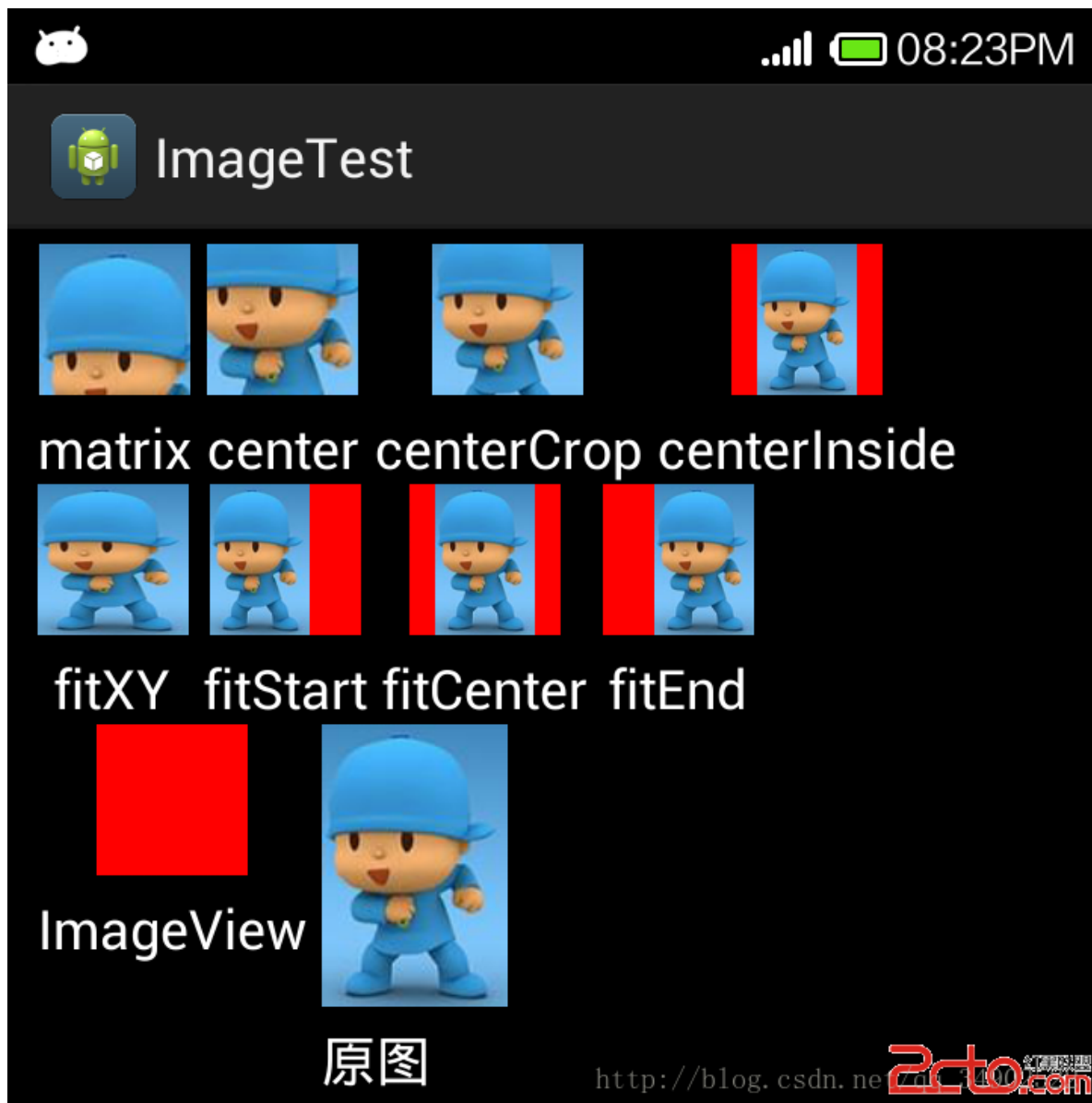
知识点积累

1. java中不支持多继承，只有多实现；(接口可以多继承)

2. 接口中的方法都是public abstract，可以省略；

3. 接口中没有变量，只能是常量，接口中定义常量默认的修饰符为public static final，可省略；

4. ImageView中的scaleType属性



5. Thread.yield(), 与Thread.sleep(long millis), 的区别

1. Thread.yield(): 是暂停当前正在执行的线程对象，并去执行其他线程。

2. Thread.sleep(long millis):则是使当前线程暂停参数中所指定的毫秒数然后在继续执行线程。

6. synchronized(this/.class/Object),synchronize方法区别

synchronize修饰的方法和 synchronize(this) 都是锁住自己本身的对象 而synchronize(class) synchronize(object) 都是锁别的对象。

synchronized的语义

每个对象都有一个**监视器monitor**，被synchronized修饰，语义就是获取这个对象的monitor，反编译后可以看到monitorenter和monitorexit。synchronized关键字有三种应用方式（其实按标题来讲应该是5种，但是其中有两种都是与另外两种等价的）：

- 修饰实例方法
- 修饰静态方法
- 修饰代码块（指定对象）

实验

```
public class SyncThread {
    private final Object lock = new Object();

    public void foo() throws Exception {
        synchronized (SyncThread.class) {
            for (int i = 0; i < 5; i++) {
                System.out.println(">>>foo: " + i);
                Thread.sleep(1000);
            }
        }
    }

    public void bar() throws Exception {
        synchronized (this) {
            for (int i = 0; i < 5; i++) {
                System.out.println("<<<bar: " + i);
                Thread.sleep(1000);
            }
        }
    }

    public void cpp() throws Exception {
        synchronized (lock) {
            for (int i = 0; i < 5; i++) {
                System.out.println("===cpp: " + i);
                Thread.sleep(1000);
            }
        }
    }

    public void der() throws Exception {
        for (int i = 0; i < 5; i++) {
            System.out.println("!!!der: " + i);
            Thread.sleep(1000);
        }
    }
}
```

运行结果：

```
>>>foo: 0
<<<bar: 0
===cpp: 0
!!!der: 0
>>>foo: 1
===cpp: 1
<<<bar: 1
!!!der: 1
>>>foo: 2
!!!der: 2
<<<bar: 2
===cpp: 2
>>>foo: 3
<<<bar: 3
===cpp: 3
!!!der: 3
>>>foo: 4
!!!der: 4
<<<bar: 4
===cpp: 4
```

分析

从以上结果来看各线程并没有发生竞争，互不影响，其实明白了synchronized语义也很好理解以上结果，几个synchronized获取的monitor都不是一个，当然相互不影响。

但是值得注意的几点：

- `synchronized(xxx.class)` 获取的是类的monitor，所以与 `public synchronized static void some()` 修饰静态方法是等价的
- `synchronized(this)` 获取的是当前实例的monitor，所以与 `public synchronized void some()` 修饰实例方法是等价的

总结：使用this和object的区别:this是对当前类做控制,即保证当前类是线程安全的,而对于非线程安全的类在调用时,为保证其线程安全性可以在调用时使用synchronized(object)确保被调用类的线程安全

总之，个人认为要理解几种不一样的地方，关键是理解清楚是获取的谁的monitor，只要是同一个monitor，当然就会发生同步！

7.java中实例和对象有什么区别

`MyClass myClass=new MyClass();`
new的时候在堆中新建了一个对象。这个对象是MyClass的实例。其中myClass是该对象的引用。这样理解就ok了。