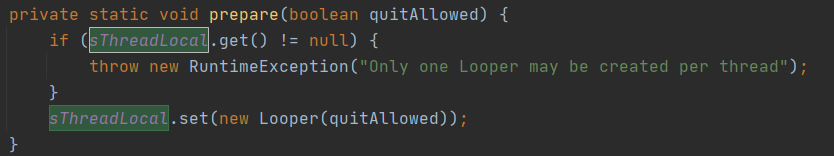
#### 1.一个线程有几个Handler?

答：有无数个，因为每个地方都可以new handler

#### 2.一个线程有几个Looper? 如何保证

答：一个，threadLocal保证，threadLocal如何保证的。是在Looper的prepare方法保证的



#### 3.Handler内存泄漏原因？为什么其他的内部类没有说过这个问题？

答：持有外部类的引用。匿名内部类会调用外部类的引用。Java基础。RecyclerView adapter ViewHolder 也是内部类，就没有内存泄漏的问题，是因为生命周期的问题。

enqueueMessage{

msg.target=this;

}

msg会持有this，this是Handler

msg持有了handelr，handler持有activity=====jc回收不了

GC:JVM的可达性分析，因为msg不能回收，所以msg持有的对象都不能回收，所以activity也不能回收了

#### 4.为何主线程可以new Handler?如果想要在子线程中new Handler要做些什么准备？

因为主线程在启动的时候就启动了Looper。ActivityThread的main方法有Looper.myLoop().

C:\Users\ZHANGW~1\AppData\Local\Temp\1636356133(1).png

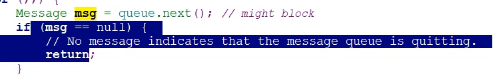
在子线程创建的时候必须要Looper.prepare()

#### 5.子线程中维护的Looper，消息队列无消息的时候处理方案是什么？有什么用？

生产者-消费者设计模式。如何消息队列无消息，就会阻塞、休眠。这是主线程

子线程的消息队列无消息的时候。就会调用quit().把mQuitting=true.就会返回一个null

Queue.next()也就会等于null



子线程无消息的时候：调用quit：唤醒线程-----messageQueue-----null------退出loop------整个loop就会结束

#### 6.既然可以存在多个Handler往MessageQueue中添加数据（发送消息时各个Handler可能处于不同线程），那它的内部是如何确保线程安全的？

答：锁

Messagequeue，对象里面的所有的函数，代码块，都会受限

Synchronized{

}:同步锁，内置锁 ？JVM加锁跟解锁自动完成

可以锁代码块，锁对象

MessageQueue 111=new MessageQueue();

111.enqueueMessage

这个时候的111.next不可以使用

MessageQueue 222=new MessageQueue();

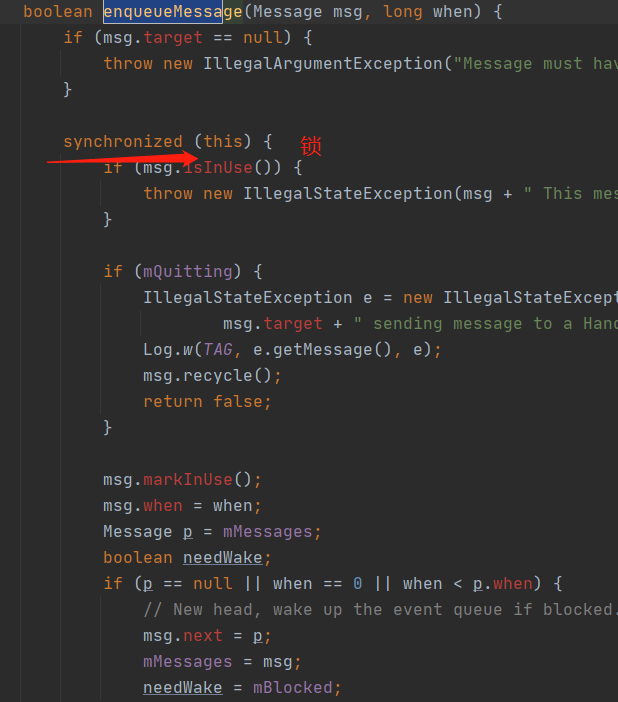
222.next可以使用

一个线程只有一个Looper，一个MessageQueue对象，而MessageQueue有一个锁，synchronize，所以这个线程只有一个地方可以调用euqueueMessage()方法。一个线程只有一个地方可以操作MessageQueue的地方

MessageQueue是保证安全的地方

然后注意一下别的锁。乐观锁。悲观锁。AQS问题

主线程不能调用quit。

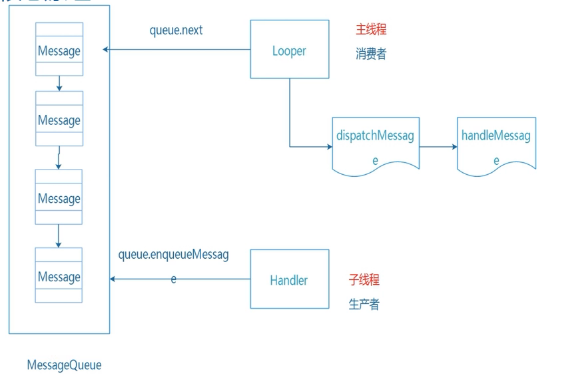


#### 7.我们使用Message时应该如何创建它？

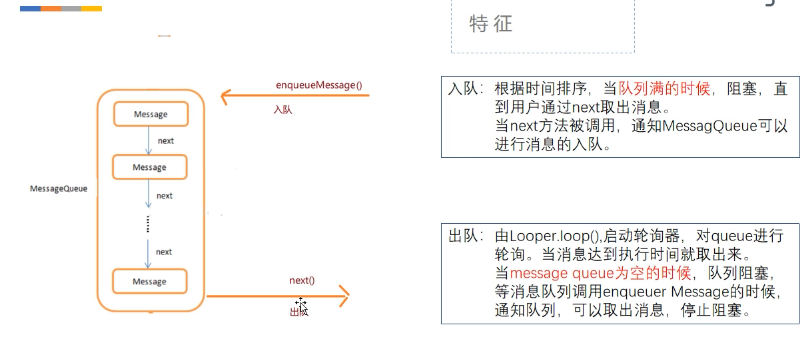
#### 8.Looper死循环为什么不会导致应用卡死？

#### 9.跨进程的核心原理

生产者-消费者设计模式：秒表，lancher,notification 1hours



#### 10.MessageQueue源码分析



生产者-消费者设计模式

两个方面的阻塞?

1. Message不到时间，自动唤醒
2. messageQueue为空，无限等待