封面

目录

[1. 功能需求说明 3](#_Toc45313168)

[1.1字符串同步显示 3](#_Toc45313169)

[2. 数据结构 3](#_Toc45313170)

[2.1字符串同步显示 3](#_Toc45313171)

[3．实验结果 3](#_Toc45313172)

[3.1字符串同步显示 3](#_Toc45313173)

[4. 关键部分逻辑 4](#_Toc45313174)

[4.1字符串同步显示 4](#_Toc45313175)

[4.1.1运行过程中设置控件不可选 4](#_Toc45313176)

[4.1.2wait notify 4](#_Toc45313177)

[4.1.3Latchdown 5](#_Toc45313178)

[4.1.4Semaphore 6](#_Toc45313179)

[5. 实验中遇到的问题及解决方案 7](#_Toc45313180)

# 功能需求说明

## 1.1字符串同步显示

1. .使用Scene Builder拖拽生成界面。
2. 完成Controller类与fxml的衔接。
3. 实现wait, notify、latchdown、semaphore进行线程的并发控制。

# 2. 数据结构

## 2.1字符串同步显示

使用了一些一维数组存储了TextField，方便调度线程操作。

# 3．实验结果

## 3.1字符串同步显示

主界面：



运行界面：（控件不可选）



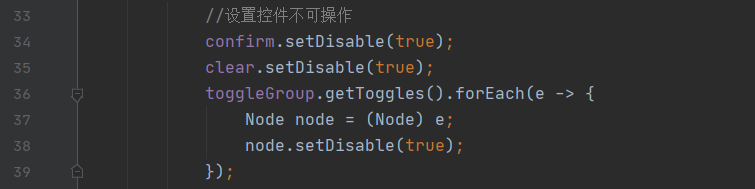
完成运行：



# 4. 关键部分逻辑

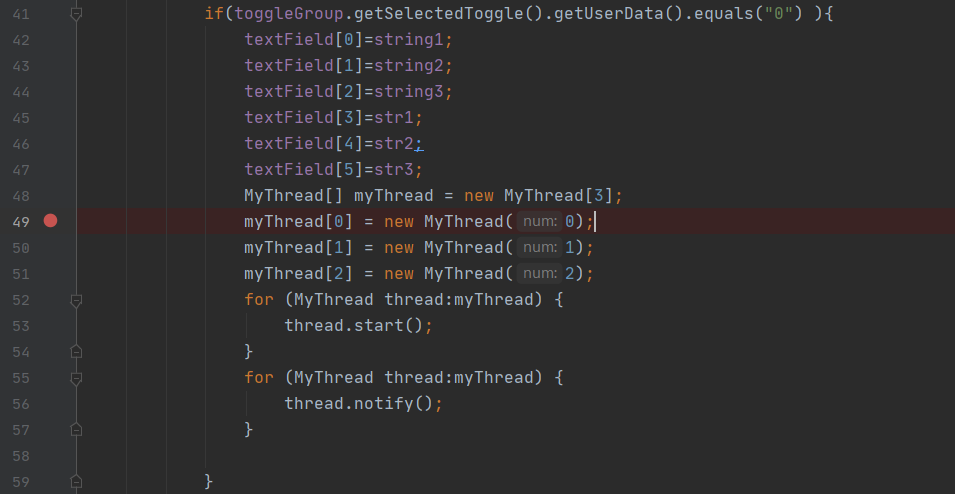
## 4.1字符串同步显示

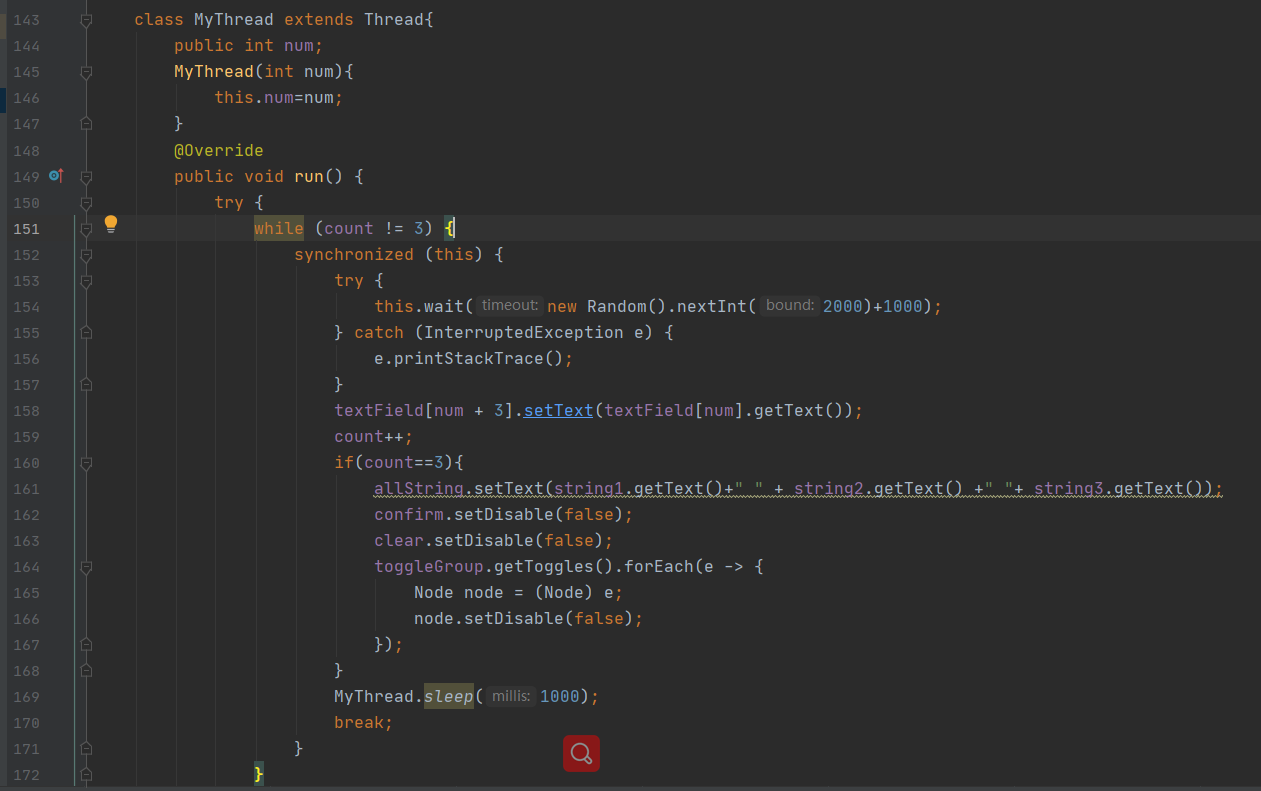
### 4.1.1运行过程中设置控件不可选



### 4.1.2wait notify

这里生成了3个线程，然后使用了notify及wait，在线程的run方法中随机生成等待的时间。





### 4.1.3Latchdown

CountDownLatch调用countDown()方法对计数减一，调用await()方法只进行阻塞，对计数没任何影响，使用latch.await() 也可以实现等待子线程先执行完，在执行主线程，类似于Thread.join()。



### 4.1.4Semaphore

Semaphore当前在多线程环境下被扩放使用，操作系统的信号量是个很重要的概念，在进程控制方面都有应用。Java 并发库 的Semaphore 可以很轻松完成信号量控制，Semaphore可以控制某个资源可被同时访问的个数，通过 acquire() 获取一个许可，如果没有就等待，而 release() 释放一个许可。比如在Windows下可以设置共享文件的最大客户端访问个数。

Semaphore维护了当前访问的个数，提供同步机制，控制同时访问的个数。在数据结构中链表可以保存“无限”的节点，用Semaphore可以实现有限大小的链表。



# 5. 实验中遇到的问题及解决方案

1.对于javafx与Controller的数据传递不太清楚

解决方法：通过百度查看一些javafx的实例，得知在Controller使用了@FXML标注，对于fxml中可以直接使用Controller类得方法。而Controller中访问fxml是通过fx:id。

2.对于多线程掌握得不太清楚

解决方法：通过查了一些线程得例题在看，并且查看线程得方法和使用用方法。然后对于学习使用wait, notify、latchdown、semaphore的使用调度方法。