****

****

**信息学院软件工程系**

**《JAVA程序设计》实验报告**

实验13

**姓名：马鑫**

**学号：37220232203780**

**学院：信息学院**

**专业：数字媒体技术**

**完成时间：2025.5.15**

**一、实验目的及要求**

* 熟悉多线程编程

**二、实验题目及实现过程**

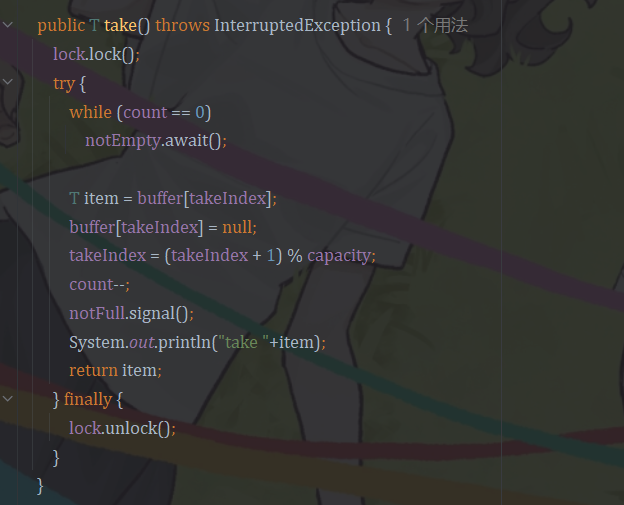
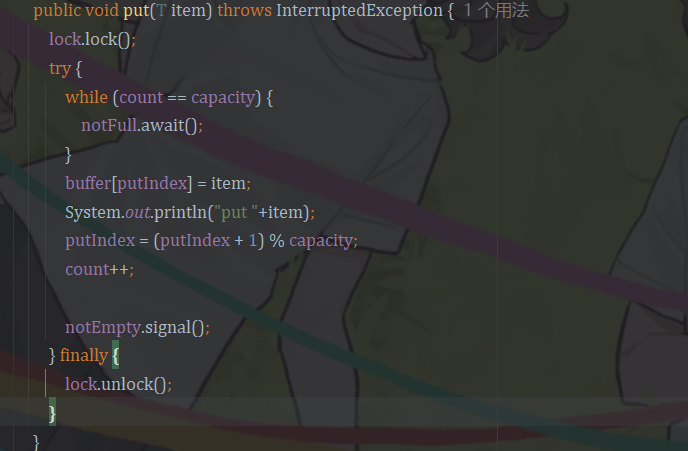
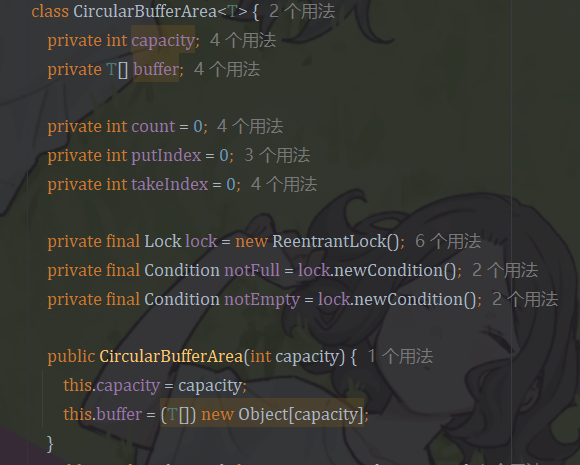
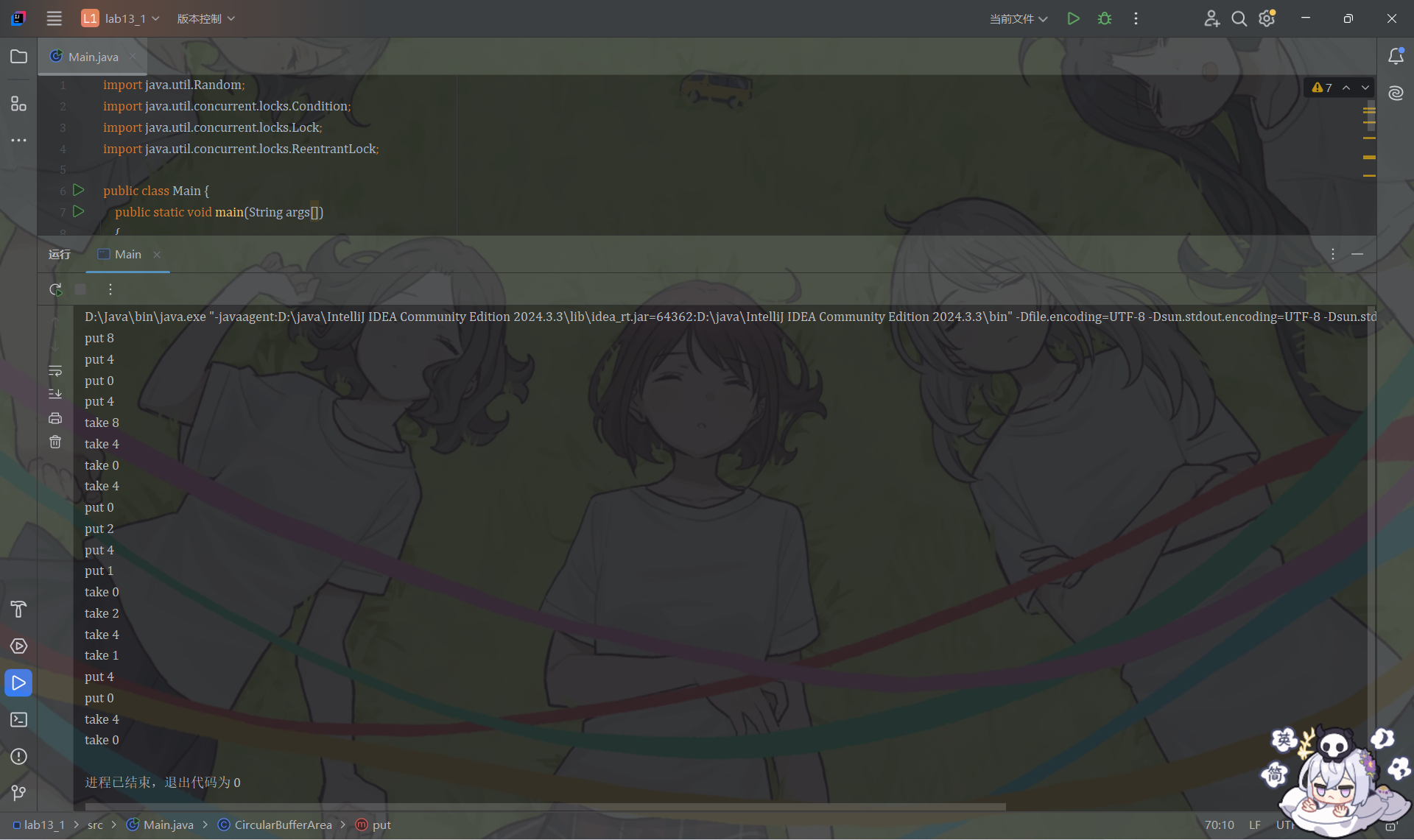
题目1：使用Lock和Condition实现循环缓冲区；

1. 实验环境（集成开发环境、jdk版本、字符编码等）

IDEA 2024.3.3、JavaSE 21、UTF-8

1. 实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

实现CircularBufferArea<T>泛型类，用数组存储数据，用两个位置变量实现循环，用Lock控制put和take的进行，用notFull和notEmpty两个Condition来控制同步。当数组满时，put函数会等待，否则添加元素，并通知notEmpty，当数组空时，take函数会等待，否则取出元素，并通知notFull。在main函数使用两个线程对同一个缓冲区分别进行存取测试。

1. 过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

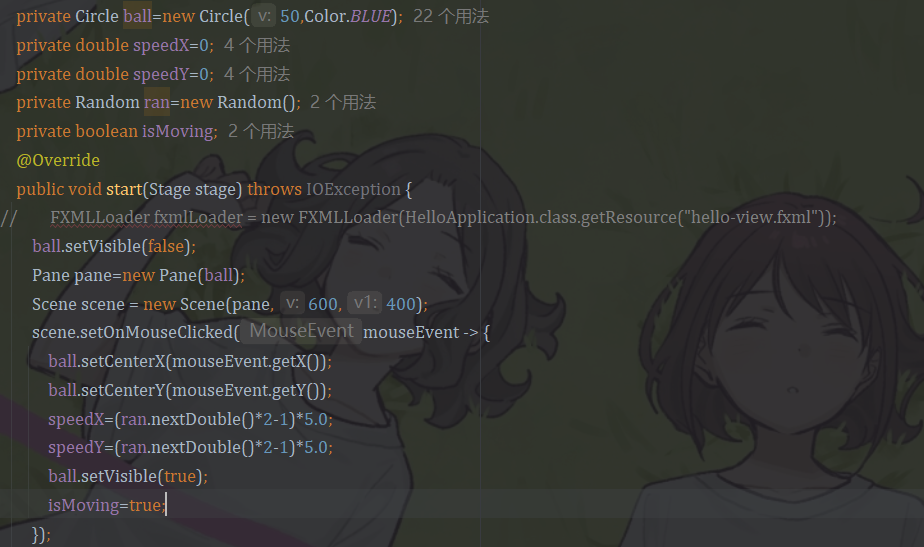
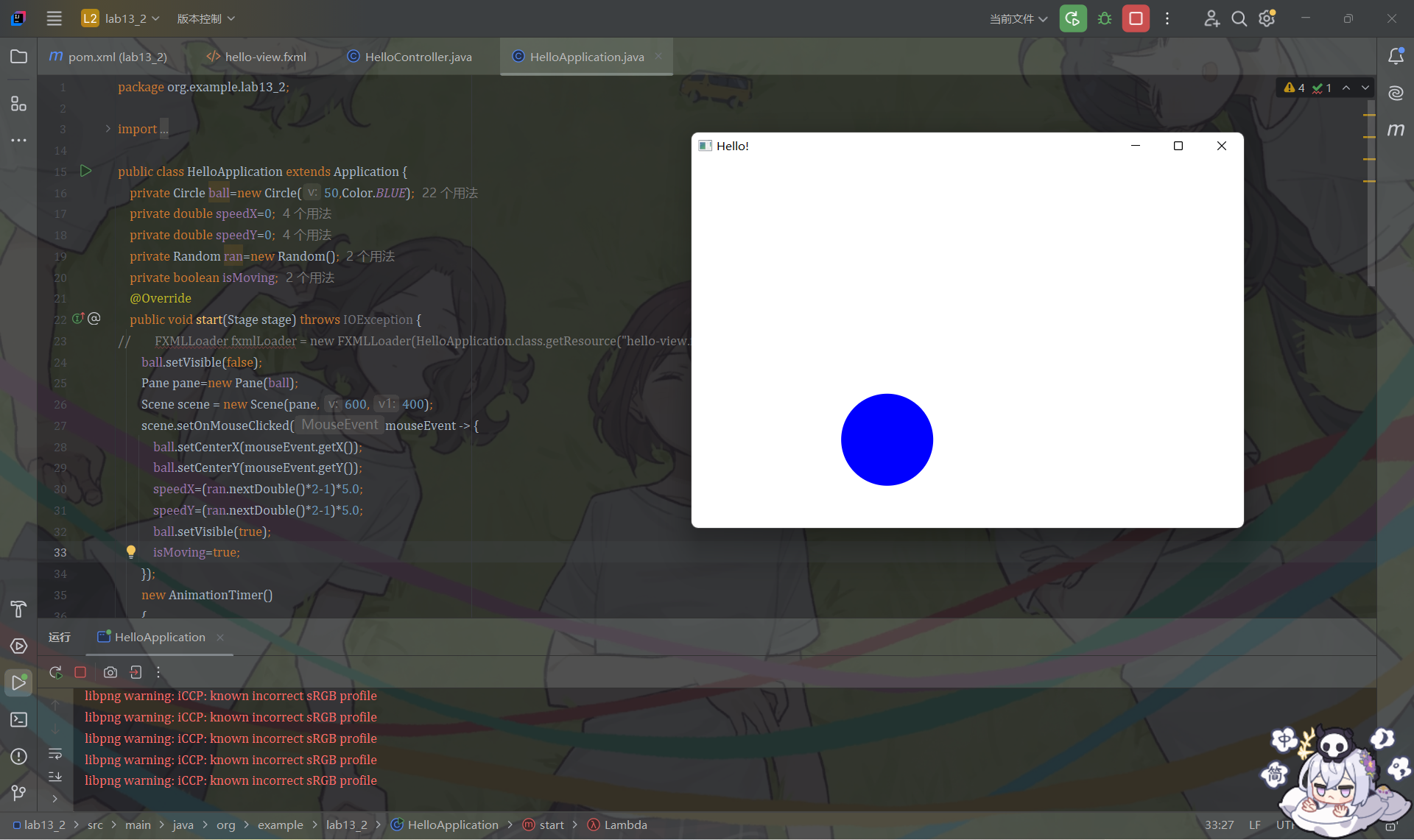
题目2：使用JavaFX编写一个程序，使一个蓝色球在面板中回弹。球应该从用户单击鼠标的位置开始移动，方向随机。当球碰到边缘时，应当反弹回来并沿相反的方向移动。

1. 实验环境（集成开发环境、jdk版本、字符编码等）

IDEA 2024.3.3、JavaSE 21、UTF-8、javafx21

1. 实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

定义全局变量球（实际上是一个二维圆），x轴速度，y轴速度，在start函数中将球设为不可见，并将其设为pane的子物体。为scene添加鼠标点击监听事件，将球的位置设在鼠标所在位置，x轴的速度和y轴速度为-5~5的随机值，并将球设为可见，isMoving改为true。用AnimationTimer事件实现动画，如果开始移动，即isMoving为true，则将球的位置在原位置x轴和y轴上分别增加speedX和speedY，当球的边缘超出的屏幕的边界，将相应轴的速度进行反向。

1. 过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

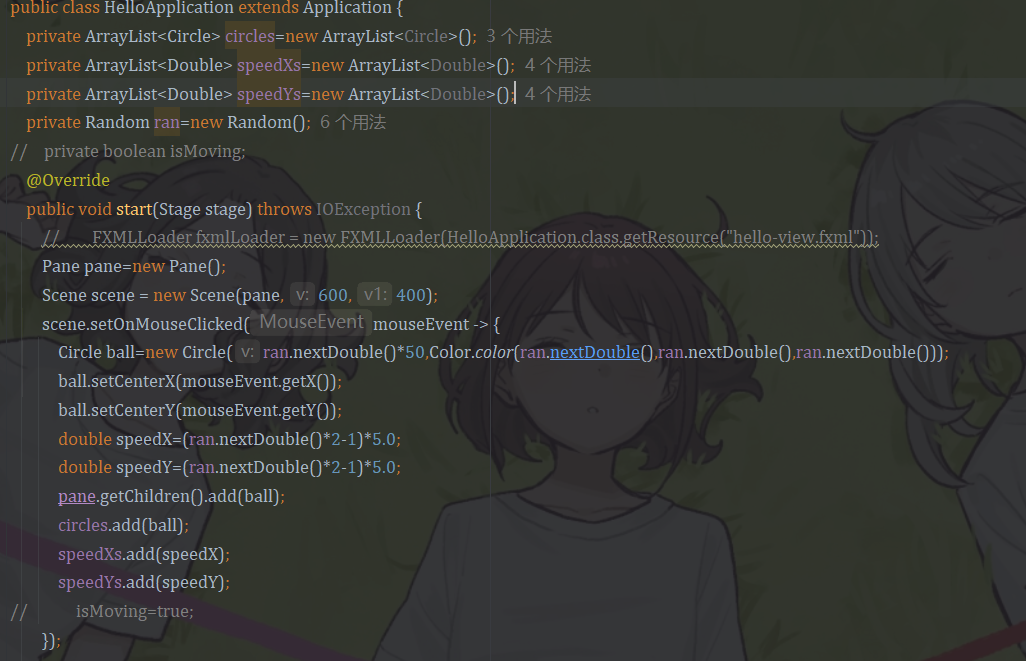
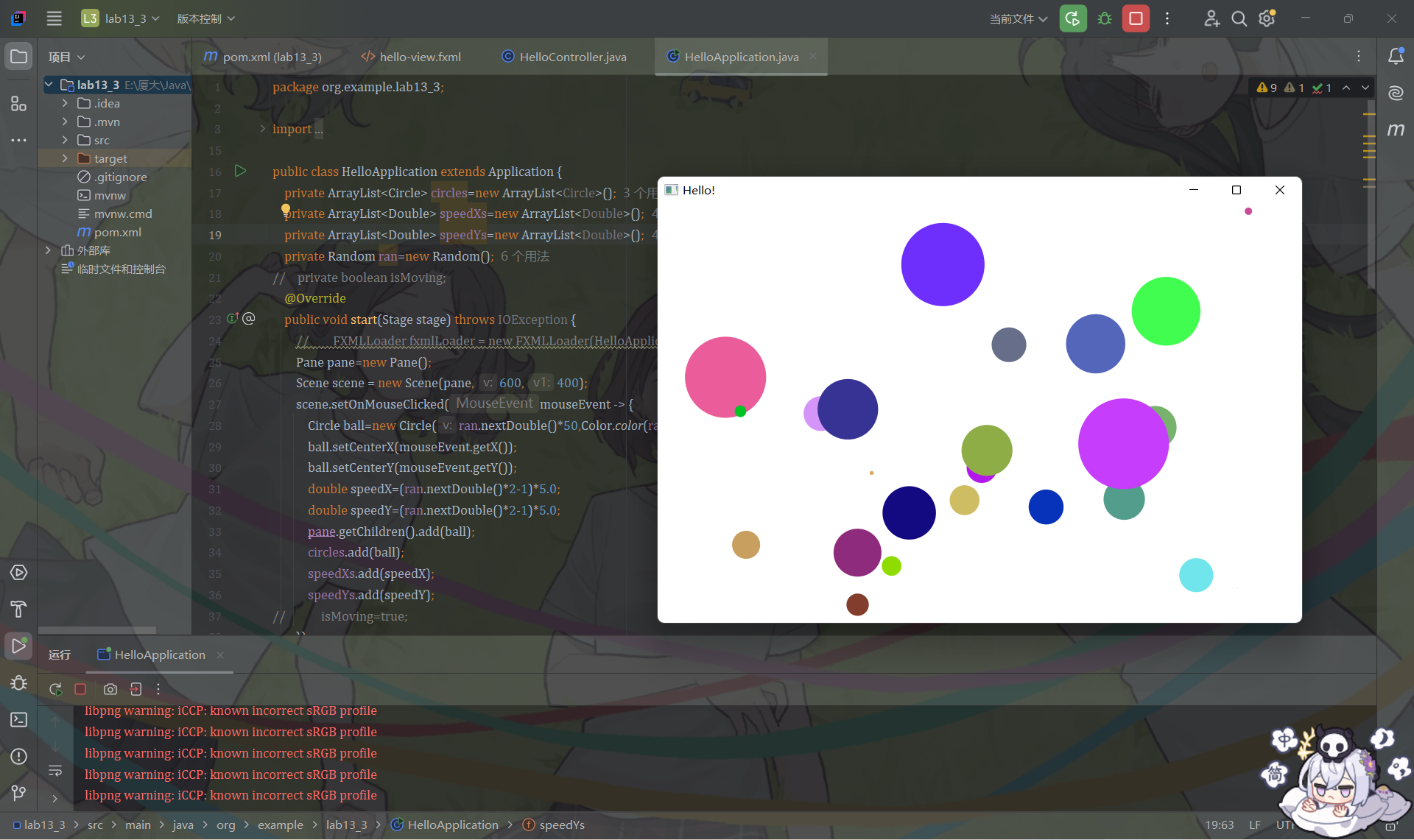
题目3：3.修改2，每次用户点击增加一个球，球的颜色随机。

1. 实验环境（集成开发环境、jdk版本、字符编码等）

IDEA 2024.3.3、JavaSE 21、UTF-8、javafx21

1. 实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

在实验2的基础上，将球用circles容器类存储，x轴速度用speedXs容器类存储，y轴速度用speedY存储。在scene的鼠标点击事件中，创建新的颜色和半径大小均随机的球，加入pane的子物体中，并加入circles容器中，并将随机生成的speedX和speedY分别加入speedXs和speedYs容器中。在AnimationTimer事件的handle函数中，遍历三个容器，对每个球进行移动和边界判断。

1. 过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

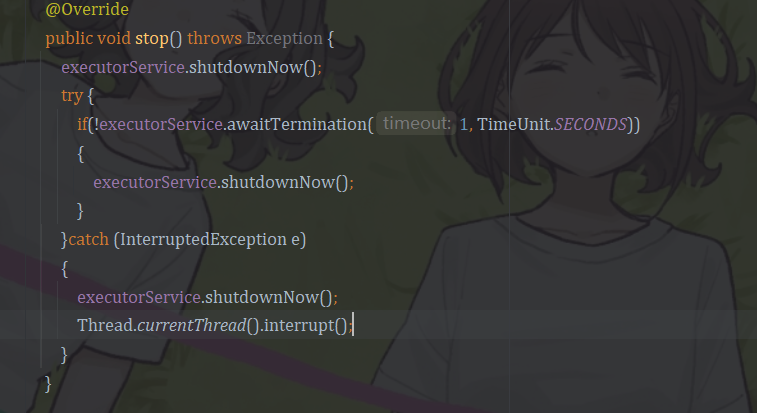
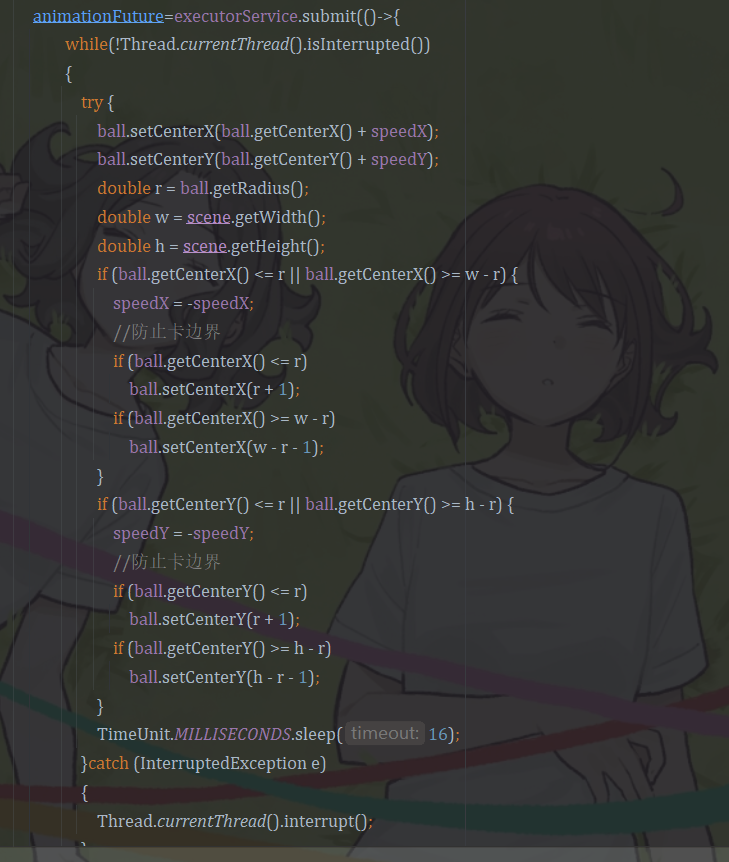
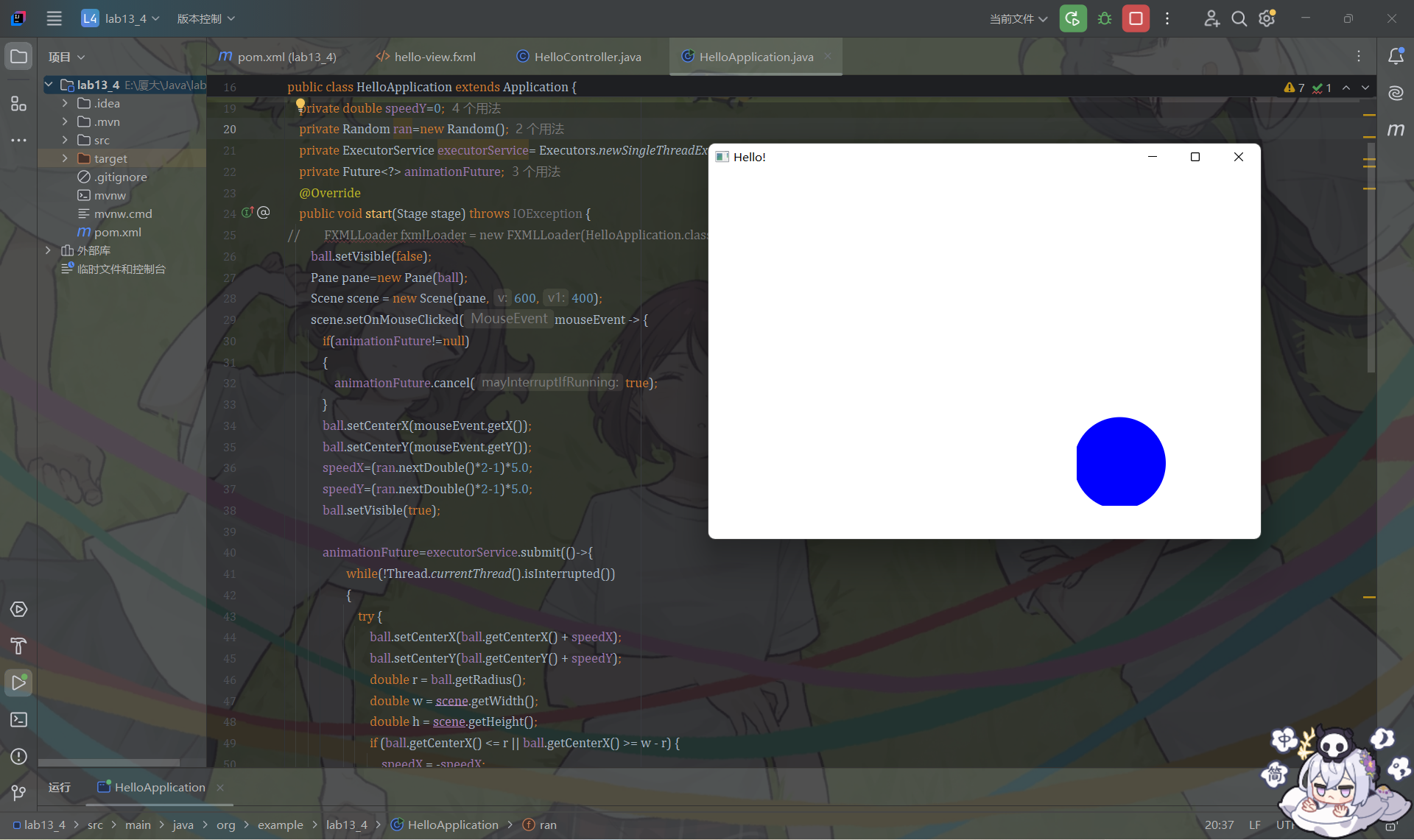
题目4：使用Callable 接口和 Future 接口改写题目2.

1. 实验环境（集成开发环境、jdk版本、字符编码等）

IDEA 2024.3.3、JavaSE 21、UTF-8、javafx21

1. 实现过程（本部分为主要评分依据，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

在实验2的基础上用线程取代AnimationTimer实现动画，当点击屏幕后，为Future接口递交Callable实现接口，用TimeUnit调整帧率。在重新点击屏幕或者停止程序时停止取消线程。

1. 过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

**三、实验总结与心得记录**

在进入线程后，java编程和其他语言不同的地方就越来越多了，线程原理其实不难理解，类似于C#的异步，但是实现方法很多，有各种继承和接口关系，函数也很多，这部分倒是挺难的。这次javafx不是在controller里进行编程，而是在生命周期中进行，有点体现动画交互与UI交互的区别了。