

**软件体系结构作业16**

姓 名 ： 洪伟鑫

专 业 ： 软件工程

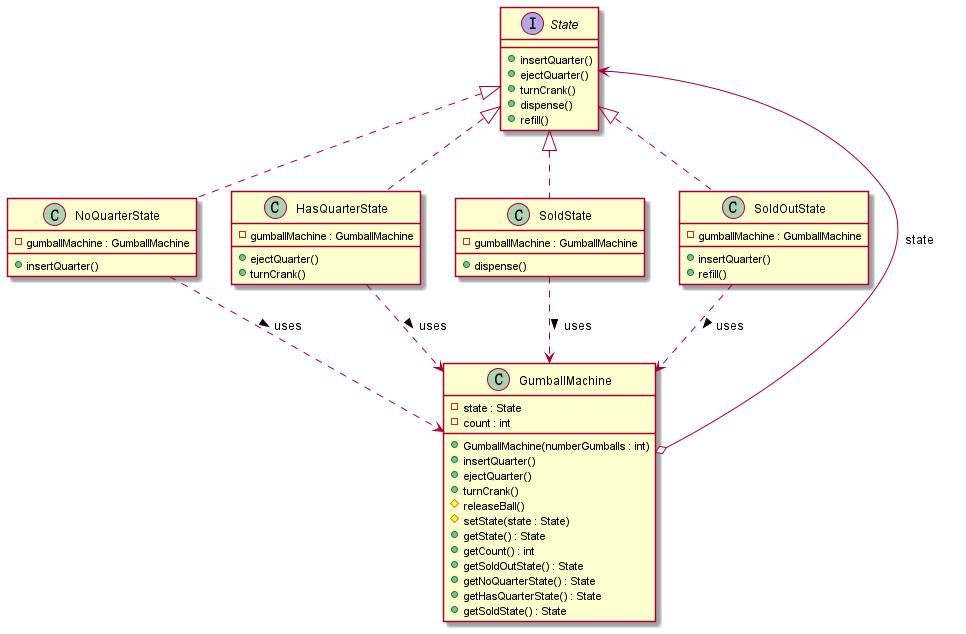
年 级 ： 2022级

学 号 ： 37220222203612

**2025年5月4日**

**1.阅读Gumballstate源码并改写成你想的（GUI ？）。**

**State模式分析:**



**该扭蛋机项目通过状态模式（State Pattern）将扭蛋机的不同状态（如无硬币、有硬币、售出糖果、售罄）封装成独立的对象，从而实现了状态与行为的分离。**

**核心类 `GumballMachine.java` 作为Context，负责维护当前状态以及所有具体状态对象的实例（ noQuarterState , hasQuarterState , soldState , soldOutState ）。它不直接处理用户的操作请求，而是将这些请求委托给当前的 state 对象。**

**所有具体状态类（如 `NoQuarterState.java` , `HasQuarterState.java` 等）均实现了统一的 `State.java` 接口，该接口定义了所有可能的操作。每个具体状态类根据自身状态实现了这些操作接口，并负责在执行操作后，通过调用 `GumballMachine.java` 的 `setState` 方法来改变扭蛋机的当前状态。这种设计使得 `GumballMachine.java` 类避免了复杂的条件判断逻辑，将状态相关的行为局部化到各个状态类中，使得系统更加灵活，易于理解和扩展。当扭蛋机的内部状态改变时，其行为也随之改变，这正是状态模式的核心思想。**

**GUI 设计：**

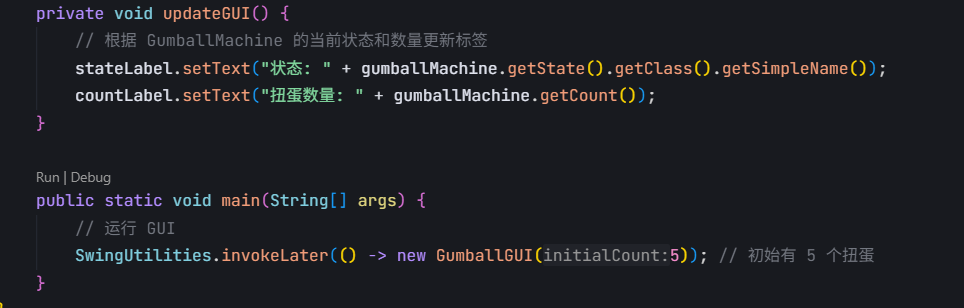
**GUI 使用 Java Swing 构建，主窗口采用 `BorderLayout` 布局，顶部放置了显示当前状态和剩余扭蛋数量的 `JLabel`，中部是一个带滚动条的 `JTextArea` 用于显示操作反馈和机器状态信息，底部则通过 `FlowLayout` 居中排列了“投入硬币”、“退回硬币”、“转动曲柄”和数量输入框。所有按钮共享同一个 `ActionListener`，在 `actionPerformed` 方法中根据事件来源调用 `GumballMachine` 相应的方法来触发状态转换，并通过调用 `updateGUI` 方法实时更新界面上的状态和数量标签，从而直观地展示了状态模式下扭蛋机随操作变化的行为。**

****

****

****

****

****

**运行截图：**

****

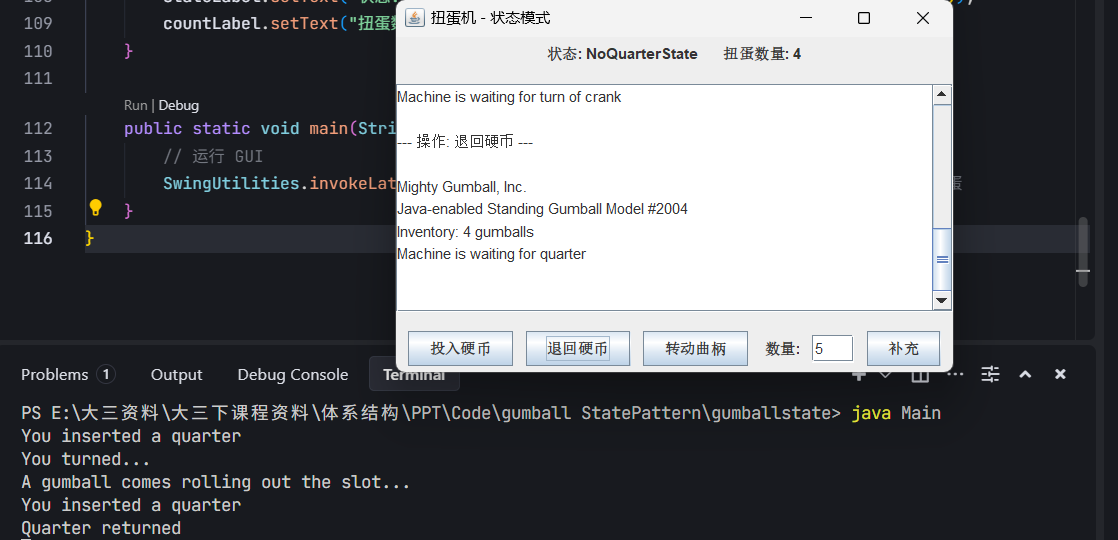
**初始状态**

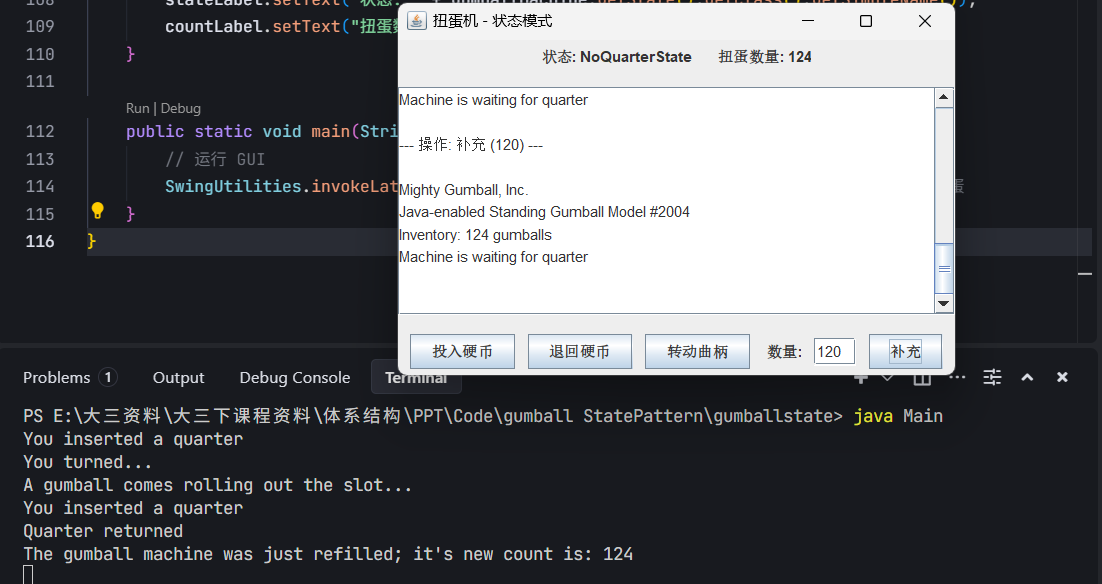
****

**投入一枚硬币**

****

**转动曲柄**

**投入后再退回**

****

**补充扭蛋币**