

实验 07：多线程

一、实验目的

通过模拟医院患者管理系统中的多线程操作，加深对 Java 线程概念、线程状态、线程调度与优先级、以及多线程编程方法的理解和应用。

二、实验学时

2 学时

三、实验类型

验证性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 IntelliJ IDEA，以及 Java 运行所需要的相关基础环境。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

设计一个简化的医院患者管理系统，其中包含一个患者等待队列和多个处理患者请求（如咨询、检查等）的线程。每个线程模拟一个医生或护士，能够处理队列中的患者请求。

六、实验内容及步骤

1、定义患者类 (Patient)：

- 包含患者的基本信息（如姓名、ID 等）。
- 实现一个方法用于模拟患者的请求处理过程（可以简单打印一条消息表示处理完成）。

2、创建等待队列：

- 使用 LinkedList 或 Queue 接口的实现类来存储等待处理的患者。
- 提供方法向队列中添加患者和从队列中移除并返回患者。

3、编写线程类 (Doctor/Nurse)：

- (1)继承自 Thread 类或实现 Runnable 接口。
- (2)在线程的 run 方法中,循环从等待队列中获取患者对象,并调用患者对象的请求处理方法。
- (3)为了模拟真实情况,可以在处理完一个患者后,让线程休眠一段时间(如使用 Thread.sleep()方法)。

4、主类设计:

- (1)创建患者等待队列。
- (2)创建并启动多个线程(医生或护士),每个线程不断从队列中取出患者并处理。
- (3)提供一种方式向队列中添加患者,以模拟新患者到达医院。

示例代码:

```
// Patient.java
public class Patient {
    private String name;
    private int id;
    public Patient(String name, int id) {
        this.name = name;
        this.id = id;
    }
    public void processRequest() {
        System.out.println("Patient " + id + " (" + name + ") is being processed.");
        // 模拟处理过程,如打印日志、执行检查等
    }
}

// Doctor.java (或 Nurse.java, 如果使用 Runnable 接口则不同)
public class Doctor extends Thread {
    private Queue<Patient> patientQueue;
    public Doctor(Queue<Patient> patientQueue) {
        this.patientQueue = patientQueue;
    }
    @Override
    public void run() {
        while (true) {
            Patient patient = patientQueue.poll(); // 尝试从队列中移除并返回患者
            if (patient != null) {
                patient.processRequest(); // 处理患者请求
                try {
                    Thread.sleep(1000); // 模拟处理时间
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt(); // 处理中断
                }
            }
        }
    }
}

// Main.java
```

```
import java.util.LinkedList;
import java.util.Queue;
public class Main {
    public static void main(String[] args) {
        Queue<Patient> patientQueue = new LinkedList<>();
        // 填充患者队列（示例）
        patientQueue.add(new Patient("Alice", 1));
        patientQueue.add(new Patient("Bob", 2));
        // ... 添加更多患者
        // 创建并启动医生线程（示例为 2 个医生）
        Doctor doctor1 = new Doctor(patientQueue);
        Doctor doctor2 = new Doctor(patientQueue); // 注意：这里共享同一个队列
        doctor1.start();
        doctor2.start();
        // 可以在此处添加代码以动态地向队列中添加患者
    }
}
```

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- （1）学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- （2）指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。
- （3）实验考核结束后，进行公布成绩。

八、创作说明

本实验指导书由河南中医药大学信息技术学院互联网技术教学团队与河南方和信息科技有限公司联合创作。

作者：黄子杰（河南方和信息科技有限公司）

审核：阮晓龙（河南中医药大学信息技术学院）

排版：刘昊鑫（河南中医药大学智能医学工程专业 2023 级）