TITLE

University Online Portal Simulation Using Multithreading in Java

VIT®

BHOPAL

Student Name: Vaibhavshri Singh

Reg no: 24BAS10058

Course: Programming in java

Date: 25 Nov 2025

AIM

To simulate a university online academic portal using Java Multithreading, where each portal task is handled by a separate thread running concurrently.

OBJECTIVES

1. Implement multiple threads in Java.

2. Simulate real-time parallel tasks of a university system.

3. Demonstrate concurrent execution of login, attendance, fee payment, marks upload, and notifications.

SOFTWARE / HARDWARE REQUIREMENTS

Software: Java JDK 8+

IDE: VS Code / IntelliJ / Eclipse / Notepad++

Hardware: Any standard PC/laptop

THEORY (Short & Simple)

Multithreading in Java allows a program to perform multiple tasks at the same time.

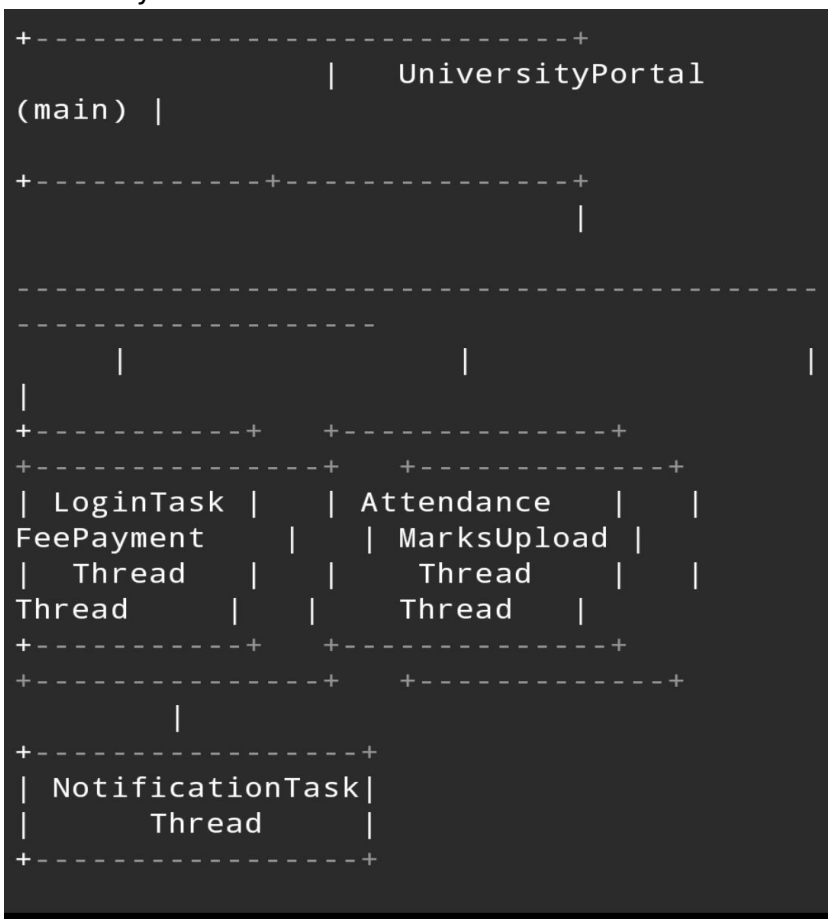
A thread is the smallest unit of a process.

Java provides two ways to create threads:

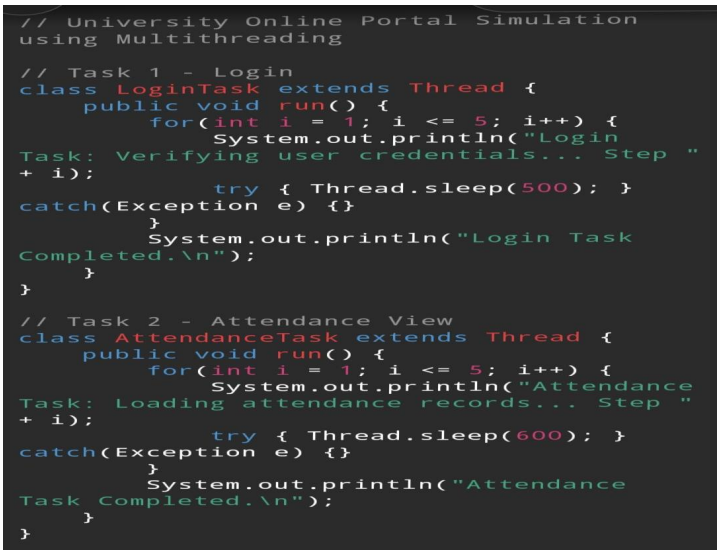
1. Extending the Thread class

2. Implementing the Runnable interface

Using threads helps simulate real-life parallel operations like a University Portal where many tasks run at the same time.



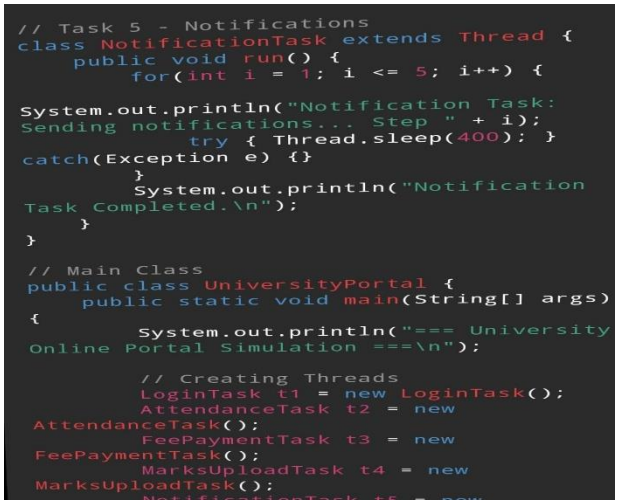
UML DIAGRAM



```
// University Online Portal Simulation using Multithreading

// Task 1 - Login
class LoginTask extends Thread {
    public void run() {
        for(int i = 1; i <= 5; i++) {
            System.out.println("Login Task: Verifying user credentials... Step " + i);
            try { Thread.sleep(500); } catch(Exception e) {}
        }
        System.out.println("Login Task Completed.\n");
    }
}

// Task 2 - Attendance View
class AttendanceTask extends Thread {
    public void run() {
        for(int i = 1; i <= 5; i++) {
            System.out.println("Attendance Task: Loading attendance records... Step " + i);
            try { Thread.sleep(600); } catch(Exception e) {}
        }
        System.out.println("Attendance Task Completed.\n");
    }
}
```

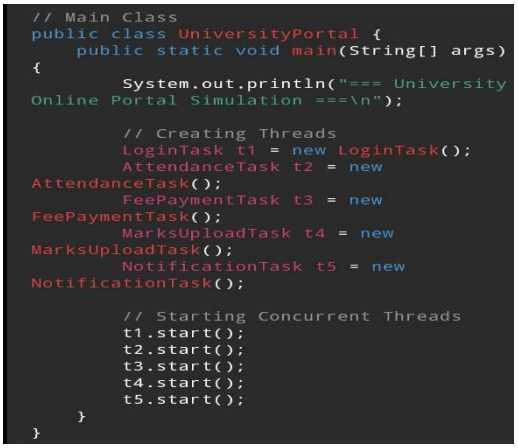


```
// Task 5 - Notifications
class NotificationTask extends Thread {
    public void run() {
        for(int i = 1; i <= 5; i++) {
            System.out.println("Notification Task: Sending notifications... Step " + i);
            try { Thread.sleep(400); } catch(Exception e) {}
        }
        System.out.println("Notification Task Completed.\n");
    }
}

// Main Class
public class UniversityPortal {
    public static void main(String[] args) {
        System.out.println("=== University Online Portal Simulation ===\n");

        // Creating Threads
        LoginTask t1 = new LoginTask();
        AttendanceTask t2 = new AttendanceTask();
        FeePaymentTask t3 = new FeePaymentTask();
        MarksUploadTask t4 = new MarksUploadTask();
```

```
// Task 4 - Marks Upload
class MarksUploadTask extends Thread {
    public void run() {
        for(int i = 1; i <= 5; i++) {
            System.out.println("Marks Upload Task: Uploading marks... Step " + i);
            try { Thread.sleep(450); } catch(Exception e) {}
        }
        System.out.println("Marks Upload Task Completed.\n");
    }
}
```



```
// Main Class
public class UniversityPortal {
    public static void main(String[] args) {
        System.out.println("=== University Online Portal Simulation ===\n");

        // Creating Threads
        LoginTask t1 = new LoginTask();
        AttendanceTask t2 = new AttendanceTask();
        FeePaymentTask t3 = new FeePaymentTask();
        MarksUploadTask t4 = new MarksUploadTask();
        NotificationTask t5 = new NotificationTask();

        // Starting Concurrent Threads
        t1.start();
        t2.start();
        t3.start();
        t4.start();
        t5.start();
    }
}
```

Output:

```
=== University Online Portal Simulation
===

Login Task: Verifying user credentials...
Step 1
Attendance Task: Loading attendance
records... Step 1
Fee Payment Task: Processing fee
payment... Step 1
Marks Upload Task: Uploading marks... Step
1
Notification Task: Sending
notifications... Step 1

... (tasks running concurrently)

Login Task Completed.
Attendance Task Completed.
Fee Payment Task Completed.
Marks Upload Task Completed.
Notification Task Completed.
```

RESULT

The University Online Portal was successfully simulated using Java Multithreading, with each task running concurrently using separate threads.

CONCLUSION

This project demonstrates the use of Java Multithreading to simulate real-world portal operations running at the same time. It shows how multiple threads can work independently yet concurrently, improving performance and efficiency.

Refrences

1. **Herbert Schildt, Java: The Complete Reference, 11th Edition, McGraw-Hill Education, 2019.**

2. **Kathy Sierra & Bert Bates, Head First Java, 2nd Edition, O'Reilly Media, 2005.**

3. **Oracle Java Documentation – Multithreading in Java.**

https://docs.oracle.com/javase/tutorial/essential/concurrency/

4. **James Gosling, Bill Joy, Guy Steele – The Java Language Specification, Oracle Press.**

5. **TutorialsPoint – Java Multithreading Tutorial.**

https://www.tutorialspoint.com/java/java_multithreading.htm

6. **GeeksforGeeks – Multithreading in Java.**

https://www.geeksforgeeks.org/multithreading-in-java/

7. **JDK Official Release Notes – Oracle Corporation.**