**Student Name:** HARINEE N J

**Registration No:** 22CSR063

**Course/Batch:** KONGU ENGINEERING COLLEGE (B.E COMPUTER SCIENCE AND ENGINEERING)

**EXERCISE 1: LOGGING ERROR MESSAGES AND WARNING LEVELS**

**Introduction:**

This Java program simulates an **ATM withdrawal system** and demonstrates how to log **errors** and **warnings** using **SLF4J**. It logs:

* Errors when the withdrawal fails due to insufficient balance.
* Warnings for low account balance after a successful withdrawal.

**Objective:**

* **Log Critical Failures**: Use logger.error() for scenarios like failed transactions or invalid PIN entries.
* **Warn About Risk Conditions**: Use logger.warn() when the user’s account balance is critically low.
* **Enhance Monitoring**: Enable developers/ops to observe system activity and potential issues in real time.

**Implementation Breakdown:**

**ATMSystem.java:**

package org.example;

import java.util.logging.Logger;

import java.util.logging.Level;

public class ATMSystem {

private static final Logger logger = Logger.getLogger(ATMSystem.class.getName());

public static void main(String[] args) {

double currentBalance = 500.0;

double withdrawAmount = 600.0;

String enteredPin = "1234";

String correctPin = "1234";

if (!enteredPin.equals(correctPin)) {

logger.log(Level.SEVERE, "Invalid PIN entered. Access denied.");

System.out.println("Error: Invalid PIN.");

} else {

if (withdrawAmount > currentBalance) {

logger.log(Level.SEVERE, "Insufficient funds. Tried to withdraw {0}, available {1}", new Object[]{withdrawAmount, currentBalance});

System.out.println("Error: Insufficient funds.");

} else {

currentBalance -= withdrawAmount;

System.out.println("Withdrawal of $" + withdrawAmount + " successful.");

System.out.println("Remaining balance: $" + currentBalance);

if (currentBalance < 100) {

logger.log(Level.WARNING, "Low balance warning: ${0}", currentBalance);

}

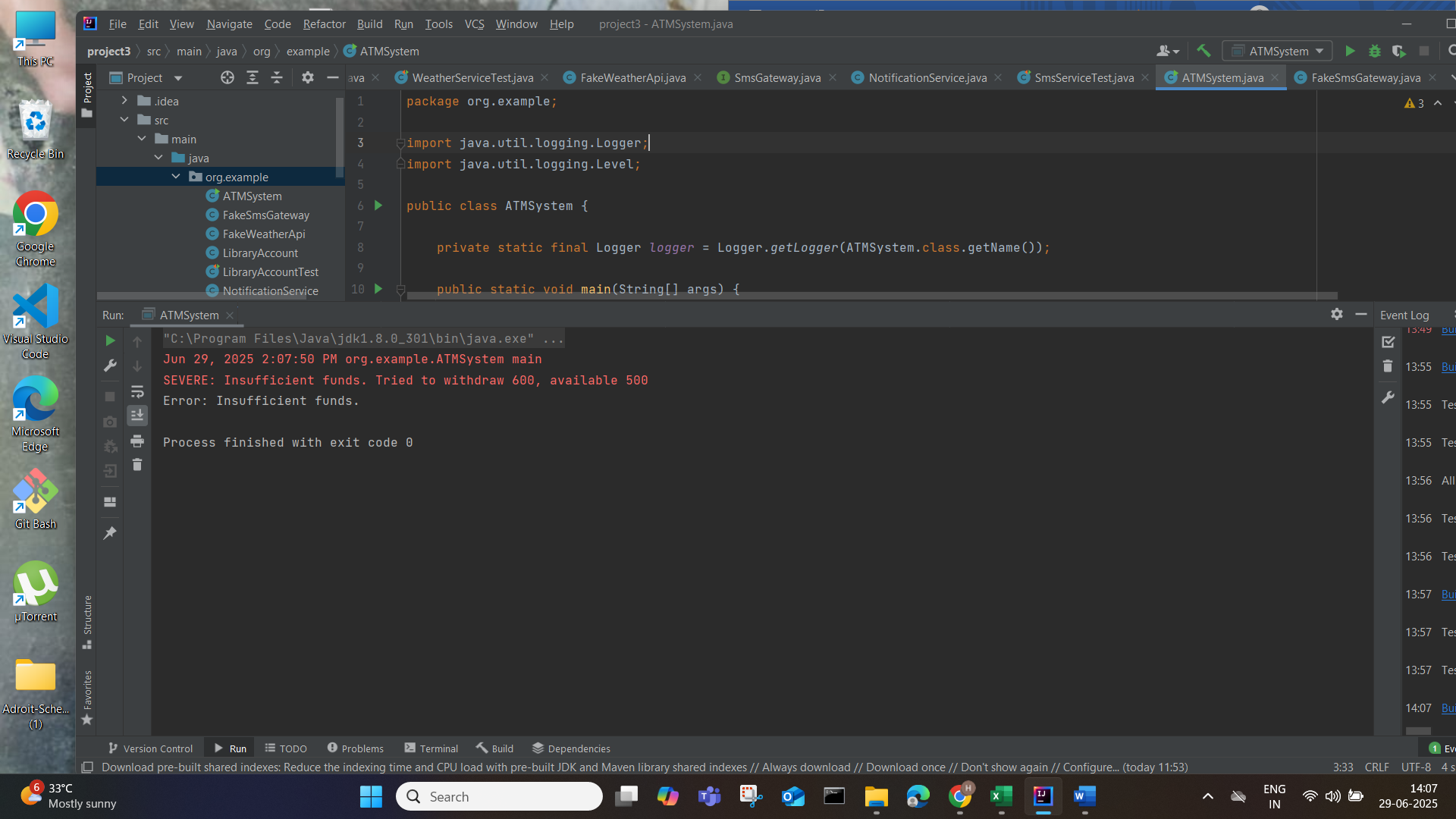
}

}

}

}

**Output:**



**Conclusion:**

This program demonstrates structured logging with SLF4J to:

* Track critical errors like incorrect PIN or insufficient funds.
* Provide warnings when balance gets low after transactions.
* Maintain observability in financial applications, ensuring security and user awareness.