**package** logger;

**import** org.junit.jupiter.api.Test;

**import** org.junit.jupiter.api.io.TempDir;

**import** java.io.IOException;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.util.List;

**import** java.util.concurrent.ExecutorService;

**import** java.util.concurrent.Executors;

**import** java.util.concurrent.TimeUnit;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**public** **class** ScoreLoggerTest {

@Test

**void** testConcurrentWrites(@TempDir Path tempDir) **throws** Exception {

Path file = tempDir.resolve("scores.csv");

ScoreLogger logger = **new** ScoreLogger(file);

List<Student> group1 = List.*of*(

**new** Student("A", "R1", 10),

**new** Student("B", "R2", 20)

);

List<Student> group2 = List.*of*(

**new** Student("C", "R3", 30),

**new** Student("D", "R4", 40)

);

ExecutorService executor = Executors.*newFixedThreadPool*(2);

executor.submit(**new** ScoreWriterRunnable(logger, group1));

executor.submit(**new** ScoreWriterRunnable(logger, group2));

executor.shutdown();

**boolean** finished = executor.awaitTermination(5, TimeUnit.***SECONDS***);

*assertTrue*(finished, "Threads did not finish properly");

**long** lineCount = Files.*lines*(file).count();

*assertEquals*(1 + group1.size() + group2.size(), lineCount, "File should contain header + entries");

}

@Test

**void** testSingleWrite(@TempDir Path tempDir) **throws** IOException {

Path file = tempDir.resolve("single.csv");

ScoreLogger logger = **new** ScoreLogger(file);

Student s = **new** Student("Zoe", "R99", 99);

logger.log(s);

List<String> content = Files.*readAllLines*(file);

*assertEquals*(2, content.size());

*assertEquals*("name,roll,marks", content.get(0));

*assertEquals*("Zoe,R99,99", content.get(1));

}

}