Sudoku Verifier - Sequence Diagram

Participants: User, App, VerifyFactory, Verifier, Checker, Result

---

Initialization

User → App: java -jar app.jar file.csv mode

Note over App: Reads command line arguments:
- CSV file path
- Verification mode (0, 3, or 27)

App → App: readBoard(csvFile)

Note over App: Parses CSV and creates 9x9 board
Validates all values are 1-9

Factory Pattern: Create Verifier

App → VerifyFactory: createVerifier(mode, board)

Note over VerifyFactory: Factory creates appropriate
verifier based on mode

alt [mode == 0 (Sequential)]

VerifyFactory → Verifier: Mode0.getInstance(board)
Verifier --> VerifyFactory: Mode0 instance
VerifyFactory --> App: verifier
App → Verifier: verify()

  loop [for i = 0 to 8]

    Verifier → Checker: Process row i
    Checker → Checker: findViolations()
    Verifier → Checker: Process column i
    Checker → Checker: findViolations()
    Verifier → Checker: Process box i
    Checker → Checker: findViolations()

[mode == 3 (Three Threads)]

VerifyFactory → Verifier: Mode3.getInstance(board)
Verifier --> VerifyFactory: Mode3 instance
VerifyFactory --> App: verifier
App → Verifier: verify()

  par [Row Thread]

    Verifier → Checker: Process all 9 rows
    Note over Checker: Thread 1: Validates all rows
    loop [for each row]
      Checker → Checker: findViolations()

  [Column Thread]

    Verifier → Checker: Process all 9 columns
    Note over Checker: Thread 2: Validates all columns
    loop [for each column]
      Checker → Checker: findViolations()

  [Box Thread]

    Verifier → Checker: Process all 9 boxes
    Note over Checker: Thread 3: Validates all boxes
    loop [for each box]
      Checker → Checker: findViolations()

Verifier → Verifier: join all threads
Note over Verifier: Wait for all 3 threads to complete

[mode == 27 (Maximum Parallelization)]

VerifyFactory → Verifier: Mode27.getInstance(board)
Verifier --> VerifyFactory: Mode27 instance
VerifyFactory --> App: verifier
App → Verifier: verify()

  par [Row Threads (9 threads)]

    loop [for each of 9 rows]
      Verifier → Checker: Thread for row i
      Checker → Checker: findViolations()

  [Column Threads (9 threads)]

    loop [for each of 9 columns]
      Verifier → Checker: Thread for column i
      Checker → Checker: findViolations()

  [Box Threads (9 threads)]

    loop [for each of 9 boxes]
      Verifier → Checker: Thread for box i
      Checker → Checker: findViolations()

Verifier → Verifier: join all 27 threads
Note over Verifier: Wait for all 27 threads to complete

Result Collection

Verifier → Checker: getResult()
Checker → Result: new Result(rowViolations, colViolations, boxViolations)
Result --> Checker: result object
Checker --> Verifier: result
Verifier --> App: result

Output

App → User: System.out.println(result)