Lab 9 — Lucky Vault: Country Edition (Scanner · Random · NIO · try-with-resources)

Build a Command-Line-Interface game that picks a **random country name** from a text file and has the player guess it.

After each guess, show (note: more details next page):

- “Wrong length (X). Need Y.”

  or

- “N letter(s) correct (right position).”

  or

- "Correct!"
  Track attempts, **save best (fewest) attempts** to a high-score file, and **log** each session’s guesses.

What you must implement

1. **Game loop**

- On start, load all country names from data/countries.txt (UTF-8 format, one per line).
- Choose one country uniformly at random.
- Print:
  - Title line: LUCKY VAULT — COUNTRY MODE. Type QUIT to exit.
  - Secret word length: <number>
  - Current best: — *or* <N> attempts (read from high score file; see #4, below).
- Repeatedly prompt: Your guess:
- Commands:
  - QUIT → exit immediately (no crash).
  - Any other input → treat as a guess (trim; compare case-insensitively).
- Responses per guess:
  - If **length differs**: print Wrong length (X). Need Y.
  - Else if **exact match**: print Correct in <attempts> attempts! Word was: <country> then:
    - If <attempts> < stored best: print NEW BEST for COUNTRY mode! and update high score.

        - End the program.
    - Else: print Not it. <N> letter(s) correct (right position).

2. **Input**

- Use Scanner(System.in) (UTF-8).
- Validate empty input → "Empty guess. Try again."

3. **Random**

- Use Random to choose the secret country.

4. **Persistence (NIO + try-with-resources)**

- **High score file**: data/highscore.txt
    - Plain text; store a single line like COUNTRY=7.
    - Read it at start (if missing or malformed → treat as "no best").
    - On win, if attempts improve the best → overwrite with new value.
- **Session log**: data/logs/YYYY-MM-DD_HH-mm-ss_COUNTRY.txt
    - Create a new file for each run.
    - Append a line per guess: timestamp + guess + outcome (wrong_length, matches=N, or CORRECT in K).
- All file I/O must use **NIO** (Path, Files, newBufferedReader/newBufferedWriter) with **try-with-resources**.
- Ensure directories exist (data/, data/logs/); create if missing.

5. **OOP structure**

- Keep logic tidy with small classes (suggested):
    - WordList (load countries),
    - HighScoreService (read/write best),
    - LoggerService (open/write session log),
    - Game (loop + rules).
- Methods short, single-purpose, clean names.

Required files & format

- data/countries.txt — one country per line (UTF-8). Use the ~200 list (same as standard UN-style list; include "Czechia", "North Macedonia", "Timor-Leste", "Eswatini", etc.).

- data/highscore.txt — created/updated by your program.
- data/logs/ — directory; each run writes a new log file.

(Tip: Normalize comparisons to lowercase; keep original for winner message if you prefer.)

Sample runs (exact phrasing required)

A) Start → Wrong length → Wrong length → Right length but wrong → Win (new best)

LUCKY VAULT — COUNTRY MODE. Type QUIT to exit.

Secret word length: 7

Current best: —

Your guess: canada

Wrong length (6). Need 7.

Your guess: united states

Wrong length (13). Need 7.

Your guess: denmark

Not it. 2 letter(s) correct (right position).

Your guess: germany

Correct in 4 attempts! Word was: germany

NEW BEST for COUNTRY mode!

B) Start with an existing best → Several wrongs → Win but not best

LUCKY VAULT — COUNTRY MODE. Type QUIT to exit.

Secret word length: 5

Current best: 3 attempts

Your guess: chile

Not it. 1 letter(s) correct (right position).

Your guess: china

Correct in 2 attempts! Word was: china

C) Quit early

LUCKY VAULT — COUNTRY MODE. Type QUIT to exit.

Secret word length: 9

Current best: —

Your guess: QUIT

Bye!

D) Bad input, then progress

LUCKY VAULT — COUNTRY MODE. Type QUIT to exit.

Secret word length: 6

Current best: —

Your guess:

Empty guess. Try again.

Your guess: france

Not it. 3 letter(s) correct (right position).

Your guess: serbia

Correct in 2 attempts! Word was: serbia

(Logs must show each guess with timestamp and outcome; highscore updates only on genuine improvement.)

Marking

- **NIO used correctly** (paths, read/write, directory create)
- **try-with-resources** for all I/O (reader/writer/scanner)
- **Scanner + Random** used properly; robust input handling
- **Persistence**: high score read/write; per-session logging
- **OOP cleanliness**: small classes, clear methods

Common pitfalls to avoid

- Forgetting UTF-8 when opening Scanner or readers/writers.
- Writing logs/highscore without closing (must use try-with-resources).
- Comparing mixed case without normalizing.
- Not validating countries.txt empty file scenario.

- Hard-coding absolute paths (use relative data/… and NIO Path).