Braille Autocorrect Java Program

Java Code for Braille Autocorrect and Suggestion System

```java
import java.util.*;

public class Main {
    private static final Map<Character, Integer> keyToDot = Map.of(
        'D', 1, 'W', 2, 'Q', 3, 'K', 4, 'O', 5, 'P', 6
    );

    private static final Map<String, Character> brailleToChar = new HashMap<>();
    private static final List<String> dictionary = List.of(
        "cat", "bat", "rat", "can", "man", "cap", "map", "mat", "cot", "cop"
    );

    static {
        brailleToChar.put("1", 'a');
        brailleToChar.put("1-2", 'b');
        brailleToChar.put("1-4", 'c');
        brailleToChar.put("1-4-5", 'd');
        brailleToChar.put("1-5", 'e');
        brailleToChar.put("1-2-4", 'f');
        brailleToChar.put("1-2-4-5", 'g');
        brailleToChar.put("1-2-5", 'h');
        brailleToChar.put("2-4", 'i');
        brailleToChar.put("2-4-5", 'j');
        brailleToChar.put("1-3", 'k');
        brailleToChar.put("1-2-3", 'l');
        brailleToChar.put("1-3-4", 'm');
        brailleToChar.put("1-3-4-5", 'n');
        brailleToChar.put("1-3-5", 'o');
        brailleToChar.put("1-2-3-4", 'p');
        brailleToChar.put("1-2-3-4-5", 'q');
        brailleToChar.put("1-2-3-5", 'r');
        brailleToChar.put("2-3-4", 's');
        brailleToChar.put("2-3-4-5", 't');
        brailleToChar.put("1-3-6", 'u');
        brailleToChar.put("1-2-3-6", 'v');
        brailleToChar.put("2-4-5-6", 'w');
        brailleToChar.put("1-3-4-6", 'x');
        brailleToChar.put("1-3-4-5-6", 'y');
        brailleToChar.put("1-3-5-6", 'z');
    }

    public static String convertBrailleToText(List<Set<Character>> brailleInput) {
        StringBuilder sb = new StringBuilder();
        for (Set<Character> cell : brailleInput) {
            List<Integer> dots = new ArrayList<>();
            for (char ch : cell) {
                if (keyToDot.containsKey(ch)) {
                    dots.add(keyToDot.get(ch));
                }
            }
            Collections.sort(dots);
            String key = dots.stream().map(Object::toString).reduce((a, b) -> a + "-" + b).
                orElse("");
```

```
            sb.append(brailleToChar.getOrDefault(key, '?'));
        }
        return sb.toString();
    }

    public static String suggestWord(String word) {
        int minDistance = Integer.MAX_VALUE;
        String closestWord = "";
        for (String dictWord : dictionary) {
            int dist = levenshteinDistance(word, dictWord);
            if (dist < minDistance) {
                minDistance = dist;
                closestWord = dictWord;
            }
        }
        return closestWord;
    }

    public static int levenshteinDistance(String a, String b) {
        int[][] dp = new int[a.length() + 1][b.length() + 1];
        for (int i = 0; i <= a.length(); i++) {
            for (int j = 0; j <= b.length(); j++) {
                if (i == 0) dp[i][j] = j;
                else if (j == 0) dp[i][j] = i;
                else if (a.charAt(i - 1) == b.charAt(j - 1)) {
                    dp[i][j] = dp[i - 1][j - 1];
                } else {
                    dp[i][j] = 1 + Math.min(dp[i - 1][j - 1],
                                Math.min(dp[i - 1][j], dp[i][j - 1]));
                }
            }
        }
        return dp[a.length()][b.length()];
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Set<Character>> brailleInput = new ArrayList<>();
        System.out.println("Enter Braille characters one by one.");
        System.out.println("For each character, type the keys pressed simultaneously (D W Q
            K O P), separated by spaces.");
        System.out.println("Press Enter on an empty line when done.");

        while (true) {
            System.out.print("Enter keys for one Braille character (or press Enter to finish
                ): ");
            String line = scanner.nextLine().trim().toUpperCase();
            if (line.isEmpty()) break;

            String[] keys = line.split("\\s+");
            Set<Character> cell = new HashSet<>();
            boolean valid = true;

            for (String key : keys) {
                if (key.length() != 1 || !"DWQKOP".contains(key)) {
                    System.out.println("Invalid key '" + key + "'. Use only D, W, Q, K, O, P
                        .");
                    valid = false;
                    break;
                }
                cell.add(key.charAt(0));
            }

            if (valid) {
                brailleInput.add(cell);
            }
        }

```

```
        if (brailleInput.isEmpty()) {
            System.out.println("No input given.");
            return;
        }

        String typedWord = convertBrailleToText(brailleInput);
        System.out.println("Typed word: " + typedWord);
        String suggestion = suggestWord(typedWord);
        System.out.println("Suggested word: " + suggestion);
    }
}
```