

```

import java.io.*;
import java.util.*;
import java.net.*;

public class URLShortener {
    private static final String DOMAIN = "http://short.url/";
    private static final String CHAR_MAP =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    private static final String FILE_NAME = "url_data.txt";

    // Thread-safe maps for storing URLs
    private static final Map<String, String> urlMap = new HashMap<>();    // short → long
    private static final Map<String, String> reverseMap = new HashMap<>(); // long → short

    public static void main(String[] args) {
        loadFromFile();
        Scanner scanner = new Scanner(System.in);

        System.out.println("=== Java Link Shortener ===");

        while (true) {
            System.out.println("\nOptions:\n1. Shorten URL\n2. Expand URL\n3. Exit");
            System.out.print("Choose an option (1-3): ");

            try {
                int choice = Integer.parseInt(scanner.nextLine().trim());

                switch (choice) {
                    case 1:
                        System.out.print("Enter URL to shorten: ");
                        String longUrl = scanner.nextLine();
                        String shortUrl = shortenURL(longUrl);
                        System.out.println("Shortened URL: " + shortUrl);
                        break;
                    case 2:
                        System.out.print("Enter short URL to expand: ");
                        String shortUrlInput = scanner.nextLine();
                        String originalUrl = expandURL(shortUrlInput);
                        System.out.println("Original URL: " + originalUrl);
                        break;
                    case 3:
                        saveToFile();
                        System.out.println("Exiting. Goodbye!");
                        scanner.close();
                        System.exit(0);
                    default:
                        System.out.println("Invalid choice. Try again.");
                }
            }
        }
    }
}

```

```

        } catch (NumberFormatException e) {
            System.out.println("Error: Enter a number (1-3).");
        }
    }
}

// Validate URL format
private static boolean isValidURL(String url) {
    try {
        new URI(url).parseServerAuthority();
        return true;
    } catch (URISyntaxException e) {
        return false;
    }
}

// Generate a 6-character hash (base62 encoded)
private static String generateHash(String url) {
    int hash = Math.abs(url.hashCode());
    StringBuilder shortHash = new StringBuilder();

    for (int i = 0; i < 6; i++) {
        shortHash.append(CHAR_MAP.charAt(hash % CHAR_MAP.length()));
        hash /= CHAR_MAP.length();
    }
    return shortHash.toString();
}

// Shorten URL with collision handling
private static String shortenURL(String longUrl) {
    if (!isValidURL(longUrl)) {
        return "Error: Invalid URL format!";
    }

    // Return existing short URL if already mapped
    if (reverseMap.containsKey(longUrl)) {
        return DOMAIN + reverseMap.get(longUrl);
    }

    // Generate unique hash
    String hash = generateHash(longUrl);
    while (urlMap.containsKey(hash)) {
        hash = generateHash(longUrl + System.currentTimeMillis()); // Append timestamp to
        avoid collision
    }

    urlMap.put(hash, longUrl);
    reverseMap.put(longUrl, hash);
}

```

```

        return DOMAIN + hash;
    }

    // Expand short URL to original
    private static String expandURL(String shortUrl) {
        if (!shortUrl.startsWith(DOMAIN)) {
            return "Error: Invalid short URL format!";
        }

        String hash = shortUrl.replace(DOMAIN, "");
        String longUrl = urlMap.get(hash);

        return (longUrl != null) ? longUrl : "Error: Short URL not found!";
    }

    // Save mappings to file
    private static void saveToFile() {
        try (PrintWriter writer = new PrintWriter(FILE_NAME)) {
            urlMap.forEach((k, v) -> writer.println(k + "," + v));
        } catch (IOException e) {
            System.err.println("Error saving data: " + e.getMessage());
        }
    }

    // Load mappings from file
    private static void loadFromFile() {
        try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",", 2);
                if (parts.length == 2) {
                    urlMap.put(parts[0], parts[1]);
                    reverseMap.put(parts[1], parts[0]);
                }
            }
        } catch (IOException e) {
            // Ignore if file doesn't exist
        }
    }
}

```