NumberGame Code Documentation

1. Introduction

The **NumberGame** is a Java-based console application that implements a number guessing game. In each round, the program generates a random number between 1 and 100. The user is given up to 10 attempts to guess the number correctly. Points are awarded based on the number of attempts used, and the game continues for multiple rounds until the user decides to stop.

2. Overview

- **Language:** Java
- **Primary Libraries:** `java.util.Scanner` for user input, `java.util.Random` for generating random numbers.
- **Game Flow:** The game consists of multiple rounds. In every round, a random number is generated, and the user makes guesses until either the correct number is found or the maximum number of attempts (10) is reached.
- **Scoring:** A score is calculated based on how quickly the correct guess is made. Fewer attempts result in a higher score.
- **User Interaction:** After each round, the user is asked whether to play another round, allowing the game to continue or end based on the user's decision.

3. Detailed Code Explanation

3.1. Initialization

- **Scanner and Random Instances:**
  - The program begins by creating an instance of `Scanner` to read input from the console.
  - A `Random` object is used to generate random numbers for each round.
- **Variables:**
  - `rounds`: Tracks the total number of rounds played.
  - `totalScore`: Accumulates the total score across rounds.

3.2. Game Loop Structure

- **Welcome Message:**
    - A welcome message is displayed to introduce the user to the game.
- **Main Game Loop:**

    - A `while (true)` loop is used to allow continuous play until the user chooses to exit.
    - Within each iteration, a new random number is generated using `random.nextInt(100) + 1`, ensuring the number is between 1 and 100.
    - The variable `attempts` is reset to zero at the beginning of each round.
    - The boolean `guessedCorrectly` is initialized to `false` to track whether the user has guessed the number correctly in the current round.

3.3. Guessing Mechanism

- **Inner Loop for Attempts:**
    - The game provides a maximum of 10 attempts (`maxAttempts`) per round.
    - The inner loop prompts the user for a guess and reads the input using the `Scanner` object.
    - After each guess:
        - The program increments the `attempts` counter.
        - It compares the user's guess with the generated number.
- **Feedback on Guesses:**
    - If the guess is correct, a congratulatory message is printed, and the number of attempts taken is displayed.
    - If the guess is too low or too high, an appropriate hint is provided to guide the user.

3.4. Scoring System

- **Score Calculation:**

When the user correctly guesses the number, the score for that round is calculated as:
(maxAttempts - attempts + 1) * 10

-
- This system rewards fewer attempts by providing a higher score for quicker guesses.
- **Failure Condition:**
    - If the user fails to guess the number within the allotted 10 attempts, the correct number is revealed.

3.5. Game Termination

- **Round Counter and Replay Prompt:**

    - After each round, the round counter (`rounds`) is incremented.
    - The user is prompted with the option to play another round.
    - If the user enters anything other than "yes" (case insensitive), the game terminates.
- **Final Output:**
    - When the game ends, a summary of the total rounds played and the accumulated score is printed in a formatted table.
    - The `Scanner` is then closed to free up system resources.