This Java program is a simple number guessing game where the player tries to guess a randomly generated target number within a specified range. The program uses a stack to keep track of the player's previous guesses.

Here's a breakdown of the elements used in the code:

1. \*\*Import Statements:\*\*

   - `java.util.Random`: This import is used to create an instance of the `Random` class for generating random numbers.

   - `java.util.Scanner`: This import is used to create an instance of the `Scanner` class for reading user input.

   - `java.util.Stack`: This import is used for creating a stack data structure to store previous guesses.

2. \*\*Variable Declarations:\*\*

   - `minNumber` and `maxNumber`: These variables define the range within which the player has to guess the number (1 to 100 in this case).

   - `Random random`: An instance of the `Random` class used to generate a random target number within the specified range.

   - `int targetNumber`: Stores the randomly generated target number.

   - `int attempts`: Keeps track of the number of attempts the player has made.

   - `int guess`: Stores the current guess made by the player.

   - `boolean isCorrect`: A flag to indicate whether the player has guessed the correct number.

   - `Scanner scanner`: An instance of the `Scanner` class used for user input.

   - `Stack<Integer> previousGuesses`: A stack to store the player's previous guesses.

3. \*\*Game Logic:\*\*

   - The program generates a random `targetNumber` within the specified range.

   - It enters a loop that continues until `isCorrect` becomes `true`.

   - Inside the loop:

     - The player is prompted to enter a guess using `scanner.nextInt()`.

     - The current guess is added to the `previousGuesses` stack.

     - The program checks if the guess is correct. If it is, `isCorrect` is set to `true`, and the loop ends. Otherwise, the program provides feedback to the player about whether the guess is too high or too low.

4. \*\*Game Outcome Display:\*\*

   - Once the player guesses the correct number, the program displays a congratulatory message along with the number of attempts it took to guess correctly.

5. \*\*Displaying Previous Guesses:\*\*

   - After the game is won, the program uses a loop to display the player's previous guesses by popping elements from the `previousGuesses` stack.

6. \*\*Closing the Scanner:\*\*

   - Finally, the `scanner` is closed to release resources.

This code demonstrates a simple text-based number guessing game, where the player can repeatedly guess numbers until they guess the correct one. The program also keeps track of and displays the player's previous guesses using a stack.