```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Random;
import java.util.Scanner;

/**
 * Pre-Conditions
 * The game starts with a list of words defined in the WORDS array,
which can be used for the unscramble task.
 * The user has access to a working console to input their guesses.
 * The user is expected to input a guess for each scrambled word
presented to them.
 * The user is expected to input either "yes" or "no" when asked if
they want to play again.
 */

public class Unscramble { //Declares the class Unscramble.
    private static final String[] WORDS = { //Declares a constant array
'WORDS' containing a list of words that will be used in the game.
            "programming", "unscramble", "computer", "keyboard",
"cookies", "random", "technology" //String lists all the words that are
going to be used in the game.
    };

    //The scrambling of the words
    public static String scramble_words(String word) { //Accepts words
as parameter.
        List<Character> letters = new ArrayList<>(); //Converts the
words in the array to a list of characters.
        for (char letter : word.toCharArray()) {
            letters.add(letter);
        }
        Collections.shuffle(letters); //Shuffles the converted list of
characters by shuffling letters.

        // Builds a scrambled version of the words by appending
shuffled characters.
        StringBuilder scrambled = new StringBuilder();
        for (char letter : letters) {
            scrambled.append(letter);
        }
        return scrambled.toString(); //Returns the scrambled word.
    }

    public static void minigameplayer() {
        Scanner170 console = new Scanner170(System.in); //Creates a
scanner object (Scanner170) to read user input through the console.
        Random random = new Random(); //Creates a random object, that
will be used to generate random letter/numbers.
        boolean playAgain = true; //Sets variable 'playAgain' to true
(meaning the game continues.)
        System.out.println("Welcome to the Unscramble Game!"); //Prints
welcome message to user in the console.

        //While loop that continues as long as 'playAgain' variable is
true.
        while (playAgain) {

            String word = WORDS[random.nextInt(WORDS.length)];
//Selects a random word from the 'WORDS' array.
            String scrambledWord = scramble_words(word); //Call
'scramble_words' to scramble the selected words and store the results
in scrambledWord.

            //Asking the user to unscramble the word
            System.out.println("Unscramble the next word: " +
scrambledWord); //Displays the scrambled word to the user in console,
and prompts user to unscramble the word.
            String userGuess = console.nextLine(); //Reads the users
guess from console input.

            //Checking if the users guess was/is correct.
            //If the users guess matches the original word in the,
print "Correct".
            if (userGuess.equalsIgnoreCase(word)) {
                System.out.println("Correct!");
            } else {
                System.out.println("Oops! You got it wrong! wooomp
woomp!\nThe correct answer was: " + word); //If not print incorrect
message to console, and show the correct answer.
            }
            //Asking if the user wants to play again
            System.out.print("Do you want to play again? (yes or no)");
            String response = console.nextLine();
            playAgain = response.equalsIgnoreCase("yes"); //If response
from the user is yes, set 'playAgain' to true and continue playing.
            // //If response from the user is anything other than yes,
set 'playAgain to false, and end the game.
        }
        System.out.println("Thanks for playing!"); //Print Thank You
message to the user.

        console.close(); //Close the scanner object.

        /** Post-Conditions
         * The user has the option to keep playing or exit after each
round based on their input.
         * After each round, if the guess is correct, the user is
informed with a "Correct!" message.
         * If the guess is incorrect, the correct answer is revealed.
         * If the user chooses to play again, the game will proceed
with a new scrambled word.
         * The game will end if the user opts to stop playing (when the
response is "no").
         * After the game ends, a "Thanks for playing!" message is
printed.
         * The Scanner object is closed, and game ends with Winner/or
Loser message displayed in console.
         */


    }
}
```