```
import java.util.Scanner;

/**
 * The user is ready to play the mini-game.
 * The words array contains a predefined set of words (e.g., "method",
"class", "software", "java", "object").
 * The program randomly selects one word from the words array to be
guessed.
 * The user has access to a console or input interface where they can
input character/symbols in order to play the game.
 * The program is initialized with 6 attempts and begins with the user
having no correct guesses.
 * The program is set up to track the userâ€™s guesses and the state of
the word being guessed, along with the stages of the hangman drawing.
 */

public class HangMan { //Defines the class Hangman.
    public static void minigameplayer() { //Defines method
minigameplayer.
        // Computer should randomly choose a word (method, class,
software, java, object).
        // User should pick a letter.
        // Computer will tell true or false.
        // If false draws character once character is complete you
lose.
        // If you get word before character the result is a win for the
user.

        Scanner170 console = new Scanner170(System.in); //Initialized
scanner object 'console' to read user input from the console.

        String[] words = {"method", "class", "software", "java",
"object"}; //Declares an array 'words' to be guessed.

        String wordToGuess = words[(int) (Math.random() *
words.length)]; //Randomly selects a word from the 'words' array as
'wordtoGuess'.
        StringBuilder maskedWord = new
StringBuilder("_".repeat(wordToGuess.length())); //Creates a
'StringBuilder' to 'maskedWords' initialized with underscores.

        int attemptsLeft = 6; //Number of user attempts before the game
is lost.
        boolean[] guessed = new boolean[26];
        String[] hangmanStages = {
                "",
                "  O",
                " O\n |",
                " O\n/|",
                " O\n/|\\",
                " O\n/|\\\n |",
                " O\n/|\\\n |\n/\\",


        };
        System.out.println("Try to guess the 5 letter word before the
man is drawn!"); //Print welcome message to console.

        while (attemptsLeft > 0) {
            System.out.println("\nWord to guess: " + maskedWord);
            System.out.println("Attempts left: " + attemptsLeft);
            System.out.println(hangmanStages[6 - attemptsLeft]);

            System.out.println("Enter a letter: ");
            char guessedLetter =
console.next().toLowerCase().charAt(0);

            if (guessedLetter < 'a' || guessedLetter > 'z' ||
guessed[guessedLetter - 'a']) {
                System.out.println("That's wrong, try a different
letter");
            }

            guessed[guessedLetter - 'a'] = true;

            boolean isCorrect = false; //Intialize boolean to track if
the guessedLetter is correct.
            for (int i = 0; i < wordToGuess.length(); i++) { //Loop
over each character/symbol in the wordToGuess.
                if (wordToGuess.charAt(i) == guessedLetter) { //If
character at current index matches to guessedLetter, update maskedWord.
                    maskedWord.setCharAt(i, guessedLetter);
                    isCorrect = true; //Set isCorrect to true.
                }
            }
            //Check if 'maskedWord' is equal to 'wordToGuess'
            if (isCorrect) {
                System.out.println("Correct!"); //If they match print
'Win message' and exit the loop.
            } else {
                System.out.println("Wrong!");
                attemptsLeft--; //If attempts left == 0, print final
hangman stage and 'Lose message' and exit the loop.
            } //You exit the loop either by winning or running out of
attemps.

            if (maskedWord.toString().equals(wordToGuess)) {
                System.out.println("You Win!"); //If matching print
'Win message' and exit game loop.
                break;
            }
        }
        if (attemptsLeft == 0) {
            System.out.println(hangmanStages[6]);
            System.out.println("You Lost! The word was " +
wordToGuess); //If no attempts are left print 'Lose message' and the
correct word.

        }
        console.close(); //Closes the scanner object console.

/**
 * If the user guesses the word before exhausting all attempts, a
message "You Win!" is displayed.
 * If the user exhausts all 6 attempts without guessing the word, the
hangman drawing is completed, and the message "You Lost! The word was
[word]" is displayed.
 * The game ends after either a win or a loss condition is met, and the
user is prompted with the result.
 * The program closes the scanner and terminates after the game ends.
 */


    }

}
```