# **CIS 36A :: LAB 06 - Strings**

#### **Student Name: Esmatullah Nickzad**

### **Task 1: Definitions & Concepts**

1. Why do Java Strings have methods but the Arrays do not?

Answer: java String have methods because they are objects and are instance of the String class which provides various built in methods for string manipulation, comparison and other operations

Arrays on the other hand , are not objects in java but rather a fundamental data type, and they have methods attached to them directly .

1. What are the differences between String length and Array length?

Answer: the length property in java String refers to the number of characters in the string, while the length in Array refers to the number of elements in the array.

String length: tells you how many letters, spaces, or symbols are in the text phrase.

Array length: tells you how many items (numbers, characters or even other arrays) are stored withing the array.

### **Task 2: Understanding Programming**

Instructions: Answer each question below. Try to understand and explain the code. **Do not put an IDE code screenshot.**

No Task 2.

### **Task 3: Programming Exercises**

Instructions: Use any text editor to write and execute below exercises from the book chapter 5. Attach Snipping photos of your source code and execution of the code in the console. Make sure to create separate files for each exercise.

**Chapter Exercises: Do the following chapter exercises.**

* Exercise 07: SimpleCipher

A screen shot of a computer program

Description automatically generated

* Exercise 20: Bubble Sort for Strings

A computer screen shot of a program

Description automatically generated

* Exercise 28: Palindrome String

A computer screen shot of a program

Description automatically generated

* Exercise 29: String to Array of characters

A screenshot of a computer program

Description automatically generated

### **Task 4: Programming Application**

**Instructions:** Use any IDE to write and execute the program below. Attach Snipping photos of your source code and execution of the code in the console.

**For this Task only, submit your .java file as well.**

**Hangman Game:** Using Strings and strings methods, design a hangman game.

1. You should have a list of words to choose from (at least 20 words between lengths of 5 and 7). When each game starts it should choose a word from the list randomly.
2. Create a blank string with the exact size of your chosen word and fill it with underscores.
3. After each correct guess, place all of the occurrences of the correct letter in the blank string and display it. (Hint: Use substring method to reconstruct the blank string)
4. If the user guesses a wrong letter, show a console-based hangman illustrating the progress. Show this illustration for correct guesses as well.

Please see the sample display below.

|  |  |  |
| --- | --- | --- |
| **Start Screen** | **Progress Screen** | **Lost Screen** |
| H A N G M A N  +---+  |  |  |  ======  Missed letters:  \_ \_ \_ \_ \_  Guess a letter: | +---+  0 |  | |  |  ===  Missed letters: o r  \_ a t  Guess a letter: | +---+  0 |  /|\ |  / \ |  ======== |

Note: You may create a method to display this illustration.

1. Keep all of the user guesses in an array of strings (or chars) and warn the user if they entered the same letter again.
2. The game should repeat until the user guesses the word or loses the game. Allow users to miss up to four or five letters.
3. Add one more feature of your own.
4. Good Luck and Have Fun!!

Answer:

import java.util.Scanner;

import java.util.Random;

public class HangmanGame {

private static final String[] WORDS = {"ahmad", "mahmood", "esmatullah", "nickzad", "hamza","qodus","hakim",

"mateen", "aslam", "nazeer", "basheer", "amina","saber","marjan",

"latifa", "siyar", "sultan", "qasim", "hashim","kabir","hashmatullah",

"manan", "monisa", "latif", "marwa", "hikmatullah", "sadaf", "osman"};

private static final int MAX\_TRIES = 5;

private static final String[] HANGMAN\_DRAWINGS = {

" |===|\n" +

" | |\n" +

" | O\n" +

" | /|\\\n" +

" | / \\\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" | O\n" +

" | /|\\\n" +

" | /\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" | O\n" +

" | /|\\\n" +

" |\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" | O\n" +

" | /|\n" +

" |\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" | O\n" +

" | |\n" +

" |\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" | O\n" +

" |\n" +

" |\n" +

" |\n" +

" /|\\=====",

" |===|\n" +

" | |\n" +

" |\n" +

" |\n" +

" |\n" +

" |\n" +

" /|\\====="

};

private static final Random random = new Random();

private static String secretWord;

private static char[] guessedLetters;

private static int triesLeft;

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Welcome to Hangman Game!");

System.out.println("Try to guess the name within " + MAX\_TRIES + " tries.");

while (true) {

secretWord = getRandomWord();

guessedLetters = new char[secretWord.length()];

triesLeft = MAX\_TRIES;

for (int i = 0; i < guessedLetters.length; i++) {

guessedLetters[i] = '\_';

}

while (triesLeft > 0) {

System.out.println("\n" + HANGMAN\_DRAWINGS[MAX\_TRIES - triesLeft]);

displayWord();

System.out.println("Tries left: " + triesLeft);

System.out.print("Missed letters: ");

displayMissedLetters();

char guess = getGuess(scanner);

if (!processGuess(guess)) {

triesLeft--;

}

if (isWordGuessed()) {

System.out.println("\nCongratulations! You guessed the name: " + secretWord);

break;

}

}

if (triesLeft == 0) {

System.out.println("\nSorry, you've run out of tries. The name was: " + secretWord);

}

System.out.print("\nDo you want to play again? (yes/no): ");

String playAgain = scanner.next().toLowerCase();

if (!playAgain.equals("yes")) {

System.out.println("Thanks for playing Hangman!");

break;

}

}

}

private static String getRandomWord() {

return WORDS[random.nextInt(WORDS.length)];

}

private static void displayWord() {

for (char letter : guessedLetters) {

System.out.print(letter + " ");

}

System.out.println();

}

private static void displayMissedLetters() {

// Display missed letters

for (char letter : guessedLetters) {

if (letter == '\_') {

System.out.print("\_ ");

}

}

System.out.println();

}

private static char getGuess(Scanner scanner) {

char guess;

while (true) {

System.out.print("Guess a letter: ");

String input = scanner.next().toLowerCase();

if (input.length() == 1 && Character.isLetter(input.charAt(0))) {

guess = input.charAt(0);

break;

} else {

System.out.println("Invalid input. Please enter a single letter.");

}

}

return guess;

}

private static boolean processGuess(char guess) {

boolean isCorrectGuess = false;

for (int i = 0; i < secretWord.length(); i++) {

if (secretWord.charAt(i) == guess) {

guessedLetters[i] = guess;

isCorrectGuess = true;

}

}

return isCorrectGuess;

}

private static boolean isWordGuessed() {

for (char letter : guessedLetters) {

if (letter == '\_') {

return false;

}

}

return true;

}

}

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated