Andy Cherney

10/16/2023

Week3Meet - 10 pts

Turn in on BBL as soon as complete, but before end of day Sunday following the lecture.

===============================

Reading a program

Read both programs. Be the computer and trace which line of code is executing in order. Keep track of what the current state of each variable is.

import java.util.Scanner;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PuzzlePage.java

\* Creates word puzzles for the Triangle

\* @author Tammy Pirmann

\* @version 20210407

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class PuzzlePage {

public static void main(String args[]){

Scanner keyboard = new Scanner(System.in);

WordScramble puzzle = new WordScramble();

System.out.print("Enter the solution word: ");

String solution = keyboard.nextLine(); // Let solution = “Andy”

System.out.printf("Your puzzle is: %s", puzzle.scramble(solution));

}

}

The main part of the code creates a WordScramble object. Then the program prompts the user to enter a word. Finally, the program scrambles the word and displays the new scrambled string to the user.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* WordScramble.java, Scrambles a given word

\* @author Tammy Pirmann

\* @version 20210407

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class WordScramble {

private String solution;

//constructor

public WordScramble() {

solution = "NA";

}

//Setter

public void setSolution(String str){

solution = str.toUpperCase();

}

//Scrambles the solution String

public String scramble(String str){

setSolution(str);

String mix;

int a = solution.indexOf("A");

if (a >= 0) {

mix = solution.substring(a).concat(solution.substring(0,a));

}

int e = solution.indexOf("E");

if (e >= 0) {

mix = solution.substring(e).concat(solution.substring(0,e));

}

int i = solution.indexOf("I");

if (i >= 0) {

mix = solution.substring(i).concat(solution.substring(0,i));

}

int oh = solution.indexOf("O");

if (oh >= 0) {

mix = solution.substring(oh).concat(solution.substring(0,oh));

}

int u = solution.indexOf("U");

if (u >= 0) {

mix = solution.substring(u).concat(solution.substring(0,u));

}

//reverse it in case it still looks like the original word

return reverse(mix);

}

//helper method to reverse the scrambled string

private String reverse(String str){

//base case

if (str.isEmpty()){

return str;

}

//Recursive call

return reverseString(str.substring(1)) + str.charAt(0);

}

1. The only attribute of the Word scramble object is the solution which is “NA”
2. There is a method that converts the scrambled string to all uppercase letters.
3. The main part of the object is the scramble method.
   1. First, it runs the setSolution method to set the solution string to the uppercase version of the string
   2. A string called mix is declared
   3. Main part of the class is the scramble method
      1. Checks the solution string and scrambles based on the indexes of the vowels:
         1. If A is in the string (index of first occurrence >= 0), then mix is equal to the string starting from that letter to the end of the string combined with the start of the string to the first index of A (equivalent to reversing)
         2. It continues to do this pattern all the way until the last vowel is found, in which case, that will be the final version of the scrambled string
         3. The string is then reversed using a helper method called reverse just in case the final scramble word looks like the original
   4. The reverse method is a recursive method that reverses the string. The reverse function is called again on the substring starting from the second character onward. The first character is concatenated with the parameter from that function call. This operation continues until there is no substring left in the parameter.

Everything below this line relates ONLY to the problem discussed in class, the code we wrote together.

I understand the problem introduced in class to be: (in your own words)

We need to update our paint program to have a wall class. This wall class must allow a user to set and get a height and width in feet. The class should also allow the user to calculate the area of the wall.

My UML Diagram for these classes: (feel free to paste in a photo of a hand done diagram)

Wall

-height double

-width double

-area() : double

The solution to the problem were the following programs: (provide the names of the .java files only)

PaintProgramWithClass.java

Wall.java

I tested the solution with at least 3 different value sets. The test data and results are:

(use this format: var1 = data, var2 = data, etc -> result)

width = 12, height = 10, -> 120.0

width = 20, height = 4, -> 80.0

width = 35, height = 34, -> 1155.0

Reflect on your problem-solving:

How confident are you in the solution? Pretty confident, but more test cases and considerations are always good to have

How often do you go to the Internet when working on your lab?

Sometimes I go on the Internet in order to look up syntax, but that is the only thing I use it for.

How often do you refer to programs from class or the text when working on your lab?

I often review them to enhance what I am learning about

Reflect on your learning and your needs. After this class meeting, what topics do you feel like you learned and what topics do you feel like you need more information on to learn?

I learned how to write classes and if statements. I would say I need to work on recursion more, but I think this is better learned through practice.