CS 1073

FR03A

Assignment #6

Ethan A. McCarthy

3573807

# Section 1

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.FlowPane;

import javafx.scene.text.Text;

import javafx.stage.Stage;

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

 public class Octal2DecimalGUI extends Application{

    private TextField inputField;

    private Text topText;

    private Text outpuText;

    public void start(Stage primaryStage){

        primaryStage.setTitle("Ocatal to Decimal Converter");

        Button convert = new Button("Convert Octal to Decimal");

        convert.setOnAction(this::convert2Decimal);

        inputField = new TextField();

        inputField.setPrefWidth(150);

        inputField.setOnAction(this::convert2Decimal);

        topText = new Text("Enter a number in octal to convert.");

        outpuText = new Text("Welcome to the converter app!");

        FlowPane pane = new FlowPane(topText, inputField, convert, outpuText);

        pane.setAlignment(Pos.CENTER);

        pane.setHgap(10);

        pane.setVgap(30);

        Scene scene = new Scene(pane, 300, 300);

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public void convert2Decimal(ActionEvent event){

        boolean invalid = false;

        int decimal = 0;

        int count = 0;

        int numberIn = Integer.parseInt(inputField.getText());

        while (numberIn > 0){

            int x = numberIn % 10;

            if(x > 7){

                invalid = true;

                break;

            }

            decimal += (int) (x \* Math.pow(8, count));

            numberIn = numberIn / 10;

            count++;

        }

        if (invalid){

            outpuText.setText("Invalid input. Please enter an octal number.");

        }

        else{

            outpuText.setText(decimal + "");

        }

    }

}

# Section 2

Graphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

# Section 3

import java.util.Scanner;

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

 public class NameAnalyzer{

    public static void main(String[] args){

        Scanner scan = new Scanner(System.in);

        boolean adjacent = false;

        boolean phonic = false;

        String longestName = "";

        System.out.println("Enter your Surname, Given Name, and then your nicknames.");

        String input = scan.nextLine();

        int size = input.split(",").length;

        String[] items = input.split(",");

        for(int i = 1; i < items[1].length(); i++){

            char char1 = items[1].charAt(i);

            char char2 = items[1].charAt(i-1);

            if (char1 == char2){

                adjacent = true;

                break;

            }

        }

        if(items[1].charAt(items[1].length() - 1) == items[0].charAt(0 )){

            phonic = true;

        }

        for(int i = 2; i < size; i++){

            if(items[i].length() > longestName.length()){

                longestName = "(" + items[i] + ")";

            }

        }

        System.out.println(items[1].toUpperCase() + " " + items[0].toUpperCase() + " " + longestName);

        System.out.println("Repeated Adjacent in Given: " + adjacent);

        System.out.println("Phonic Flow: " + phonic);

        System.out.println("Amount of Nicknames: " + (size - 2));

    }

 }

# Section 4

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated