Stone Barrett

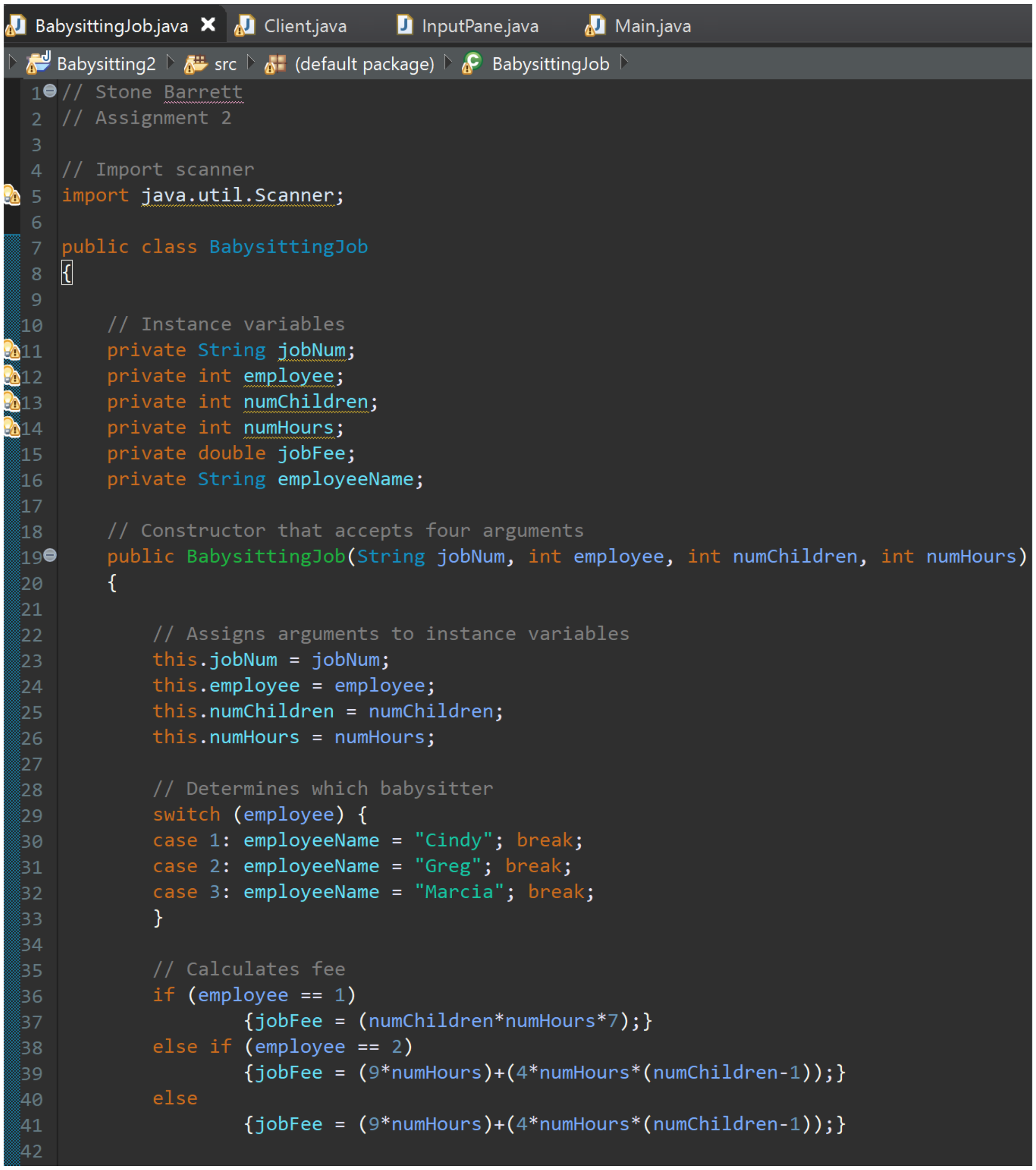
Assignment 02

6/9/19

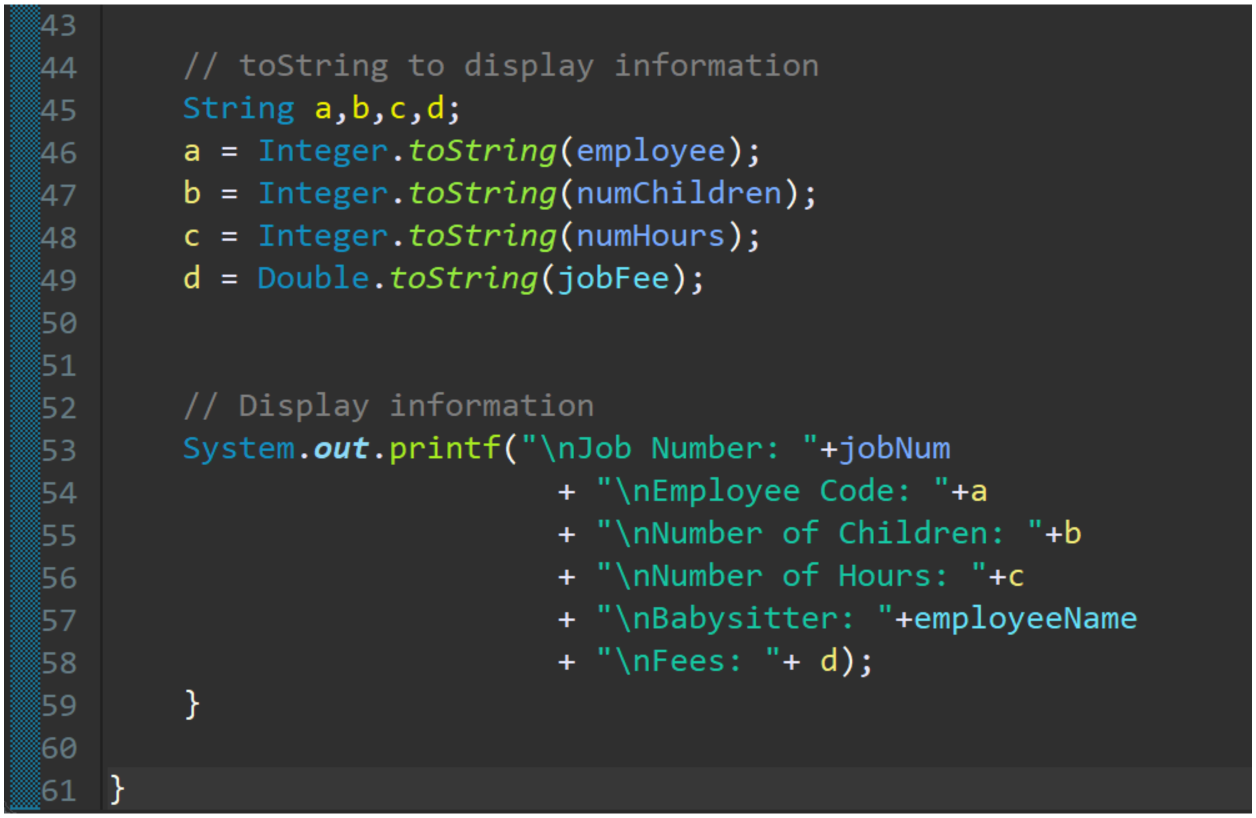
Problem 1

As the prompt instructed, I created a class for the babysitting job that uses four arguments and assigns them to instance variables. This class also assigns the babysitter numbers to their respective names and displays all the information as strings after the user interaction is finished. Per the problems statement again, I also created a client class that interacts with a user to receive the information associated with the babysitting job. This information is the babysitter number, the year, the job number, the number of children, and the hours spent babysitting.

Screenshots of the code:

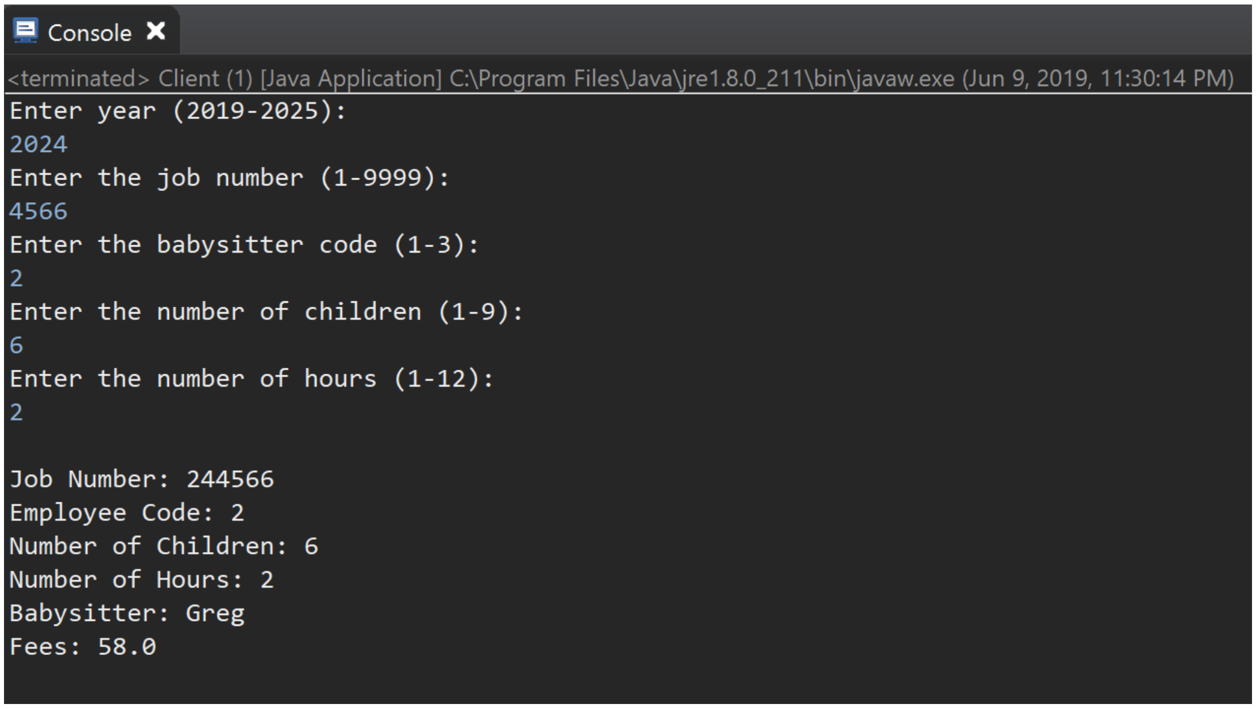


```
// Stone Barrett
// Assignment 2

// Import scanner
import java.util.Scanner;

public class BabysittingJob
{

    // Instance variables
    private String jobNum;
    private int employee;
    private int numChildren;
    private int numHours;
    private double jobFee;
    private String employeeName;

    // Constructor that accepts four arguments
    public BabysittingJob(String jobNum, int employee, int numChildren, int numHours)
    {

        // Assigns arguments to instance variables
        this.jobNum = jobNum;
        this.employee = employee;
        this.numChildren = numChildren;
        this.numHours = numHours;

        // Determines which babysitter
        switch (employee) {
        case 1: employeeName = "Cindy"; break;
        case 2: employeeName = "Greg"; break;
        case 3: employeeName = "Marcia"; break;
        }

        // Calculates fee
        if (employee == 1)
                {jobFee = (numChildren*numHours*7);}
        else if (employee == 2)
                {jobFee = (9*numHours)+(4*numHours*(numChildren-1));}
        else
                {jobFee = (9*numHours)+(4*numHours*(numChildren-1));}

```



```java

    // toString to display information
    String a,b,c,d;
    a = Integer.toString(employee);
    b = Integer.toString(numChildren);
    c = Integer.toString(numHours);
    d = Double.toString(jobFee);


    // Display information
    System.out.printf("\nJob Number: "+jobNum
                        + "\nEmployee Code: "+a
                        + "\nNumber of Children: "+b
                        + "\nNumber of Hours: "+c
                        + "\nBabysitter: "+employeeName
                        + "\nFees: "+ d);
    }

}
```

Screenshot of a sample run:

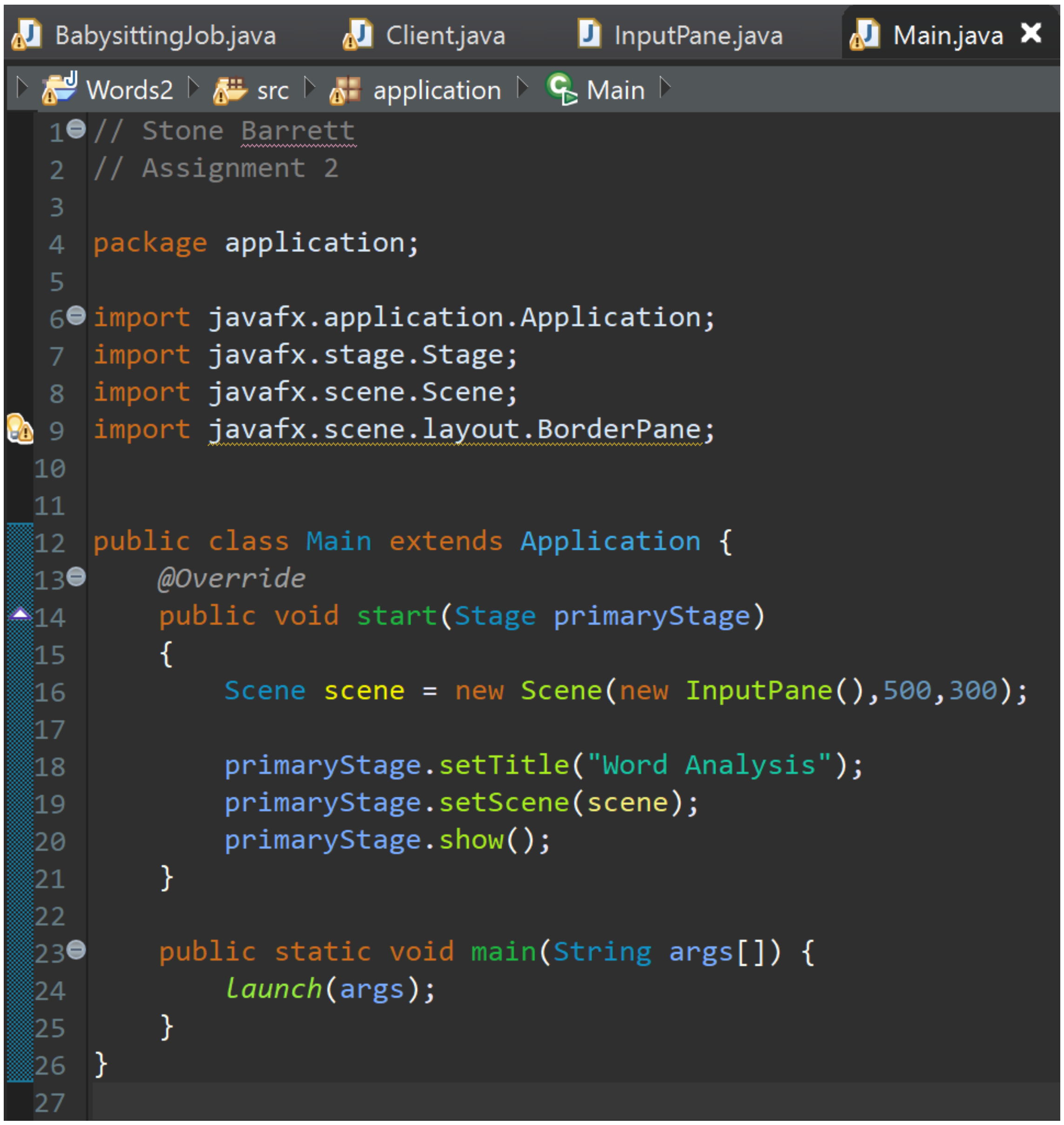


```
<terminated> Client (1) [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (Jun 9, 2019, 11:30:14 PM)
Enter year (2019-2025):
2024
Enter the job number (1-9999):
4566
Enter the babysitter code (1-3):
2
Enter the number of children (1-9):
6
Enter the number of hours (1-12):
2

Job Number: 244566
Employee Code: 2
Number of Children: 6
Number of Hours: 2
Babysitter: Greg
Fees: 58.0
```

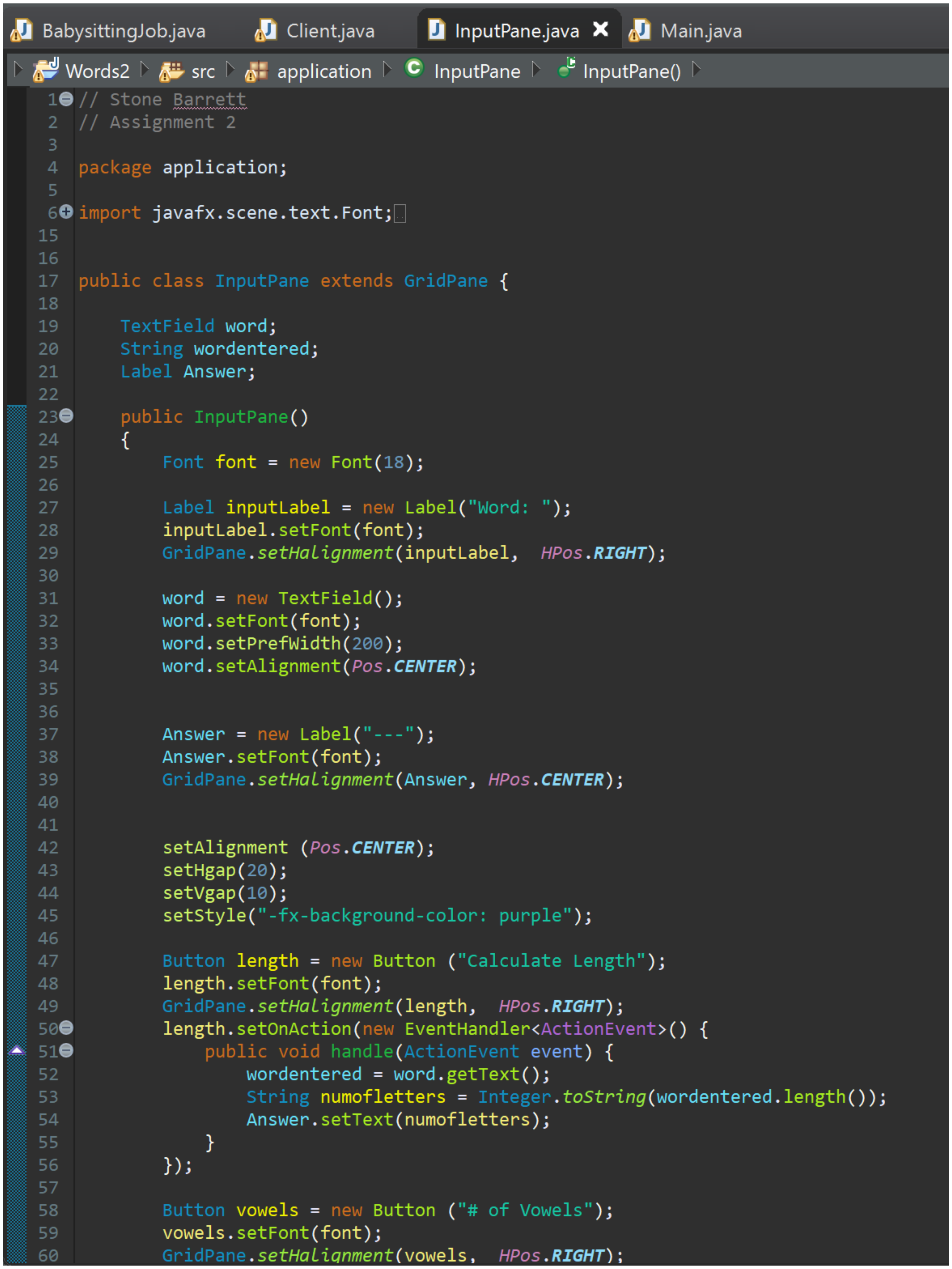
Problem 2

After the immense trouble myself and many others went through getting JavaFX to work within Eclipse, finally successfully compiling and running the second problem’s program was a great relief. For the main class, I imported everything necessary to work with FX. I created a scene for the application with a title, and then the program runs the show. The input window class is what the user sees and interacts with. It is positioned, sized, and designed along with its attributes (input boxes, fonts, buttons, etc.). Just like asked for, this UI has a text box for users to enter, then the program returns values based on what the user requests; number of letters, number of uppercase letters, number of vowels.

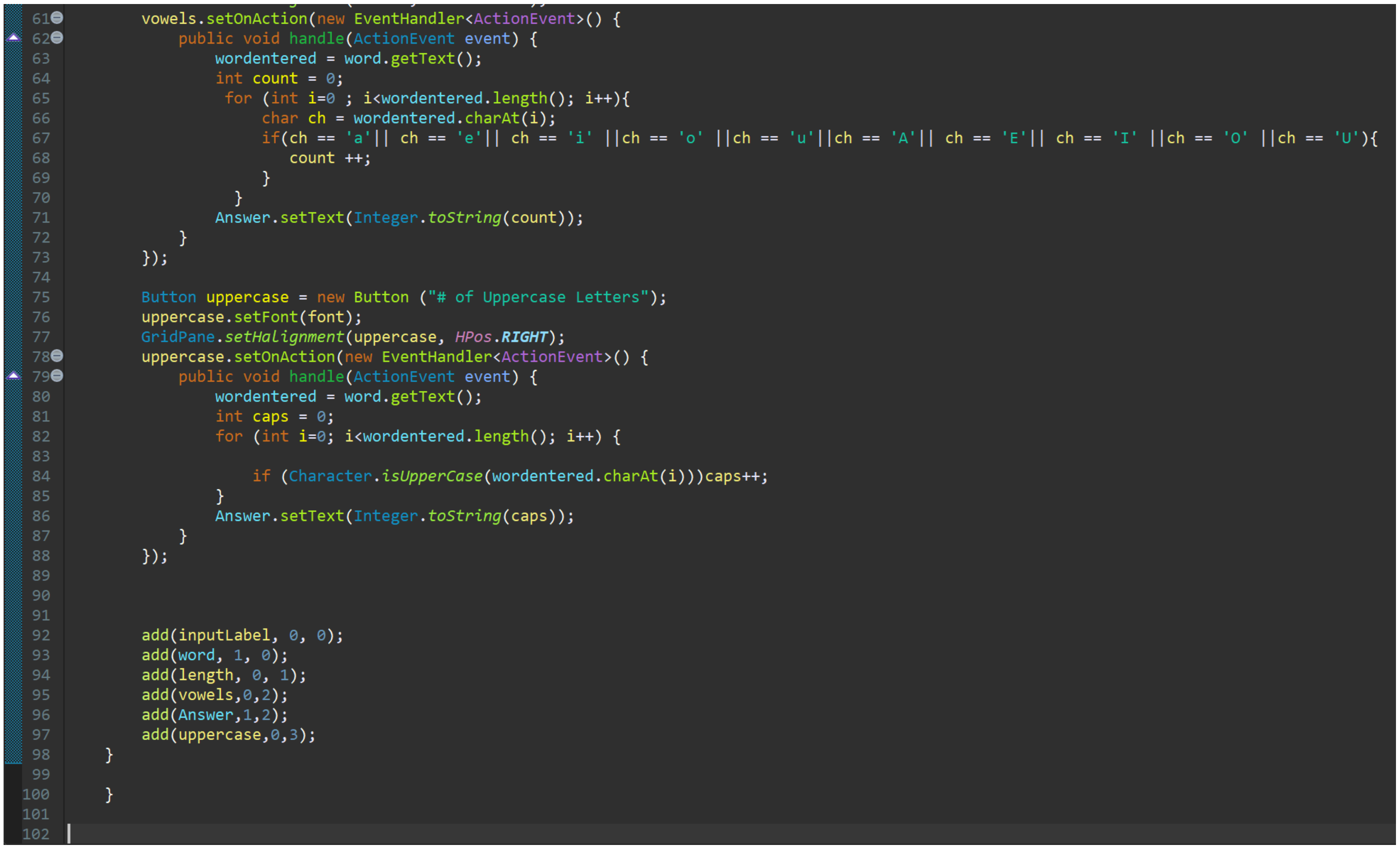
Screenshots of the code:



```java
// Stone Barrett
// Assignment 2

package application;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.BorderPane;


public class Main extends Application {
    @Override
    public void start(Stage primaryStage)
    {
        Scene scene = new Scene(new InputPane(),500,300);

        primaryStage.setTitle("Word Analysis");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String args[]) {
        launch(args);
    }
}

```

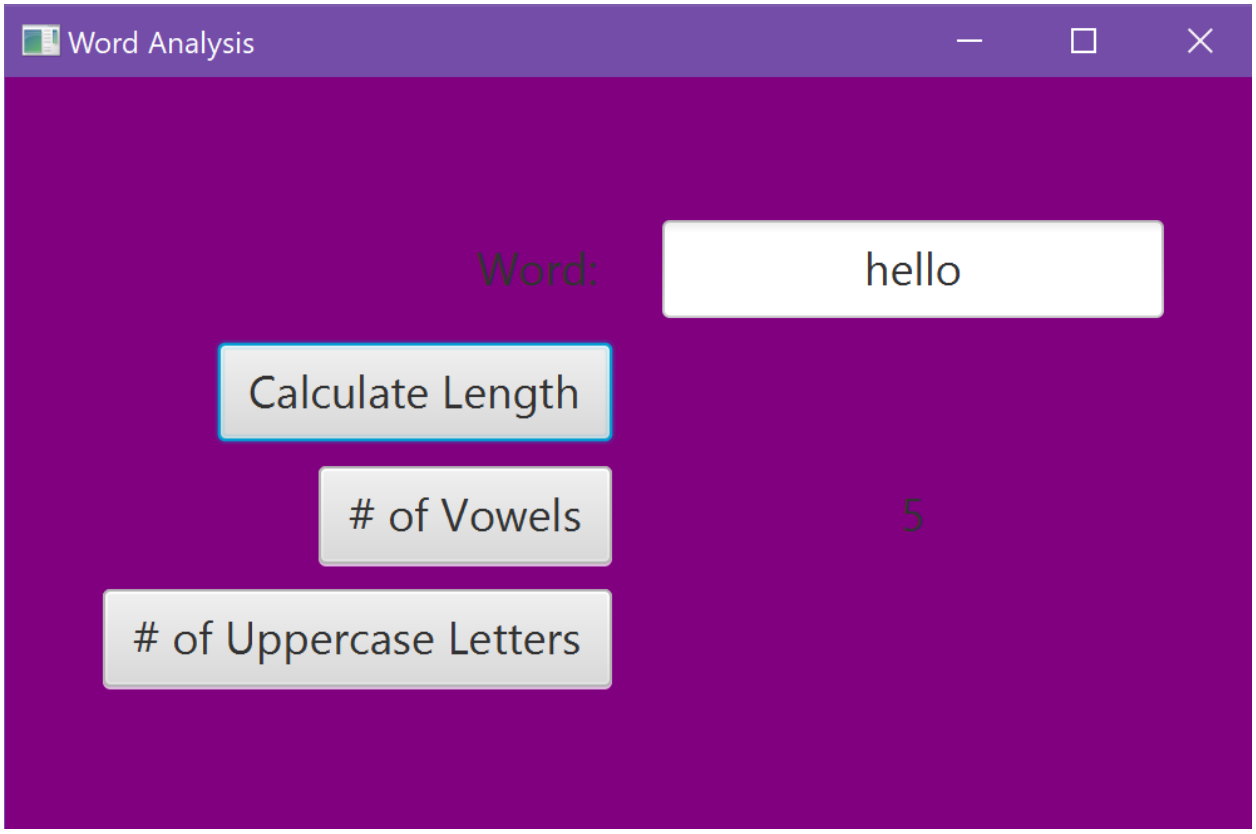


```java
// Stone Barrett
// Assignment 2

package application;

import javafx.scene.text.Font;


public class InputPane extends GridPane {

    TextField word;
    String wordentered;
    Label Answer;

    public InputPane()
    {
        Font font = new Font(18);

        Label inputLabel = new Label("Word: ");
        inputLabel.setFont(font);
        GridPane.setHalignment(inputLabel,  HPos.RIGHT);

        word = new TextField();
        word.setFont(font);
        word.setPrefWidth(200);
        word.setAlignment(Pos.CENTER);


        Answer = new Label("---");
        Answer.setFont(font);
        GridPane.setHalignment(Answer, HPos.CENTER);


        setAlignment (Pos.CENTER);
        setHgap(20);
        setVgap(10);
        setStyle("-fx-background-color: purple");

        Button length = new Button ("Calculate Length");
        length.setFont(font);
        GridPane.setHalignment(length,  HPos.RIGHT);
        length.setOnAction(new EventHandler<ActionEvent>() {
            public void handle(ActionEvent event) {
                wordentered = word.getText();
                String numofletters = Integer.toString(wordentered.length());
                Answer.setText(numofletters);
            }
        });

        Button vowels = new Button ("# of Vowels");
        vowels.setFont(font);
        GridPane.setHalignment(vowels,  HPos.RIGHT);
```



```
        vowels.setOnAction(new EventHandler<ActionEvent>() {
            public void handle(ActionEvent event) {
                wordentered = word.getText();
                int count = 0;
                 for (int i=0 ; i<wordentered.length(); i++){
                     char ch = wordentered.charAt(i);
                     if(ch == 'a'|| ch == 'e'|| ch == 'i' ||ch == 'o' ||ch == 'u'||ch == 'A'|| ch == 'E'|| ch == 'I' ||ch == 'O' ||ch == 'U'){
                        count ++;
                     }
                  }
                Answer.setText(Integer.toString(count));
            }
        });

        Button uppercase = new Button ("# of Uppercase Letters");
        uppercase.setFont(font);
        GridPane.setHalignment(uppercase, HPos.RIGHT);
        uppercase.setOnAction(new EventHandler<ActionEvent>() {
            public void handle(ActionEvent event) {
                wordentered = word.getText();
                int caps = 0;
                for (int i=0; i<wordentered.length(); i++) {

                    if (Character.isUpperCase(wordentered.charAt(i)))caps++;
                }
                Answer.setText(Integer.toString(caps));
            }
        });


        add(inputLabel, 0, 0);
        add(word, 1, 0);
        add(length, 0, 1);
        add(vowels,0,2);
        add(Answer,1,2);
        add(uppercase,0,3);
    }

    }

```

Screenshot of a sample run:


Word Analysis
Word:
hello
Calculate Length
# of Vowels
5
# of Uppercase Letters