University of Taipei

Computer Science

Summer

Homework 1

Student ID: U10916024

Student: Cheng-Hao, Zhang

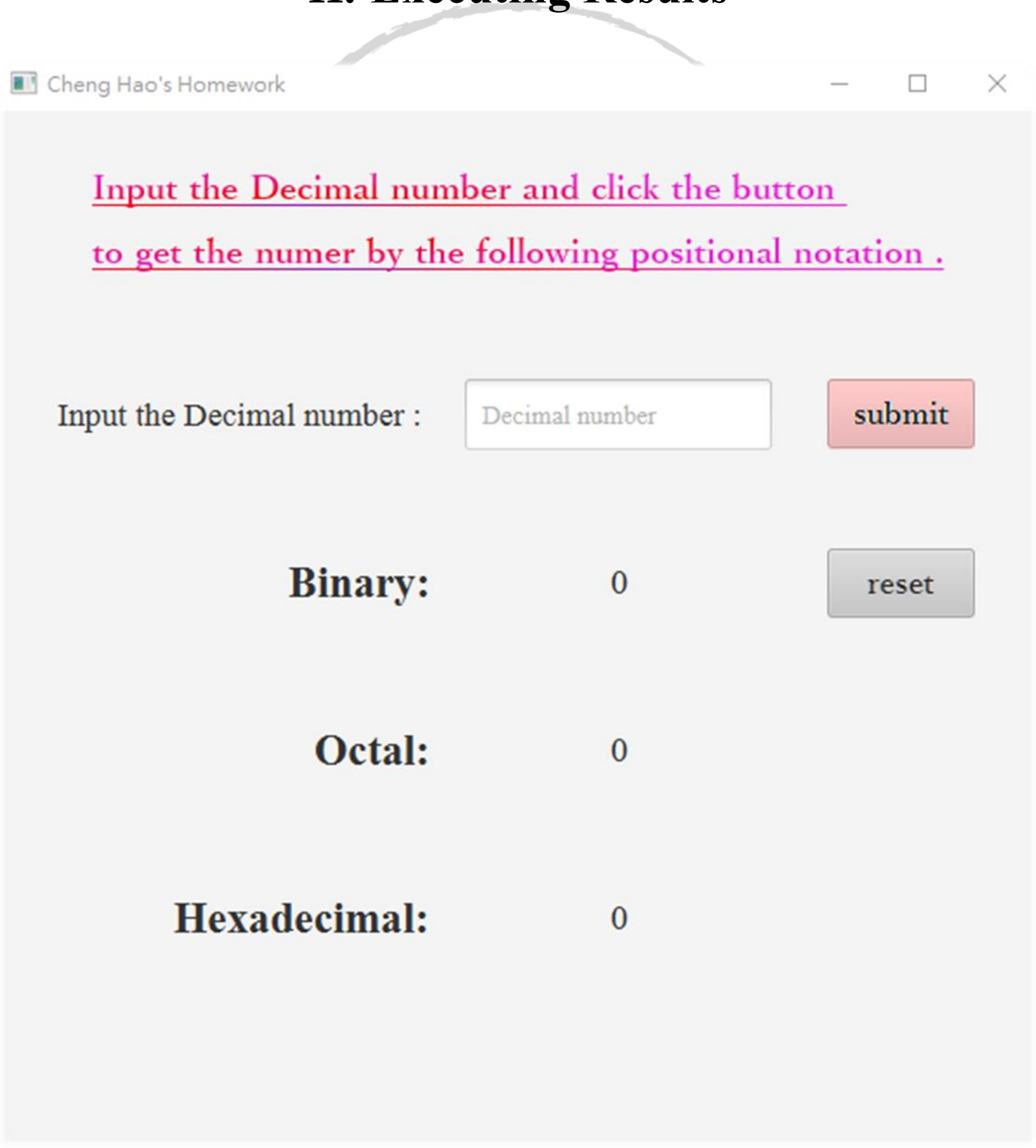
張呈顥

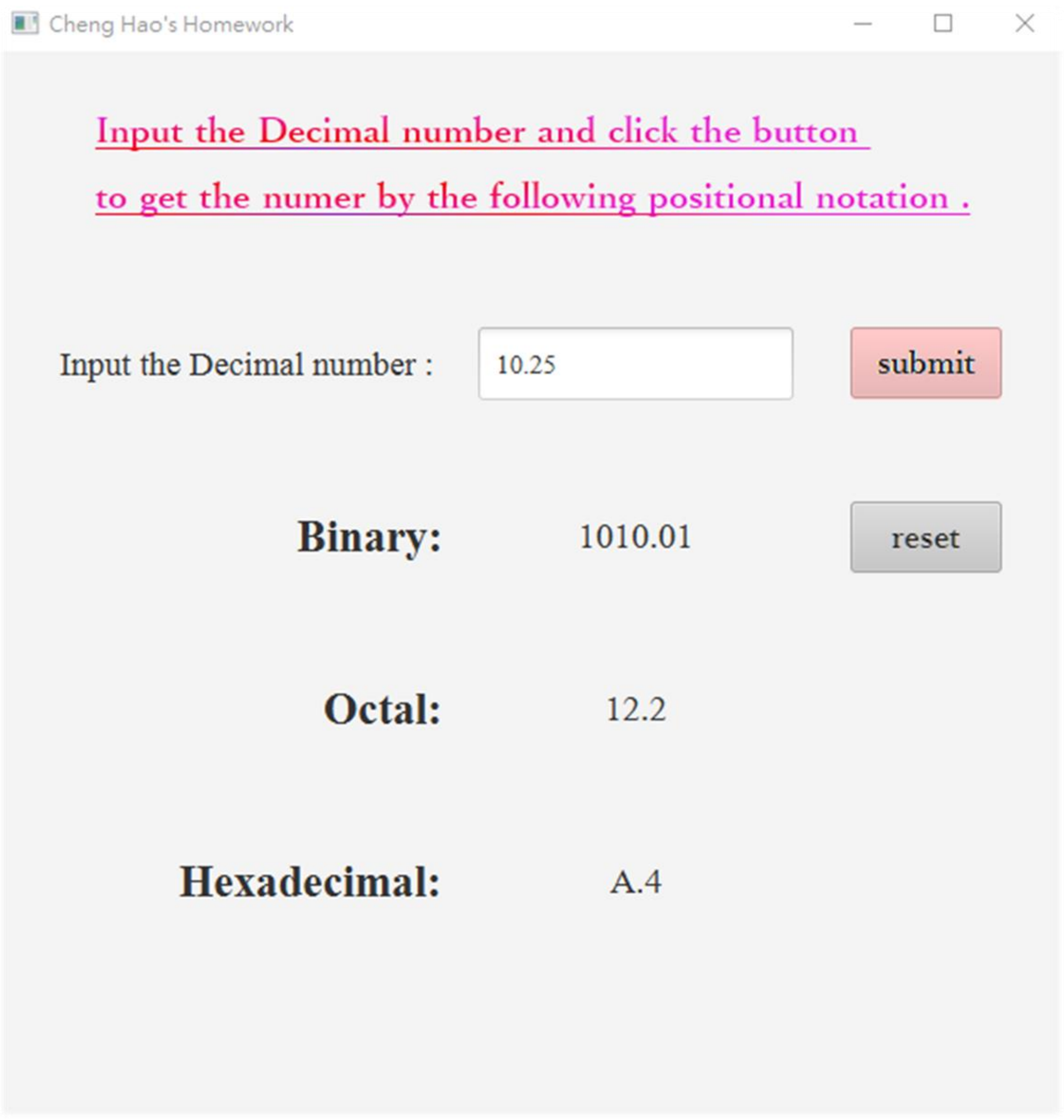
August 2021

I. Introduction

The application provides a text field for inputting a decimal number, and two buttons for create the events to drive the programs. After clicking the **submit** button, it will compute the target numbers based on the three radixes and show these results. To initialize the variables in the application, the button **reset** was created.

II. Executing Results


Cheng Hao's Homework
Input the Decimal number and click the button
to get the numer by the following positional notation .
Input the Decimal number :
Decimal number
submit
Binary:
0
reset
Octal:
0
Hexadecimal:
0



Cheng Hao's Homework

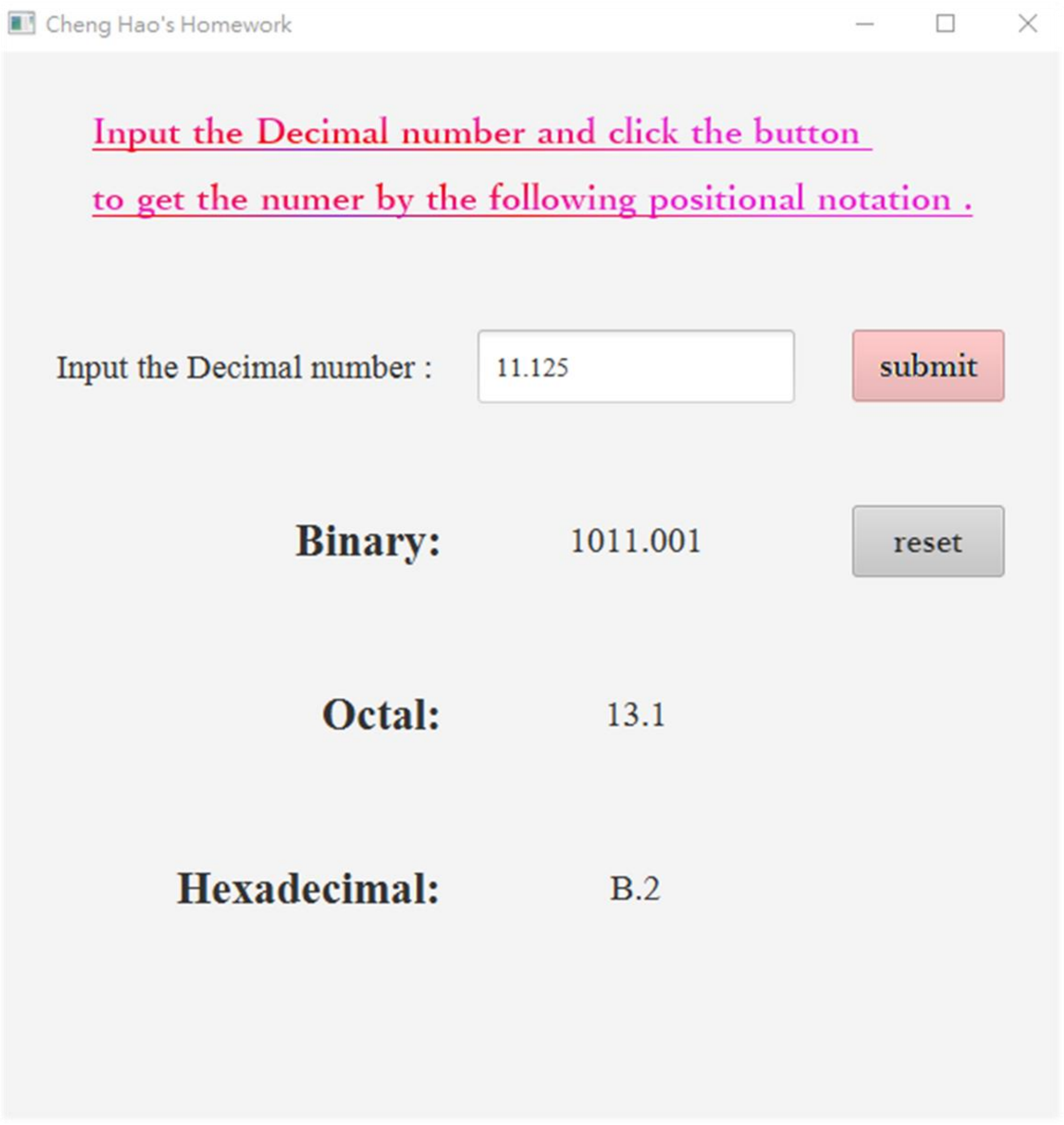
Input the Decimal number and click the button
to get the numer by the following positional notation .

Input the Decimal number : 10.25

Binary: 1010.01

Octal: 12.2

Hexadecimal: A.4



Cheng Hao's Homework

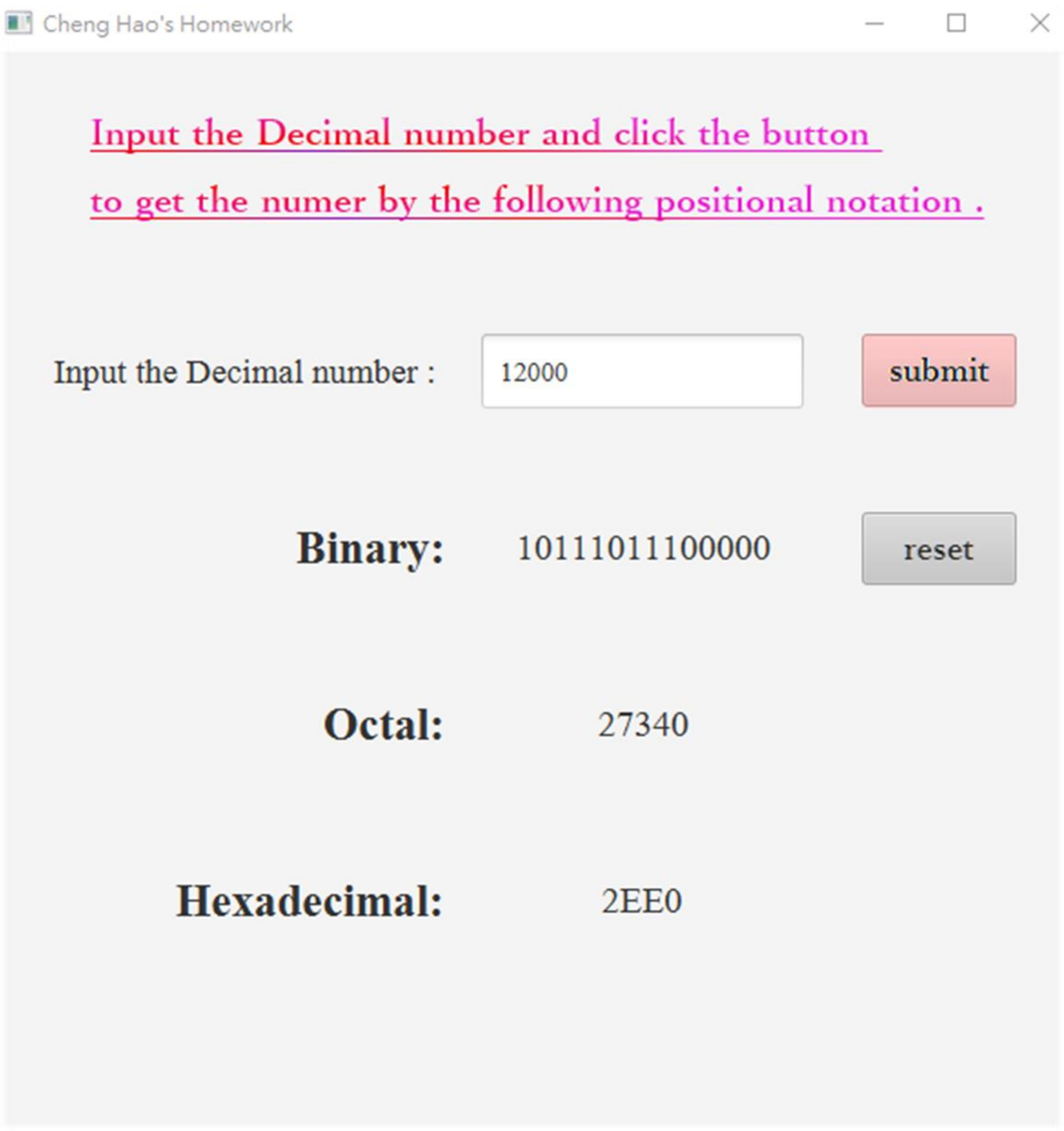
Input the Decimal number and click the button
to get the numer by the following positional notation .

Input the Decimal number : 11.125

Binary: 1011.001

Octal: 13.1

Hexadecimal: B.2


Cheng Hao's Homework

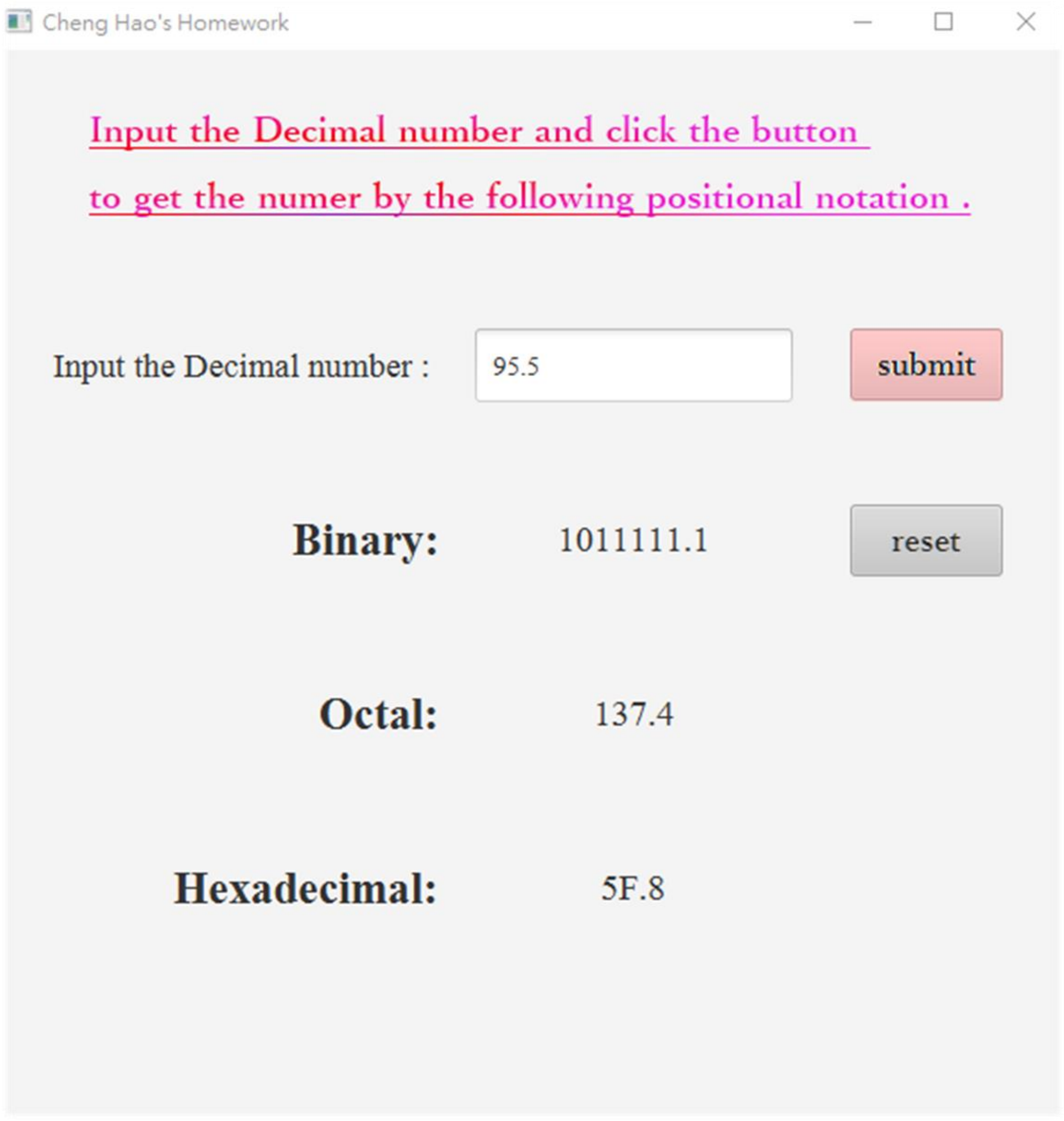
Input the Decimal number and click the button
to get the numer by the following positional notation .

Input the Decimal number : 12000

Binary: 10111011100000

Octal: 27340

Hexadecimal: 2EE0



Cheng Hao's Homework

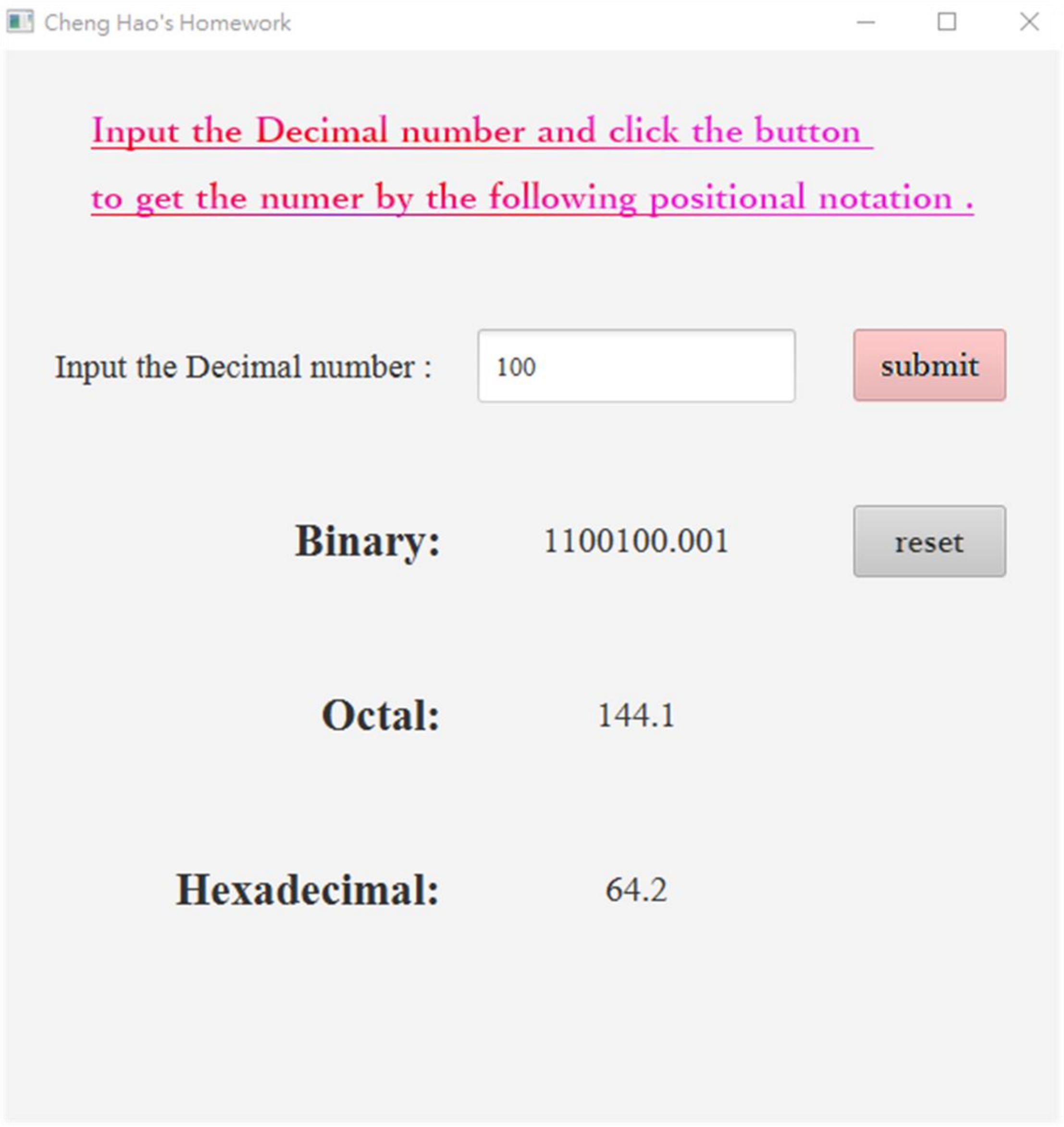
Input the Decimal number and click the button
to get the numer by the following positional notation .

Input the Decimal number : 95.5

Binary: 1011111.1

Octal: 137.4

Hexadecimal: 5F.8



Input the Decimal number and click the button
to get the numer by the following positional notation .

Input the Decimal number : 100

Binary: 1100100.001

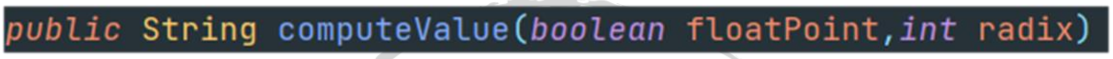
Octal: 144.1

Hexadecimal: 64.2

III. Architecture And Algorithm

Let us talk about my algorithm for the application. Based on MVC architecture, this is a javaFX application with the GUI functions.

Of algorithms converting the positional notation from the one radix to another one, the only difference is the shift value. Hence, I integrated the algorithms into a method.



```
public String computeValue(boolean floatPoint,int radix)
```

diagram 1

Following is the algorithm introduction, e.g., transform decimal into binary, octal number, or hexadecimal:

i. Initialization
   A. Set Integers for 0
   B. Set Double for 0.0
   C. Set Boolean for false
   D. Set Labels for “0”

ii. Check input exceptions
   A. Empty input
   B. Illegal input
   C. Minus number
   D. Floating-point number

iii. Transform the Integer number

(Use a character array for getting the changed value)

```
char[] ch_ref1 = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
```

A. As a buffer, declaring a string builder (Object) to append character

B. Declare a temporary variable to save the inputted Integer value

C. A while loop for the temporary variable bigger than zero

1. Get the remainder to divide the temporary variable by radix(圖表 1 parameter)
2. The remainder is the index of the array **ch_ref_1**
3. According as the index, find the element of **ch_ref_1**, and append it to the string builder.
4. Assign the temporary variable to that dividing the temporary variable by radix.

D. Reversing the string builder, and transforming into String

E. Complete

iv. Transform the number after floating-point

(Use a character array for getting the changed value like iii)

A. As a buffer, declaring a string builder (Object) to append character

B. Declare a temporary variable to save the inputted Integer value

C. A while loop for the temporary variable bigger than zero

1. Assign the temporary variable to that multiplying the temporary variable by radix
2. The integer part of product is the index of the array
3. According as the index, find the element of the array, and append it to the string builder.
4. Assign the temporary variable to that the temporary variable minus the integer part of the temporary variable.

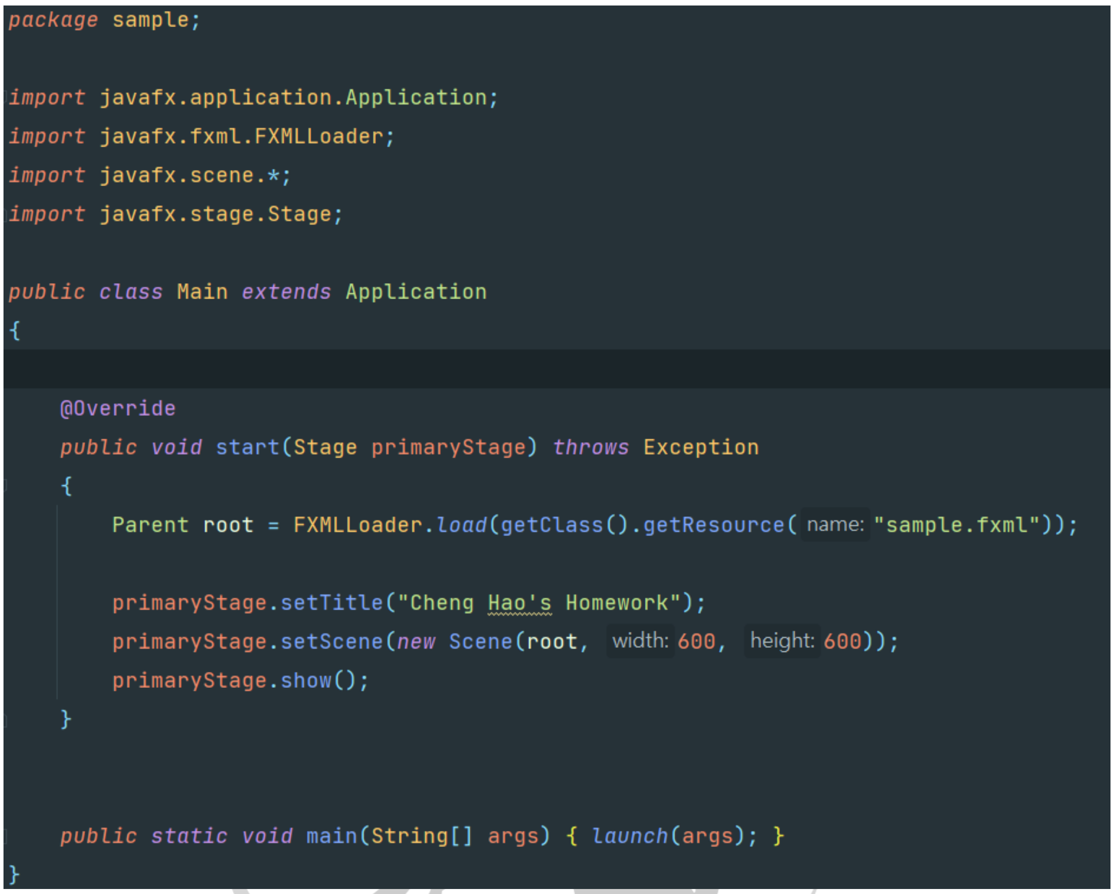
D. No reversing the string builder, but transforming into String

E. Don’t forget the floating-point, and Complete

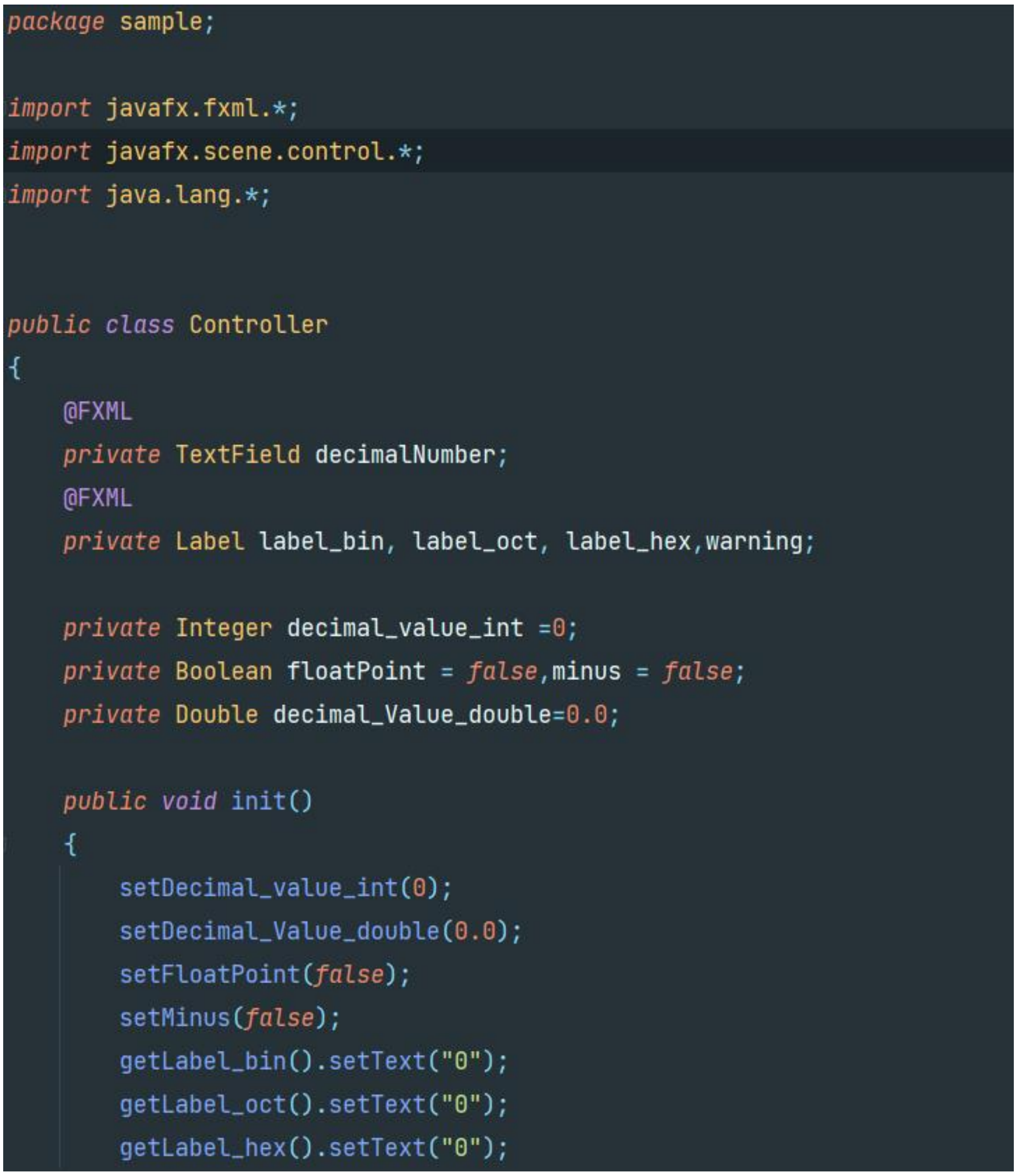
IV. Reflection

During doing the homework, I reviewed the technique and documentation of Java GUI (Javafx, OpenJFX). Moreover, the positional notation conversion algorithms are also practiced. Using the APIs is convenience, Nevertheless, being a programmer cannot forget the hand-on axioms. The core competency is the most important.

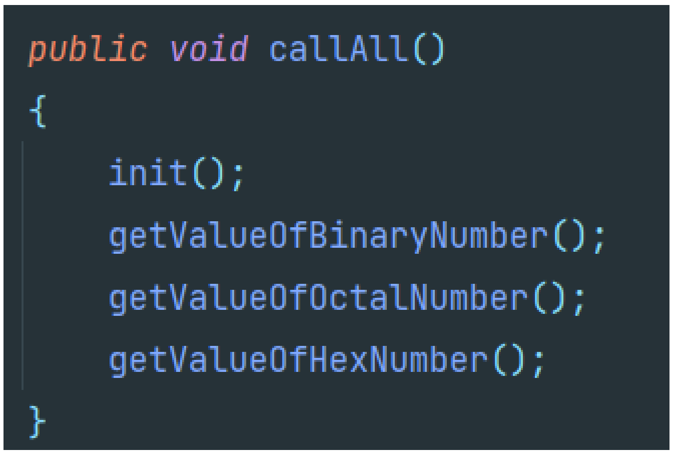
V. Source Code



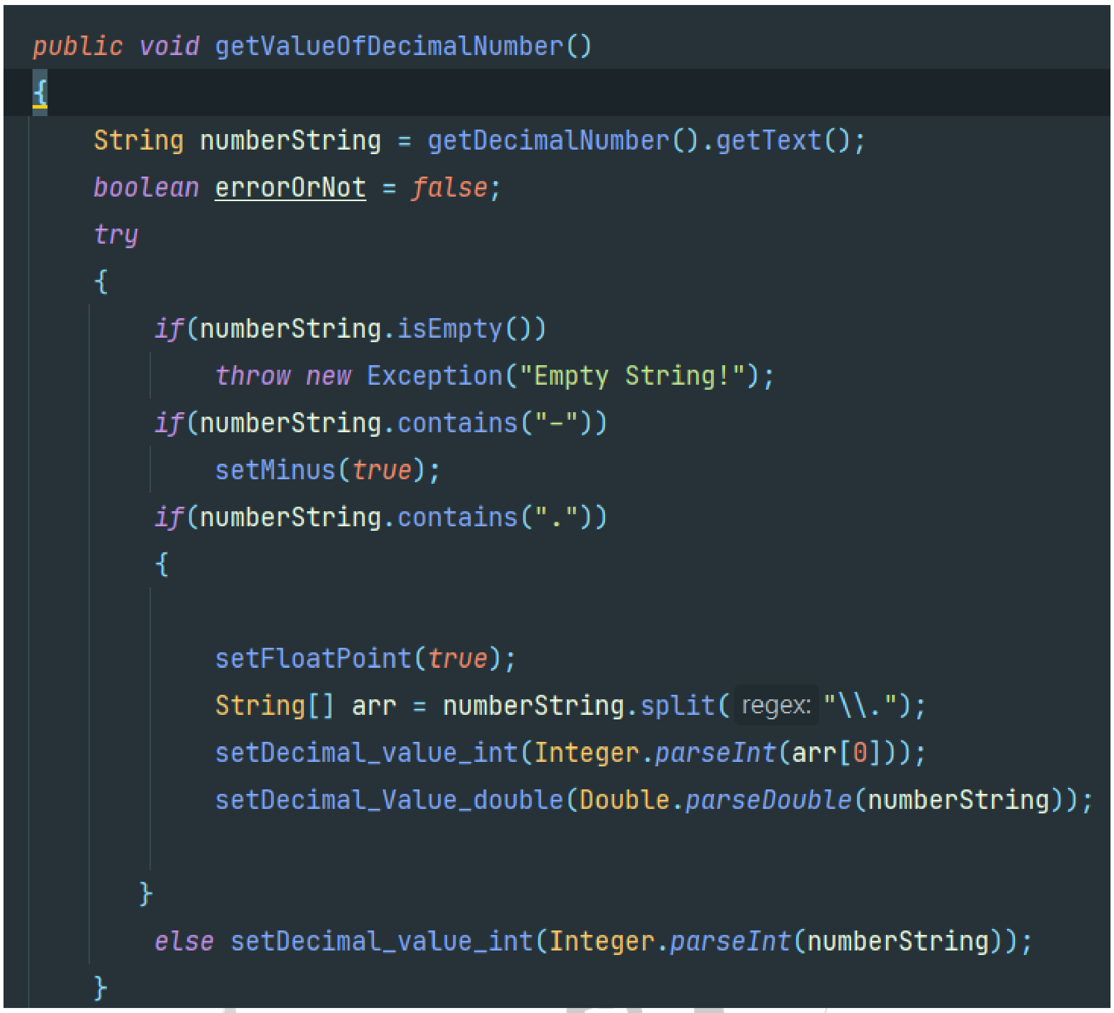
```
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.*;
import javafx.stage.Stage;

public class Main extends Application
{

    @Override
    public void start(Stage primaryStage) throws Exception
    {
        Parent root = FXMLLoader.load(getClass().getResource( name: "sample.fxml"));

        primaryStage.setTitle("Cheng Hao's Homework");
        primaryStage.setScene(new Scene(root,  width: 600,  height: 600));
        primaryStage.show();
    }


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



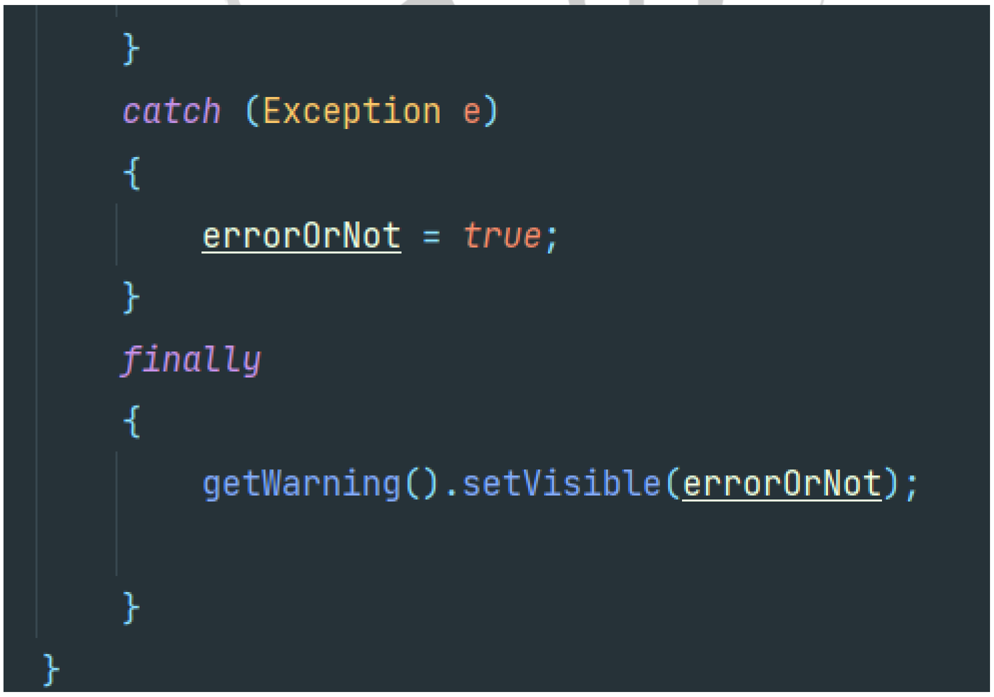
```
package sample;

import javafx.fxml.*;
import javafx.scene.control.*;
import java.lang.*;


public class Controller
{
    @FXML
    private TextField decimalNumber;
    @FXML
    private Label label_bin, label_oct, label_hex,warning;

    private Integer decimal_value_int =0;
    private Boolean floatPoint = false,minus = false;
    private Double decimal_Value_double=0.0;

    public void init()
    {
        setDecimal_value_int(0);
        setDecimal_Value_double(0.0);
        setFloatPoint(false);
        setMinus(false);
        getLabel_bin().setText("0");
        getLabel_oct().setText("0");
        getLabel_hex().setText("0");
```



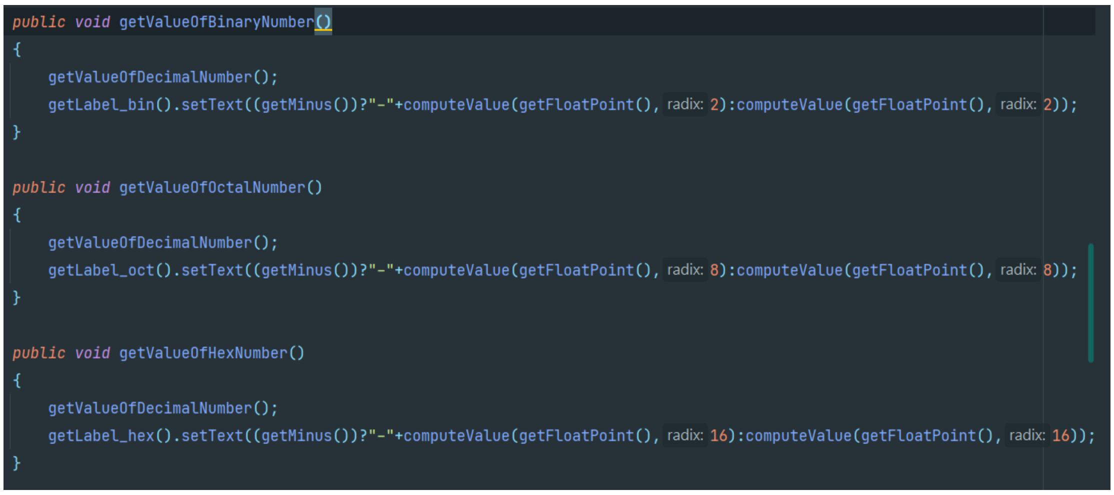
```
public void callAll()
{
    init();
    getValueOfBinaryNumber();
    getValueOfOctalNumber();
    getValueOfHexNumber();
}
```



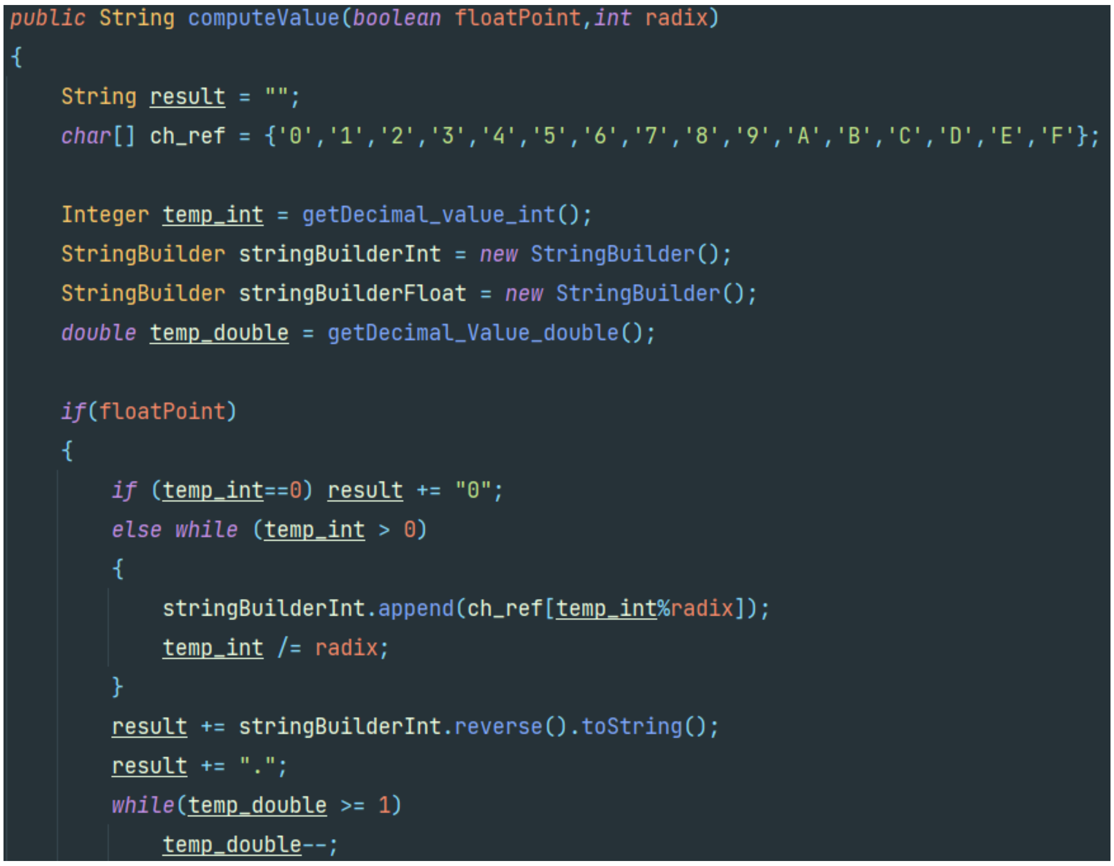
```java
public void getValueOfDecimalNumber()
{
    String numberString = getDecimalNumber().getText();
    boolean errorOrNot = false;
    try
    {
        if(numberString.isEmpty())
            throw new Exception("Empty String!");
        if(numberString.contains("-"))
            setMinus(true);
        if(numberString.contains("."))
        {

            setFloatPoint(true);
            String[] arr = numberString.split( regex: "\\.");
            setDecimal_value_int(Integer.parseInt(arr[0]));
            setDecimal_Value_double(Double.parseDouble(numberString));

       }
        else setDecimal_value_int(Integer.parseInt(numberString));
    }
```



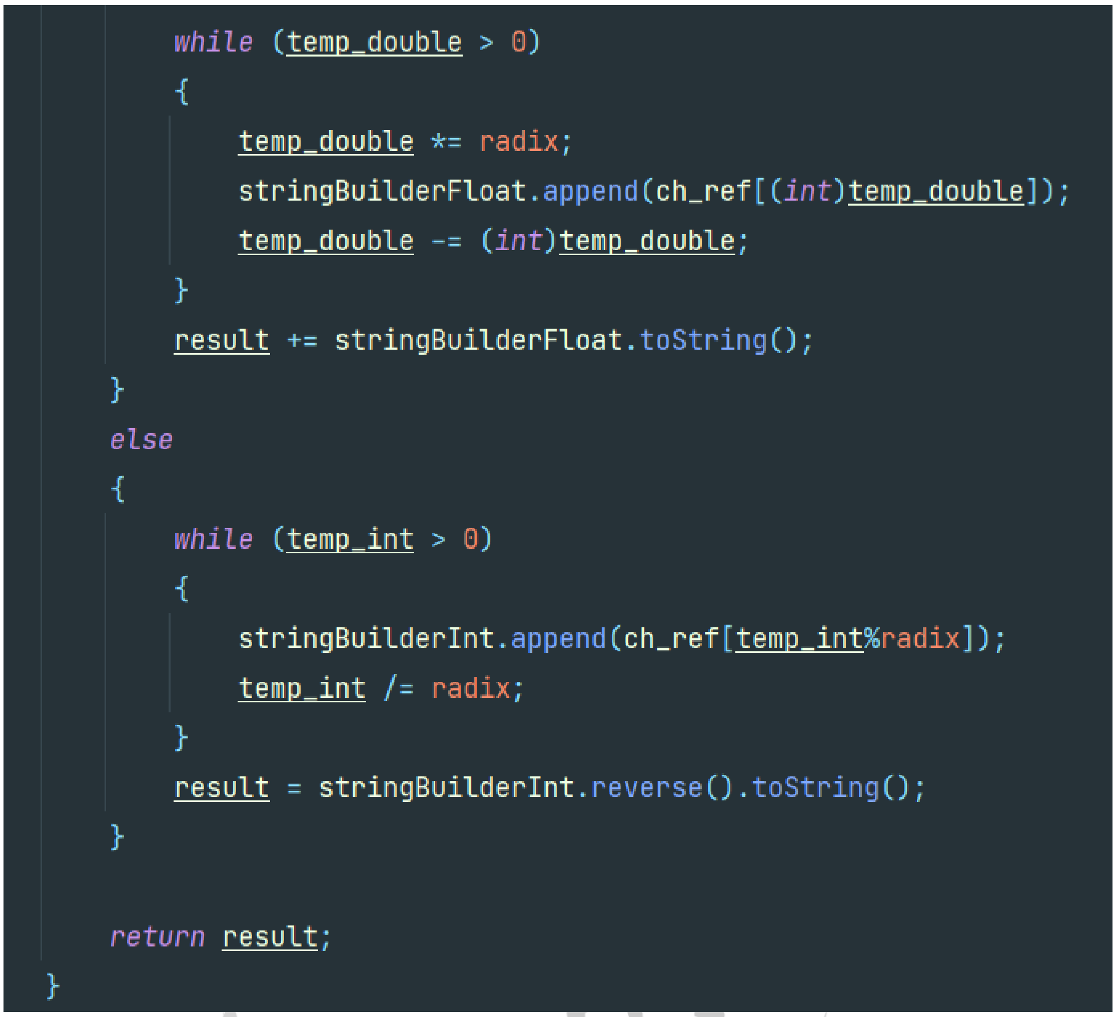
```java
    }
    catch (Exception e)
    {
        errorOrNot = true;
    }
    finally
    {
        getWarning().setVisible(errorOrNot);

    }
}
```



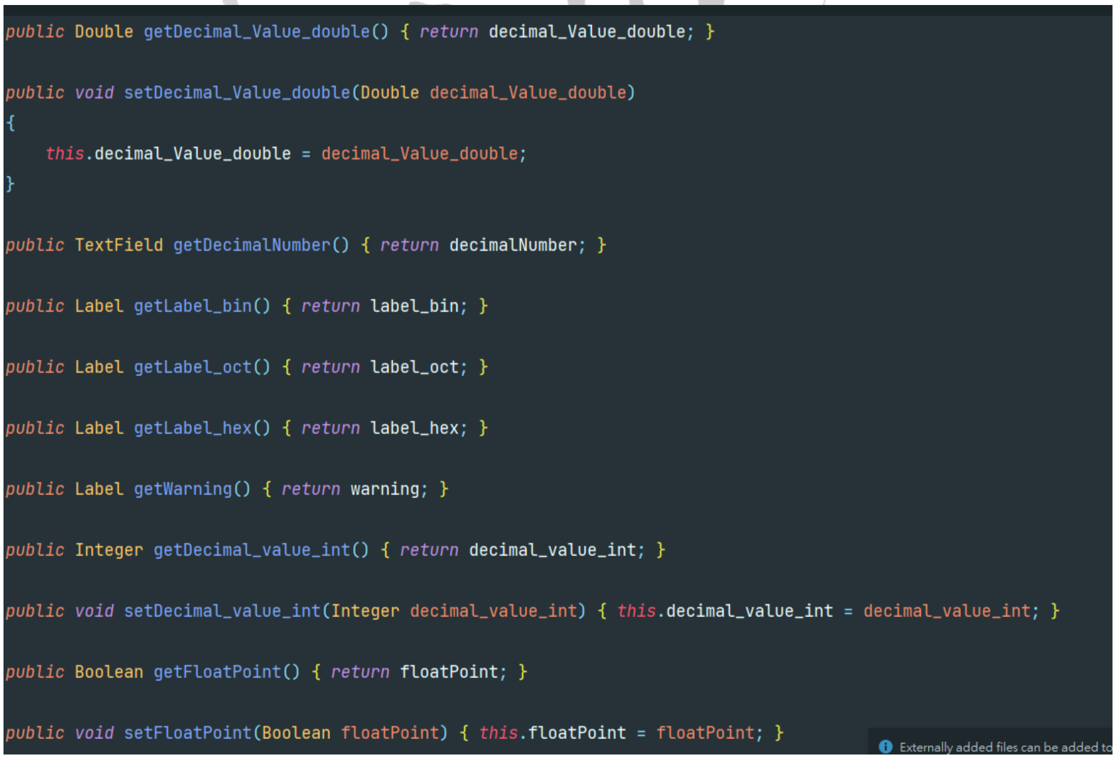
```
public void getValueOfBinaryNumber()
{
    getValueOfDecimalNumber();
    getLabel_bin().setText((getMinus())?"-"+computeValue(getFloatPoint(), radix: 2):computeValue(getFloatPoint(), radix: 2));
}

public void getValueOfOctalNumber()
{
    getValueOfDecimalNumber();
    getLabel_oct().setText((getMinus())?"-"+computeValue(getFloatPoint(), radix: 8):computeValue(getFloatPoint(), radix: 8));
}

public void getValueOfHexNumber()
{
    getValueOfDecimalNumber();
    getLabel_hex().setText((getMinus())?"-"+computeValue(getFloatPoint(), radix: 16):computeValue(getFloatPoint(), radix: 16));
}
```



```
public String computeValue(boolean floatPoint,int radix)
{
    String result = "";
    char[] ch_ref = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

    Integer temp_int = getDecimal_value_int();
    StringBuilder stringBuilderInt = new StringBuilder();
    StringBuilder stringBuilderFloat = new StringBuilder();
    double temp_double = getDecimal_Value_double();

    if(floatPoint)
    {
        if (temp_int==0) result += "0";
        else while (temp_int > 0)
        {
            stringBuilderInt.append(ch_ref[temp_int%radix]);
            temp_int /= radix;
        }
        result += stringBuilderInt.reverse().toString();
        result += ".";
        while(temp_double >= 1)
            temp_double--;
```

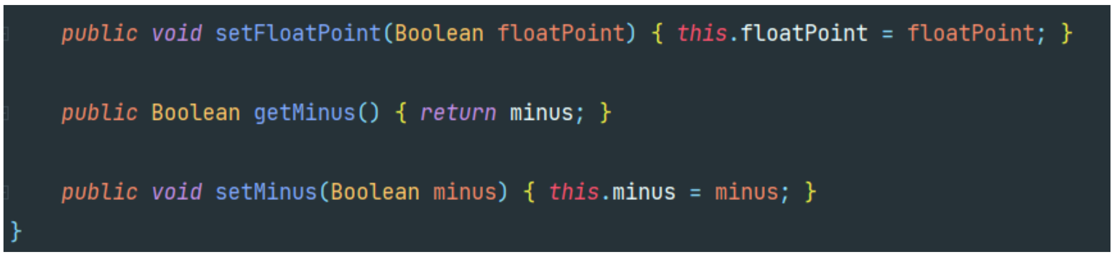


```
            while (temp_double > 0)
            {
                temp_double *= radix;
                stringBuilderFloat.append(ch_ref[(int)temp_double]);
                temp_double -= (int)temp_double;
            }
            result += stringBuilderFloat.toString();
        }
        else
        {
            while (temp_int > 0)
            {
                stringBuilderInt.append(ch_ref[temp_int%radix]);
                temp_int /= radix;
            }
            result = stringBuilderInt.reverse().toString();
        }

        return result;
    }
```



```
public Double getDecimal_Value_double() { return decimal_Value_double; }

public void setDecimal_Value_double(Double decimal_Value_double)
{
    this.decimal_Value_double = decimal_Value_double;
}

public TextField getDecimalNumber() { return decimalNumber; }

public Label getLabel_bin() { return label_bin; }

public Label getLabel_oct() { return label_oct; }

public Label getLabel_hex() { return label_hex; }

public Label getWarning() { return warning; }

public Integer getDecimal_value_int() { return decimal_value_int; }

public void setDecimal_value_int(Integer decimal_value_int) { this.decimal_value_int = decimal_value_int; }

public Boolean getFloatPoint() { return floatPoint; }

public void setFloatPoint(Boolean floatPoint) { this.floatPoint = floatPoint; }
```

Externally added files can be added to



```java
    public void setFloatPoint(Boolean floatPoint) { this.floatPoint = floatPoint; }

    public Boolean getMinus() { return minus; }

    public void setMinus(Boolean minus) { this.minus = minus; }
}
```