Stone Barrett

Assignment 01

5/28/19

Problem 1

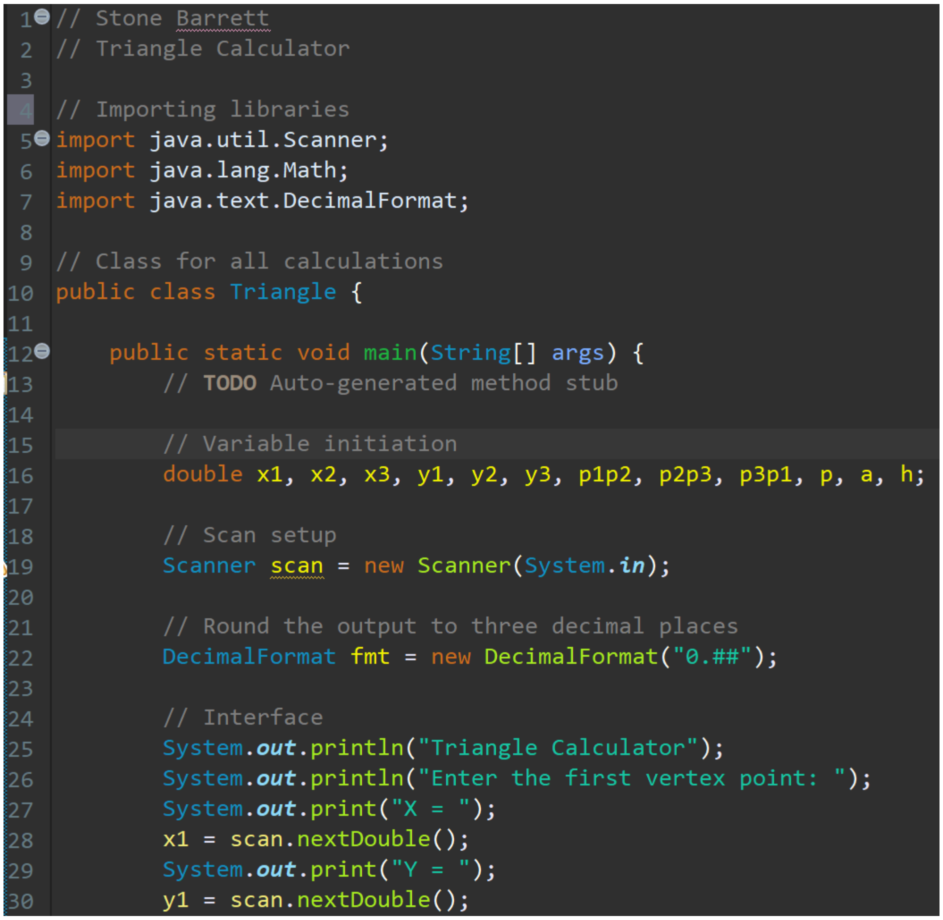
When reading the prompt for the first problem, I immediately knew I'd need to utilize variables for each of the values in the coordinates. My solution simply asks for the X and Y values for each vertex of the triangle, uses the points and the distance formula to find the lengths of all three legs, then adds the legs for perimeter and uses the given Heron's formula for area.

The program then displays all of that information, formatted nicely to two decimal places for the user to see.

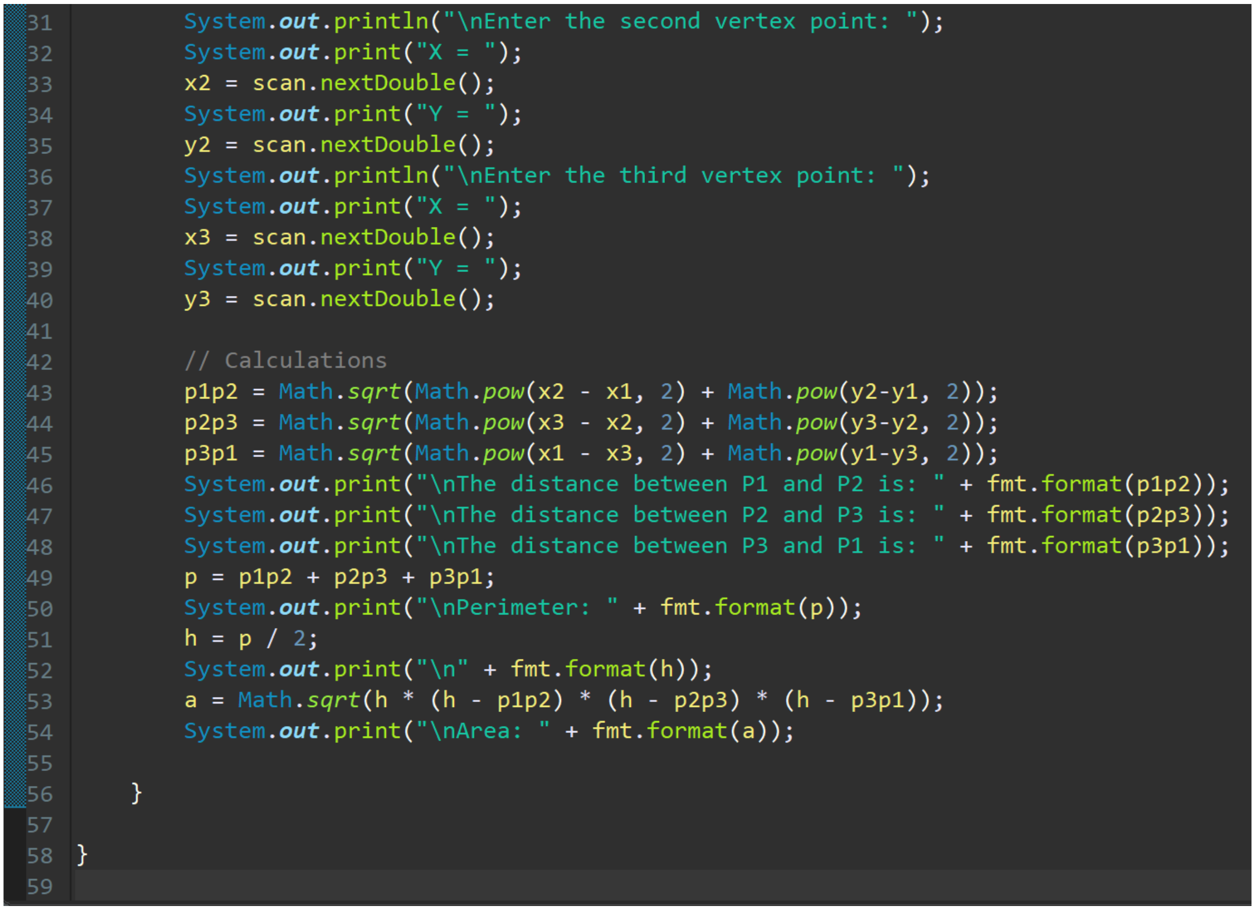
Some pseudocode logic would look like:

```
Print triangle calculator
Initialize variables
Print enter points
Scan x and y values for each point
Distance =
P = d1 + d2 + d3
A = herons formula
Print perimeter, area, distances
```

Screenshots of the code:

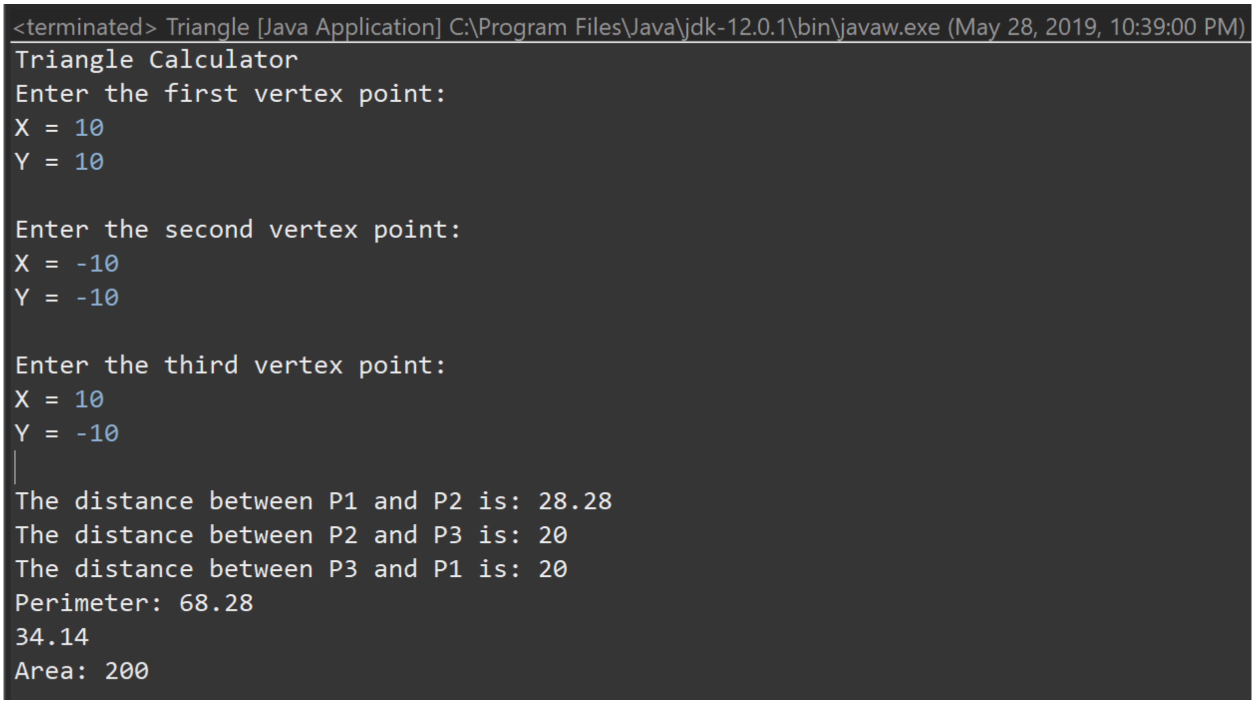


```java
// Stone Barrett
// Triangle Calculator

// Importing libraries
import java.util.Scanner;
import java.lang.Math;
import java.text.DecimalFormat;

// Class for all calculations
public class Triangle {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // Variable initiation
        double x1, x2, x3, y1, y2, y3, p1p2, p2p3, p3p1, p, a, h;

        // Scan setup
        Scanner scan = new Scanner(System.in);

        // Round the output to three decimal places
        DecimalFormat fmt = new DecimalFormat("0.##");

        // Interface
        System.out.println("Triangle Calculator");
        System.out.println("Enter the first vertex point: ");
        System.out.print("X = ");
        x1 = scan.nextDouble();
        System.out.print("Y = ");
        y1 = scan.nextDouble();
```



```java
        System.out.println("\nEnter the second vertex point: ");
        System.out.print("X = ");
        x2 = scan.nextDouble();
        System.out.print("Y = ");
        y2 = scan.nextDouble();
        System.out.println("\nEnter the third vertex point: ");
        System.out.print("X = ");
        x3 = scan.nextDouble();
        System.out.print("Y = ");
        y3 = scan.nextDouble();

        // Calculations
        p1p2 = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2-y1, 2));
        p2p3 = Math.sqrt(Math.pow(x3 - x2, 2) + Math.pow(y3-y2, 2));
        p3p1 = Math.sqrt(Math.pow(x1 - x3, 2) + Math.pow(y1-y3, 2));
        System.out.print("\nThe distance between P1 and P2 is: " + fmt.format(p1p2));
        System.out.print("\nThe distance between P2 and P3 is: " + fmt.format(p2p3));
        System.out.print("\nThe distance between P3 and P1 is: " + fmt.format(p3p1));
        p = p1p2 + p2p3 + p3p1;
        System.out.print("\nPerimeter: " + fmt.format(p));
        h = p / 2;
        System.out.print("\n" + fmt.format(h));
        a = Math.sqrt(h * (h - p1p2) * (h - p2p3) * (h - p3p1));
        System.out.print("\nArea: " + fmt.format(a));

    }

}

```

Screenshot of a sample run:



```
<terminated> Triangle [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (May 28, 2019, 10:39:00 PM)
Triangle Calculator
Enter the first vertex point:
X = 10
Y = 10

Enter the second vertex point:
X = -10
Y = -10

Enter the third vertex point:
X = 10
Y = -10

The distance between P1 and P2 is: 28.28
The distance between P2 and P3 is: 20
The distance between P3 and P1 is: 20
Perimeter: 68.28
34.14
Area: 200
```

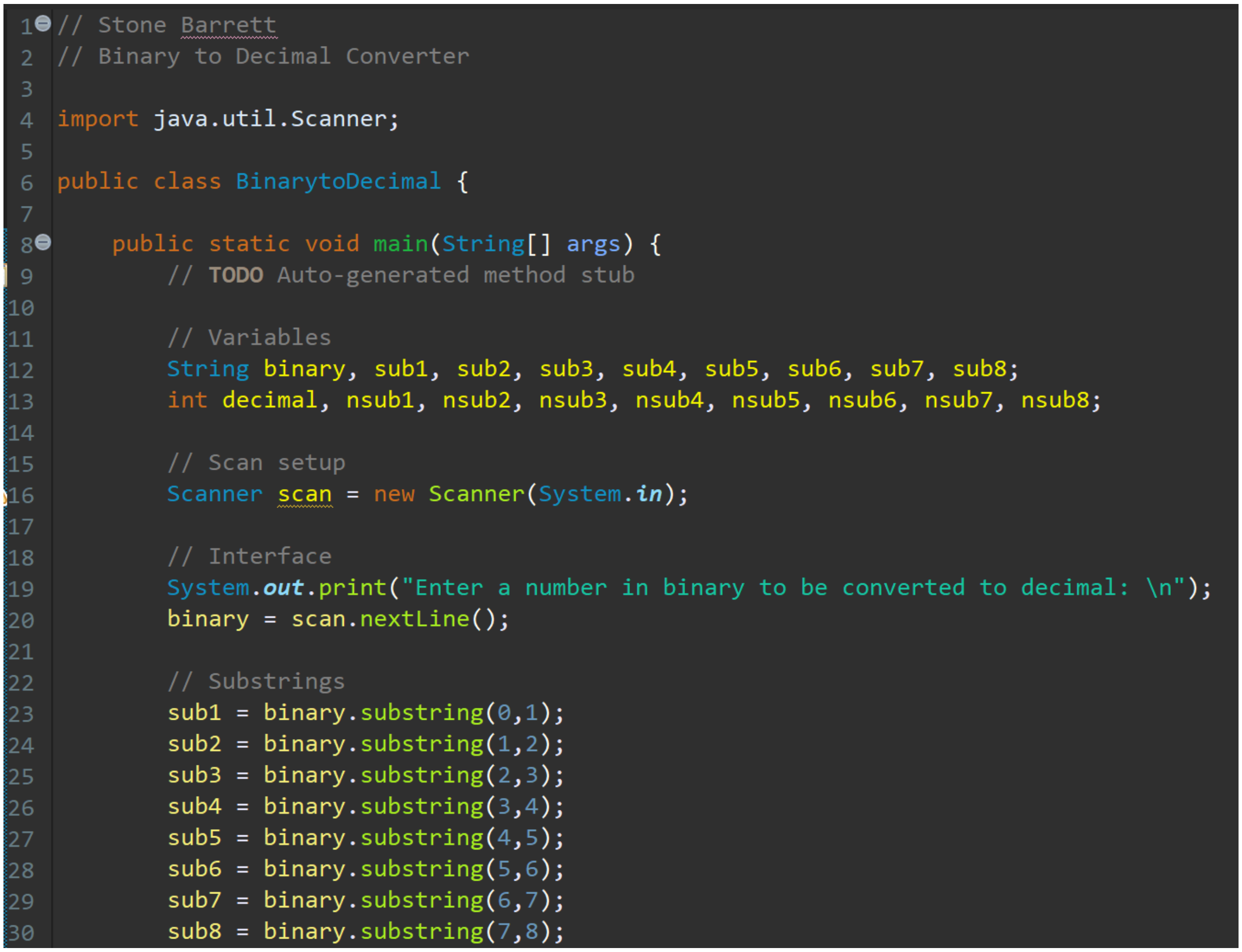
Problem 2

For this program, I decided to create variables for the first input and final output. Then I figured I needed to create variables for each substring of the binary string and their integer parsed counterparts. I ask the user for a number in binary, then assign that value to a string. I break the string into eight substrings (one for each 1 or 0) and parse each to an integer. Then I use the given formula to convert each 1 or 0 into their base-10 equal and add them all together to print the final result.

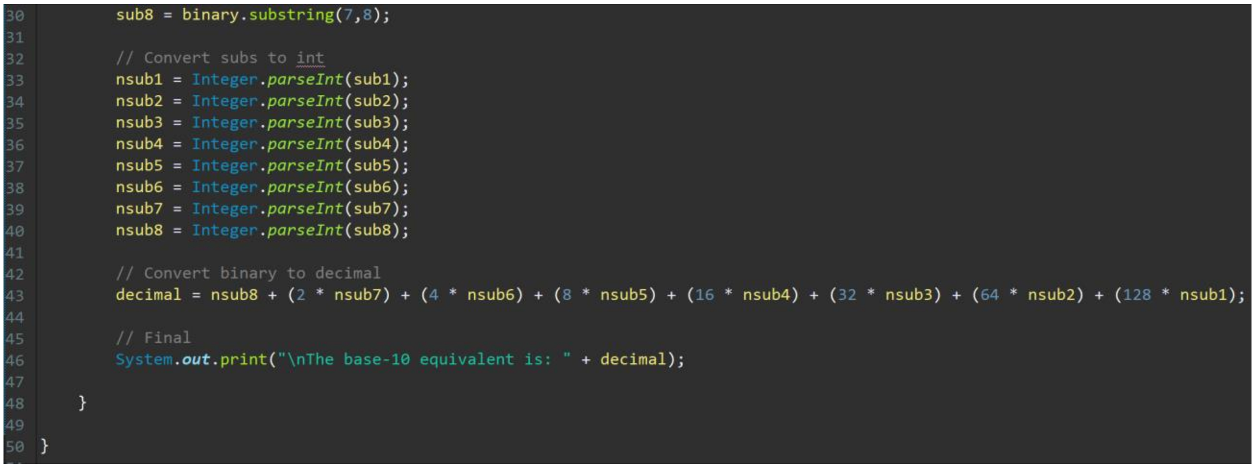
Some pseudocode logic would look like:

```
Variable initiation
Print enter a number in binary
Scan for binary, assign as string
Binary.substring(a,b) for each number
Parse each substring to integer, integer.parseint(var)
Decimal = n1+2n2+4n3+8n4+16n5+32n6+64n7+128n8
Print decimal
```

Screenshots of code:

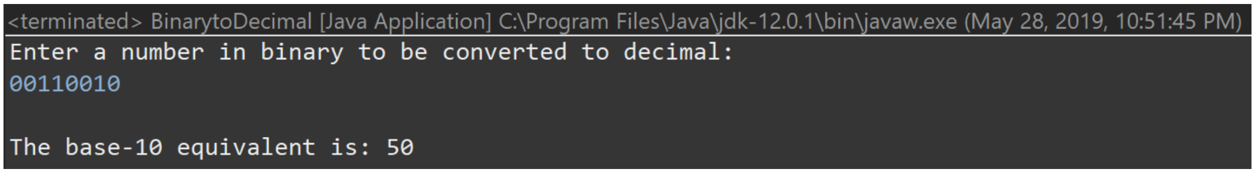


```java
// Stone Barrett
// Binary to Decimal Converter

import java.util.Scanner;

public class BinarytoDecimal {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // Variables
        String binary, sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8;
        int decimal, nsub1, nsub2, nsub3, nsub4, nsub5, nsub6, nsub7, nsub8;

        // Scan setup
        Scanner scan = new Scanner(System.in);

        // Interface
        System.out.print("Enter a number in binary to be converted to decimal: \n");
        binary = scan.nextLine();

        // Substrings
        sub1 = binary.substring(0,1);
        sub2 = binary.substring(1,2);
        sub3 = binary.substring(2,3);
        sub4 = binary.substring(3,4);
        sub5 = binary.substring(4,5);
        sub6 = binary.substring(5,6);
        sub7 = binary.substring(6,7);
        sub8 = binary.substring(7,8);
```



```java
        sub8 = binary.substring(7,8);

        // Convert subs to int
        nsub1 = Integer.parseInt(sub1);
        nsub2 = Integer.parseInt(sub2);
        nsub3 = Integer.parseInt(sub3);
        nsub4 = Integer.parseInt(sub4);
        nsub5 = Integer.parseInt(sub5);
        nsub6 = Integer.parseInt(sub6);
        nsub7 = Integer.parseInt(sub7);
        nsub8 = Integer.parseInt(sub8);

        // Convert binary to decimal
        decimal = nsub8 + (2 * nsub7) + (4 * nsub6) + (8 * nsub5) + (16 * nsub4) + (32 * nsub3) + (64 * nsub2) + (128 * nsub1);

        // Final
        System.out.print("\nThe base-10 equivalent is: " + decimal);

    }

}
```

Sample run:



```
<terminated> BinarytoDecimal [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (May 28, 2019, 10:51:45 PM)
Enter a number in binary to be converted to decimal:
00110010

The base-10 equivalent is: 50
```