CMPE 150, Fall ’19

Nurlan Dadashov

Section 6

Assignment 1

October 2019

1 Abstract

In this report, I will be outlining my solution to a programming challenge that had been set as part of an introductory programming course. The program prints out stickman climbing stairs, the heights of stairs and stickman depending on arguments from the user. The challenge has been completed to specification and used the least number of methods and for loops.

Contents

2 Stickman and Stairs

2.1 Problem Description

I’m going to model “Stickman and Stairs”. The idea is to simulate the motion of an ASCII stickman on stairs. It starts at the bottom of the stairs and keeps on climbing until it reaches the top. Every time a stickman takes a step higher, an image of it in exactly that position is printed. The main purpose of this project is to show how for loops can be used to create ASCII images.

The variables stickmanHeight and stairHeight are given as arguments to the main method and are used to determine the structure of each frame or image (what frames are, how they are created would be discussed in further chapters), number of frames (or steps which stickman climbs towards the top), etc. It was required to create a program using these variables, which will show the exact position of stickman in each frame. It is evident from the sample solutions for this assignment that position of stairs is the same in all frames; whereas, the position of stickman is changing with each successive frame printed. Further chapters will discuss how this program was written, which techniques were used, followed by some sample outputs.

2.2 Problem Solution

After a detailed review of sample solutions, it was decided that the program should work in the following steps:

1) Print empty lines above head	emptyLines()
2) Print the line with head	printHead_Arms()
3) Print the line with arms	printHead_Arms()
4) Print lines with torso (but without stairs)	printTorso_Stairs()
5) Print lines with torso (with stairs)	printTorso_Stairs()
6) Print the line with legs	printTorso_Stairs()
7) Print lines below legs	printTorso_Stairs()
8) Print 3 empty lines after	printImage()

This program makes use of 6 methods and 5 for loops.

2.3 Implementation

2.3.1 main Method

Like every other Java program, this program has *main* method where 2 arguments from the user are assigned to 2 integer variables, *stickmanHeight*, and *stairHeight*. Besides, main method also prints the final image into the console using the *printImage* method.

```java
public static void main(String[] args) {
	// Assigning arguments to variables
	int stickmanHeight = Integer.parseInt(args[0]);
	int stairHeight = Integer.parseInt(args[1]);
	//printing final image
	printImage(stairHeight, stickmanHeight);
}
```

2.3.2 printImage Method

Variables are passed to *printImage* method (*printImage*(stairHeight, stickmanHeight);), and there they are copied into **int** stairheight, **int** SMheight respectively. In *printImage* method, these new variables are also passed to other methods (For instance, *emptyLines*(stairheight, SMheight);).

Frames were mentioned a few times in the introduction. Each step stickman takes, or each position stickman is as it climbs stairs can be defined as a frame.
In *printImage* method, the variable *count* is used to determine the number of frames printed. It was deduced from sample solutions that the number of frames is always *stairheight + 1*. *Count* variable is essential for this program to run because as program prints a new frame the number of spaces between each element changes; these changes are controlled by *count*. As *count* is used in most methods, it could not be declared in *for* loop in *printImage* method. Therefore, it was decided to declare it as (**public static int** *count*;) and thus extend its scope to the whole program. From sample solutions, it was deduced that the total number of characters in each line of each frame is 8 + 3 *stairheight.(this result will be used in some parts of the program)

```
public static void printImage(int stairheight, int SMheight) {
	// count variable determines number of frames
	for(count = 0; count < stairheight + 1; count++) {
	//prints empty lines above stickman's head
	emptyLines(stairheight, SMheight);
	//prints sticman's head and hands
	printHead_Arms(stairheight, SMheight);
	//prints stickmas torso with stairs as well as legs
	printTorso_Stairs(stairheight, SMheight);
	//prints 3 empty lines after first stair
	System.out.println("\n\n");

	}
}
```

2.3.3 repeat Method

This method is used for decreasing the number of for loops. It takes a string and an integer as parameters and prints that string given number of times.

```
public static void repeat(String s, int times) {
// prints given string  given number of times
	for(int i = 0; i < times; i++) {
		System.out.print(s);
	}
}
```

2.3.4 emptyLines Method

It is clear from sample solutions that there are empty lines above the stickman’s head, and that the number of empty lines in each successive frame decreases by one. To print those empty lines, *emptyLines* method is used. The number of empty lines is calculated using stairheight - *count*. For instance(stairheight = 2), when count = 0(1st frame), 2 empty lines are printed; when count = 1(2nd frame); 1 empty line is printed; when count = 2(3rd frame); no empty lines are printed (for sample solution).

```
public static void emptyLines(int stairheight, int SMheight) {
	//prints empty lines above stickman's head
	repeat("\n",stairheight - count);
}
```

2.3.5 printHead_Arms Method

After empty lines are printed, this program prints the stickman's head and arms using *printHead_Arms* method. It is given in description that stairHeight + 2 < stickmanHeight; therefore, no stairs can be placed after head and arms in the same line, which makes it easier to print them. With each frame, the number of spaces before head is increasing by three and in the first frame this number being 1; thus, the number of spaces can be calculated by *count* * 3 + 1. After printing head and 1 space, a new line is printed. With each frame, the number of spaces before head is increasing by three and in the first frame this number being 0; thus, the number of spaces can be calculated by *count* * 3.

```java
public static void printHead_Arms(int stairheight, int SMheight) {
	//empty spaces before and after head
	repeat(" ", count * 3 + 1);
	System.out.print("O");

	System.out.print(" ");
	//new line
	System.out.println();
	//empty spaces before and after arms
	repeat(" ",count * 3);
	System.out.print("/|\\");

	//new line
	System.out.println();
}
```

2.3.6 printTorso_Stairs Method

The most difficult part of the program is implemented in *printTorso_Stairs* method. If all sample solutions are closely examined, then it is evident that some torso characters (|) have stairs after them in the same line; however, some do not. This method's first for loop prints lines where there are no stairs after torso characters (|). It is clear that stairs always occupy 1 line more than stairheight's value, and that the other 2 lines are occupied by head and arms. Thus, if these lines are subtracted from SMheight, the formula is SMheight - (stairheight + 1) - 2; however, it can be also noticed that as stickman climbs (subsequent frame is printed), the number of empty lines above head decreases by one and number of lines only having '|' increases by one. As a consequence, the final form of the formula for calculating the number of these lines looks like SMheight - (stairheight + 1) - 2 + *count*. Empty lines before '|'

is *count* * 3 + 1. (The variable i is declared inside for loop and makes for loop run `SMheight - (stairheight + 1) - 2 + count` number of times)

```
public static void printTorso_Stairs(int stairheight, int SMheight) {
	//prints | that are in lines without stairs
	for(int i = 0; i < SMheight - (stairheight + 1) - 2 + count ; i++) {
			//empty spaces before and after torso(|)
			repeat(" ", count * 3 + 1);
			System.out.print("|");

			System.out.print(" ");
			//new line
			System.out.println();
	}
……………………………………….code………………………...……………….
}
```

In this part, the method prints those '|' which have stairs after them. Stairs occupy the total number of (stairheight + 1) lines, one of which is (legs + stairs); whereas, all others are ('|' + stairs). Besides, with each subsequent frame, one line with ('|' + stairs) becomes a line with only '|' (and that line is printed by the previous part of code). Therefore, the number of lines ('|' + stairs) is (stairheight + 1) – 1 – *count* = stairheight - *count*. (The variable j is declared inside for loop and makes for loop run stairheight - *count* number of times)

The number of spaces before '|' is *count* * 3 + 1 (the same as in previous part); however, the number of spaces after '|' is different because this number is influenced by how many stair characters are in that line. To find the formula for spaces, we should subtract (spaces before '|' + '|' + stair parts) from the total number of characters. Stairs in each line have "___" and two "|"s, and the number of stars which is 0 in the first stair and increases by 3. Thus, 8 + 3 * stairheight – ((*count* * 3 + 1) – 1 – 3 – 2 - j * 3) = 8 + 3 * stairheight - 6 - (*count* * 3 + 1) - j * 3.

```
public static void printTorso_Stairs(int stairheight, int SMheight) {
……………………………………….code………………………...……………….
	//prints all torso lines (excluding legs and stairs after legs)
	for(int j = 0; j < stairheight - count; j++) {
	// empty spaces before torso and after torso but before stairs
```

```
            repeat(" ", count * 3 + 1);
            System.out.print("|");
            repeat(" ", 8 + 3 * stairheight - 6 - (count * 3 + 1) - j * 3);
            //part of stairs
            System.out.print("___");
            System.out.print("|");
            //stars
            repeat("*", j * 3);
            //end of line
            System.out.print("|");
            System.out.println();
        }
```

……………………………………….code……………………….....………………

```
}
```

This part of the method prints out the line with legs and stairs. The number of spaces before legs is *count* * 3(Same logic as for head). Then, the essential parts of the stairs are printed. The number of stars is determined by subtracting "/ \"(3 characters), number of empty spaces before legs, "___", '|' and '|': 8 + 3 * stairheight - 3 – *count* * 3 – 3 – 2 = 8 + 3 * stairheight - 8 - *count* * 3

```
public static void printTorso_Stairs(int stairheight, int SMheight) {
```

……………………………………….code……………………….....………………

```
//prints legs
        repeat(" ",count * 3);
        System.out.print("/ \\");
        //part of stairs
        System.out.print("___");
        System.out.print("|");
        //stars
        repeat("*", 8 + 3 * stairheight - 8 - count * 3);
        //end of line
        System.out.print("|");
        System.out.println();
```

……………………………………….code……………………….....………………

```
}
```

The last part of the code is responsible for printing lines containing stairs that are below the line with legs. From sample solutions, it can be deduced that the number of lines with stairs below legs always equals to *count*. (The variable k is declared inside for loop and makes for loop run *count* number of times) The number of spaces increases with each successive frame by 3; however, if there is more than one line with stairs below legs, then the number of spaces decreases by 3 with each line. Therefore, the number of spaces can be determined by *count* * 3 - k * 3. The number of stars is determined by subtracting the number of spaces before stairs and "___" + '|' + '|' from the total number of characters, the result is 8 + 3 * stairheight - 5 - *count* * 3 + k * 3.

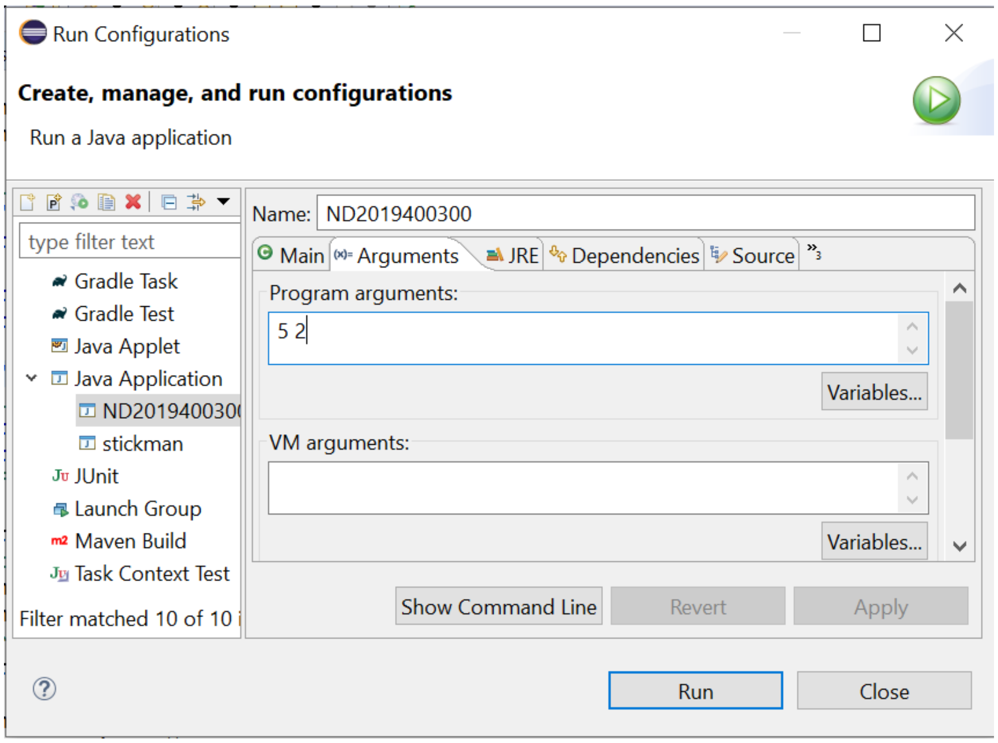
```
public static void printTorso_Stairs(int stairheight, int SMheight) {
……………………………….……..code……………………...………………
      //lines of stairs after legs
      for(int k = 0; k < count ; k++) {
            repeat(" ",count * 3 - k * 3 );
            //part of stairs
            System.out.print("___");
            System.out.print("|");
            //stars
            repeat("*", 8 + 3 * stairheight - 5 - count * 3 + k * 3);
            //end of line
            System.out.print("|");
            System.out.println();
      }
}
```

2.4 Output of the program

```java
public class ND2019400300 {

	public static int count;

	public static void main(String[] args) {
		// Assigning arguments to variables
		int stickmanHeight = Integer.parseInt(args[0]);
		int stairHeight = Integer.parseInt(args[1]);
		//printing final image
		printImage(stairHeight, stickmanHeight);
	}

	public static void repeat(String s, int times) {
		// prints given string  given number of times
		for(int i = 0; i < times; i++) {
			System.out.print(s);
		}
	}

	public static void printImage(int stairheight, int SMheight) {
		// count variable determines number of frames
		for(count = 0; count < stairheight + 1; count++) {
			//prints empty lines above stickman's head
			emptyLines(stairheight, SMheight);
			//prints sticman's head and hands
			printHead_Arms(stairheight, SMheight);
			//prints stickmas torso with stairs as well as legs
			printTorso_Stairs(stairheight, SMheight);
			//prints 3 empty lines after first stair
			System.out.println("\n\n");
		}
	}

	public static void emptyLines(int stairheight, int SMheight) {
		//prints empty lines above stickman's head
		repeat("\n",stairheight - count);
	}

	public static void printHead_Arms(int stairheight, int SMheight) {
		//empty spaces before and after head
		repeat(" ", count * 3 + 1);
		System.out.print("O");

		System.out.print(" ");
		//new line
		System.out.println();
		//empty spaces before and after arms
		repeat(" ",count * 3);
		System.out.print("/|\\");

		//new line
```

```
                System.out.println();
        }

        public static void printTorso_Stairs(int stairheight, int SMheight) {
                //prints | that are in lines without stairs
                for(int i = 0; i < SMheight - (stairheight + 1) - 2 + count ; i++) {
                        //empty spaces before and after torso(|)
                        repeat(" ", count * 3 + 1);
                        System.out.print("|");

                        System.out.print(" ");
                        //new line
                        System.out.println();
                }
                //prints all torso lines (excluding legs and stairs after legs)
                for(int j = 0; j < stairheight - count; j++) {
                        // empty spaces before torso and after torso but before stairs
                        repeat(" ", count * 3 + 1);
                        System.out.print("|");
                        repeat(" ", 8 + 3 * stairheight - 6 - (count * 3 + 1) - j * 3);
                        //part of stairs
                        System.out.print("___");
                        System.out.print("|");
                        //stars
                        repeat("*", j * 3);
                        //end of line
                        System.out.print("|");
                        System.out.println();
                }
                //prints legs
                repeat(" ",count * 3);
                System.out.print("/ \\");
                //part of stairs
                System.out.print("___");
                System.out.print("|");
                //stars
                repeat("*", 8 + 3 * stairheight - 5 - (count + 1)* 3);
                //end of line
                System.out.print("|");
                System.out.println();
                //lines of stairs after legs
                for(int k = 0; k < count ; k++) {
                        repeat(" ",count * 3 - k * 3 );
                        //part of stairs
                        System.out.print("___");
                        System.out.print("|");
                        //stars
                        repeat("*", 8 + 3 * stairheight - 5 - count * 3 + k * 3);
                        //end of line
                        System.out.print("|");
                        System.out.println();
                }
        }
}
```

stickmanHeight = 5; stairHeight = 2



Run Configurations

Create, manage, and run configurations

Run a Java application

type filter text

- Gradle Task
- Gradle Test
- Java Applet
- Java Application
  - ND2019400300
  - stickman
- JUnit
- Launch Group
- Maven Build
- Task Context Test

Filter matched 10 of 10 i

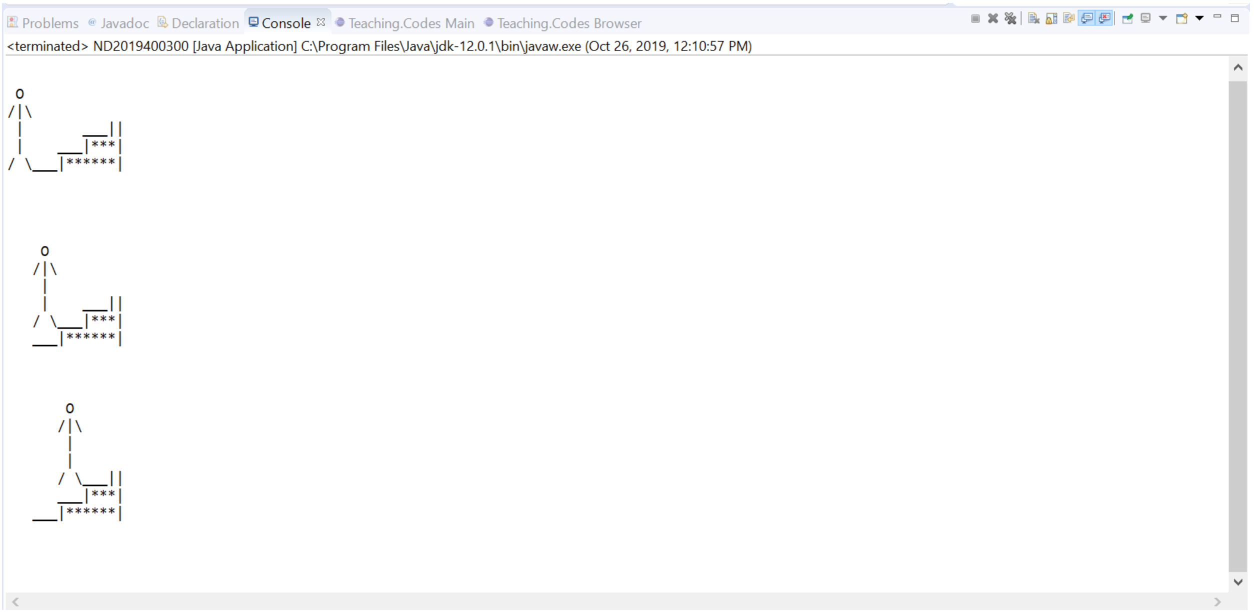
Name: ND2019400300

Main | Arguments | JRE | Dependencies | Source

Program arguments:

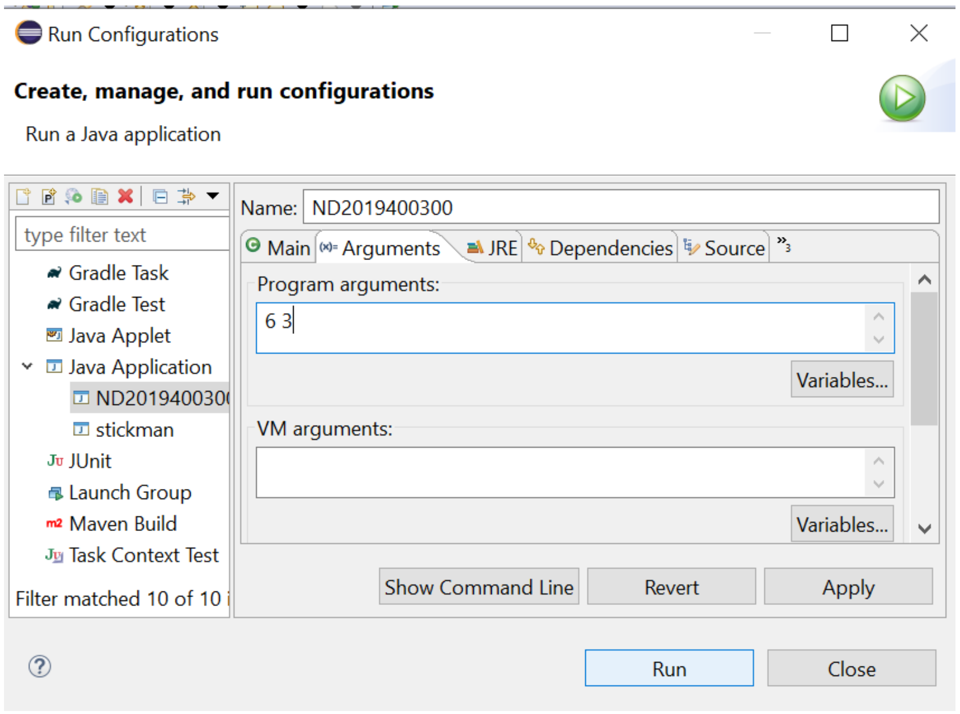
```
5 2
```

VM arguments:



```
<terminated> ND2019400300 [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Oct 26, 2019, 12:10:57 PM)

 O
/|\
 |       ___||
 |    ___|***|
/ \___|******|


    O
   /|\
    |
    |    ___||
   / \___|***|
   ___|******|


       O
      /|\
       |
       |
      / \___||
      ___|***|
   ___|******|
```

stickmanHeight = 6; stairHeight = 3



Run Configurations

Create, manage, and run configurations

Run a Java application

type filter text
- Gradle Task
- Gradle Test
- Java Applet
- Java Application
  - ND201940030
  - stickman
- JUnit
- Launch Group
- Maven Build
- Task Context Test

Filter matched 10 of 10

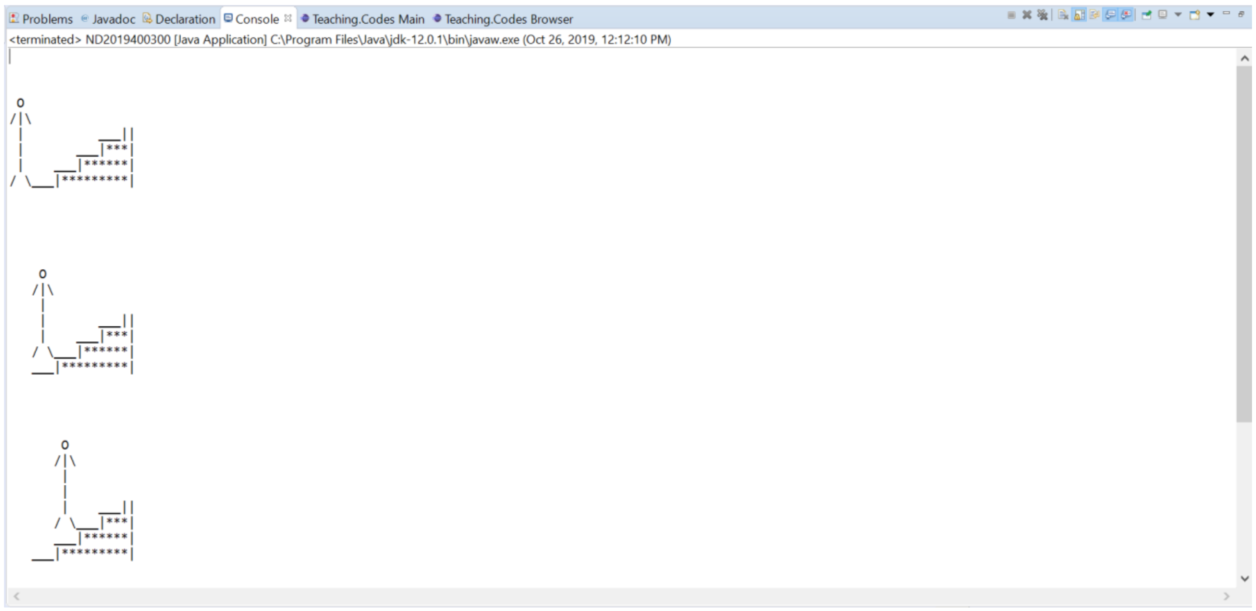
Name: ND2019400300

Main | Arguments | JRE | Dependencies | Source

Program arguments:

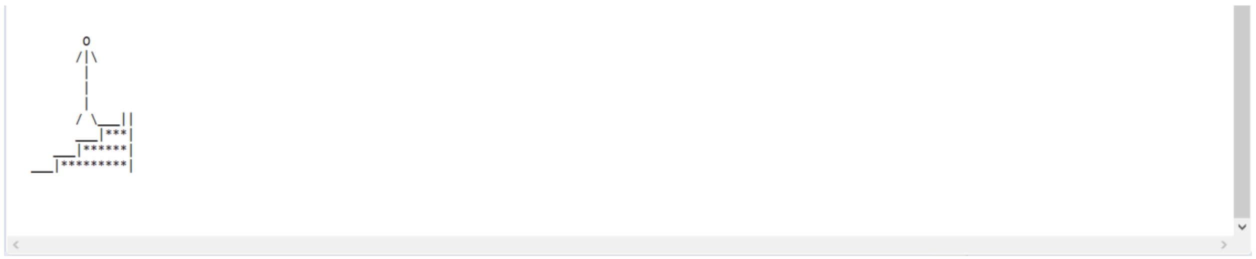
```
6 3
```

VM arguments:



<terminated> ND2019400300 [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Oct 26, 2019, 12:12:10 PM)

```
 o
/|\
 |          ___||
 |       ___|***|
 |    ___|******|
/ \___|*********|


    o
   /|\
    |
    |       ___||
    |    ___|***|
   / \___|******|
   ___|*********|


       o
      /|\
       |
       |
       |    ___||
      / \___|***|
      ___|******|
   ___|*********|
```



```
          o
         /|\
          |
          |
          |
         / \___||
         ___|***|
      ___|******|
   ___|*********|
```

3 Conclusion

All in all, the assignment was solved, and the code was written to the best of my ability. I found the assignment quite challenging at first. However, after understanding the logic behind it, I was able to quickly write the code. It was an interesting project, and I look forward to the next challenge. I hope that the solution has been satisfactory, and the reading interesting.

Thank you for reading.