CmpE 150 Introduction to Computing, Fall 2019 - Karahan Sarıtaş 2018400174

Assignment 1

Problem Description

We are asked to generate a stickman climbing the stairs by using Java. We need to have two parameters named as **stickmanHeight** and **stairHeight.** According to the values assigned to these parameters, the height of the stickman and the height of the stair will be declared, and our program is expected to generate the movement by complying with the assigned values. For example, when our values are assigned as 5 and 2 respectively, our code is expected to generate a figure with a height of 5 units and stairs with a height of 2 units – it's also important to notice that the head of the stairs is not included in the value of stairHeight. Empty lines from above should decrease throughout the implementation, whereas the total number of lines in a frame should stay same with the other frames for each case. Figure should start from the beginning of the stairs and start to climb step by step. It will stop when it arrives the last step with stars.

Problem Solution

First of all, in order to design an appropriate algorithm, I have created some tables that include the values of different variables for each unique case. I putted one of my tables below which inspects the case 5-2. To ease the writing part, I used abbreviations listed below.

stairHeight = m stair = n stars = s interspace = i

Example 5-2	Empty lines from above	Blank spaces between left side and the figure	Blank spaces between the figure (including legs) and the steps	Stars filling the steps	Spaces between the left side and the steps which stickman has passed.
Frame 1	2	0	6/3/0	0/3/6	-
Frame 2	1	3	3/0	0/3/6	3
Frame 3	0	6	0	0/3/6	6/3
Generalization	n	(m-n)*3	3*i	3*s	(m-n)*3
Change in the **first frame**	-	-	i=2 i=1 i->0 i=0	s=0 s=1 s -> n s=2	-
Change in the beginning of the different frames	n=2 n=1 n=0	n=2 n=1 n=0	i=2 i=1 i->0 i=0	s=0 s=0 s=0	-
Change in general	n -> 0	n -> 0	i = n for each frame i -> 0 throughout a frame	s=0 for each frame s -> n throughout a frame	(m-n) -> 1 for each frame

Although the table looks so confusing at the beginning, it helped me a lot to create a modular algorithm that holds for all cases. I used 5 methods excluding main method and 14 for loops including the for loops that come from methods each time I used them.

Methods

- I used a method named **Stars_or_Spaces** that generates stars filling the steps or spaces according to the assigned values. Since there were lots of different cases in which we need to leave spaces such as left side of the frame and the distance between stickman and the stairs, this method made my work easier.
- I used a method named **Stickman** that generates the figure according to the assigned String value. It creates the " | " (body), " O "(head), "/ \\" (legs) and "/|\\" (arms). This method was so helpful to improve readability.
- I used a method named **Stairs** that prints out the stairs. It included **Stars_or_Spaces** method two times with different variables. This method created spaces between the figure and the stairs and filled the stairs with appropriate number of stars for each line.
- I used a method named **Legs** which was literally composed of **Stickman** and **Stairs.** It generates the leg part of the stickman the corresponding step of the stairs. Although this method is not called more than once in the main method, I created this method to improve readability.
- I used a method named **Head_and_Torso** which calls the **Stickman** method two times. Similar to the **Legs** method, it is not used more than once in the main method.

For Loops

- I used 14 for loops throughout my code. I used one for loop to count the number of frames, one loop to leave empty lines, one loop to generate body part (" | "), one loop to generate the steps that correspond to the body part of the figure, one loop to generate the steps which the stickman passes for each case by climbing up. Other than these five for loops, I used loops in methods for two main purposes, leaving spaces and filling the stairs with stars.

Implementation

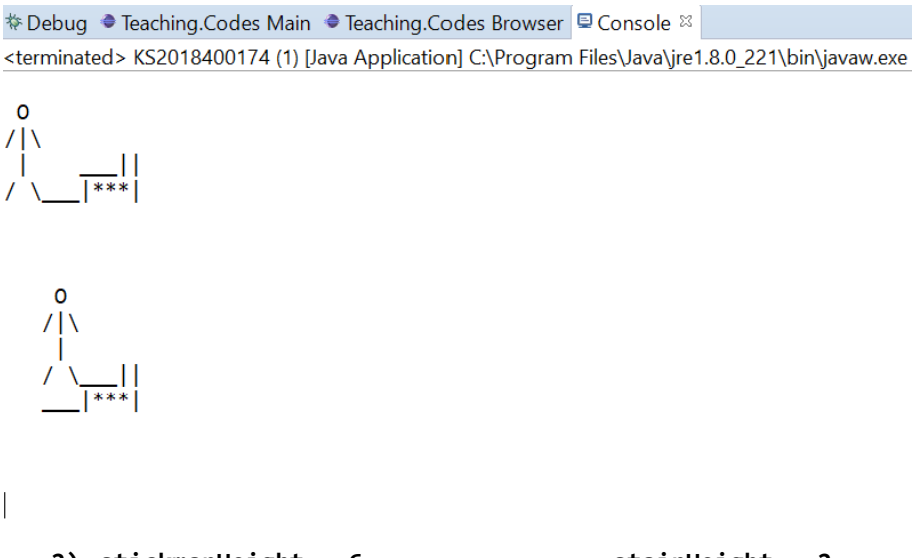
```
public class KS2018400174 {
	public static void main(String[] args) {
		int stickmanHeight = Integer.parseInt(args[0]);
		int stairHeight = Integer.parseInt(args[1]);
/* We have 4 major variables other than stickmanHeight and stairHeight
throughout the process.
* They are listed below with their tasks.

* 1) stair ==> Main purpose of the stair variable is to determine how many frames
there should be. It will decrease one by one each time a frame is finished and a new
one starts. It is also going to be used to calculate the number of empty lines from
above for each frame and spaces between the figure and the left side of each frame
since number of empty lines varies according to the frame number and the number
of spaces from left side is proportional to the (stairHeight-stair)*3.
*
```

```
* 2) interspace ==> This variable is used to generate the spaces between the stickman
and the stairs. It takes the value of the stair variable at the very beginning of the
first for loop and decreases by one each time a line, which includes spaces between
the figure and the step of the stairs, is completed.
*
* 3) line_count ==> This variable is just used to count the completed lines and give
the number of the current line. When it reaches to the value of the stickmanHeight,
steps of the stairs start to appear. At the beginning of the code, line_count is
declared as (2 + stair), since the number of empty lines equals to the value of stair
variable and line_count automatically increases by 2 with the lines including head
and torso.
*
* 4) stars ==> stars variable is used to determine the number of stars which are
going to fill the stairs for each frame. It takes the value of 0 at the beginning and
increases by one each time a step is completely finished. Number of stars in a step
is equal to stars*3 for each case.
*/
            int stair = stairHeight;
            for (; stair >= 0; stair--) {
// Value of the stair variable will decrease each time a new frame starts.
                  int interspace = stair, line_count = 2 + stair, stars = 0;
                  for (int empty_lines = stair; empty_lines > 0; empty_lines--) {
// empty_lines represents the number
// of blank lines from above.
                        System.out.println();
                  }
                  Head_and_Torso(stairHeight, stair);
// body represents the number of body " | " parts.
                  for (int body = (stickmanHeight - 3); body >= 1; body--) {
// body represents the number of body " | " parts.
                        Stickman(stairHeight, stair, " | ");
                        line_count++;
// line_count increases by one each time a " | " is completed.
                        for (; line_count >= stickmanHeight; line_count--,
interspace--, stars ++) {
                              Stairs(stars, interspace);
// When the value of line_count is equal to stickmanHeight,

                        }
// program starts to generate the stairs. Each time a step
                        System.out.println();
// is completed, interspace and line_count decreases, whereas
                  }
// the value of stars increases.
                  Legs(stairHeight, stair, stars,interspace);
// This part generates the line with legs and value of stars increases by one.
                  stars++;
                  int left_side = stairHeight - stair;
// left_side represents the number of blank spaces
                  for (; stars<= stairHeight; stars ++, left_side--) {
// between left side of the frame and the steps
                        Stairs(stars, left_side);
// which stickman has passed so far.
                        System.out.println();
                  }
```

```
                    System.out.println("\n\n");
// It leaves 3 empty lines after each frame.
                }
        }
        /*  Stars_or_Spaces method is used to generate the stars or spaces according
to the assigned values. When b is assigned as " ", and a is assigned as one of the
followings ( stairHeight - stair, interspace, left_side ) it generates the spaces
between the left side and the figure, figure and the stairs, left side and the stairs
according to the value of b respectively. When a is assigned as stars and b is
assigned as "*", it fills the steps with stars.
         */
        public static void Stars_or_Spaces(int a, String b) {
                for (int e = (a) * 3; e > 0; e--) {
                        System.out.print(b);
                }
        }
        /* Stickman is used to generate the parts of the figure. According to the
value assigned to the a variable, it prints out the corresponding part of the
stickman. For example, when a is " O ", it generates the head part.
         */
        public static void Stickman(int stairHeight, int stair, String a) {
                Stars_or_Spaces(stairHeight - stair, " ");
                System.out.print(a);
        }
        /* Stairs method is used to generate stairs and it calls Stars_of_Spaces
method two times with different variables assigned in it. Variable value is assigned
as left_side and interspace throughout the code.
         */
        public static void Stairs(int stars, int a) {
                Stars_or_Spaces(a, " ");
                System.out.print("___|");
                Stars_or_Spaces(stars, "*");
                System.out.print("|");
        }
        /* This method is used to create the part with legs. It calls Stickman method
to generate figure and Stairs method to generate the corresponding step. Then it
starts a new line.
         */
        public static void Legs(int stairHeight, int stair, int stars, int interspace)
{
                Stickman(stairHeight, stair, "/ \\");
                Stairs(stars, interspace);
                System.out.println();
        }
        /* This method is used to generate head and arms of the figure. It calls the
Stickman method two times with different String values.
         */
        public static void Head_and_Torso(int stairHeight, int stair) {
                Stickman(stairHeight, stair, " O ");
                System.out.println();
                Stickman(stairHeight, stair, "/|\\");
                System.out.println();

        }
}
```

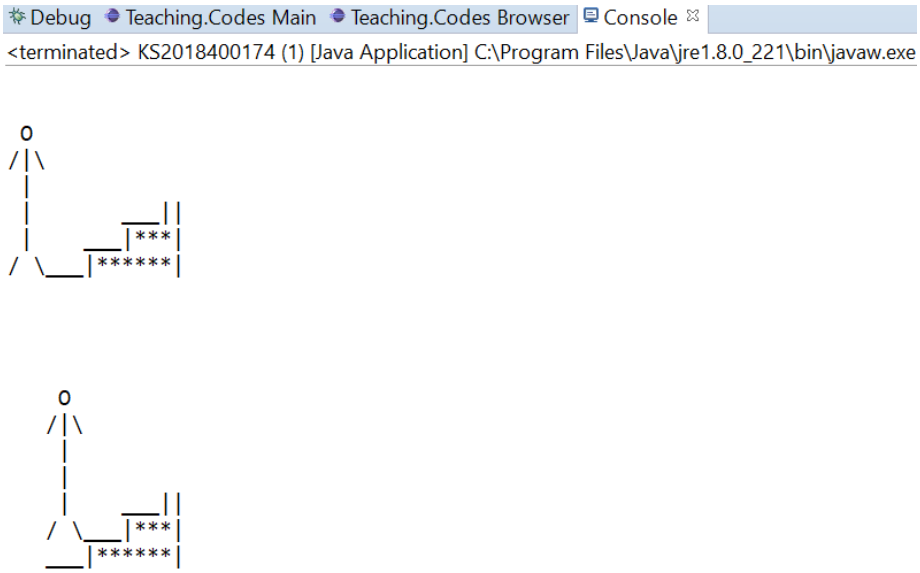
Example Runs

1) stickmanHeight = 4 stairHeight = 1




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


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

2) stickmanHeight = 6 stairHeight = 2




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


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


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

Conclusion

I have solved the problem, my code generates all possible situations with different **stickmanHeight** and **stairHeight** values. I tried to improve the readability as much as I could. Actually, some expressions such as if / else statements would be so useful for this project, but we are not allowed to use them. Moreover, I had to increase or decrease some values manually after calling a method, since we are not allowed to use instance variables which have wider scopes than local variables.