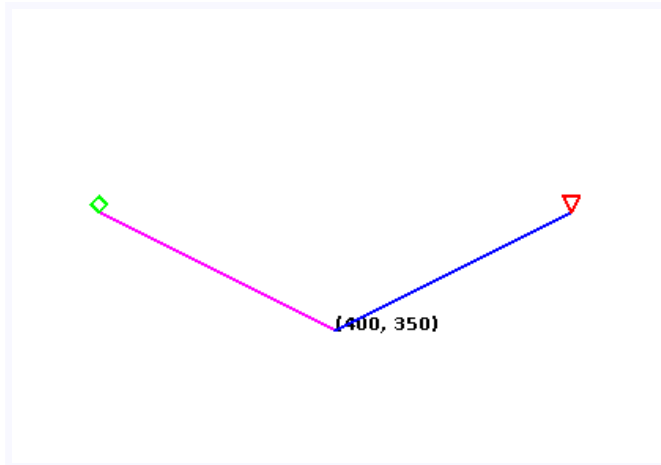


Model 1 The “Vee” Tree

Open *VeeTree.java* and run the program. Adjust the DELAY in *Drawing.java*, as needed, to notice the order. Then answer the questions below to explore and discuss the source code as a team.

Drawing (cropped)



Console output

```
Starting vee(400, 350)
diamond(250, 275)
triangle(550, 275)
Finished vee(400, 350)
```

Questions (15 min)

Start time:

1. The `vee()` method uses `g2` to draw strings and lines. Where is this variable declared?

In *Drawing.java*; it's an inherited attribute.

2. Review the Javadoc comment for the `vee()` method. In terms of its variables, what are the following coordinates?

a) The bottom of the “Vee”: (`x` , `y`)

b) The diamond on the left: (`left` , `top`)

c) The triangle on the right: (`right` , `top`)

3. Add the following line at the end of the left branch, after the trace for diamond. What happens when you run the program now? Explain both the drawing and the Console output.

```
vee(level + 1, left, top);
```

The left branch continues drawing “forever” (until a `StackOverflowError`). Based on the output, the `vee` method calls itself repeatedly with an increasing `level` each time.

4. Add the following code at the beginning of the method, right after the first trace. What happens when you run the program now? Explain both the drawing and the Console output.

```
if (level == 3) {  
    trace(level, "Aborting vee(%d, %d)", x, y);  
    return;  
}
```

The program draws a “Vee” shape at the top of each left branch, but it stops after 3 levels to avoid going on forever. The output increases to level 3, and then it decreases back to level 0.

5. Based on the most recent Console output:

- | | | | |
|-------------------------------------|---|-----------------------------------|---|
| a) How many times was vee() called? | 4 | c) How many diamonds were drawn? | 3 |
| b) How many times did vee() return? | 4 | d) How many triangles were drawn? | 3 |

6. Add the following line at the end of the right branch, after the trace for triangle. What happens when you run the program now? Explain both the drawing and the Console output.

```
vee(level + 1, right, top);
```

The drawing is now symmetrical—there is a “Vee” shape at the top of each branch. Along the top of the drawing, there are eight branches that abort the recursion. The output both increases and decreases in level as the drawing moves up and down.