CMPE 150.04

19.10.2016

CMPE 150 PROJECT 1

Problem Description:

This project contains shape like a soccer ball. Such shape has 3 items including “_”, “$”, “ “ and we are supposed to write a program which will use just these items to generate ASCII Soccer Ball and also find a pattern in order to write code easily. But, due to fact that sequential lines of this shape do not have certain order, we are supposed to write each line individually and find out how many dollar sign, underscore and blank will be printed.

Problem Solution:

To overcome this problem, I used five static methods (one for each problem) and two of them have for loop and also three of them have parameter.

The first method was line (int countAltCizgi, int countDollar). It has two methods; writingAltCizgi(int countAltCizgi) and writing(int countDollar). When line(expression1, expression2) calls, such methods specify the integer code to print. Each line has dollar signs and underscores with changing numbers and blank. int countAltCizgi and int countDollar represent how many dollar sign and underscore will be printed. Here, we have to make countAltCizgi and countDollar functional. To do it, we should generate methods for them; writingAltCizgi (int countAltCizgi) and writingDollar (int countDollar). These methods contains one for loop and for loop’s variable i is limited by the countAltCizgi and countDollar.

The second method was writingSpace(). Some lines have one blank between dollar sign and underscore and such methods which are mentioned above do not include blank. So, we have to generate method so as to put the blank between two outputs.

The third method was newline(). ASCII soccer ball given has no blank at the end of the lines and we have to declare new method to pass the other line. To pass the other line, we should write a println statement and since each line finishes with one dollar sign, println statement includes one dollar sign.

The fourth method was writingAltCizgi (int countAltCizgi). It contains one for loop and for loop’s variable is i is limited by the countAltCizgi. Firstly, line(expression1, expression2) is called, expression1 is essential for this method. Then, expression1 make such method functional and at the end, the number of expression 1 is printed as underscore.

The fifth method was writingDollar (int countDollar). It was very similar to the fourth one. Differently from the latter, it prints dollar sign.

Implementation:

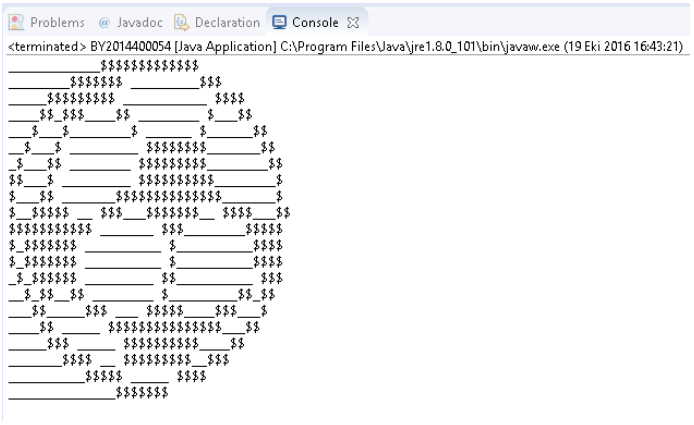
```java
//BY2014400054.java

public class BY2014400054 {


	public static void main(String[] args) {
		// 1
		line(12, 12);
		newLine();

		// 2
		line(8, 7);
		writingSpace();
		line(9, 2);
		newLine();

		// 3
		line(5, 9);
		writingSpace();
		line(11, 0);
		writingSpace();
		line(0, 3);
		newLine();

		// 4
		line(4, 2);
		line(1, 3);
		line(4, 2);
		writingSpace();
		line(8, 0);
		writingSpace();
		line(0, 1);
		line(3, 1);
		newLine();

		// 5
		line(3, 1);
		line(3, 1);
		line(8, 1);
		writingSpace();
		line(6, 0);
		writingSpace();
		line(0, 1);
		line(6, 1);
		newLine();

		// 6
		line(2, 1);
		line(3, 1);
		writingSpace();
		line(9, 0);
		writingSpace();
		line(0, 8);
```

```
line(7, 1);
newLine();

// 7
line(1, 1);
line(3, 2);
writingSpace();
line(8, 0);
writingSpace();
line(0, 9);
line(8, 1);
newLine();

// 8
line(0, 2);
line(3, 1);
writingSpace();
line(9, 0);
writingSpace();
line(0, 10);
line(8, 0);
newLine();

// 9
line(0, 1);
line(3, 2);
writingSpace();
line(7, 14);
line(7, 0);
newLine();

// 10
line(0, 1);
line(2, 5);
writingSpace();
line(2, 0);
writingSpace();
line(0, 3);
line(3, 7);
line(2, 0);
writingSpace();
line(0, 4);
line(3, 1);
newLine();

// 11
line(0, 11);
writingSpace();
line(7, 0);
writingSpace();
line(0, 3);
line(8, 4);
newLine();

// 12
```

```
line(0, 1);
line(1, 7);
writingSpace();
line(10, 0);
writingSpace();
line(0, 1);
line(10, 3);
newLine();

// 13
line(0, 1);
line(1, 7);
writingSpace();
line(10, 0);
writingSpace();
line(0, 1);
line(10, 3);
newLine();

// 14
line(1, 1);
line(1, 6);
writingSpace();
line(9, 0);
writingSpace();
line(0, 2);
line(10, 0);
writingSpace();
line(0, 2);
newLine();
// 15
line(2, 1);
line(1, 2);
line(2, 2);
writingSpace();
line(8, 0);
writingSpace();
line(0, 1);
line(9, 2);
line(1, 1);
newLine();

// 16
line(3, 2);
line(5, 3);
writingSpace();
line(3, 0);
writingSpace();
line(0, 5);
line(4, 3);
line(3, 0);
newLine();

// 17
line(4, 2);
```

```
            writingSpace();
            line(5, 0);
            writingSpace();
            line(0, 15);
            line(3, 1);
            newLine();

            // 18
            line(5, 3);
            writingSpace();
            line(5, 0);
            writingSpace();
            line(0, 10);
            line(4, 1);
            newLine();

            // 19
            line(7, 4);
            writingSpace();
            line(2, 0);
            writingSpace();
            line(0, 9);
            line(2, 2);
            newLine();

            // 20
            line(10, 5);
            writingSpace();
            line(5, 0);
            writingSpace();
            line(0, 3);
            newLine();

            // 21
            line(14, 6);
            newLine();
      }
    //This method ease to write underscores are sequenced.
      public static void writingAltCizgi(int countAltCizgi) {  //int countAltCizgi
represents how many underscore will be printed.
            for (int i = 1; i <= countAltCizgi; i++) {
                  System.out.print("_");
            }
      }
    //This method ease to write dollar signs are sequenced.
      public static void writingDollar(int countDollar) {      //int countDollar
represents how many dollar sign will be printed.
            for (int i = 1; i <= countDollar; i++) {
                  System.out.print("$");
            }
      }

      //Here, writingSpace() method make spacing easy.
      public static void writingSpace() {
            System.out.print(" ");
```

```
        }
        /*ASCII soccer ball shape is given has no blank at the end of the all lines
          so, we have to declare new method to pass the other line.*/
        public static void newLine() {
            System.out.println("$");
        }
    /*This method ease to write sequencing underscores and dollar signs
      without using for loop.*/
        public static void line(int countAltCizgi, int countDollar) {  //
              writingAltCizgi(countAltCizgi);
              writingDollar(countDollar);
        }
}
```



Conclusion:

I have solved the problem correctly, because when I run BY2014400054.java, I do not get any compiler errors and I get the same output as the one in the example.