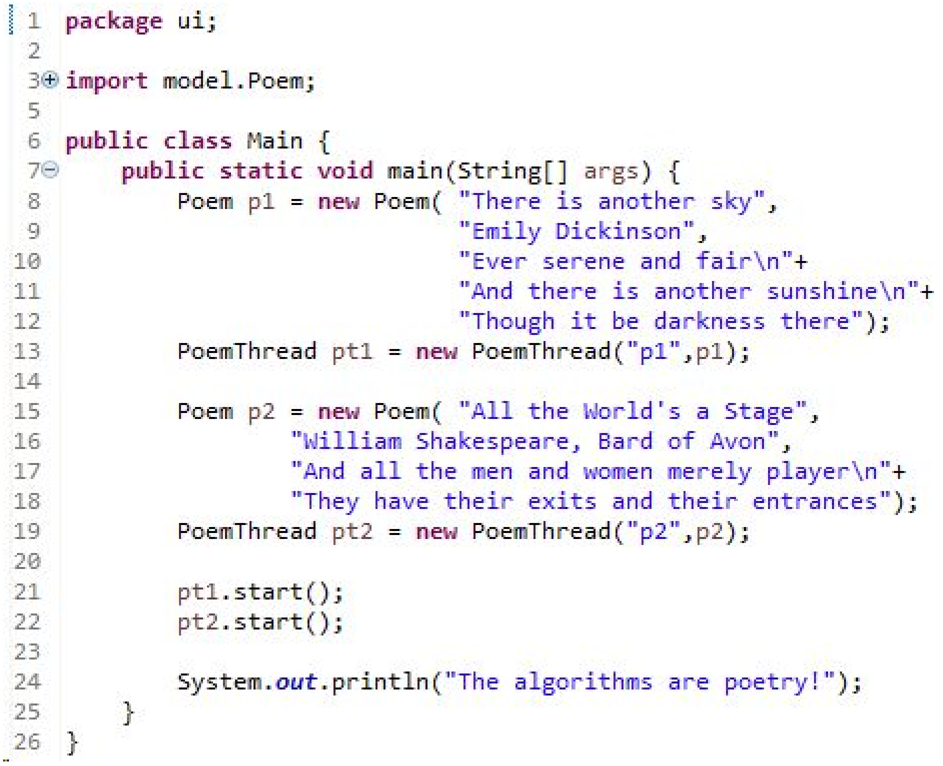
Algoritmos y Programación II
2019-1 - Seguimiento sobre Hilos

Nombre:		**Código:**
Profesor:	**Grupo:**	**Fecha: 4** de Marzo de 2019

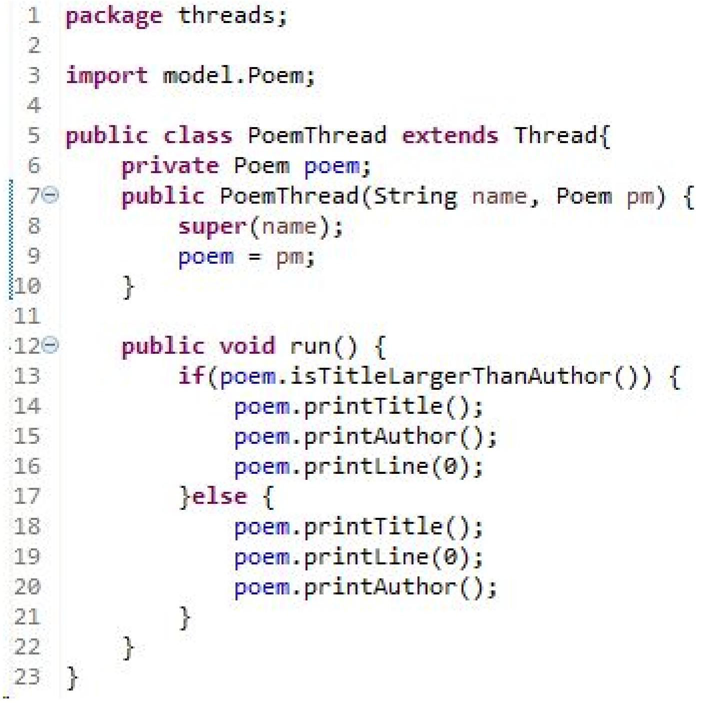
Unidad 3: Concurrencia, Dibujo Básico en 2D y Persistencia
Al finalizar esta unidad, el estudiante estará en capacidad de:
OE3.1 Desarrollar un programa que maneje concurrencia, de manera que sea posible que ejecute más de una parte del programa de manera simultánea, utilizando hilos de ejecución (threads).

Enunciado

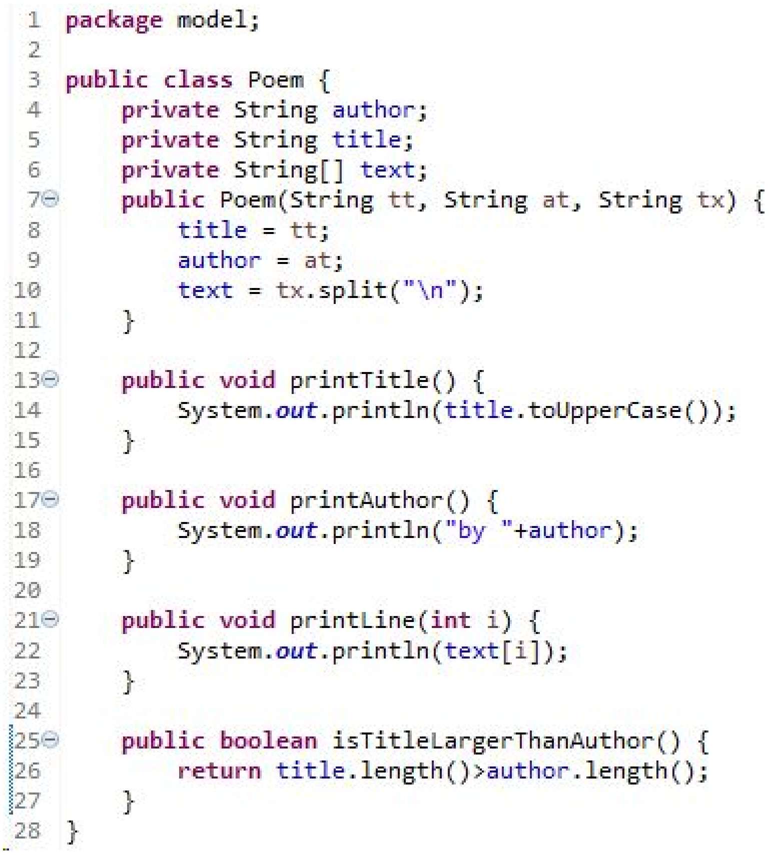
Teniendo en cuenta la ejecución del siguiente código, indique debajo de cada una de las salidas mostradas a continuación si es una salida posible del programa o si no lo es, y justifique su respuesta. Indique también al lado de cada línea el nombre del hilo que la imprimió.



```java
package ui;

import model.Poem;

public class Main {
    public static void main(String[] args) {
        Poem p1 = new Poem( "There is another sky",
                            "Emily Dickinson",
                            "Ever serene and fair\n"+
                            "And there is another sunshine\n"+
                            "Though it be darkness there");
        PoemThread pt1 = new PoemThread("p1",p1);

        Poem p2 = new Poem( "All the World's a Stage",
                "William Shakespeare, Bard of Avon",
                "And all the men and women merely player\n"+
                "They have their exits and their entrances");
        PoemThread pt2 = new PoemThread("p2",p2);

        pt1.start();
        pt2.start();

        System.out.println("The algorithms are poetry!");
    }
}
```



```java
package threads;

import model.Poem;

public class PoemThread extends Thread{
    private Poem poem;
    public PoemThread(String name, Poem pm) {
        super(name);
        poem = pm;
    }

    public void run() {
        if(poem.isTitleLargerThanAuthor()) {
            poem.printTitle();
            poem.printAuthor();
            poem.printLine(0);
        }else {
            poem.printTitle();
            poem.printLine(0);
            poem.printAuthor();
        }
    }
}
```



```java
package model;

public class Poem {
    private String author;
    private String title;
    private String[] text;
    public Poem(String tt, String at, String tx) {
        title = tt;
        author = at;
        text = tx.split("\n");
    }

    public void printTitle() {
        System.out.println(title.toUpperCase());
    }

    public void printAuthor() {
        System.out.println("by "+author);
    }

    public void printLine(int i) {
        System.out.println(text[i]);
    }

    public boolean isTitleLargerThanAuthor() {
        return title.length()>author.length();
    }
}
```

Salida 1	**Salida 2**	**Salida 3**
The algorithms are poetry! THERE IS ANOTHER SKY by Emily Dickinson Ever serene and fair ALL THE WORLD'S A STAGE And all the men and women merely player by William Shakespeare, Bard of Avon	THERE IS ANOTHER SKY by Emily Dickinson And all the men and women merely player ALL THE WORLD'S A STAGE The algorithms are poetry! Ever serene and fair by William Shakespeare, Bard of Avon	THERE IS ANOTHER SKY ALL THE WORLD'S A STAGE And all the men and women merely player by Emily Dickinson Ever serene and fair by William Shakespeare, Bard of Avon The algorithms are poetry!
Es una salida posible?:	Es una salida posible?:	Es una salida posible?:
Justificación:	Justificación:	Justificación: