

Nombre:		Código:
Profesor:	Grupo:	Fecha: 16 de noviembre 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 con claridad. Indique también al lado de cada línea el nombre del hilo que la imprimió.

```

1 package ui;
2 import model.Poem;
3
4
5 public class Main {
6     public static void main(String[] args) throws InterruptedException {
7         Poem p1 = new Poem( "When You Come",
8                             "Maya Angelou",
9                             "When you come to me, unbidden\n" +
10                            "Beckoning me\n" +
11                            "To long-ago rooms\n" +
12                            "Where memories lie.");
13         PoemThread pt1 = new PoemThread("p1",p1);
14         Poem p2 = new Poem( "It is Here",
15                             "Harold Pinter",
16                             "What sound was that?\n" +
17                             "I turn away, into the shaking room.");
18         PoemThread pt2 = new PoemThread("p2",p2);
19         pt1.start();
20         pt2.start();
21         pt1.join();
22         System.out.println("The algorithms are poetry?");
23     }
24 }

```

```

1 package threads;
2
3 import model.Poem;
4
5 public class PoemThread extends Thread{
6     private Poem poem;
7     public PoemThread(String name, Poem pm) {
8         super(name);
9         poem = pm;
10    }
11
12    public void run() {
13        if(poem.isTitleLargerThanAuthor()) {
14            poem.printTitle();
15            poem.printAuthor();
16            poem.println(0);
17        }else {
18            poem.printTitle();
19            poem.println(0);
20            poem.printAuthor();
21        }
22    }
23 }

```

```

1 package model;
2
3 public class Poem {
4     private String author;
5     private String title;
6     private String[] text;
7     public Poem(String tt, String at, String tx) {
8         title = tt;
9         author = at;
10        text = tx.split("\n");
11    }
12
13    public void printTitle() {
14        System.out.println(title.toUpperCase());
15    }
16
17    public void printAuthor() {
18        System.out.println("by "+author);
19    }
20
21    public void printLine(int i) {
22        System.out.println(text[i]);
23    }
24
25    public boolean isTitleLargerThanAuthor() {
26        return title.length()>author.length();
27    }
28 }

```

Salida 1		Salida 2		Salida 3	
IT IS HERE What sound was that? WHEN YOU COME by Maya Angelou When you come to me, unbidden by Harold Pinter The algorithms are poetry?		IT IS HERE What sound was that? WHEN YOU COME by Maya Angelou When you come to me, unbidden The algorithms are poetry? by Harold Pinter		WHEN YOU COME by Maya Angelou When you come to me, unbidden The algorithms are poetry? IT IS HERE What sound was that? by Harold Pinter	
Es una salida posible?:		Es una salida posible?:		Es una salida posible?:	
Justificación:		Justificación:		Justificación:	