ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

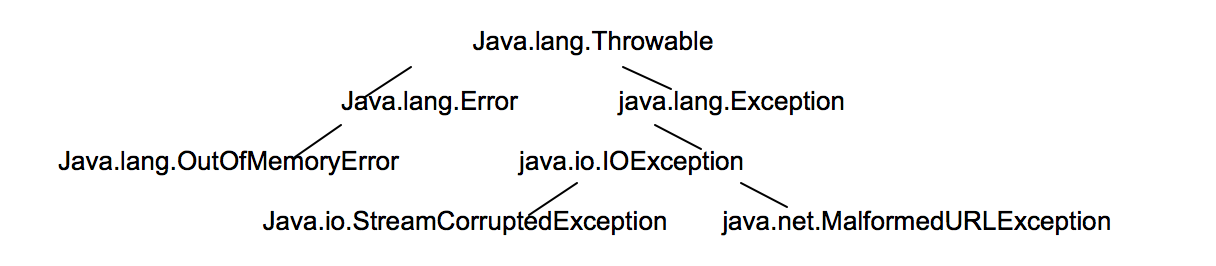
**EXAMEN FINAL 2016-II**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

1. **(5 puntos) Responda Verdadero (V) o Falso (F)**

|  |  |
| --- | --- |
| Los eventos son acciones iniciadas por el usuario, que al llevarse a cabo provocan la creación de un objeto cuyo tipo depende del tipo de componente sobre el que se haya llevado a cabo la acción | V F |
| Se puede crear un Thread, creando una clase que  implemente la interfase Runnable, hacer el overriding del método start() y luego ejecutar el método run(), siendo estos dos métodos de la interfase Runnable. | V F |
| El administrador de diseño StackPane organiza a los nodos de manera consecutiva, tal que se reorganizan en base a la forma de la ventana. | V F |
| En excepciones, la ejecución del bloque finally solo se da cuando no se ejecuta una excepción | V F |
| Una condición de carrera en hilos ocurre si el efecto de múltiples hilos sobre datos compartidos no depende del orden en que los hilos son programados. | V F |

1. **(15 puntos) Encierre la respuesta correcta**
   1. **(5 puntos) Considere la siguiente jerarquía de clases:**



**Y dado el siguiente código:**

1. try{
2. //asuma que s esta previamente definida
3. URL u = new URL(s);
4. //in es un ObjectInputStream
5. Object o = in.readObject();
6. System.out.println(“Success”);
7. }
8. catch (MalformedURLException e){
9. System.out.println(“Bad URL”);
10. }
11. catch (StreamCorruptedException e) {
12. System.out.println(“Bad file contents”);
13. }
14. catch (Exception e) {
15. System.out.println(“General Exception”);
16. }
17. finally {
18. System.out.println(“Doing finally part”);
19. }
20. System.out.println(“Carrying on”);

**¿Qué líneas son las salidas si el método en la línea #5 dispara un StreamCorruptedException? (Escoja todas las que aplican)**

1. Success
2. Bad URL
3. Bad files contents
4. General Exception
5. Doing finally part
6. Carrying on
   1. **(5 puntos) Dado el siguiente código indique cuál es la salida**
7. import java.util.Map;
8. import java.util.Set;
9. import java.util.TreeMap;
10. public class MapClass {
11. public static void main(String[] args) {
12. Map<String, String> partList = new TreeMap<>();
13. partList.put("P002", "Large Widget");
14. partList.put("P001", "Widget");
15. partList.put("P002", "X-Large Widget");
16. Set<String> keys = partList.keySet();
17. for (String key : keys) {
18. System.out.println(key + " " + partList.get(key));
19. }
20. }
21. }

|  |  |  |
| --- | --- | --- |
| i. P001 Widget  P002 X-Large Widget | iii. P002 X-Large Widget  P001 Widget | v. Compilación Falla |
| ii. P002 Large Widget  P001 Widget | iv. P001 Widget  P002 Large Widget |  |

* 1. **(5 puntos) Seleccione la respuesta correcta y justifique su respuesta.**

Whimsical Toys Inc (WTI) quiere crear una nómina con sus empleados. Está interesado en obtener un reporte de sus empleados por departamento en el que ellos trabajan. Para dicha problema cuál de las siguientes interfaces sería la más apropiada y explique cómo la utilizaría.

* + 1. Set
    2. Map
    3. List
    4. Queue

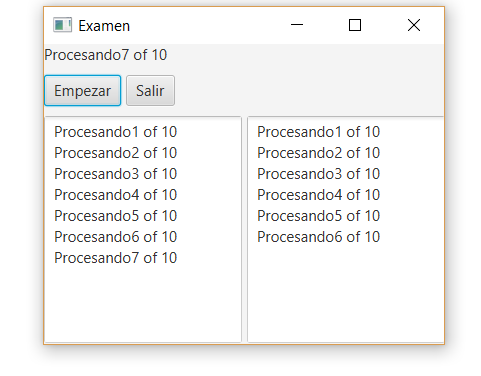
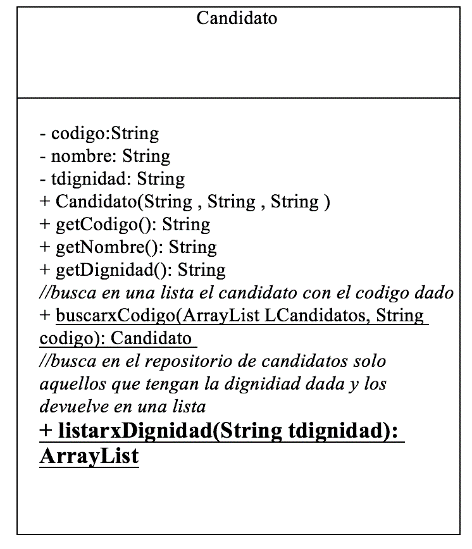
1. **(30 puntos) Evaluación de código.** 
   1. **(10 puntos) Dado el siguiente código, muestre la salida y justifique:**
2. class Plane {
3. static String s = "-";
4. public static void main(String[] args) {
5. new Plane().s1();
6. System.out.println(s);
7. }
8. void s1() {
9. try {
10. s2();
11. } catch (Exception e) {
12. s += "c";
13. }
14. }
15. void s2() throws Exception {
16. s3();
17. s += "2";
18. s3();
19. s += "2b";
20. }
21. void s3() throws Exception {
22. throw new Exception();
23. }
24. }
    1. **(10 puntos) Muestre la salida del siguiente código y justifique:**
25. public class Dog implements Comparable<Dog>{
26. private int age;
27. private String name;
28. public Dog(int age, String name){
29. this.age = age;
30. this.name = name;
31. }
32. public int compareTo(Dog d){
33. return Integer.valueOf(age).compareTo(Integer.valueOf(d.age));
34. }
35. public static void main(String[] args){
36. TreeSet<Dog> td = new TreeSet<Dog>();
37. td.add(new Dog(12,"firulais"));
38. td.add(new Dog(10,"bobby"));
39. td.add(new Dog(8,"totto"));
40. TreeSet<Integer> ti = new TreeSet<Integer>();
41. ti.add(new Integer(1));
42. ti.add(new Integer(3));
43. ti.add(new Integer(1));
44. for(Dog d : td){
45. System.out.println(d.name+" "+d.age);
46. }
47. for(Integer i : ti){
48. System.out.println(i);
49. }
50. }
51. }
    1. **(10 puntos) Usted quiere usar la siguiente clase como un tipo en una colección HashSet<T>. ¿Qué está faltante y cómo lo arreglaría?**

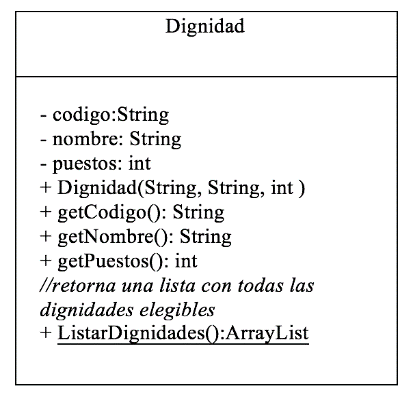
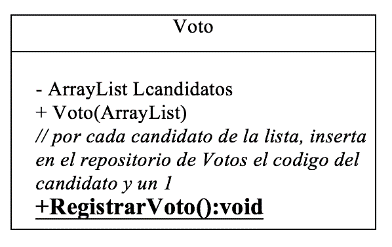
public class Customer{  
       private String name;  
       private String address;  
       public Customer(String nom, String dir)  
       {

           name = nom; address = dir;  
       }  
 public String toString()  
       {

         return name + " : " + address;  
       }

}

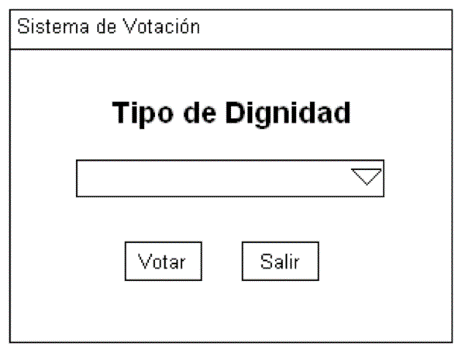
1. **( 15 puntos) Implemente la siguiente pantalla**
   1. **(5 puntos)** Cree la interfaz de usuario usando JavaFX
   2. **(5 puntos)** Cree una clase ExamenHilo que actualice una caja de texto y una etiqueta (Label)
   3. **(5 puntos)** Cree 2 Threads de la clase ExamenHilo que actualice cada caja de texto, el segundo Thread deberá esperar 1 segundo para poder comenzar a ejecutarse.
2. **( 35 puntos) Se necesita implementar un sistema para votación electrónica, para lo cual ya se tiene disponible un repositorio de datos (archivos) y las clases del siguiente diagrama UML:**



* 1. **(15 puntos)** Implemente los dos métodos en negrita del diagrama de clases previo

**Nota:** Para esto Ud. dispone del archivo Candidatos y del archivos Votos.

* 1. **(5 puntos)** Construya la aplicación EleccionDignidad que represente la siguiente ventana. En esta, el votante podrá elegir la dignidad por la que desea votar. El botón Votar le muestra una ventana EleccionCandidatos y  el botón Salir cierra la aplicación.



* 1. **(15 puntos)** Construya la clase EleccionCandidatos que represente la siguiente ventana. Esta recibe el objeto Dignidad elegido anteriormente y muestra sólo los candidatos que participan para esa dignidad. Para elegir un candidato, el votante debe presionar el botón con el código respectivo y este se deshabilitará. El votante no puede elegir más de los puestos disponibles para esa dignidad. El botón Votar registrará los votos en el repositorio de datos.

