

Certamen I ELO-329 Diseño y Programación Orientada a Objetos (1s2023)

En este certamen usted no podrá hacer preguntas. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.

Primera parte, **sin apuntes** (40 minutos):

Primera pregunta (30 pts.)

A. (5 pts.)

- i) Se dice que Java permite desarrollar una solución en un computador y luego esta solución se puede ejecutar en otro con distinto sistema operativo. ¿Qué hace posible esta portabilidad?

Esto es posible gracias a que el código Java corre sobre una máquina virtual. Si contamos con la máquina virtual Java para una arquitectura de computador y sistema operativo, podremos correr allí la solución Java desarrollada en otro sistema operativo y tipo de computador.

- ii) Considere 2 usuarios: uno utiliza Linux en dos computadores, uno con arquitectura de 64 bits (x64) y otro con arquitectura ARM; el otro usuario tiene el mismo computador de 64 bits del primero pero le instaló Microsoft Windows.

¿Cuántas versiones del JDK (Java Development Kit) debe preparar Oracle para su uso en los computadores de estos dos usuarios? Explique su respuesta.

Se requiere preparar 3 versiones del JDK. Se requiere desarrollar una por cada combinación de sistema operativo y arquitectura. En este caso son: Linux en arquitectura x64, Linux en ARM, y Microsoft Windows en x64.

- B. (5 pts.) Se necesita crear una matriz triangular superior de enteros como la mostrada en la figura. Muestre el segmento de código para crearla suponiendo que los valores son leídos desde consola y se dispone del objeto `in`, instancia de `Scanner`. Los números no mostrados son cero y no requieren ser ingresados ni almacenados. Notar que, si la matriz es `m`, `m[1][2]` permite acceder al elemento de la segunda fila, tercera columna.

$$M = \begin{bmatrix} 1 & 8 & 3 & 4 \\ & 5 & 15 & 7 \\ & & 8 & 9 \\ & & & 10 \end{bmatrix}, M[1][2] \text{ corresponde a } 15$$

Para generar esta matriz, la ejecución de su Código debería ser algo del tipo:

Ingrese la fila 0 de la matriz: 1 8 3 4

Ingrese la fila 1 de la matriz: 5 15 7

Ingrese la fila 2 de la matriz: 8 9

Ingrese la fila 3 de la matriz: 10

Scanner in = new Scanner (System.in) // continúe con lo que falte.

```
int [][] m = new int[4][];
for (int i=0; i< m.length; i++){
    m[i]= new int[m.length-i];
    System.out.print("Ingrese la fila "+i+" de la matriz:");
    for(int j=0; j<m[i].length; j++)
        m[i][j]= in.nextInt();
}
```

Nota: con este almacenamiento, para acceder a M[i][j], se debe usar (i>j ? 0:m[i][j-i])

C. (5 pts.) Dado el conjunto de clases presentes en la caja de la izquierda ¿Qué imprime el segmento de código de la derecha?

| | |
|---|--|
| <pre>public class Bird { String fly(){ return "The bird try to fly"; } } public class Swan extends Bird { String fly() { return "Fly with slow wingbeats and outstretched necks"; } } public class Condor extends Bird { String fly() { return "Fly long distances without moving their wings"; } }</pre> | <pre>. . Bird obBird1 = new Bird(); Swan obSwan1 = new Swan(); Condor obCondor1 = new Condor(); System.out.println(obBird1.fly()); obBird1 = obSwan1; System.out.println(obBird1.fly()); System.out.println(obCondor1.fly()); . .</pre> |
|---|--|

The bird tries to fly

Fly with slow wingbeats and outstretched necks

Fly long distances without moving their wings

D. (5 pts.) Considere que usted tiene la clase A y de esta deriva la clase B. A su vez, B adopta los comportamientos que ofrece la interface C.

| | |
|---|---|
| <pre>public class A { protected double b, a; public A(double b, double a) { this.a = a; this.b = b; } public double getCalculate() { return -b / (2*a); } }</pre> | <pre>public class B extends A { private double c; public B(double a, double b, double c) { super(a,b); } public double getCalculate() { return getCalculate() + Math.sqrt(b*b - 4*a*c) / (2*a); } @Override</pre> |
|---|---|

| | |
|---|---|
| } | <pre>public void check() { System.out.println("Check Calculation"); }</pre> |
|---|---|

- i) ¿Qué modificación debe hacerse a las clases anteriores para que el siguiente código sea válido? Identifique la interface.

```
C c = new B(4,5,3);
c.check();
```

La clase B debe implementar la interface de la siguiente manera:

```
class B extends A implements C
```

Definición de la Interfaz

```
interface C {
    void check();
}
```

- ii) Argumente sobre el correcto funcionamiento del método *getCalculate()* en la clase B.

Como el método *getCalculate()* de B depende del método *getCalculate()* de la clase A entonces se debe utilizar dicho método invocándolo utilizando *super*, de la siguiente manera:

```
return super.getCalculate() + Math.sqrt(b*b - 4*a*c) / (2*a);
```

- E. (5 pts.) ¿Cuál es la relación entre Stage y Scene en JavaFX y cuál es el propósito de cada una en la creación de interfaces gráficas de usuario?

Stage corresponde a la ventana principal, mientras que Scene corresponde a un contenedor de elementos gráficos que se despliega en una Stage en un cierto momento. Debido a esta relación, puede haber múltiples Scene conectadas a una misma Stage, pero solo una de ellas se puede mostrar en un cierto momento.

- F. (5 pts.) Dos estudiantes del ramo ELO329 realizan las siguientes declaraciones:

Estudiante A: "Toda clase anónima puede ser reescrita como expresión lambda"

Estudiante B: "Toda expresión lambda puede ser reescrita como clase anónima"

Comente sobre la veracidad de dichas declaraciones. Justifique.

La primera declaración es falsa, pues las clases anónimas pueden poseer múltiples métodos y atributos, lo cual no es posible representar con una expresión lambda, por lo que la frase es falsa. Sin embargo, la segunda declaración es verdadera, pues las expresiones lambda son un caso particular de una clase anónima con solo un método implementado.