ELO329 1s2022, 1er certamen. 9.05.22

Certamen I ELO-329 Diseño y Programación Orientada a Objetos

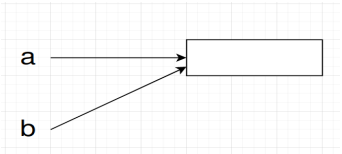
En este certamen usted **no podrá hacer preguntas**. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.
Tiempo: **70 minutos, sin extensión.**

Primera pregunta (10 pts)

A. ¿Cuál es la diferencia entre un lenguaje compilado y un lenguaje interpretado? ¿En qué clasificación está el lenguaje java?
   Un lenguaje compilado debe ser compilado para generar un programa en lenguaje de máquina el cual es ejecutado directamente por el hardware. Un lenguaje interpretado necesita otro programa para poder ejecutarse. Java es un lenguaje interpretado, aun cuando previo a ser interpretado debe ser compilado para generar el código bytes que luego es interpretado por una máquina virtual. Hay maneras de trabajar en lenguaje de máquina. 2pts
B. Escriba tres características de los lenguajes orientados a objetos.
   Permiten expresar herencia.
   Se pueden usar subtipos
   Permite usar abstracción. 2 pts
C. Anote 6 tipos primitivos en java.
   boolean, int, short, long, byte, float. 3 pts
D. Si se tiene lo siguiente:

```
int [] a = new int[5];
int [] b = a;
b[3] = 67;
```

- ¿El arreglo b es independiente del arreglo a? Realice un diagrama de los bloques de memoria.
  Ambos arreglos apuntan al mismo espacio de memoria



- ¿Cuánto vale el cuarto elemento del arreglo a?
  67 d) 3pts.

Segunda pregunta (10 pts)

Dada la siguiente clase:

```
class Estudiante {
   private int nota_math, nota_lang, nota_science;
   public Estudiante() {
       nota_math = nota_lang = nota_science = 0;
   }
   public Estudiante(int m, int l, int s) {
       nota_math = m;
       nota_lang = l;
       nota_science = s;
   }
   public Estudiante(int m, int l) {
```

```
        nota_math = m;
        nota_lang = l;
        nota_science = 0;
    }
    public Estudiante(int m) {
        nota_math = m;
        nota_lang = 0;
        nota_science = 0;
    }
    public int getMath() {
        return nota_math;
    }
    public int getLang() {
        return nota_lang;
    }
    public int getScience() {
        return nota_science;
    }
}
```

A. ¿Qué imprimirá por pantalla el siguiente bloque de código?

```
Estudiante x = new Estudiante(12,13)
System.out.println(x.getMath());
System.out.println(x.getLang());
System.out.println(x.getScience());
12
13
0          3pts
```

B. ¿Cuáles son los valores de los atributos de los siguientes dos objetos?

```
Estudiante x = new Estudiante(78,10)
Estudiante y = new Estudiante(44)
```

Objeto	nota_math	nota_lang	nota_science
x	78	10	0
y	44	0	0

4pts.

C. ¿Hay un problema en el siguiente código? Justifique su respuesta.

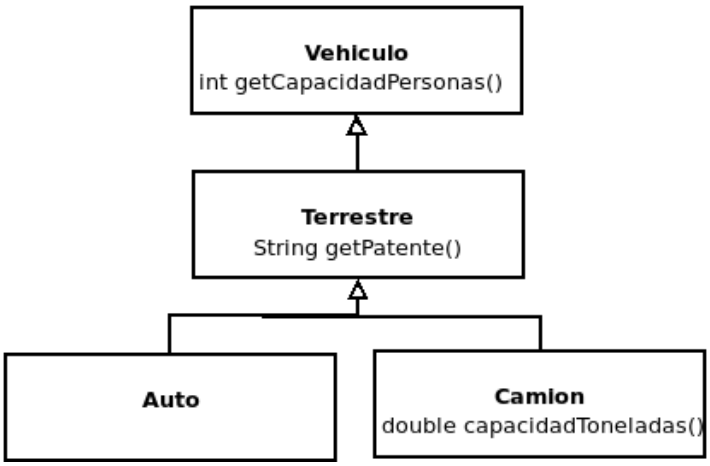
```
Estudiante x = new Estudiante(78,10)
System.out.println(x.nota_science);
```

No corre, nota_science es private. 3pts.

Tercera pregunta (10 puntos)

A continuación, se presenta un diagrama de clases

ELO329 1s2022, 1er certamen. 9.05.22



Implemente el código de las clases Vehículo, Terrestre, Auto y Camión. **OBSÉRVESE QUE** los constructores no fueron presentados a propósito para que sean implementados y pensados por usted, además tampoco se muestran los parámetros que usted debe definir en las clases para que esté todo según como se espera que funcionen los métodos propuestos en las clases.

SOLUCIÓN:

```
class Vehiculo{
        public Vehiculo(int cap){capacidadPersonas=cap;}
        int getCapacidadPersonas(){return capacidadPersonas;}
        private int capacidadPersonas;
}   3pts
class Terrestre extends Vehiculo{
        public Terrestre(String patente, int capacidadPersonas ){t
                super(capacidadPersonas);
                his.patente = patente;
        }
        public String getPatente(){return patente;}
        private String patente;
}.  3pts
class Camion extends Terrestre{
        public Camion(double tonelaje, String patente, int capacidadPersonas){
                super(patente, capacidadPersonas)
                this.tonelaje = tonelaje;
        }
        public double getTonelaje(){return tonelaje;}
        private double tonelaje;
} 2pts
class Autoextends Terrestre{
        public Auto(String patente, int capacidadPersonas){
                super(patente, capacidadPersonas)
        }
}.  2pts
```

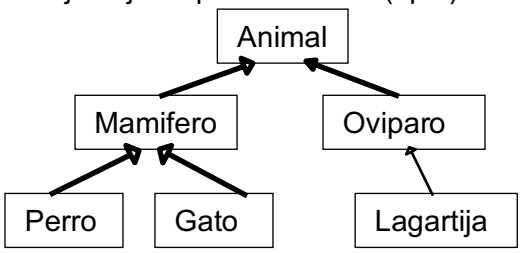
Cuarta pregunta (10’ puntos)

ELO329 1s2022, 1er certamen. 9.05.22

A continuación se presentarán las siguientes clases

```
class Animal{}
class Mamifero extends Animal{}
class Oviparo extends Animal{}
class Perro extends Mamifero {}
class Gato extends Mamifero {}
class Lagartija extends Oviparo{}
```

A. Dibuje la jerarquía de clases (3pts) :



B. ¿Qué clases podrían ser Abstractas? (entiéndase que ninguna fue explícitamente declarada abstracta sin embargo puede haber clases que en un **buen diseño** sí deberían ser abstractas), explique el criterio usado. (4ptos)

Solución: Los nodos terminales son por lógica no Abstractos, y más se refuerza por el ejemplo planteado, sin embargo, Animal, Mamifero o Oviparo pueden ser considerables Abstractas.

C. De las siguientes líneas de código ¿cuáles compilan? ¿cuál es la clase de la **referencia**?¿cuál es la clase de la **instancia**? si no compila no debe explicitar referencia ni instancia. (3ptos) (0.5ptos c/u)

Gato a = new Perro().	¿Compila? F	Clase Referencia: -	Clase Instancia: -
Oviparo b = new Gato().	¿Compila? F	Clase Referencia: -	Clase Instancia: -
Animal c = new Perro().	¿Compila? V	Clase Ref.: Animal	Clase Instancia: Perro
Lagartija d = new Oviparo().	¿Compila? F	Clase Ref: -	Clase Instancia: -
Mamifero e = new Object().	¿Compila? F	Clase Ref: -	Clase inst: -
Object e = new Lagartija ().	¿Compila? V	Clase Ref: Object	Clase inst: Lagartija

Quinta pregunta (10 puntos)

En la siguiente definición de clase:

```
public class Box{
      private int x;
      private int y;

      static class Filler{
          void Filler(int pos_x, int pos_y){
                x = pos_x;
```

```
                y = pos_y;
            }
        }
}
```

A. ¿Compila?, fundamente su respuesta.
No compila. La razón es que las clases anidadas estáticas (Filler) no necesitan una instancia de la clase anfitriona (Box) y por lo tanto no pueden acceder a los atributos de la clase anfitriona. Además el constructor de Filler no puede retornar un valor "void" no procede. 5pts

B. Escriba el código para crear un objeto de la clase Filler. Suponga que no hay error en el código.

```
Box.Filler fill = new Box.Filler();    // 5pts.
```

Sexta pregunta (10 puntos)

Para el siguiente código:

```
interface Drawable{
    public void draw();
}
public class MiClase {
    public static void main(String[] args) {
        int width=10;


        Drawable d=new Drawable(){
            public void draw(){System.out.println("Drawing "+width);}
        };
        d.draw();
    }
}
```

Reescriba el código usando una expresión lambda.
Nota: Las expresiones lambda que no tienen argumentos se escriben de la forma : ()->{}
Respuesta: 10 pts.

```
interface Drawable{
    public void draw();
}
public class MiClase {
    public static void main(String[] args) {
        int width=10;
        Drawable d= () -> System.out.println("Drawing "+width);
        d.draw();
    }
}
```

Séptima pregunta (10 puntos)

Un desarrollador hace dos afirmaciones:

a) “Si usted usa JavaFX, que permite la programación basada en eventos, usted podrá programar incluso no basado en eventos”.

b) “Si usted usa un lenguaje cualquiera, no podrá programar basado en eventos”

¿Está usted de acuerdo con cada una de estas afirmaciones? Explique en cada caso.

R: a) Sí. Un programa Java FX parte con el método start. Si no se define y registra algún handler o listener, será equivalente a un “main” de un programa sin eventos. (5 pts.)
b) No. La programación basada en eventos se puede realizar en cualquier lenguaje a través de un loop infinito que espere por la llegada de algún evento y luego ejecute las tareas asociadas a esos eventos. (5 pts.) (Obs.: No se especifica el tipo de lenguaje, si usted da como ejemplo un lenguaje declarativo, se asignó puntaje máximo al señalar acuerdo con la afirmación.)

Octava pregunta (10 puntos)

Se tiene la siguiente aplicación JavaFX:

```
public class ButtonInPane extends Application { // 1
  public void start(Stage primaryStage) {       // 2
    StackPane pane = new StackPane();           // 3
    Button btOK = new Button("OK");             // 4
    pane.getChildren().add(btOK);               // 5
    Scene scene = new Scene(pane, 200, 50);     // 6
    primaryStage.setScene(scene);               // 8
    primaryStage.show();                        // 9
  }                                             // 10
}                                               // 11
```

Liste los pasos (no el código) que le faltan a esta aplicación para que al presionar btOK aparezca en consola el mensaje “OK presionado”. Muestre en qué parte codificaría esos pasos.

R: En cualquier parte posterior a la línea 4 y antes de la línea 9, se debe (4 pts.):

1.- Crear un objeto instancia de una clase que implemente la interfaz requerida por botón para reaccionar al evento de presionado. El método implementado debe generar la impresión en consola de “OK presionado”. (3 pts.)

2.- Registrar el objeto previo en btOK asociado al evento de presionar el botón. (3 pts.)