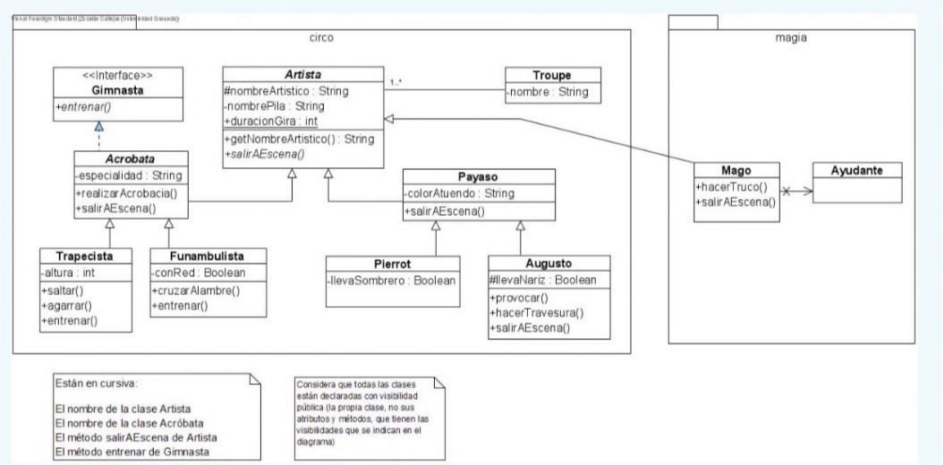

Pregunta **1**

Sin responder aún

Puntúa como 1,00

Marcar pregunta

Supón que todas las clases del diagrama tienen un constructor que inicializa todos sus atributos utilizando valores que le llegan como argumento.

Implementa en Ruby la cabecera de la clase Pierrot y su constructor.

```
class Pierrot < Payaso
  def initialize(nombreArtistico,nombrePila,colorAtuendo,sombrero)
    super(NombreArtistico,nombrePila,colorAtuendo)
    @llevaSombrero=sombrero
  end
end
```

Pregunta **2**

Sin responder aún

Puntúa como 1,00

Marcar pregunta

Implementa en Ruby los métodos salirAEscena de Payaso y de Augusto de modo que:

- en Payaso imprima por pantalla "¿Cómo están ustedes?"
- en Augusto haga lo que hace un Payaso y a continuación haga una travesura.

```
def salirAEscena() #Payaso
puts"¿Cómo están ustedes??"
end

def salirAEscena()#Augusto
  puts super+" y hago una travesura"
end
```

Pregunta **3**

Sin responder aún

Puntúa como 0,25

Marcar pregunta

Supón que tanto en Artista como en la subclase Trapecista tenemos los siguientes métodos de clase:

```
public static void setDuracionGira(int d){ duracionGira=d;}
public static int getDuracionGira(){ return duracionGira;}
```

¿Qué se imprimiría con las siguientes líneas de código?

```
Artista.setDuracionGira(2);
System.out.println(Artista.getDuracionGira());
System.out.println(Trapecista.getDuracionGira());
Trapecista.setDuracionGira(4);
System.out.println(Artista.getDuracionGira());
System.out.println(Trapecista.getDuracionGira());
```

- a. 2, 2, 2, 4
- b. 2, 2, 2, 2
- c. 2, 4, 2, 4
- d. 2, 2, 4, 4
- e. 2, 4, 4, 4

d. Porque las dos clases comparte el atributo(La hija lo hereda del padre). Entonces al modificarlo, se modifica en las 2.

Pregunta **4**

Sin responder aún

Puntúa como 0,25

Marcar pregunta

¿Y si la subclase tuviese su propio atributo de clase duracionGira inicializado a 4?

- a. 2, 2, 2, 2
- b. 2, 4, 2, 4
- c. 2, 2, 2, 4
- d. 2, 4, 4, 4
- e. 2, 2, 4, 4

b. En este caso ya no comparten atributo. Además al modificar trapecista su atributo, ya no modifica el del padre.

Supón que en la clase Pierrot existen los siguientes métodos adicionales donde compi es otro objeto Pierrot:

Ruby:

```
def presentacionCompi(compi)
 puts "Hoy está conmigo #{compi.nombreArtistico} vestido de #{compi.colorAtuendo}"
end
```

Java:

```
void autoPresentacion(Pierrot compi){
  System.out.println("Me llamo "+nombreArtistico+" y voy vestido de "+colorAtuendo);
}
```

Corrige los fallos en los métodos (si los hay). Si es necesario hacer algún cambio en alguna clase indícalo.

Pregunta **6**
Sin responder aún
Puntúa como 0.25
Marcar pregunta

Artista x1 = new Acrobata();

- a. Error de ejecución
- b. Sin errores
- c. Error de compilación

6. b

Pregunta **7**
Sin responder aún
Puntúa como 0.25
Marcar pregunta

Gimnasta x2 = new Funambulista();

7.b

- a. Sin errores
- b. Error de ejecución
- c. Error de compilación

Pregunta **8**

Payaso x3 = new Payaso();

Pregunta **8**

Sin responder aún

Puntúa como 0.25

Marcar pregunta

Payaso x3 = new Payaso();

String s = x3.getNombreArtistico();

- a. Sin errores
- b. Error de compilación
- c. Error de ejecución

8. a

Pregunta **9**

Sin responder aún

Puntúa como 0.25

Marcar pregunta

Acrobata x4 = new Trapecista();

x4.agarrar();

- a. Sin errores
- b. Error de compilación
- c. Error de ejecución

9.b

Pregunta **10**

Sin responder aún

Puntúa como 0.25

Marcar pregunta

Payaso pa2 = new Pierrot();
((Augusto) pa2).salirAEscena();

- a. Error de ejecución
- b. Sin errores
- c. Error de compilación

10.b

Pregunta **11**

Sin responder aún

Puntúa como 0.25

Marcar pregunta

Artista pay4 = new Augusto();
((Payaso) pay4).salirAEscena();

- a. Error de compilación
- b. Sin errores
- c. Error de ejecución

11.b

Pregunta **12**

Sin responder aún

Puntúa como 0.50

Marcar pregunta

Supón que en nuestro sistema hay algunos magos mentalistas que no hacen trucos como cualquier otro mago, sino que leen la mente.

Además, son capaces de hipnotizar, cosa que no cualquier mago puede hacer.

¿Cómo incluirías estas novedades en el sistema?

```
class Mentalista extends Mago{

  Mentalista(Strring nombreArtistico, String nombrePila){
    super(nombreArtistico,nombrePila);
  }
  @Override
  public hacerTruco(){
    System.out.println("El mago te está leyendo la mente");
  }
  public hipnotizar(){
    System.out.println("El mago te dice que mires
    fijamente al péndulo");
  }

}
```

```
class Mentalista < Mago{

  def initialize(Strring nombreArtistico, String nombrePila){
    super(nombreArtistico,nombrePila);
  end

  def hacerTruco(){
    puts "El mago te está leyendo la mente"
  end

  def hipnotizar(){
    puts "El mago te dice que mires fijamente al péndulo"
  end

end
```

Pregunta **13**
Sin responder aún
Puntúa como 0,15
Marcar pregunta

En Gimnasta podríamos declarar una constate que indicase la edad mínima de jubilación

Seleccione una:
- Verdadero
- Falso

13. Verdadero. En las interfaces se pueden declarar constantes

Pregunta **14**
Sin responder aún
Puntúa como 0,15
Marcar pregunta

La clase Troupe tiene un atributo de tipo colección de artistas

Seleccione una:
- Verdadero
- Falso

14.Verdadero

Pregunta **15**
Sin responder aún
Puntúa como 0,20
Marcar pregunta

Dado el código:

```
ArrayList<Trapecista> lista = new ArrayList();
ArrayList<Trapecista> otralista = new ArrayList();
for(Trapecista t: lista)
 otralista.add(t);
```

¿Se está haciendo una copia profunda?

Seleccione una:
- Verdadero
- Falso

15.Falso. Para que fuera una copia profunda se tendría que crear un objeto nuevo con las mismas características que el que se quiere copiar y añadirlo al Array