

Teorico - Preguntas

- ¿Qué es una clase? ¿Es una clase igual a un objeto? Justifique.
R: Una clase es un conjunto de variables y de métodos. Se utilizan para instanciar objetos con determinados valores. Es por esto ultimo que los objetos y las clases no son iguales.
- ¿Existe diferencia entre una clase abstracta y una interfaz? Justifique.
R: Si. Una de las principales diferencias radica en sus métodos, ya que en la clase abstracta los métodos pueden contener operaciones, pero en una interfaz solo se colocan las firmas. Además, la interfaz tiene sus variables con valores ya declarados e inalterables (constantes), mientras las variables en la clase abstracta, si bien puede recibir o no un valor por defecto, sus valores no son estrictamente constantes.
- ¿Existe la herencia múltiple? Dar un ejemplo
R: Si. Aunque no todos los lenguajes lo soportan. Un ejemplo de un lenguaje que soporta la herencia múltiple es Python.
- ¿Que diferencia existe entre las variables const, let y var en Javascript?
R: let se utiliza para declarar variables que pueden ser alteradas pero que solo existirán dentro del bloque de código donde fue declarada. Las variables var, luego de ser declaradas, se pueden modificar desde fuera del bloque donde fueron creadas. Como variable, const se declara con un valor y este no puede ser alterado.
- ¿Existe alguna diferencia entre DOM y Html? Justifique.
R: Si. El DOM es la estructura del HTML y es la interfaz utilizada para modificar dicho HTML
- ¿Cuál es la diferencia entre asincronía y paralelismo? ¿Son conceptos excluyentes?
R: Que, en la asincronía, en algún punto, se espera alguna respuesta de una llamada, pero esto no detiene la ejecución de la aplicación. No precisamente son conceptos excluyentes ya que la asincronía puede mostrar el resultado de una tarea paralela como resultado de la mencionada llamada.
- El paralelismo y la concurrencia. ¿Son lo mismo?
R: No. La concurrencia es cuando hay mas de un proceso ejecutándose al mismo tiempo, pero no precisamente iniciaron a la vez. En el paralelismo los procesos iniciaron al mismo tiempo y se pueden ejecutar a la vez. Por lo que, si hay paralelismo, hay concurrencia, pero esto no es reciproco.

- ¿Qué es `__init__` en Python?

R: Es un método que se utiliza como “constructor” en una clase.

- ¿Puede explicar `*args` y `**kwargs` en Python?

R: En las funciones, `*args` es utilizado para referirse a los parámetros que va a recibir y `**kwargs` se usa para hacer referencia al diccionario recibe como parámetro.

- ¿Que es un decorador en Python?

R: Son funciones que agregan otras funcionalidades a otras funciones.

- ¿Cuáles de las siguientes frases crean un diccionario en el lenguaje Python? (Múltiples respuestas correctas posible)

- a) `d = {}`
- b) `d = {"juan":40, "pedro":45}`
- c) `d = {40: "juan", 45: "pedro"}`
- d) `d = (40: "juan", 45: "50")`

R: a, b, c

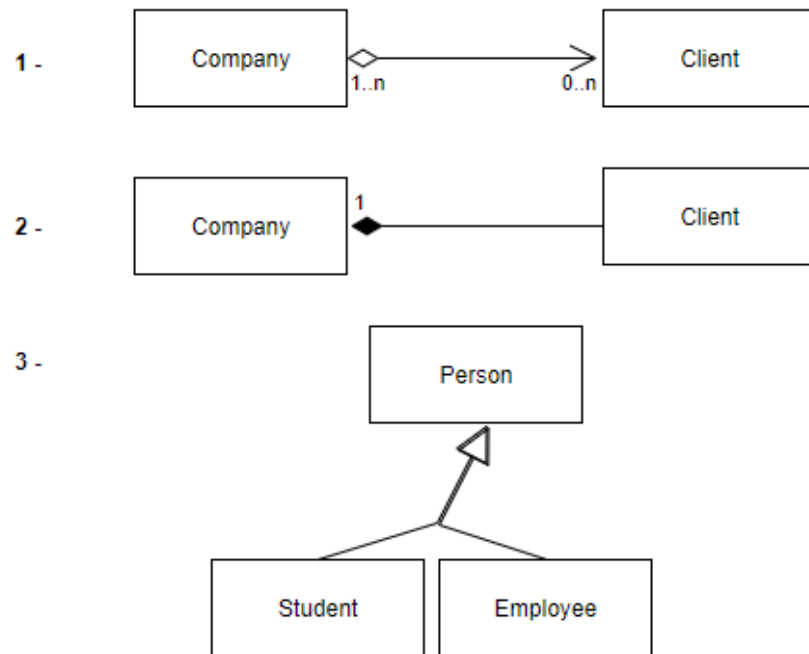
- ¿Qué imprime este código?

```
try:
    if '1' != 1:
        raise "algún error"
    else:
        print("no se ha producido algún error")
except "algún error":
    print ("se ha producido algún error")
```

- a) se ha producido algún error
- b) no se ha producido algún error
- c) código inválido
- d) ninguno de los anteriores

R: d (tomando en cuenta que la opción c devuelve exactamente “código inválido”)

- Dado los siguientes diagramas UML. Indique cual es correspondiente a composición, cual a herencia y cual define una agregación.



R: 1-Agregación, 2-Composición, 3-Herencia