Departamento de Programación
Facultad de Informática
Universidad Nacional del Comahue

Introducción a la Programación Orientada a Objetos-

Trabajo Práctico 1

1. Definir los siguientes términos: objeto, clase , método, atributos.
2. Que diferencia existe entre: *variable instancia* e *instancia de una clase.* El significado es el mismo? Justificar.
3. Clasificar las siguientes palabras como posibles candidatas a representar Clases **(C)**, atributos **(A)** o métodos (M):

mouse (C)	reloj	persona	inalambrico
televisor	mover	darHoraActual	ponerHora
darCanalActual	fechaNacimiento	irCanalTv	documento
configurarDespertador	encender	fechaNacimiento	darPosicionActual

4. Enumere 5 objetos con los que interactúa durante su rutina diaria. Enumere sus atributos y las acciones que realiza en la interacción con cada objeto.
5. Si tuviera que implementar la clase **Persona**, con qué atributos y métodos debería contar la clase. Diseñar la clase Persona con sus atributos y métodos.
6. Cuáles son los métodos que debemos encontrar implementados en una clase ?
7. Cuál es la razón de implementar el método ***_ _toString()*** en una clase?
8. Cómo se denomina y con qué nombre debe implementarse en PHP, al método que se ejecuta cuando se crea una instancia de una clase?
9. Cuando utilizamos ***$this*** como parámetro de un método, a que se esta referenciando ?
10. Para cada una de las siguientes clases implementar los métodos de acceso de cada una de las variables instancias , el método _ _toString() (que permite visualizar los valores que poseen las variables instancia) y por último, implementar la clase ***TestNombreClase*** para probar cada uno de los métodos implementados en cada clase.
    a) Diseñar e implementar la clase **Calculadora** que permite realizar las operaciones básicas: + , - , *, /
    b) Diseñar e implementar la clase **Reloj** que simule el comportamiento de un cronómetro digital (con las características puesta_a_cero, incremento, etc.). Cuando el contador llegue a 23:59:59 y reciba el mensaje de incremento deberá pasar a 00:00:00.
    c) Diseñar e implementar la clase **Fecha**. La clase deberá disponer de los atributos mínimos y necesarios para almacenar el día, el mes y el año, además de métodos que devuelvan un String con la fecha en forma abreviada (16/02/2000) y extendida (16 de febrero de 2000) . Implementar una función incremento, que reciba como parámetros un entero y una fecha, que retorne una nueva fecha, resultado de incrementar la fecha recibida por parámetro en el numero de días definido por el parámetro entero.
    Nota 1: Son años bisiestos los múltiplos de cuatro que no lo son de cien, salvo que lo sean de cuatrocientos, en cuyo caso si son bisiestos.
    Nota 2: Para la solución de este problema puede ser útil definir un método incrementa_un_dia.
    d) Implementar una clase **Login** que almacene el nombreUsuario, contraseña, frase que permite recordar la contraseña ingresada y las ultimas 4 contraseñas utilizadas. Implementar un método que permita validar una contraseña con la almacenada y un método para cambiar la contraseña actual por otra nueva, el sistema deja cambiar la contraseña siempre y cuando esta no haya sido usada recientemente (es decir no se encuentra dentro de las cuatro almacenadas). Implementar el método recordar que dado el usuario, muestra la frase que permite recordar su contraseña.
11. Crear una clase ***Cuadrado*** que modele cuadrados por medio de cuatro puntos (los vértices). Puede utilizar arreglos asociativos para implementar cada uno de los vértices. La clase dispondrá de los siguientes métodos:
    a) Método constructor ***_ _construct ()*** que recibe como parámetros las coordenadas de los puntos.
    b) Los métodos de acceso de cada uno de los atributos de la clase.

c) ***area()***: método que calcula el área del cuadrado.

d) ***desplazar*($d)**: método que recibe por parámetro un punto y desplaza el cuadrado en el plano desde su punto mas inferior izquierdo.

e) ***aumentarTamaño*($t)**: método que recibe por parámetro el tamaño que debe aumentar el cuadrado.

f) Redefinir el método ***_ _toString()*** para que retorne la información de los atributos de la clase.

g) Crear un script ***Test_Cuadrado*** que cree un objeto **Cuadrado** e invoque a cada uno de los métodos implementados.

12. Definir una clase ***Linea*** con cuatro atributos: pA , p B, pC y pD donde (pA , p B) y (pC,pD) son 2 puntos por los que pasa la línea en un espacio de dos dimensiones. La clase dispondrá de los siguientes métodos:
    a) Método constructor ***_ _construct()*** que recibe como parámetros las coordenadas de los puntos.
    b) Los métodos de acceso de cada uno de los atributos de la clase.
    c) ***mueveDerecha*($d)**: Desplaza la línea a la derecha la distancia(d) que se recibe por parámetro.
    d) ***mueveIzquierda*($d)**: Desplaza la línea a la izquierda la distancia(d) que se recibe por parámetro.
    e) ***mueveArriba*($d)**: Desplaza la línea hacia arriba la distancia que se recibe por parámetro.
    f) ***mueveAbajo*($d)**: Desplaza la línea hacia abajo la distancia que se recibe por parámetro.
    g) Redefinir el método ***_ _toString()*** para que retorne la información de los atributos de la clase.
    h) Crear un script ***Test_Linea*** que cree un objeto **Linea** e invoque a cada uno de los métodos implementados.
13. Desarrollar una clase ***Cafetera*** con los atributos *capacidadMaxima* (la cantidad máxima de café que puede contener la cafetera) y *cantidadActual* (la cantidad actual de café que hay en la cafetera). Implementar los siguientes métodos:
    a) Método constructor ***_ _construct()*** que recibe como parámetros los valores iniciales para los atributos de la clase.
    b) Los método de acceso de cada uno de los atributos de la clase.
    c) ***llenarCafetera*()**: la cantidad actual debe ser igual a la capacidad de la cafetera.
    d) ***servirTaza($cantidad)***: simula la acción de servir una taza con la capacidad indicada. Si la cantidad actual de café "no alcanza" para llenar la taza, se sirve lo que quede y se envía un mensaje al consumidor para que sepa porque no se le sirvió un taza completa.
    e) ***vaciarCafetera*()**: pone la cantidad de café actual en cero.
    f) ***agregarCafe*($cantidad)**: añade a la cafetera la cantidad de café indicada.
    g) Redefinir el método ***_ _toString()*** para que retorne la información de los atributos de la clase.
    h) Crear un script ***Test_Cafetera*** que cree un objeto **Cafetera** e invoque a cada uno de los métodos implementados.
14. Crea una clase **CuentaBancaria** con los siguientes atributos: *número de cuenta, el DNI del cliente, el saldo actual y el interés anual* que se aplica a la cuenta. Define en la clase los siguientes métodos:
    a) Método constructor ***_ _construct()*** que recibe como parámetros los valores iniciales para los atributos de la clase.
    b) Los método de acceso de cada uno de los atributos de la clase.
    c) ***actualizarSaldo*()**: actualizará el saldo de la cuenta aplicándole el interés diario (interés anual dividido entre 365 aplicado al saldo actual).
    d) ***depositar*($cant)**: permitirá ingresar una cantidad de dinero en la cuenta.
    e) ***retirar*($cant)**: permitirá sacar una cantidad de dinero de la cuenta (si hay saldo).
    f) Redefinir el método ***_ _toString()*** para que retorne la información de los atributos de la clase.
    g) Crear un script ***Test_CuentaBancaria*** que cree un objeto **CuentaBancaria** e invoque a cada uno de los métodos implementados.
15. Un teatro se caracteriza por su nombre y su dirección y en él se realizan 4 funciones al día. Cada función tiene un nombre y un precio. Realice el diseño de la clase **Teatro** e indique qué métodos tendría que tener la clase, teniendo en cuenta que se pueda cambiar el nombre del teatro y la dirección, el nombre de un función y el precio. Implementar las 4 funciones usando array de array asociativo. Cada función es un array asociativo con las claves "nombre" y "precio".

16. Cree una clase **Libro** que tenga los atributos *ISBN, titulo, año de edición, editorial, nombre y apellido del autor*. Defina en la clase los siguientes métodos:
    a) Método constructor ***_ _construct()*** que recibe como parámetros los valores iniciales para los atributos de la clase.
    b) Los método de acceso de cada uno de los atributos de la clase.
    c) Método***__toString()*** que retorne la información de los atributos de la clase.
    d) ***perteneceEditorial($peditorial)****:* indica si el libro pertenece a una editorial dada. Recibe como parámetro una editorial y devuelve un valor verdadero/falso.
    e) ***iguales($plibro,$parreglo)****:* dada una colección de libros, indica si el libro pasado por parámetro ya se encuentra en dicha colección.
    f) ***aniosdesdeEdicion()****:* método que retorna los años que han pasado desde que el libro fue editado.
    g) ***librodeEditoriales($arregloautor, $peditorial)****:* método que retorna un arreglo asociativo con todos los libros publicados por la editorial dada.
    h) Cree un script ***TestLibro*** que cree al menos tres libros e invoque a cada uno de los métodos implementados.