



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

1. Desarrolle los criterios de encapsulamiento y acceso disponibles ¿Por qué se hace esto?
2. En base a lo desarrollado en el punto anterior, corrija la siguiente clase para que cumpla con los criterios del Encapsulamiento y acceso necesarios dada la situación de uso posterior.

Python:

```
class Cancion:
    nombre = ''
    autor = ''
    duracion = 0.0
```

```
import Cancion

cancion = Cancion.Cancion()
cancion.nombre = "Un velero llamado libertad"
cancion.autor = "Jose Luis Perales"
cancion.duracion = 222.0

print(cancion.nombre)
print(cancion.autor)
print(cancion.duracion)
```

Java:

```
public class Cancion {
    public String nombre = "";
    public String autor = "";
    public Double duracion = 0.0;
}
```



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
public class App {  
    public static void main(String[] args) throws Exception {  
        Cancion cancion = new Cancion();  
        cancion.nombre = "Un velero llamado libertad";  
        cancion.autor = "Jose Luis Perales";  
        cancion.duracion = 222.0;  
  
        System.out.println(cancion.nombre);  
        System.out.println(cancion.autor);  
        System.out.println(cancion.duracion);  
    }  
}
```

3. Corrija el siguiente programa para que funcione realizando la **mínima cantidad de cambios y solo en la clase Alumno**.

Python:

```
class Alumno:  
    __nombre = ''  
    __apellido = ''  
    dni = 0  
  
    def __init__(self):  
        pass  
  
    @classmethod  
    def iniciar_con_nom_ape(cls, nombre, apellido):  
        alumno = cls.__new__(cls)  
        alumno.__nombre = nombre  
        alumno.__apellido = apellido  
        return alumno
```

```
from Alumno import Alumno  
  
alumno1 = Alumno()  
alumno1 = alumno1.iniciar()  
alumno1.setNombre("Alejandro")  
alumno1.setApellido("Rojas")  
alumno1.setDni(11111111)  
  
alumno2 = Alumno().iniciar_con_nom_ape("Martin", "Rosales")  
alumno2.setDni(11111112)  
  
print(f'{alumno1.getNombreYapellido()}{ " - DNI: "}{alumno1.getdni()}' )  
print(f'{alumno2.getNombreYapellido()}{ " - DNI: "}{alumno2.getdni()}' )
```

Java:



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
public class Alumno {  
  
    private String nombre = "";  
    private String apellido = "";  
    public Integer dni = 0;  
  
    public Alumno(){  
  
    }  
  
    public Alumno(String nombre, String apellido){  
        this.nombre = nombre;  
        this.apellido = apellido;  
    }  
}
```

```
public class App {  
    public static void main(String[] args) throws Exception {  
        Alumno alumno1 = new Alumno();  
        alumno1.setNombre("Alejandro");  
        alumno1.setApellido("Rojas");  
        alumno1.setDni(11111111);  
  
        Alumno alumno2 = new Alumno("Martin", "Rosales");  
        alumno2.setDni(11111112);  
  
        System.out.println(alumno1.getNombreYapellido()+" - DNI: "+  
            alumno1.getDni());  
        System.out.println(alumno2.getNombreYapellido()+" - DNI: "+  
            alumno2.getDni());  
    }  
}
```

4. Dado el siguiente código, copie y corrija todo aquello necesario para que el programa funcione de forma adecuada.

La salida esperada en la ejecución es:

Profesor: Hernandez Pedro
Materias:
POO
Algebra
Profesor: Alvarez Romina
Materias:
Introducción a la computación
Algoritmica
Profesor: Perez Laura
Materias:

Código Python:



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
class Materia:
    __nombre = ''
    __codigo = ''

    def __init__(self, nombre, codigo):
        self.__nombre = nombre
        self.__codigo = codigo

    def getNombre(self):
        return True

    def set_nombre(self, nombre):
        self.__nombre = nombre

    def set_codigo(self, codigo):
        self.__codigo = codigo
```

```
class Profesor:
    __nombre = ''
    __apellido = ''
    __materia = []

    def __init__(self, nombre, apellido):
        self.__nombre = nombre
        self.__apellido = apellido

    def set_nombre(self, nombre):
        self.__nombre = nombre

    def get_nombre(self):
        return self.__nombre

    def set_apellido(self, apellido):
        self.__apellido = apellido

    def get_apellido(self):
        return self.__apellido

    def set_materia(self, materia):
        self.__materia = materia

    def get_materia(self):
        return self.__materia

    def add_materia(self, materia):
        self.__materia.add(materia)
```



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
from Materia import Materia
from Profesor import Profesor

poo = Materia("P00", "IF153")
algebra = Materia("Algebra", 183)
introduccion = Materia("Introduccion a la computacion")
introduccion.set_codigo("IF300")
algoritmica = Materia("Algoritmica", "500")

profesores = []

profesor1 = Profesor("Pedro", "Hernandez")
profesor1.add_materia(poo)
profesor1.add_materia(algebra)
profesores.add(profesor1)
profesor2 = Profesor("Romina", "Alvarez")
profesor2.add_materia(introduccion)
profesor2.add_materia(algoritmica)
profesores.add(profesor2)
profesor3 = Profesor("Laura", "Perez")
profesores.add(profesor3)

for pro in profesores:
    print(f'{"Profesor:"}{pro.get_nombre()}{" "}{pro.get_apellido}')
    print("Materias:")
    for mat in pro.get_materia():
        print(mat.get_nombre())
```

Codigo Java:

```
public class Materia {

    private String nombre;
    private String codigo;

    public Materia(String nombre, String codigo){
        this.nombre = nombre;
        this.codigo = codigo;
    }

    public boolean getNombre(){
        return true;
    }

    public void setNombre(String nombre){
        this.nombre = nombre;
    }

    public void setCodigo(String codigo){
        this.codigo = codigo;
    }

}
```



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
import java.util.ArrayList;

public class Profesor {
    private String nombre = "";
    private String apellido = "";
    private ArrayList<Materia> materias = new ArrayList<>();

    public Profesor(String nombre, String apellido){
        this.nombre = nombre;
        this.apellido = apellido;
    }

    public void setNombre(String nombre){
        this.nombre = nombre;
    }

    public String getNombre(){
        return this.nombre;
    }

    public void setApellido(String apellido){
        this.apellido = apellido;
    }

    public String getApellido(){
        return this.apellido;
    }

    public void setMateria(Materia materia){
        this.materias.add(materia);
    }

    public ArrayList<Materia> getMateria(){
        return this.materias;
    }

    public void addMateria(Materia materia){
        this.materias.add(materia);
    }
}
```



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

```
import java.util.ArrayList;

public class App {
    public static void main(String[] args) throws Exception {
        Materia poo = new Materia("POO", "IF153");
        Materia algebra = new Materia("Algebra", 183);
        Materia introduccion = new Materia(
            "Introduccion a la computacion");
        introduccion.setCodigo("IF300");
        Materia algoritmica = new Materia("Algoritmica", "500");

        ArrayList<Profesor> profesores = new ArrayList<>();

        Profesor profesor1 = new Profesor("Pedro", "Hernandez");
        profesor1.addMateria(poo);
        profesor1.addMateria(algebra);
        profesores.add(profesor1);
        Profesor profesor2 = new Profesor("Romina", "Alvarez");
        profesor2.addMateria(introduccion);
        profesor2.addMateria(algoritmica);
        profesores.add(profesor2);
        Profesor profesor3 = new Profesor("Laura", "Perez");
        profesores.add(profesor3);

        for(Profesor pro: profesores){
            System.out.println(pro.getNombre()+" "+pro.
                getApellido());
            System.out.println("Materias: ");
            for(Materia mat:pro.getMateria()){
                System.out.println(mat.getNombre());
            }
        }
    }
}
```

5. Defina la clase Persona con los atributos

- Nombre de tipo string
- Apellido de tipo string
- Fecha de nacimiento de tipo Date

Genere además un constructor que reciba tres parámetros (uno por atributo), los métodos de acceso (getters y setters) de cada atributo y un método toString que retorne una cadena con el valor de todos los atributos.

- Instancie tres veces la clase persona y por cada instancia imprima en consola el valor devuelto por el método toString

6. Desarrolle el diagrama de clases del ejercicio anterior.

7. A la clase Persona del ejercicio 6 añada un nuevo método que retorne la edad haciendo uso de la fecha de nacimiento. Imprima en consola la edad de tres instancias de la clase

8. Como parte de un pequeño censo, para relevar la cantidad de personas de una comuna se solicita un sistema que permita cargar familias, donde cada una está compuesta por



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

varias personas. De estas personas es importante conocer su edad, sexo, si estudia y si trabaja.

Desarrolle y diseñe las clases que compondrían el sistema y su respectivo diagrama de clases.

9. Utilizando lo implementado en el ejercicio anterior, desarrolle una simulación que cense entre tres y cinco familias y finalmente imprima en pantalla los siguientes datos
 - a. Cantidad de familias censadas
 - b. Cantidad de personas
 - c. Promedio de edad
 - d. Cantidad de personas que trabajan
10. Usando lo desarrollado en el ejercicio 9, implemente la lógica necesaria para que sea posible consultar a las personas, en base a su edad:
 - a. sí tienen permitido trabajar
 - b. Si tienen permitido manejar un vehículo particular

En Argentina, si bien necesitan permiso, las personas están habilitadas para trabajar a partir de los 16 años. Respecto a la licencia B para vehículos particulares, es posible obtenerla a partir de los 17 años.

11. En un nuevo proyecto, copie la clase Persona resultante del ejercicio 8.
Defina una nueva clase Empresa que posea los atributos
 - Nombre de tipo string
 - Dirección de tipo string
 - Empleados, que sea una lista de la clase Persona.

Defina un constructor que reciba como parámetros el nombre y la dirección, además de los métodos de acceso (getters & setters) para cada atributo.

Luego cree un método para añadir un nuevo empleado a la empresa y otro para obtener el número total de empleados.

Para probar la clase instancie una vez y añada cinco empleados. Finalmente, imprima en pantalla la cantidad total de empleados, itere sobre el arreglo y para cada uno de ellos imprima sus datos.



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°2

12. Cree un nuevo proyecto que incluya las clases Persona y Empresa del ejercicio 11. Defina una nueva clase Puesto con un solo atributo nombre, su constructor y los métodos de acceso. Sume a la clase Persona un atributo Puesto que se recibe en el constructor y es parte de la respuesta del método toString.

Finalmente, para probar los nuevos cambios modifique la prueba existente de la siguiente forma: de los cinco empleados de prueba que se crean, los tres primeros deben pertenecer al puesto “Administrativo”, el cuarto al puesto “Gerente” y el quinto al puesto “Tesorero”.