



Argentina Programa 4.0

Universidad Nacional de San Luis

DESARROLLADOR PYTHON

Práctico Nro. 6.2

*Lenguaje de Programación Python: Soporte Orientado
a Objetos*

Autor:

Dr. Mario Marcelo Berón

Argentina Programa 4.0

Universidad Nacional de San Luis

Práctico Nro. 6.2: *Propiedades*

Ejercicio 1: Defina la clase Persona la cual permite almacenar el nombre y la edad de una persona. Resuelva este ejercicio utilizando propiedades.

Ejercicio 2: Defina la clase Botón que permite almacenar el estado (Presionado, Libre) de un botón. Resuelva este ejercicio utilizando propiedades.

Ejercicio 3: Defina la clase semáforo la cual tiene tres variables: rojo, verde, amarillo. La clase simula el comportamiento de un semáforo de la vida real. Para resolver este ejercicio utilice propiedades.

Ejercicio 4: Defina la clase SemaforoConBotón la cual tiene un botón y un semáforo. El semáforo funciona de la siguiente manera: Se presiona el botón se enciende la luz roja, se presiona nuevamente se apaga la luz roja, cuando se presiona de nuevo se prende la luz verde, si se presiona nuevamente se apaga la luz verde, se presiona nuevamente se prende la luz amarilla y luego de otra presión se apaga la luz amarilla y comienza el ciclo nuevamente.

Ejercicio 5: Defina la clase Password la cual tiene un atributo de solo escritura privado. Cuando un usuario desea leer el atributo se devuelve un error de lectura.

Ejercicio 6: Defina la clase MatrizCuadrada la cual permite almacenar números flotantes. La clase permite realizar las siguientes operaciones:

1. Cargar una matriz.
2. Asignar a una variable una matriz.
3. Sumar dos matrices.
4. Restar dos matrices.

5. Construir una representación textual de una matriz.

Nota: Para resolver este ejercicio sobrecargue los operadores que crea conveniente.

Ejercicio 7: Dado el siguiente programa:

```
class Powers:
    def __init__(self, square, cube):
        self._square = square    # _square is the base value
        self._cube = cube       # square is the property name

    def getSquare(self):
        return self._square ** 2

    def setSquare(self, value):
        self._square = value
        square = property(getSquare, setSquare)

    def getCube(self):
        return self._cube ** 3
        cube = property(getCube)

X = Powers(3, 4)
print(X.square)      # 3 ** 2 = 9
print(X.cube)        # 4 ** 3 = 64
X.square = 5
print(X.square)      # 5 ** 2 = 25
```

Se pide:

1. Diga si el programa usa propiedades.
2. Qué hace el programa?

