

Estructuras de Datos y Algoritmos II UPE 2016

Trabajo práctico n°1

1. Escriba tres funciones dentro de un archivo llamado practica1.py que reciban por parámetro dos números enteros a y b retornando todos los números enteros comprendidos entre a y b (inclusive) de menor a mayor. Las funciones son:

- funcionFor(a,b):** realiza lo pedido con un for.
- funcionWhile(a,b):** realiza lo pedido con un while.
- funcionNoIter(a,b):** realiza lo pedido sin utilizar estructuras de control

iterativas (for, while).

Tener en cuenta que no necesariamente $a < b$.

Por último, corra los test provistos por la cátedra y verifique que los resultados sean correctos.

2. Escriba una función llamada multiplos dentro del archivo creado en el punto 1 que dado un número n devuelva una nueva lista de tamaño n con los n primeros múltiplos enteros de n mayores o iguales que 1.

Ejemplo: **multiplos**(5) = [5, 10, 15, 20, 25]

Luego corra el test provisto por la cátedra y verifique que los resultados sean correctos.

3.¿Qué imprime el siguiente programa ?

- Intente averiguarlo sin ejecutar el programa en su computadora.
- Ejecute el ejercicio en su computadora, y compare su resultado con lo esperado en el inciso anterior.

```
def swap1(x,y):
```

```
    if x < y:
        tmp = x
        x = y
        y = tmp
```

```
def swap2(lista):
```

```
    if lista[0]< lista[1]:
        newList = [0,0]
        lista = newList
```

```
a = 1
```

```
b = 2
```

```
l = [a,b]
```

```
swap1(a,b)
```

```
swap2(l)
```

```
print a
```

print b
print l

4. Dado una lista de valores tipo entero se desea calcular el valor máximo, mínimo, y promedio en una única función. Escriba dos funciones llamadas **conReturn(lista)** y **conParametro(lista,resultado)**, dentro del archivo creado en 1 ,donde respectivamente:

a. Devuelva lo pedido por el mecanismo de retorno ("return") a través de una lista.

b. Devuelva lo pedido interactuando con algún parámetro a través de una lista. Corra los tests provistos por la cátedra para asegurar corrección en su solución.

5. Realice a través de funciones el comportamiento de una pila, las operaciones necesarias son:

pila_sacar(lista) : saca un elemento al tope de la pila y lo retorna

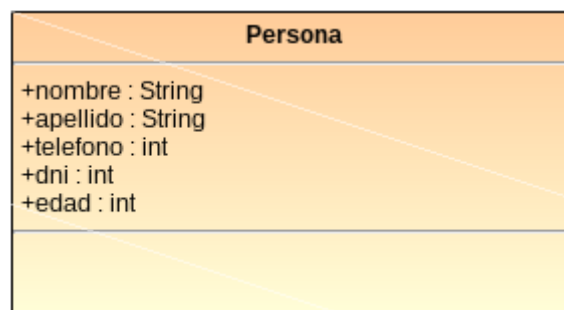
pila_poner(lista,elemento): pone un elemento en la pila

pila_esVacia(lista): devuelve true si es vacía, false caso contrario

pila_tope(lista): devuelve el elemento que está al tope de la pila

6. Realice las mismas operaciones pero para el comportamiento de una cola.

7. Realice la clase Persona de acuerdo al siguiente diagrama.

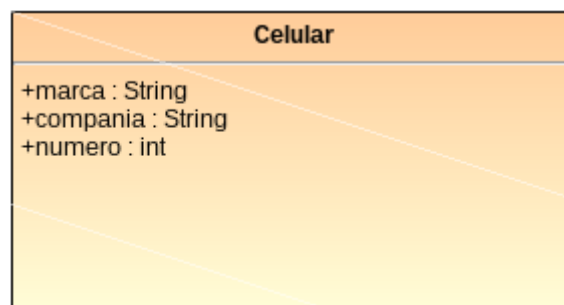


8. Realice un script que construya una lista de personas e imprima sus datos en pantalla.

9. Realice un script que con los mismos datos del punto anterior construya una lista de personas y las ordene según su dni.

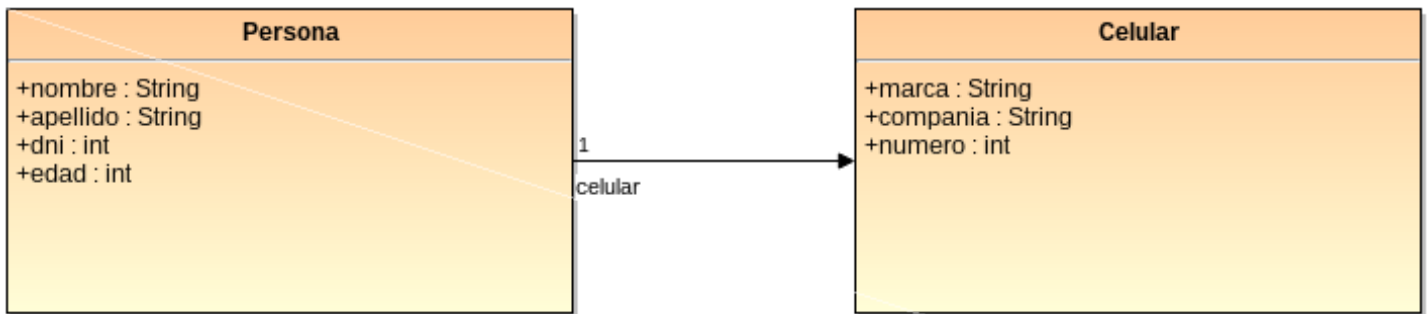
Nota: tenga en cuenta el método **__cmp__** y la funcion **cmp()**

10. Realice la clase Celular de acuerdo al siguiente diagrama



haga lo mismo que en punto 8 pero con Celulares, y el punto 9 pero que los ordene por número.

11. Modifique la clase Persona, para que ahora tenga un atributo celular, luego construya una lista de Personas con objetos celulares y relacionelos.



Tiempo de Ejecución:

12. Calcule el tiempo de ejecución de las siguientes funciones:

a) **for** i **in** range(1,n+1):
 print i

b) j = 1
 while (j <=n):
 print j
 j += 1

c) **for** i **in** range(1, n+1):
 for j **in** range(1, n+1):
 print j+i

d) j , i = 1 , 1
 while (j <= n):
 while(i <= n):
 print j+i
 i +=1
 j+=1

e) **for** i **in** range(1, n+1, 2):
 print i

f) i = 1
 while (i <= n/2):
 print i