

ESTRUCTURAS DE DATOS

Trabajo Práctico N° 3 – Repaso Listas y TDA

Ejercicio N° 1: Dada la siguiente lista [1, 14, 56, 43, 23, 46, 58, 123, 67] escribir una función que muestre el número más alto, escriba el programa que invoque a la función.

Ejercicio 2: Realizar un programa que pida al usuario ingresar el nombre de un empleado, los años de antigüedad, el sueldo y el área de trabajo Por ejemplo: [“Juan Pérez”, 14, 890000, “Contabilidad”] El programa deberá mostrar el nuevo sueldo del empleado incrementado en un 15%.

Ejercicio N° 3: Definir una función que reciba como parámetro una lista de números (se sabe que la lista tiene 3 números, la función debe retornar la suma del 1er número y el último número de la Lista. Por ejemplo: Lista= [4,3,8,9,2,11] La función deberá devolver 15.

Ejercicio 4: Definir la clase Punto. Sus atributos serán las coodenadas en x e y del plano cartesiano. Un Punto reconoce las siguientes operaciones (métodos):

distancia(p1, p2): distancia entre los puntos (devuleve un número)

resta(p1, p2): resultado de la resta de dos puntos (devuelve un punto)

norma(p): módulo del vector desde el origen hacia el punto p

Ejercicio 5: Definir un TDA llamada fracción. Debe entenderse como fracción: $\frac{3}{4}$. Como lo representaría? Pensar los métodos definen el comportamiento de la fracción.

Ejercicio 6: Definir la clase Rectangulo. Un rectangulo pude ser definido de varias maneras, por ejemplo:

4 puntos: las 4 esquinas (sirve para cualquier poligono).

2 puntos: Idem anterior, pero usamos dos.

1 punto de referencia, base y altura.

(Elejir solo una e implementarla)

La clase debe contener los siguientes métodos:

Calcular el perimetro del rectangulo.

Clacular el area del rectangulo.

Chequear si el rectangulo es un cuadrado.

Ejercicios Complementarios:

Resolver la práctica aplicando los siguientes conceptos cuando sea necesario:

- Modularización y definición adecuada de argumentos
- Estructuras de control
- Pasaje de parámetros
- Condicionales simples y compuestos

Ejercicio 1: Dado el siguiente programa (ver .py en el campus) completar las funciones que permitan resolver las opciones definidas en el Menu.

#Area de Funciones

```
def agregar(lista):
    nuevoproducto=input("Ingrese el nombre del nuevo producto: ")
    precio= float(input("Ingrese el precio del nuevo producto: "))
    stock= int(input("Ingrese la cantidad de productos (stock): "))
    prod=[nuevoproducto,precio,stock]
    lista.append(prod)
    return lista
```

#Programa Principal

```
print ("Bienvenidos a la tienda ALOISE Burzaco")
print ()
productos= [{"HELADERA", 230000, 12 },{"LUSTRADORA", 100000,4 }, {"HORNO",240000, 4 }, {"LAVADORA", 120000, 5 },
{"SECADORA", 130000, 8 }]
```

menu="""

```
(1) Mostrar la lista de productosy precios
(2) Actualice el stock de un producto
(3) Precio promedio de los productos
4. Ingrese un nuevo producto
5. Cantidad total de productos que hay en ALOISE FV
6. Salir
"""
```

```
print (menu)
opcion= int(input("ingrese una opcion del menu:"))
while opcion != 6:

    if(opcion==1):
        print("Eligio la opcion 1")
        print(productos)

    elif(opcion==2):
        print("Vamos a actualizar el stock de un producto")
        #productos=actualizarstock(productos)
        #print (productos)

    elif(opcion==3):
        print("Eligio la opcion 3")
```

```
elif(opcion==4):
    print("Vamos a ingresar un nuevo producto")
    productos= agregar(productos)
    print(productos)

elif(opcion==5):
    print("Eligio la opcion 5")

else:
    print ("No elegiste una opcion del menu.")

print(menu)
opcion= int(input("ingrese una opcion del menu:"))

print ("Gracias por usar el Sistema")
```

Ejercicio 2: Agregar una función al programa anterior llamada `carga_datos` , que no recibe ningún parámetro, la función deberá cargar los datos de la tienda ALOISE Burzaco" , la carga de datos termina cuando se ingresa como nombre del producto “zzz”.

Nota: los datos a cargar deberán ser el nombre del producto, el precio y el stock

```
productos= [ ["HELADERA", 230000, 12 ], ["LUSTRADORA", 100000,4 ], ["HORNO",240000, 4 ], ["LAVADORA", 120000, 5 ],
["SECADORA", 130000, 8 ] ]
```