

## ▼ Auxiliar 6

P1. Considere que tiene la estructura Animal que tiene como atributos Especie, Clase, Dieta y Edad(en meses). Se le entregará una lista de Animales y se le solicita que realice lo siguiente:

1. Cree la función esClase(A,clase) que verifique si el animal A es de la clase indicada
2. Cree un filtro que reciba una Lista de animales y una clase, y entregue una lista con los animales que son de la clase indicada
3. Cree Agnos(A), que reciba un Animal y convierta su edad a años (puede truncar) y además cree mapaAgnos(L) que obtenga una lista de animales y entregue la lista con las edades de los animales en años.
4. Cambie, para todos los animales de la lista, su dieta de por ejemplo Carnivoro a Come Carne.

Clases de Animales: Mamífero, Pez, Ave, Anfibio y Reptil

Dietas de Animales: Carnivora, Hervivora y Omnivora

```
import estructura as es
#Animal: especie(str) clase(str) dieta(str) edad(int)
es.crear('Animal', 'especie clase dieta edad')

from abstraccion import *
from lista import *

#esClase: Animal str -> bool
#Entrega true en el caso de que el animal entregado sea de la clase entregada
#ej: esClase(Animal('delfin','mamifero','carnicora',120), 'mamifero') entrega True
def esClase(A,clase):
    return A.clase == clase

#test
assert esClase(Animal('delfin','mamifero','carnivora',120), 'mamifero')

#filtroClase: función lista str -> lista
#Entrega una lista de los animales que son de la clase indicada
def filtro(L, x):
    assert type(L)==lista or vacia(L)
    if vacia(L):
        return None
    else:
        if esClase(cabeza(L),x):
            return crearLista(cabeza(L),filtroClase(cola(L),x))
        else:
            return filtroClase(cola(L),x)
```

```

#test
LAnimales = crearLista(Animal('delfin','mamifero','carnivora',120),\
                        crearLista(Animal('gato','mamifero','carnivora',36),\
                                    crearLista(Animal('serpiente','reptil','carnivora',15),\
                                                crearLista(Animal('salamandra','anfibio','carnivor

assert filtroClase(LAnimales,'mamifero') == crearLista(Animal('delfin','mamifero','carnivora'
                                                crearLista(Animal('gato','mamifero','c
assert filtroClase(LAnimales,'reptil') == crearLista(Animal('serpiente','reptil','carnivora',

#Agnos: Animal -> Animal
#Trunca la edad del animal que se le entrega
def Agnos(A):
    return Animal(A.especie,A.clase,A.dieta,A.edad//12)

#test
delfin = Animal('delfin','mamifero','carnivora',120)
assert Agnos(delfin) == Animal('delfin','mamifero','carnivora',10)

#mapaAgnos: lista -> lista
def mapaAgnos(L):
    return mapa(Agnos,L)

#test
LAnimalesAgnos = crearLista(Animal('delfin','mamifero','carnivora',10),\
                             crearLista(Animal('gato','mamifero','carnivora',3),\
                                         crearLista(Animal('serpiente','reptil','carnivora',1),\
                                                     crearLista(Animal('salamandra','anfibio','carnivor

assert mapaAgnos(LAnimales) == LAnimalesAgnos

#cambioDieta: Animal -> Animal
#Cambia la el str de la dieta del animal
def cambioDieta(A):
    assert A.dieta == 'carnivora' or A.dieta == 'hervivora' or A.dieta == 'omnivora'
    if A.dieta == 'carnivora':
        nueva = 'come carne'
    elif A.dieta == 'hervivora':
        nueva = 'come plantas'
    elif A.dieta == 'omnivora':
        nueva = 'come todo'
    return Animal(A.especie, A.clase, nueva, A.edad)

#test
assert cambioDieta(delfin)==Animal('delfin','mamifero','come carne',120)

#cambioDietaLista: lista(animales) -> lista(animales)
def cambioDietaLista(L):
    assert vacia(L) or type(L)==lista
    return mapa(cambioDieta,L)

```

```
#test
LAnimalesDieta = crearLista(Animal('delfin','mamifero','come carne',120),\
                             crearLista(Animal('gato','mamifero','come carne',36),\
                                           crearLista(Animal('serpiente','reptil','come carne',15),\
                                                         crearLista(Animal('salamandra','anfibio','come car
assert cambioDietaLista(LAnimales)==LAnimalesDieta
```

P2. El Kino es un popular juego de lotería chileno que consiste en elegir 14 números del 1 al 25 (distintos) y recibir un premio en función de un sorteo aleatorio. A continuación crearemos un programa para jugar Kino.

1. Cree la estructura Kino, la cual consta de una lista de 14 enteros (del 1 al 25, sin repetición y ordenada ascendentemente), un string Fecha de forma "DD/MM/AAAA" (que representa la fecha en que el boleto fue comprado) y un string Comprador que representa de quien efectuó la compra.
2. Cree la función esKino(kino), que se asegure que los atributos de un Kino nombrados anteriormente sean de la forma correcta (retorna True si lo son, False si no).
3. Cree la función xEstaEnL(x,L) que verifique si el numero x esta en la lista L L
4. Cree la funcion numerosGanadores() que entregue un lista con 14 numeros distintos, los que corresponderán a los numeros ganadores del kino
5. Cree la funcion numerosIguales(L1,L2) que recibe dos listas y entrega la cantidad de numeros de la lista 1 que estan en la lista 2
6. Cree la función imprimir lista, que imprima todos los elementos de una lista
7. Cree la función sortearKino(L) que, recibiendo una lista de Kinos, simule un sorteo aleatorio del Kino e imprima lo siguiente: si un Kino coincide con entre 14 y 10 números, debe imprimir en pantalla 'Premio X' (si coincide con 14 números, la función muestra 'Premio 1', si coincide con 13 muestra 'Premio 2' y así hasta el 5to premio), junto con el nombre de el/los ganadores, y en cualquier otro caso la función muestra "No hay ganadores". Por ejemplo (caso varios ganadores):

```
listaConocida = lista(Kino1,...)
```

```
sortearKino(listaConocida)
```

```
Premio 1 Elmer Figueroa
```

```
Premio 3 Luis M. Gallego
```

```
import estructura as es
#Kino: numeros(lista) Fecha(str) Comprador(str)
es.crear('Kino','numeros Fecha Comprador')
```

```

from lista import *
# esNumerosDeKino : lista(int) -> bool
# True si lo ingresado es lista y corresponde a numeros de un Kino, False si no
# por ejemplo esNumerosDeKino("hola") retorna False
def esNumerosDeKino(L, cont1 = 0):
    assert type(L)==lista or vacia(L)
    if L == None:
        return cont1 == 14
    if type(cabeza(L)) != int or cabeza(L) < 1 or cabeza(L) > 25:
        return False
    return esNumerosDeKino(cola(L), cont1 + 1)

#test
L = crearLista(1,crearLista(2,crearLista(3,\
                                crearLista(4,crearLista(5,lista(6,crearLista(7,\
                                crearLista(8,crearLista(9,crearLista(10,crearLista(11,\
                                crearLista(12,crearLista(13,crearLista(14,None))))))))))
assert esNumerosDeKino(L)

# esKino : Kino -> bool
# True si lo ingresado es un Kino (y cumple con sus requerimientos)
def esKino(kino):
    if type(kino) != Kino:
        return False
    if type(kino.numeros) != lista:
        return False
    if type(kino.Fecha) != str or type(kino.Comprador) != str:
        return False
    if len(kino.Fecha) != 10:
        return False
    return esNumerosDeKino(kino.numeros)

#test
kino = Kino(L,'15/10/2020','Chayanne')
assert esKino(kino)

# xEstaEnL : int lista(int) -> bool
# True si x es parte de la lista L, False si no
# por ejemplo xEstaEnL(1, lista(2,lista(1,None))) retorna True
def xEstaEnL(x,L):
    if L == None:
        return False
    if cabeza(L) == x:
        return True
    return xEstaEnL(x,cola(L))

#test

```

```

assert xEstaEnL(1,lista(2,lista(1,None)))

# numerosGanadores : None -> lista(int)
# retorna lista de 14 numeros aleatorios sin repeticion
def numerosGanadores(listaParcial = None, cont1 = 0):
    if cont1 == 14:
        assert esNumerosDeKino(listaParcial)
        return listaParcial
    enteroAleatorio = random.randint(1,25)
    if xEstaEnL(enteroAleatorio,listaParcial):
        return numerosGanadores(listaParcial, cont1)
    else:
        return numerosGanadores(lista(enteroAleatorio,listaParcial),cont1 + 1)

# numerosIguales: lista(int) lista(int) -> int
# retorna cuantos elementos de L1 estan en L2
# por ejemplo numerosIguales(lista(1,lista(2,None)), lista(2,None)) retorna 1
def numerosIguales(L1, L2):
    if L1 == None:
        return 0
    if xEstaEnL(cabeza(L1),L2):
        return 1 + numerosIguales(cola(L1),L2)
    return numerosIguales(cola(L1),L2)

#test
L1 = crearLista(15,crearLista(20,crearLista(18,\
                                crearLista(21,crearLista(17,lista(24,crearLista(7,\
                                crearLista(8,crearLista(9,crearLista(23,crearLista(11,\
                                crearLista(12,crearLista(13,crearLista(14,None))))))))))
assert numerosIguales(L,L1) == 7

# imprimirLista : lista(str) -> None
# imprime en pantalla una lista de strings (cada elemento en una linea)
def imprimirLista(L):
    if L == None: return
    print(cabeza(L))
    return imprimirLista(cola(L))

# sortearKino : lista(Kino) -> None
# sortea un juego de Kino y muestra los ganadores de una lista de Kinos
def sortearKino(L, numerosSorteados = None, listaGanadores = None):
    if numerosSorteados == None:
        numerosSorteados = numerosGanadores()
    if vacia(L):
        if listaGanadores == None:
            print ("No hay ganadores")
            return
        else:
            imprimirLista(listaGanadores)

```

```

        return
    assert type(L) == lista
    assert type(cabeza(L)) == Kino
    n = numerosIguales(numerosSorteados, cabeza(L).numeros)
    if n == 14:
        listaGanadores = lista("Premio 1 " + cabeza(L).Comprador, listaGanadores)
        return sortearKino(cola(L),numerosSorteados, listaGanadores)

    elif n == 13:
        listaGanadores = lista("Premio 2 " + cabeza(L).Comprador, listaGanadores)
        return sortearKino(cola(L),numerosSorteados,listaGanadores)

    elif n == 12:
        listaGanadores = lista("Premio 3 " + cabeza(L).Comprador, listaGanadores)
        return sortearKino(cola(L),numerosSorteados,listaGanadores)

    elif n == 11:
        listaGanadores = lista("Premio 4 " + cabeza(L).Comprador, listaGanadores)
        return sortearKino(cola(L),numerosSorteados,listaGanadores)

    elif n == 10:
        listaGanadores = lista("Premio 5 " + cabeza(L).Comprador, listaGanadores)
        return sortearKino(cola(L),numerosSorteados, listaGanadores)

    return sortearKino(cola(L),numerosSorteados,listaGanadores)

# test 50 juegos:
import random
L = lista(1,lista(2,lista(3,lista(4,lista(5,lista(6,lista(7,lista(8,lista(9,lista(10,lista(11
Kino1 = Kino(L,"24/05/1997","Luismi")
listaKinos = lista(Kino1,None)

def repite(n=50):
    if n <= 0: return
    sortearKino(listaKinos)
    repite(n-1)

repite()

```

[illegible]

