Algoritmos T.P. N° 4 – Ejercitación Adicional de Listas Juan Cruz Ambrosini

import algo1 as Algo

import mylinkedlist as List

import myqueue as Queueu

import mystack as Stack

import MyArray as MyArray

import mypriorityqueue as PQueue

import random

"""----------------------------------------------------------------------------------"""

class EntryDate:

    def \_\_init\_\_(self, day : int = None, month : int = None, year : int = None):

        self.day = day

        self.month = month

        self.year = year

    day = None

    month = None

    year = None

class Employee:

    def \_\_init\_\_(self, name : str = None, age : int = None, id : int = None, entryDate : EntryDate = None):

        self.name = name

        self.age = age

        self.id = id

        self.entryDate = entryDate

    name = None

    age = None

    id = None

    entryDate = None

"""----------------------------------------------------------------------------------"""

arreglo\_nombres = Algo.Array(10,"")

MyArray.insert(arreglo\_nombres, "Juliana", 0)

MyArray.insert(arreglo\_nombres, "Gerbacio", 1)

MyArray.insert(arreglo\_nombres, "Federico", 2)

MyArray.insert(arreglo\_nombres, "Bartolomeo", 3)

MyArray.insert(arreglo\_nombres, "Esteban", 4)

MyArray.insert(arreglo\_nombres, "Fernando", 5)

MyArray.insert(arreglo\_nombres, "Pancracia", 6)

MyArray.insert(arreglo\_nombres, "Mahoma", 7)

MyArray.insert(arreglo\_nombres, "Jorge", 8)

MyArray.insert(arreglo\_nombres, "Domingo", 9)

generatedNumbers = List.LinkedList()

"""----------------------------------------------------------------------------------"""

def generateId ():

    id = random.randrange(0,1000)

    exists = False

    currentGenNum = generatedNumbers.head

    while currentGenNum != None:

        if id == currentGenNum.value :

            exists = True

        currentGenNum = currentGenNum.nextNode

        if exists :

            return generateId()

    List.add(generatedNumbers,id)

    return id

"""----------------------------------------------------------------------------------"""

def generateEntryDate(age):

    day = random.randint(1,31)

    month = random.randint(1,12)

    year = random.randint(2025 - age + 20, 2025)

    entryDate = EntryDate(day, month, year)

    return entryDate

"""----------------------------------------------------------------------------------"""

def createEmployeeList():

    employees = List.LinkedList()

    for i in range (0, 15):

        name = arreglo\_nombres[random.randint(0,9)]

        age = random.randint(20, 70)

        id = generateId()

        entryDate = generateEntryDate(age)

        newEmployee = Employee(name, age, id, entryDate)

        List.add(employees, newEmployee)

    return employees

"""----------------------------------------------------------------------------------"""

def delete\_lowest\_five(list : List.LinkedList):

    length = List.lengthList(list)

    if length < 5 :

        list.head = None

    else:

        for i in range(0,5):

            lowest\_id = list.head.value.id

            current = list.head

            while current != None:

                if lowest\_id > current.value.id :

                    lowest\_id = current.value.id

                current = current.nextNode

            current = list.head

            while current != None:

                if current.value.id == lowest\_id:

                    List.delete(list, current.value)

                current = current.nextNode

"""----------------------------------------------------------------------------------"""

def showEmployeeList(list):

    current = list.head

    while current != None:

        print("Nombre: ", current.value.name)

        print("Edad: ", current.value.age)

        print("Legajo: ", current.value.id)

        print("Fecha de ingreso: ", current.value.entryDate.day, "/",current.value.entryDate.month, "/", current.value.entryDate.year)

        print("---------------------------------------------------------------------------------------------------------------------")

        current = current.nextNode

"""----------------------------------------------------------------------------------"""

employees = createEmployeeList()

showEmployeeList(employees)

print("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||")

print("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||")

print("||||||||||||||||||||||||||||||||||||||||||LISTA DE EMPLEADOS ELIMINADOS||||||||||||||||||||||||||||||||||||||||||")

print("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||")

print("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||")

delete\_lowest\_five(employees)

showEmployeeList(employees)

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.