

Guía de Ejercicios Prácticos 25

Sitio: [Universidad Virtual UTN FRC](https://uv.frc.utn.edu.ar)
Curso: Algoritmos y Estructuras de Datos (2020)
Libro: Guía de Ejercicios Prácticos 25

Imprimido por: Luciana Lisette Montarce
Día: lunes, 23 de noviembre de 2020, 21:30



Tabla de contenidos

1. Salón de Fiestas

- 1.1. reserva.py
- 1.2. manejador_archivo.py
- 1.3. main.py
- 1.4. main para clase



1. Salón de Fiestas

Un salón dedicado a la organización de fiestas infantiles nos solicita un sistema para gestionar las reservas de un mes determinado. De cada reserva de festejo se conoce:

- Número de reserva (es un dato tipo String)
- Nombre del cumpleaños (es un dato tipo String)
- Edad del cumpleaños (0 a 13 años) (es un dato tipo Integer)
- Tipo de servicio solicitado (0: salón - 1: salón y animación - 2: salón, animación y comida niños - 3: salón, animación, comida niños y sorpresitas) (es un dato tipo Integer)
- Cantidad Invitados (es un dato tipo Integer)
- Monto (es un dato tipo Float)

Al iniciar el programa se cargan todas las reservas almacenadas en un archivo de texto llamado [reservas.csv](#) (el cual es provisto). Se debe cargar dicho archivo en un arreglo de registros de tipo Reserva. Una vez cargado, programe los siguientes puntos en un menú de opciones:

1. Mostrar el contenido del vector, incluyendo la descripción del tipo de servicio
2. Agregar una nueva reserva al arreglo de registro, con un número que se ingresa por teclado. Validar que dicho número de reserva sea único y solicitar el resto de los datos.
3. Determinar el monto total que el salón ha obtenido por tipo de servicio (vector de acumulación). Mostrar los resultados e informar qué servicio presenta el mayor monto.
4. Crear un nuevo vector con todas las reservas donde la edad del cumpleaños sea mayor a un valor x pasado por parámetro y la cantidad de asistentes mayor a un valor y también pasado por parámetro. Mostrar el contenido.
5. Salir del programa, pero antes de terminar, grabar nuevamente todas las reservas en el archivo **reservas.csv** (sobrescribir el archivo)

1.1. reserva.py

```
__author__ = 'Algoritmos y Estructuras de Datos'

class Reserva:
    def __init__(self, numero, nombre_cumple, edad, tipo_servicio, invitados, monto):
        self.numero = numero
        self.nombre_cumple = nombre_cumple
        self.edad = edad
        self.tipo_servicio = tipo_servicio
        self.invitados = invitados
        self.monto = monto

    def descripcion_servicio(tipo_servicio):
        descripcion_servicio = ('Salón', 'Salón y animación', 'Salón, animación y comida niños',
                                'Salón, animación, comida niños y sorpresitas')
        return descripcion_servicio[tipo_servicio]

    def to_string(reserva):
        cad = '{:<11} | {:<30} | {:^4} | {:<45} | {:^10} | ${:>10.2f}'
        return cad.format(reserva.numero, reserva.nombre_cumple, reserva.edad, descripcion_servicio(reserva.tipo_servicio), \
                           reserva.invitados, reserva.monto)

    def to_lineacsv(reserva):
        cad = '{},{},{},{},{},{}\n'
        return cad.format(reserva.numero, reserva.nombre_cumple, reserva.edad, reserva.tipo_servicio, \
                           reserva.invitados, reserva.monto)

    def str_toreserva(linea):
        token = linea.split(',')
        numero = token[0]
        nombre = token[1]
        edad = int(token[2])
        tipo_servicio = int(token[3])
        invitados = int(token[4])
        monto = float(token[5])
        reserva = Reserva(numero, nombre, edad, tipo_servicio, invitados, monto)
        return reserva
```

1.2. manejador_archivo.py

```
from reserva import *
import os

__author__ = 'Algoritmos y Estructuras de Datos'

def generar_arreglo():
    v = []
    if os.path.exists("reservas.csv"):
        m = open('reservas.csv', 'rt')
        for linea in m:
            reserva = str_toreserva(linea)
            v.append(reserva)
        m.close()
    return v

def grabar_arreglo(vector):
    m = open('reservas.csv', "wt")
    for res in vector:
        cad = to_lineacsv(res)
        m.write(cad)
    m.close()

if __name__ == '__main__':
    v = generar_arreglo()
    print(titulo())
    for r in v:
        print(to_string(r))
```

1.3. main.py



```
from manejador_archivo import generar_arreglo, grabar_arreglo
from reserva import *

__author__ = 'Algoritmos y Estructuras de Datos'

def menu():
    cad = 'Menu de Opciones\n' + \
        '=' * 80 + '\n' + \
        '1 ----- Mostrar Arreglo de Reservas\n' + \
        '2 ----- Agregar nuevo reserva\n' + \
        '3 ----- Determinar total por monto de servicio\n' + \
        '4 ----- Crear y mostrar nuevo vector de reservas\n' + \
        '0 ----- Salir\n' + \
        'Ingrese su opcion: '
    return int(input(cad))

def validar_rango(min, max, mensaje='Ingrese un numero: '):
    valor = min - 1
    while valor < min or valor > max:
        valor = int(input(mensaje))
        if valor < min or valor > max:
            print('Error!!!! El valor esta fuera del rango solicitado [' + str(min) + ', ' + str(max) + ']')
    return valor

def validar_mayor(min, mensaje='Ingrese un numero: '):
    valor = min - 1
    while valor < min:
        valor = int(input(mensaje))
        if valor < min:
            print('Error!!!! El valor ser mayor a ' + str(min))
    return valor

def display(vector):
    print('Listado de Reservas')
    print('=' * 126)
    titulo = '{:<11} | {:<30} | {:>4} | {:<45} | {:>10} | {:>11}\n{'
    print(titulo.format('Numero', "Nombre Cumpleañoero", 'Edad', 'T. Scio', 'Cant Inv.', 'Monto', '=' * 126))

    for elem in vector:
        print(to_string(elem))

def existe(vector, num_reserva):
    existe = False
    for elem in vector:
        if elem.numero == num_reserva:
            existe = True
            break
    return existe

def agregar_nueva_reserva(vector):
    numero = input('Ingresar el nuevo numero de la reserva: ')
    if not existe(vector, numero):
        nombre = input('Ingrese el nombre del cumpleañero: ')
        edad = validar_rango(0, 13, 'Ingrese la edad del cumpleañero: ')
        tipo_servicio = validar_rango(0, 3, 'Ingrese el tipo de servicio a prestar: ')
        invitados = validar_mayor(0, 'Ingrese la cantidad de invitados: ')
        monto = validar_mayor(0, 'Ingrese el monto a abonar por la fiesta: ')
        reserva = Reserva(numero, nombre, edad, tipo_servicio, invitados, monto)
        vector.append(reserva)
```

```
def acumular_por_tipo(vector):
    va = [0] * 4
    for res in vector:
        va[res.tipo_servicio] += res.monto
    return va

def buscar_mayor_tipo_servicio(vector):
    mayor_tipo = 0
    monto_mayor = vector[0]
    for i in range(len(vector)):
        if vector[i] > monto_mayor:
            monto_mayor, mayor_tipo = vector[i], i
    return mayor_tipo

def mostrar_acumulados(vector, may_tipo_servicio):
    print('Acumulados por Tipo de Servicio')
    print('=' * 65)
    print('{:<45} | {:>15}'.format('Descripcion Tipo', 'Acumulado'))
    print('=' * 65)
    for i in range(len(vector)):
        cad = '{:<45} | S{:>15.2f}'.format(descripcion_servicio(i), vector[i])
        print(cad)
    print('El tipo de servicio que mas recaudo fue', descripcion_servicio(may_tipo_servicio))

def generar_nuevo_por_edad_monto(reservas, edad, invitados):
    vector = []
    for res in reservas:
        if res.edad > edad and res.invitados > invitados:
            vector.append(res)
    return vector

def principal():
    opcion = -1
    reservas = generar_arreglo()
    while opcion != 0:
        print()
        opcion = menu()
        if opcion == 1:
            display(reservas)
        elif opcion == 2:
            agregar_nueva_reserva(reservas)
        elif opcion == 3:
            va = acumular_por_tipo(reservas)
            may_tipo_servicio = buscar_mayor_tipo_servicio(va)
            mostrar_acumulados(va, may_tipo_servicio)
        elif opcion == 4:
            edad = validar_rango(0, 13, 'Ingrese la edad del cumpleaños: ')
            invitados = validar_mayor(0, 'Ingrese la cantidad de invitados: ')
            vector = generar_nuevo_por_edad_monto(reservas, edad, invitados)
            display(vector)
        elif opcion == 0:
            print('Fin del Programa')
            grabar_arreglo(reservas)

if __name__ == '__main__':
    principal()
```


1.4. main para clase

```
from manejador_archivo import generar_arreglo, grabar_arreglo
from reserva import *

__author__ = 'Algoritmos y Estructuras de Datos'

def menu():
    cad = 'Menu de Opciones\n' + \
        '=' * 80 + '\n' + \
        '1 ----- Mostrar Arreglo de Reservas\n' + \
        '2 ----- Agregar nuevo reserva\n' + \
        '3 ----- Determinar total por monto de servicio\n' + \
        '4 ----- Crear y mostrar nuevo vector de reservas\n' + \
        '0 ----- Salir\n' + \
        'Ingrese su opcion: '
    return int(input(cad))

def validar_rango(min, max, mensaje='Ingrese un numero: '):
    valor = min - 1
    while valor < min or valor > max:
        valor = int(input(mensaje))
        if valor < min or valor > max:
            print('Error!!!! El valor esta fuera del rango solicitado [' + str(min) + ',' + str(max) + ']')
    return valor

def validar_mayor(min, mensaje='Ingrese un numero: '):
    valor = min - 1
    while valor < min:
        valor = int(input(mensaje))
        if valor < min:
            print('Error!!!! El valor ser mayor a ' + str(min))
    return valor

def principal():
    opcion = -1
    reservas = generar_arreglo()
    while opcion != 0:
        opcion = menu()
        if opcion == 1:
            pass
        elif opcion == 2:
            pass
        elif opcion == 3:
            pass
        elif opcion == 4:
            pass
        elif opcion == 0:
            print('Fin del Programa')

if __name__ == '__main__':
    principal()
```