ACTIVIDAD 2

Seguridad Ofensiva

Lección 2.
Python II

Armando Elorriaga Muñoz

armando.elorriaga@gmail.com

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

INDICE

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

INTRODUCCION A LA ACTIVIDAD

Ha llegado el momento de realizar el ejercicio, en el que vas a utilizar todos los conocimientos que has aprendido durante las dos unidades dedicadas a Python. Crearás una agenda siguiendo estas instrucciones:

Código fuente y ejecución

En este apartado vamos a explicarte el código fuente del proyecto con cada uno de sus componentes y realizaremos una ejecución de éste para que veas cómo funciona.

El proyecto Agenda estará compuesto por las siguientes clases:

- **Dirección**: contendrá toda la información referente a la dirección.
- **Persona**: contendrá toda la información referente a la persona.
- **Teléfono**: contendrá toda la información referente a los teléfonos.
- **Contacto**: hereda de las tres clases anteriores para conformar una clase que contendrá toda la información de un contacto junta.
- **Agenda**: contendrá toda la información de todos los contactos.

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

DESARROLLO DE LA PRÁCTICA

A continuación el código de la actividad, también disponible en github junto con el archivo de test y este documento. (más abajo su acceso).

```
# -*- coding: utf-8 -*-
#!/usr/bin/Python

#Importamos libreria para limpiar la pantalla
import os
#Para logs
import logging
logging.basicConfig(filename='app.log', filemode='w', format='%(name)s - %(levelname)s - %(message)s')


##DEFINICION DE LAS CLASES
class Direccion:
    def __init__ (self):
        self.__Calle = ""
        self.__Piso = ""
        self.__Ciudad = ""
        self.__CodigoPostal = ""

    def GetCalle(self):
        return self.__Calle

    def SetCalle(self, calle):
        self.__Calle = calle


    def GetPiso(self):
        return self.__Piso

    def SetPiso(self, piso):
        self.__Piso = piso


    def GetCiudad(self):
        return self.__Ciudad

    def SetCiudad(self, ciudad):
        self.__Ciudad = ciudad


    def GetCodigoPostal(self):
        return self.__CodigoPostal

    def SetCodigoPostal(self, codigoPostal):
        self.__CodigoPostal = codigoPostal


class Persona:
    def __init__ (self):
        self.__Nombre = ""
        self.__Apellidos = ""
        self.__FechaNacimiento = ""

    def GetNombre(self):
        return self.__Nombre

    def SetNombre(self, nombre):
        self.__Nombre = nombre


    def GetApellidos(self):
        return self.__Apellidos

    def SetApellidos(self, apellidos):
```

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

```
        self.__Apellidos = apellidos


    def GetFechaNacimiento(self):
        return self.__FechaNacimiento

    def SetFechaNacimiento(self, fechaNacimiento):
        self.__FechaNacimiento = fechaNacimiento



class Telefono:
    def __init__ (self):
        self.__TelefonoFijo = ""
        self.__TelefonoMovil = ""
        self.__TelefonoTrabajo = ""


    def GetTelefonoFijo(self):
        return self.__TelefonoFijo

    def SetTelefonoFijo(self, tfijo):
        self.__TelefonoFijo = tfijo


    def GetTelefonoMovil(self):
        return self.__TelefonoMovil

    def SetTelefonoMovil(self, tmovil):
        self.__TelefonoMovil = tmovil


    def GetTelefonoTrabajo(self):
        return self.__TelefonoTrabajo

    def SetTelefonoTrabajo(self, ttrabajo):
        self.__TelefonoTrabajo = ttrabajo



class Contacto(Persona, Direccion, Telefono):
    def __init__ (self):
        self.__Email = ""

    def GetEmail(self):
        return self.__Email

    def SetEmail(self, email):
        self.__Email = email


    def MostrarContacto(self):
        print("--- CONTACTO ---")
        print("Nombre completo: ", self.GetNombre(), " ", self.GetApellidos())
        print("Fecha de nacimiento: ", self.GetFechaNacimiento())
        print("Direccion: ", self.GetCalle() + " Piso: " + self.GetPiso() + " - " + self.GetCiudad() +
"("+self.GetCodigoPostal()+")")
        print("Telefono fijo: ", self.GetTelefonoFijo())
        print("Telefono movil: ", self.GetTelefonoMovil())
        print("Telefono trabajo: ", self.GetTelefonoTrabajo())
        print("Email: ", self.GetEmail())



class Agenda():
    def __init__ (self, path):
        self.ListaContactos = []
        self.Path = path

    #Cargar contactos desde fichero
    def CargarContactos(self):
        logging.info('Cargando contactos desde archivo.')
        print("Algo")

        #El fichero contiene la informacion de los contactos
```

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

```
        #Cada contacto en una linea separando los datos por ';'
        try:
            fichero = open(self.Path, 'r')
            Lineas = fichero.readlines()
            for linea in Lineas:
                print("linea: ", linea)
                contenidoContacto = linea.split(';')
                print("len(contenidoContacto): ", len(contenidoContacto))
                print("contenidoContacto: ", contenidoContacto)
                #Comprobamos que hay 11 bloques de datos, es decir 10 ';'
                if len(contenidoContacto) == 11:
                    contacto = Contacto()
                    contacto.SetNombre(contenidoContacto[0])
                    print("contacto nombre:", contacto.GetNombre())
                    contacto.SetApellidos(contenidoContacto[1])
                    contacto.SetFechaNacimiento(contenidoContacto[2])
                    contacto.SetTelefonoFijo(contenidoContacto[3])
                    contacto.SetTelefonoMovil(contenidoContacto[4])
                    contacto.SetTelefonoTrabajo(contenidoContacto[5])
                    contacto.SetEmail(contenidoContacto[6])
                    contacto.SetCalle(contenidoContacto[7])
                    contacto.SetPiso(contenidoContacto[8])
                    contacto.SetCiudad(contenidoContacto[9])
                    contacto.SetCodigoPostal(contenidoContacto[10].replace('\n',''))

                    self.ListaContactos.append(contacto)
                    logging.info('Contacto creado.')
            logging.info('Carga de contactos finalizada con exito.')
            fichero.close()
        except Exception as exc:
            logging.error('Fallo al cargar contactos desde archivo: ', exc)


    #Crar un contacto desde teclado
    def CrearNuevoContacto(self, contacto):
        try:
            logging.info('Almacenando contacto')
            self.ListaContactos.append(contacto)
            logging.info('Contacto alamcenado.')

        except Exception as exc:
            logging.error('Fallo al crear contacto: ', exc)


    #Gaurdar los contactos en el archivo
    def GuardarContactos(self):
        logging.info('Guardando contactos en fichero.')

        #El fichero contiene la informacion de los contactos
        #Cada contacto en una linea separando los datos por ';'
        try:
            fichero = open(self.Path, 'w') #Sobre escribimos el contenido

            for contacto in self.ListaContactos:
                linea = ''
                linea   =   contacto.GetNombre()   +   ';'   +   contacto.GetApellidos()   +   ';'   +
contacto.GetFechaNacimiento() + ';' + \
                        contacto.GetTelefonoFijo()  +  ';'  +  contacto.GetTelefonoMovil()    +  ';'  +
contacto.GetTelefonoTrabajo()  + ';' + \
                        contacto.GetEmail() + ';' + contacto.GetCalle() + ';' + contacto.GetPiso() +
';' + \
                        contacto.GetCiudad() + ';' +contacto.GetCodigoPostal()+'\n'
                fichero.write(linea)

            logging.info('Contactos guardado en fichero con exito.')
            fichero.close()
        except Exception as exc:
            logging.error('Fallo al guardar contactos en fichero: ', exc)


    def MostrarAgenda(self):
        if len(self.ListaContactos) > 0:
            for contacto in self.ListaContactos:
                contacto.MostrarContacto()
        else:
```

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

```
            print("La agenda está vacía.")

    def BuscarContactoPorNombre(self, nombre):
        for contacto in self.ListaContactos:
            if contacto.GetNombre().lower() == nombre.lower():
                return contacto
        return None

    def BuscarContactoPorTelefono(self, telefono):
        for contacto in self.ListaContactos:
            if contacto.GetTelefonoFijo() == telefono or contacto.GetTelefonoMovil() == telefono or \
                contacto.GetTelefonoTrabajo() == telefono:
                return contacto
        return None


    def BorrarContactoPorNombre(self, nombre):
        posicion = 0
        encontrado = False
        for contacto in self.ListaContactos:
            if contacto.GetNombre().lower() == nombre.lower():
                encontrado = True
                break
            posicion += 1

        if encontrado:
            del self.ListaContactos[posicion]
            print("Contacto: " + nombre + " eliminado!")
        else:
            print("Contacto: " + nombre + " no encontrado.")


    def BorrarContactoPorTelefono(self, telefono):
        posicion = 0
        encontrado = False
        for contacto in self.ListaContactos:
            if contacto.GetTelefonoFijo() == telefono or contacto.GetTelefonoMovil() == telefono or \
                contacto.GetTelefonoTrabajo() == telefono:
                encontrado = True
                break
            posicion += 1

        if encontrado:
            del self.ListaContactos[posicion]
            print("Contacto con telefono: " + telefono + " eliminado!")
        else:
            print("Telefono " + telefono + " no encontrado.")
```

Definición de funciones principales:

```
def MostrarMenu():
    print ("""Menu Agenda
    1) Buscar contacto
    2) Crear contacto
    3) Borrar contacto
    4) Mostrar agenda
    5) Cargar agenda desde fichero
    6) Guardar agenda en fichero
    7) Salir""")


def MostrarMenuBuscar():
    print ("""
    1) Buscar por nombre
    2) Buscar por telefono
    3) Volver""")


def MostrarMenuBorrar():
```

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

```
    print ("""
    1) Borrar por nombre
    2) Borrar por telefono
    3) Volver""")


def ObtenerOpcion():
    return int(input("Seleccione opcion: "))



def BuscarContacto(agenda, opcion):
    if opcion == 1:
        nombre = input("Introduce nombre del contacto a buscar: ")
        contacto = agenda.BuscarContactoPorNombre(nombre)
    else:
        telefono = input("Introduce telefono del contacto a buscar: ")
        contacto = agenda.BuscarContactoPorTelefono(telefono)

    if contacto != None:
        print("Contacto encontrado")
        contacto.MostrarContacto()
    else:
        print("Contacto NO encontrado")

def BorrarContacto(agenda, opcion):
    if opcion == 1:
        nombre = input("Introduce nombre del contacto a borrar: ")
        agenda.BorrarContactoPorNombre(nombre)
    else:
        telefono = input("Introduce telefono del contacto a borrar: ")
        agenda.BorrarContactoPorTelefono(telefono)


def CrearContacto():
    contacto = Contacto()
    contacto.SetNombre(input("Introduzca nombre: "))
    contacto.SetApellidos(input("Introduzca apellidos: "))
    contacto.SetFechaNacimiento(input("Introduzca fecha de nacimiento: "))
    contacto.SetTelefonoFijo(input("Introduzca telefono fijo: "))
    contacto.SetTelefonoMovil(input("Introduzca telefono movil: "))
    contacto.SetTelefonoTrabajo(input("Introduzca telefono de trabajo: "))
    contacto.SetEmail(input("Introduzca email: "))
    contacto.SetCalle(input("Introduzca calle: "))
    contacto.SetPiso(input("Introduzca piso: "))
    contacto.SetCiudad(input("Introduzca ciudad: "))
    contacto.SetCodigoPostal(input("Introduzca codigo postal: "))
    return contacto
```

Definición de la función principal main:

```
def Main():
    path = "C://Mio//Master//11 Seguridad Ofensiva//Lección 2. Python II//Actividad 2//agenda.txt"
    agenda = Agenda(path)
    continuar = True

    while continuar:
        #Lipiamos la pantalla segun sistema operatico en el que se ejecuta
        os.system('clear') if (os.name == 'posix') else os.system('cls')
        MostrarMenu()
        opcion = ObtenerOpcion()
        #Busqueda de contactos
        if opcion ==  1:
            MostrarMenuBuscar()
            opcion2 = ObtenerOpcion()
            if opcion2 == 1 or opcion2 == 2:
                BuscarContacto(agenda, opcion2)

        #Crear contacto
```

Armando Elorriaga Muñoz
armando.elorriaga@gmail.com

```
        elif opcion == 2:
            contacto = CrearContacto()
            agenda.CrearNuevoContacto(contacto)

        #Borrar contacto
        elif opcion == 3:
            MostrarMenuBorrar()
            opcion2 = ObtenerOpcion()
            if opcion2 == 1 or opcion2 == 2:
                BorrarContacto(agenda, opcion2)

        #Mostrar la agenda
        elif opcion == 4:
            agenda.MostrarAgenda()

        #Cargar agenda desde fichero
        elif opcion == 5:
            agenda.CargarContactos()

        #Guardar agenda en fichero
        elif opcion == 6:
            agenda.GuardarContactos()

        #Salir
        elif opcion == 7:
            continuar = False
            print("Bye Bye")

        if opcion != 7:
            input("Pulse intro para continuar.")


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

Para obtener el código directamente, puedes obtenerlo en mi github:
https://github.com/vtrex3/SOActividad2.git