Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра КІ та КБ

Група: КБ-21-1

Програмування мовою Python

Лабораторна робота №7

«КЛАСИ. Ч. 1»

Виконала: Поліщук К. Р.

Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися з ООП в мові Python

**Хід роботи**

Завдання 1: Реалізувати клас Person, який відображає запис в книзі контактів. Клас має 4 атрибута:

- surname - рядок - прізвище контакту (обов'язковий)

- first\_name - рядок - ім'я контакту (обов'язковий)

- nickname - рядок - псевдонім (опціональний)

- birth\_date - об'єкт datetime.date (обов'язковий)

Кожен виклик класу повинен створювати екземпляр (інстанс) класу із зазначеними атрибутами.

Також клас має 2 методи:

- get\_age() - рахує вік особи в повних роках на дату виклику і повертає рядок виду: "25";

- get\_fullname() - повертає рядок, що відображає повне ім'я (прізвище + ім'я) контакту;

Примітка:

при створенні атрибута birth\_date з рядка типу "2002-12-31" необхідно:

- визначити яка інформація потрібна для створення об'єкта datetime.date,

- отримати ці дані з рядка

- розібрати її (дістати з неї окремо, рік, місяць, число),

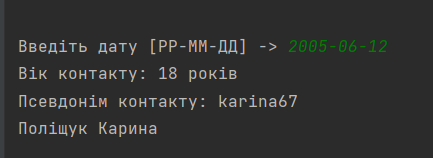
- на підставі цієї інформації створити спеціальний об'єкт datetime.date,

- помістити цей спец.об'ект в атрибут екземпляра класу

Лістинг програми:

import datetime  
import codecs  
from tabulate import tabulate  
from class\_person import ClassPerson  
  
def task1():  
 test = ClassPerson ('Поліщук', 'Карина',  
 datetime.datetime.strptime(str(input('Введіть дату [РР-ММ-ДД] -> ')), "%Y-%m-%d").date(),  
 'karina67')  
  
 print(f"Вік контакту: {test.get\_age()} років")  
 print(f"Псевдонім контакту: {test.nickname}")  
 print(test.get\_fullname())

Результат виконання програми:



Завдання 2. Написати функцію modifier(filename), яка приймає ім'я файлу і повинна:

- прочитати дані з переданого файлу;

- створити об'єкти класу Person на підставі отриманих даних;

- модифікувати дані в файлі:

а) додати графу повного імені (fullname) після графи з ім'ям (name)

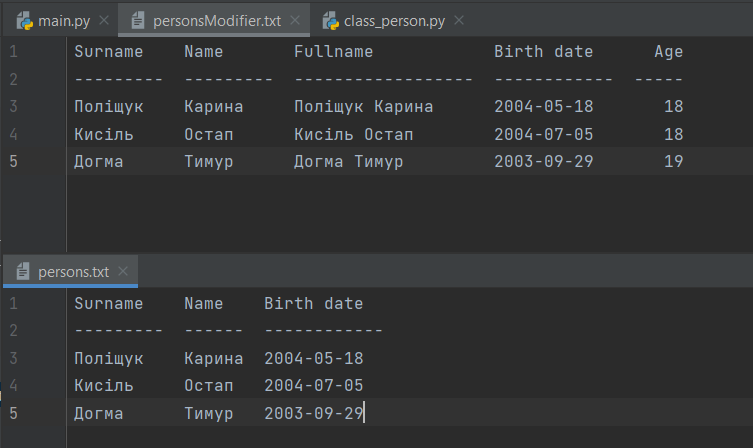
б) додати графу з віком (age) в кінець.

На виході отримати файл, розширений зазначеним чином..

Лістинг програми:

def task2():  
 with codecs.open('files\\persons.txt', 'w', "utf-8") as fileText:  
 table = [['Поліщук', 'Карина', '2004-05-18'],  
 ['Кисіль', 'Остап', '2004-07-05'],  
 ['Догма', 'Тимур', '2003-09-29']]  
 fileText.writelines(tabulate(table, headers=["Surname", "Name", "Birth date"]))  
 persons = ClassPerson.modifier('persons')

Результат виконання програми:



Лістинг класу Person:

from dateutil.relativedelta import relativedelta  
import datetime  
import codecs  
from tabulate import tabulate  
  
class ClassPerson:  
 *"""  
 Клас Person, який відображає запис в книзі контактів.  
 Клас має 4 атрибута:  
 - surname - рядок - прізвище контакту (обов'язковий)  
 - first\_name - рядок - ім'я контакту (обов'язковий)  
 - nickname - рядок - псевдонім (опціональний)  
 - birth\_date - об'єкт datetime.date (обов'язковий)  
 """* def \_\_init\_\_(self, surname, first\_name, birth\_date, nickname=''):  
 self.surname = surname  
 self.first\_name = first\_name  
 self.nickname = nickname  
 self.birth\_date = birth\_date  
  
 def get\_age(self):  
 return (datetime.datetime.now() - relativedelta(years=self.birth\_date.year)).year  
  
 def get\_fullname(self):  
 return self.surname + ' ' + self.first\_name  
  
 @staticmethod  
 def modifier(filename):  
 persons = []  
 with codecs.open(f'files\\{filename}.txt', 'r', "utf-8") as fileRead:  
 with codecs.open(f'files\\{filename}Modifier.txt', 'w', "utf-8") as fileWrite:  
 table = []  
 count = 0  
 for i in fileRead.readlines():  
 if count > 1:  
 line = i.split()  
 persons.append(  
 ClassPerson(line[0], line[1], datetime.datetime.strptime(line[2], "%Y-%m-%d").date()))  
 table.append([line[0], line[1], persons[count - 2].get\_fullname(), line[2],  
 persons[count - 2].get\_age()])  
 count += 1  
 headers = ['Surname', 'Name', 'Fullname', 'Birth date', 'Age']  
 fileWrite.writelines(tabulate(table, headers=headers))  
 return persons

**Висновок:** у даній лабораторній роботі ми ознайомилися з ООП в мові Python.