**ЛАБОРАТОРНА РОБОТА № 7**

**КЛАСИ. Ч. 1**

***Мета***: 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;

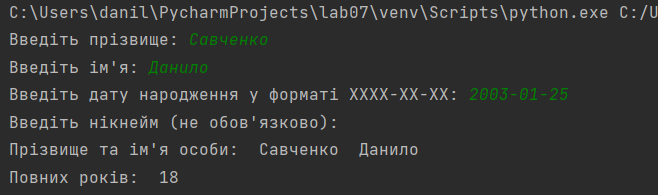
Примітка: при створенні атрибута birth\_date з рядка типу "2002-12-31" необхідно:

* 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 re  
import datetime  
  
  
class Person:  
 def \_\_init\_\_(self, surname, first\_name, birth\_date, nickname=''):  
 self.surname = surname  
 self.first\_name = first\_name  
 self.nickname = nickname  
 self.search\_date(birth\_date)  
  
 def search\_date(self, birth\_date):  
 regex = r'(\d{4})-(\d{2})-(\d{2})'  
 result = re.match(regex, birth\_date)  
 self.birth\_date = datetime.date(int(result.group(1)), int(result.group(2)), int(result.group(3)))  
  
 def get\_age(self):  
 year = int(datetime.datetime.today().year - self.birth\_date.year)  
 month = int(datetime.datetime.today().month - self.birth\_date.month)  
 day = int(datetime.datetime.today().day - self.birth\_date.day)  
 if self.birth\_date.month > month or self.birth\_date.day > day and self.birth\_date.month == month:  
 year -= 1  
 return year  
  
 def get\_fullname(self):  
 return self.surname + " " + self.first\_name  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 surname = input("Введіть прізвище: ")  
 first\_name = input("Введіть ім'я: ")  
 birth\_date = input("Введіть дату народження у форматі ХХХХ-ХХ-ХХ: ")  
 nickname = input("Введіть нікнейм (не обов'язково): ")  
if len(nickname) > 0:  
 person = Person(surname, first\_name, birth\_date, nickname)  
else:  
 person = Person(surname, first\_name, birth\_date)  
print("Прізвище та ім'я особи: ", person.get\_fullname())  
print("Повних років: ", person.get\_age())

Результат виконання програми:



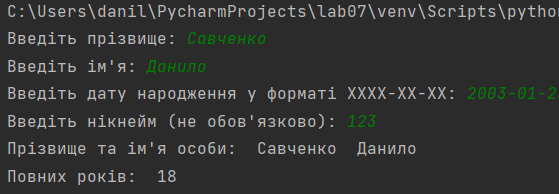


Рис. 1. Результат виконання програми

**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:

1. додати графу повного імені (fullname) після графи з ім'ям (name)
2. додати графу з віком (age) в кінець.

На виході отримати файл, розширений зазначеним чином.

Лістинг програми

def modifier(filename):  
 arr = []  
 with open(f"{filename}", "r", encoding="UTF-8") as a:  
 for line in a:  
 tmp1 = line.rstrip("\n")  
 arr.append(tmp1.split(", "))  
 for i in range(len(arr)):  
 user = (Person(arr[i][0], arr[i][1], arr[i][2]))  
 arr[i].append(str(user.get\_age()))  
 with open(r'rez\_task02.txt', "w", encoding="UTF-8") as rez\_file:  
 for info in arr:  
 info = ', '.join(info)  
 rez\_file.write(info + '\n')  
  
  
modifier('task02.txt')

Результат виконання програми:

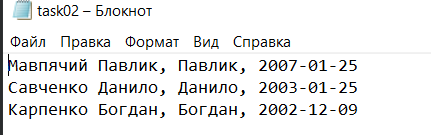


Рис. 2. Початковий файл

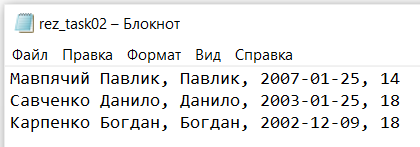


Рис. 3. Кінцевий файл

***Висновки:*** ознайомилися з ООП в мові Python