Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра Бі та Т

Група: ЗІПЗ-21-1

Програмування мовою Python

Лабораторна робота № 7

«Основи мови Python»

Виконав: Ліман П. О.

Прийняв: Морозов Д. С.

**Мета роботи:** Ознайомитися з ООП в Python

**Виконання роботи:**

**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**

class Person:

def \_\_init\_\_(self, surname: str, first\_name: str, birth\_date: str, nickname=""):

self.\_surname = surname

self.\_first\_name = first\_name

self.\_nickname = nickname

self.\_birth\_date = self.\_date\_string\_to\_datetime(birth\_date)

@property

def surname(self):

return self.\_surname

@property

def first\_name(self):

return self.\_first\_name

@property

def nickname(self):

if self.\_nickname != "":

return self.\_nickname

else:

print("This person does not have nickname!")

@property

def birth\_date(self):

return self.\_birth\_date

@staticmethod

def \_date\_string\_to\_datetime(birth\_date) -> datetime.date:

date\_split = birth\_date.split('-')

year = int(date\_split[0])

month = int(date\_split[1])

day = int(date\_split[2])

return datetime.date(year, month, day)

def get\_age(self) -> str:

# Даний метод повертає вік користувача контакту

to\_day = datetime.date.today()

age = (to\_day - self.\_birth\_date)

return str(math.trunc(age.days / 365))

def get\_fullname(self) -> str:

# Даний метод повертає повне ім'я користувача контакту

return "{0} {1}".format(self.\_surname, self.\_first\_name)

**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.**

def modifier(filename):

with open(filename, "r") as PersonFile:

data = PersonFile.readlines()

i = 0

while i < len(data):

data[i] = data[i].replace('\n', '')

i += 1

person = Person(data[0], data[1], data[2], data[3])

with open(filename, "w") as PersonFile:

PersonFile.write(person.first\_name + "\n")

PersonFile.write(person.surname + "\n")

PersonFile.write(person.get\_fullname() + "\n")

PersonFile.write(str(person.birth\_date) + "\n")

PersonFile.write(person.nickname + "\n")

PersonFile.write(person.get\_age())

**Результат виконання програми:**

Файл до:



Файл після:



**Висновок:** Було огляно основи ООП у мові програмування Python.