

Міністерство освіти і науки України
Державний університет „Житомирська політехніка”

Кафедра ІПЗ
Група: ІПЗ-23-1

Програмування мовою Python
Лабораторна робота №7
«КЛАСИ. Ч. 1»

Виконав:

Риженко Я.В.

Прийняв:

Желізко В. В.

					«Житомирська політехніка».24.121.05.000				
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи		Літ.	Арк.	Аркушів
Розроб.		Риженко Я.В.							
Перевір.		Желізко В. В.						1	5
Керівник							ФІКТ, гр. ІПЗ-23-1		
Н. контр.									
Затверд.									

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'єкт в атрибут екземпляра класу

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.

					«Житомирська політехніка».24.121.05.000	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Листинг програми:

```
import csv
from datetime import date

# Task 1
class Person:
    def __init__(self, surname: str, first_name: str, birth_date: str, nickname: str = None):
        if not surname.strip():
            raise ValueError("Прізвище не може бути порожнім.")
        if not first_name.strip():
            raise ValueError("Ім'я не може бути порожнім.")

        self.surname = surname.strip()
        self.first_name = first_name.strip()
        self.nickname = nickname.strip() if nickname else None

        year, month, day = map(int, birth_date.split('-'))
        birth_date_obj = date(year, month, day)

        if birth_date_obj > date.today():
            raise ValueError("Дата народження не може бути в майбутньому.")

        self.birth_date = birth_date_obj

    def get_age(self) -> str:
        today = date.today()
        age = today.year - self.birth_date.year - (
            (today.month, today.day) < (self.birth_date.month,
self.birth_date.day)
        )
        return str(age)

    def get_fullname(self) -> str:
        return f"{self.surname} {self.first_name}"

def create_person(filename):
    print("Введіть дані для створення нового контакту.")
    surname = input("Прізвище: ").strip()
    first_name = input("Ім'я: ").strip()
    birth_date = input("Дата народження (у форматі YYYY-MM-DD): ").strip()
    nickname = input("Псевдонім (необов'язково): ").strip() or None

    try:
        person = Person(surname=surname, first_name=first_name,
birth_date=birth_date, nickname=nickname)
        print("\nКонтакт успішно створено!")
        print(f"Повне ім'я: {person.get_fullname()}")
        print(f"Вік: {person.get_age()} років")
        if person.nickname:
            print(f"Псевдонім: {person.nickname}")

        with open(filename, 'a', newline='', encoding='utf-8') as file:
            writer = csv.DictWriter(file, fieldnames=["surname", "name",
```

					«Житомирська політехніка».24.121.05.000	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

"birth_date", "nickname"]])

    if file.tell() == 0:
        writer.writeheader()

    writer.writerow({
        "surname": person.surname,
        "name": person.first_name,
        "birth_date": person.birth_date,
        "nickname": person.nickname or ''
    })

    print("Контакт успішно додано до файлу.")
except ValueError as e:
    print(f"Помилка: {e}")
except Exception as e:
    print(f"Невідома помилка: {e}")

#Task 2
def modifier(filename):
    try:
        with open(filename, 'r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            fieldnames = reader.fieldnames

            if not {'surname', 'name', 'birth_date'}.issubset(set(fieldnames)):
                raise ValueError("Файл повинен містити графи 'surname', 'name',
'birth_date'.")

            extended_fieldnames = fieldnames.copy()
            if 'fullname' not in extended_fieldnames:
                extended_fieldnames.insert(extended_fieldnames.index('name') + 1,
'fullname')
            if 'age' not in extended_fieldnames:
                extended_fieldnames.append('age')

            modified_rows = []
            for row in reader:
                surname = row['surname'].strip()
                name = row['name'].strip()
                birth_date = row['birth_date'].strip()

                person = Person(surname, name, birth_date)

                row['fullname'] = person.get_fullname()
                row['age'] = person.get_age()

                modified_rows.append(row)

            with open(filename, 'w', encoding='utf-8', newline='') as file:
                writer = csv.DictWriter(file, fieldnames=extended_fieldnames)
                writer.writeheader()
                writer.writerows(modified_rows)

            print("Файл успішно модифіковано.")
    except FileNotFoundError:

```

					«Житомирська політехніка».24.121.05.000	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        print(f"Помилка: Файл {filename} не знайдено.")
    except ValueError as e:
        print(f"Помилка: {e}")
    except Exception as e:
        print(f"Невідома помилка: {e}")

if __name__ == "__main__":
    create_person('contacts.csv')
if __name__ == "__main__":
    modifier('contacts.csv')

```

Результат виконання:

```

Введіть дані для створення нового контакту.
Прізвище: Приклад Виконання
Ім'я: Лабораторної
Дата народження (у форматі YYYY-MM-DD): 2024-06-12
Псевдонім (необов'язково):

Контакт успішно створено!
Повне ім'я: Приклад Виконання Лабораторної
Вік: 0 років
Контакт успішно додано до файлу.
Файл успішно модифіковано.

```

Рис. 1. Результат.

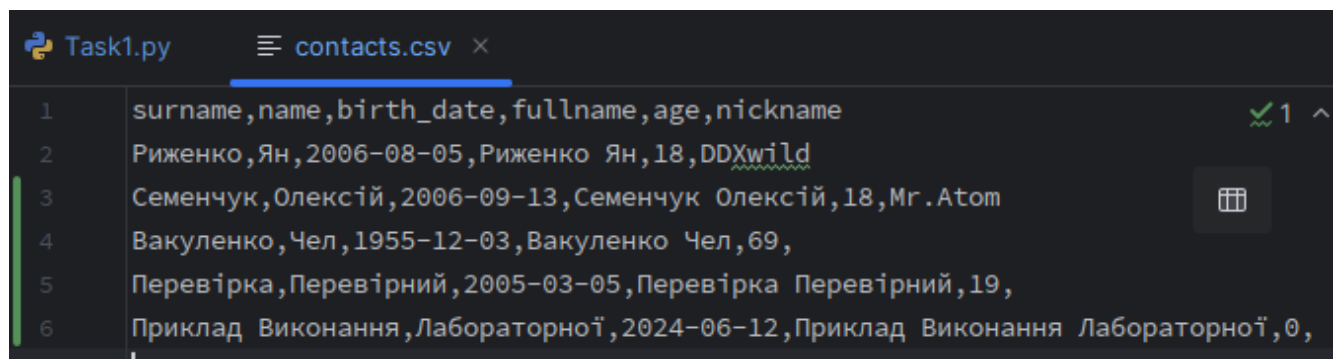


Рис. 2. Створений та модифікований файл.

Висновок: У цій лабораторній роботі ми ознайомились з основами ООП мови Python.

Посилання на Git: : https://github.com/JanRizhenko/Python_Labs

					«Житомирська політехніка».24.121.05.000	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		