**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет інформаційних технологій**

**Кафедра комп’ютерних систем і мереж**

**Звіт з лабораторної роботи №4**

# Курс «Кросплатформне програмування (Python)»

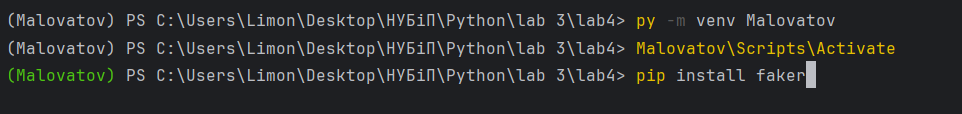
**Робота з файлами. Серіалізація**

**Виконав:**

Студент групи КІ-21013Б Маловатов Дмитро Валерійович

**Київ – 2024**

1. Створити віртуальне оточення

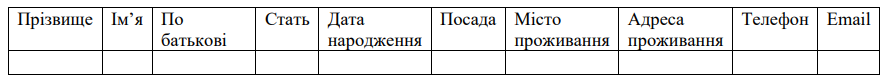


Створити каталог проекту та перехід до нього

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

2. В проекті створити три програми. Перша зберігає в файл CSV таблицю



import csv  
from faker import Faker  
from datetime import datetime, timedelta  
import random  
  
fake = Faker('uk\_UA')  
  
po\_batkovi = {  
 'male': ['Олександрович', 'Іванович', 'Петрович', 'Миколайович', 'Васильович', 'Сергійович', 'Андрійович',  
 'Вікторович', 'Юрійович', 'Романович', 'Максимович', 'Дмитрович', 'Артемович', 'Ігорович', 'Володимирович',  
 'Олексійович', 'Євгенович', 'Денисович', 'Владиславович', 'Павлович'],  
 'female': ['Олександрівна', 'Іванівна', 'Петрівна', 'Миколаївна', 'Василівна', 'Сергіївна', 'Андріївна',  
 'Вікторівна', 'Юріївна', 'Романівна', 'Максимівна', 'Дмитрівна', 'Артемівна', 'Ігорівна',  
 'Володимирівна', 'Олексіївна', 'Євгенівна', 'Денисівна', 'Владиславівна', 'Павлівна']  
}  
  
with open('employees.csv', 'w', newline='', encoding='utf-8') as csvfile:  
 fieldnames = ['Прізвище', 'Ім\'я', 'По батькові', 'Стать', 'Дата народження', 'Посада', 'Місто',  
 'Адреса проживання', 'Телефон', 'Email']  
 writer = csv.DictWriter(csvfile, fieldnames=fieldnames)  
 writer.writeheader()  
  
 for \_ in range(2000):  
 gender = 'female' if random.random() < 0.4 else 'male'  
 if gender == 'female':  
 first\_name = fake.first\_name\_female()  
 po\_batkovi\_name = random.choice(po\_batkovi['female'])  
 else:  
 first\_name = fake.first\_name\_male()  
 po\_batkovi\_name = random.choice(po\_batkovi['male'])  
  
 birth\_date = fake.date\_between\_dates(date\_start=datetime(1938, 1, 1), date\_end=datetime(2008, 1, 1))  
  
 employee = {  
 'Прізвище': fake.last\_name\_nonbinary() if gender not in ("male", "female") else fake.last\_name(),  
 'Ім\'я': first\_name,  
 'По батькові': po\_batkovi\_name,  
 'Стать': gender,  
 'Дата народження': birth\_date.strftime('%Y-%m-%d'),  
 'Посада': fake.job(),  
 'Місто': fake.city(),  
 'Адреса проживання': fake.address(),  
 'Телефон': fake.phone\_number(),  
 'Email': fake.email()  
 }  
 writer.writerow(employee)

Изображение выглядит как текст, снимок экрана, Шрифт, информация

Автоматически созданное описание

3. Друга програма створює файл XLSX. В ньому 5 аркушів – “all”, “younger\_18”, “18-45”, “45-70”, “older\_70”. В аркуш “all” записує всі данні із CSV файлу, який створено в другому пункті. На інших аркушах відображає данні про співробітників, які належать до відповідної вікової категорії наступної структури:

import openpyxl  
from openpyxl.styles import Font  
import csv  
from datetime import datetime, date, timedelta  
  
try:  
 workbook = openpyxl.Workbook()  
 sheets = {  
 "all": workbook.active,  
 "younger\_18": workbook.create\_sheet("younger\_18"),  
 "18-45": workbook.create\_sheet("18-45"),  
 "45-70": workbook.create\_sheet("45-70"),  
 "older\_70": workbook.create\_sheet("older\_70")  
 }  
  
 for sheet\_name in sheets:  
 sheets[sheet\_name].append(['№', 'Прізвище', 'Ім\'я', 'По батькові', 'Дата народження','Вік'])  
  
  
 try:  
 with open('employees.csv', 'r', newline='', encoding='utf-8') as csvfile:  
 reader = csv.DictReader(csvfile)  
 index = 1  
 for row in reader:  
 birth\_date\_str = row['Дата народження']  
 birth\_date = datetime.strptime(birth\_date\_str, '%Y-%m-%d').date()  
 age = (date.today() - birth\_date) // timedelta(days=365.2425) # More accurate age calculation  
  
 row\_data = [index, row['Прізвище'], row['Ім\'я'], row['По батькові'], row['Дата народження'], age]  
  
 sheets["all"].append(row\_data)  
  
 if age < 18:  
 sheets["younger\_18"].append(row\_data)  
 elif 18 <= age <= 45:  
 sheets["18-45"].append(row\_data)  
 elif 45 < age <= 70:  
 sheets["45-70"].append(row\_data)  
 else:  
 sheets["older\_70"].append(row\_data)  
  
 index += 1  
  
  
 workbook.save('employees.xlsx')  
 print("Ok")  
  
 except FileNotFoundError:  
 print("Повідомлення про відсутність, або проблеми при відкритті файлу CSV.")  
  
except Exception as e:  
 print(f"Повідомлення про неможливість створення XLSX файлу: {e}")

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

4. Третя програма зчитує дані з CSV файлу і виконує наступні функції:

import csv  
from datetime import datetime, date, timedelta  
import matplotlib.pyplot as plt  
import pandas as pd  
  
try:  
 with open('employees.csv', 'r', newline='', encoding='utf-8') as csvfile:  
 reader = csv.DictReader(csvfile)  
  
 gender\_counts = {'male': 0, 'female': 0}  
 age\_groups = {'younger\_18': 0, '18-45': 0, '45-70': 0, 'older\_70': 0}  
 gender\_age\_groups = {  
 'male': {'younger\_18': 0, '18-45': 0, '45-70': 0, 'older\_70': 0},  
 'female': {'younger\_18': 0, '18-45': 0, '45-70': 0, 'older\_70': 0}  
 }  
  
 for row in reader:  
 gender\_counts[row['Стать']] += 1  
  
 birth\_date\_str = row['Дата народження']  
 birth\_date = datetime.strptime(birth\_date\_str, '%Y-%m-%d').date()  
 age = (date.today() - birth\_date) // timedelta(days=365.2425)  
  
 if age < 18:  
 age\_groups['younger\_18'] += 1  
 gender\_age\_groups[row['Стать']]['younger\_18'] += 1  
 elif 18 <= age <= 45:  
 age\_groups['18-45'] += 1  
 gender\_age\_groups[row['Стать']]['18-45'] += 1  
  
 elif 45 < age <= 70:  
 age\_groups['45-70'] += 1  
 gender\_age\_groups[row['Стать']]['45-70'] += 1  
  
 else:  
 age\_groups['older\_70'] += 1  
 gender\_age\_groups[row['Стать']]['older\_70'] += 1  
 print("Ok")  
  
 print("Кількість співробітників за статтю:", gender\_counts)  
 plt.figure(figsize=(6, 4))  
 plt.bar(gender\_counts.keys(), gender\_counts.values())  
 plt.title('Розподіл співробітників за статтю')  
 plt.xlabel('Стать')  
 plt.ylabel('Кількість')  
  
 print("Кількість співробітників за віковими групами:", age\_groups)  
 plt.figure(figsize=(8, 5))  
 plt.bar(age\_groups.keys(), age\_groups.values())  
 plt.title('Розподіл співробітників за віковими групами')  
 plt.xlabel('Вікова група')  
 plt.ylabel('Кількість')  
  
 for gender in gender\_age\_groups:  
 print(f"Кількість співробітників {gender} статі за віковими групами:", gender\_age\_groups[gender])  
 plt.figure(figsize=(8, 5))  
 plt.bar(gender\_age\_groups[gender].keys(), gender\_age\_groups[gender].values())  
 plt.title(f'Розподіл співробітників {gender} статі за віковими групами')  
 plt.xlabel('Вікова група')  
 plt.ylabel('Кількість')  
  
 plt.show()  
  
  
except FileNotFoundError:  
 print("Повідомлення про відсутність або проблеми при відкритті файлу CSV.")  
  
except Exception as e:  
 print(f"Помилка: {e}")

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

5. Створити файл requirments.txt, в якому зберігаються всі встановлені модулі та пакети.



Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

6. Завантажити проект на сервіс GitHub.