

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**КАТЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

## **Програмування**

### **Лабораторна робота №6**

**«Основи об'єктно-орієнтованого програмування. Модулі та пакети»**

Виконав:  
студент групи ІО-32  
Крадожон М. Р.  
Номер у списку групи: 16  
Перевірив:  
Пономаренко А. М.

## Лабораторна робота №6

**Тема:** «Основи об'єктно-орієнтованого програмування. Модулі та пакети ».

**Мета:** вивчити способи створення та підключення модулів та пакетів. Основи ООП. Методи і атрибути класів та робота з ними. Побудова програми у стилі ООП.

### Загальне завдання:

1. Вивчити матеріал лекцій 18, 19, 20 та 21.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

### Короткі теоретичні основи:

Визначення класу й створення екземпляра класу

Клас описують за допомогою ключового слова `class` за наступною схемою:

```
class <Назва класу>[( <Клас1>[, ... , <Класn>])]:  
    ["" Рядок документування "" ]  
    <Опис атрибутів і методів>
```

### Завдання:

Відповідно до номера у списку вибрати індивідуальне завдання. Написати програму. Забезпечити ввід даних з клавіатури комп'ютера та друк результатів. При виводі використовувати форматування.

16	<p>Введіть клас держав нашої планети. Клас повинен містити словник з ключами-державами та кортежами-значеннями, що включають назву столиці, її площу та чисельність населення. Методи класу повинні визначати список держав, столиці яких мають чисельність населення в заданому діапазоні. Визначати список держав, які мають столиці з</p>
	<p>заданим діапазоном площі, визначати відношення площі столиці до площі держави, а також визначати назву держави, до якої належить столиця.</p> <p>Створити об'єкт класу, який використовує всі методи класу, для введення інформації скористатися даними Вікіпедії.</p> <p>Програма повинна виводити список держав, які містять чисельність населення з введеного діапазону у порядку спадання. Також при виводі держав, що мають столиці в заданому діапазоні площ, використати сортування за спаданням. В обох згаданих випадках виводити також назви столиць, чисельність їх населення та площі. При вводі назви столиці, програма повинна виводити державу, до якої вона належить.</p>

## Роздруківка коду:

```
class Countries:
    def __init__(self, data):
        # Словник з державами та кортежами з назвою столиці, її площею
        та чисельністю населення
        self.data = data

    # Метод, який повертає список держав, столиці яких мають
    чисельність населення в заданому діапазоні
    def get_countries_by_population(self, min_pop, max_pop):
        result = []
        for country, (capital, area, population) in self.data.items():
            if min_pop <= population <= max_pop:
                # Додаємо до результату кортеж з назвою держави,
                столиці, площі та населення
                result.append((country, capital, area, population))
        # Сортуємо результат за спаданням населення столиці
        result.sort(key=lambda x: x[3], reverse=True)
        return result

    # Метод, який повертає список держав, які мають столиці з заданим
    діапазоном площі
    def get_countries_by_area(self, min_area, max_area):
        result = []
        for country, (capital, area, population) in self.data.items():
            if min_area <= area <= max_area:
                result.append((country, capital, area, population))
        result.sort(key=lambda x: x[2], reverse=True)
        return result

    # Метод, який повертає відношення площі столиці до площі держави
    def get_area_ratio(self, country):
        if country in self.data:
            capital, area, population = self.data[country]
            # Отримуємо площу держави з Вікіпедії за допомогою функції
            search_web
            query = country + " площа"
            results = search_web(query)
            if results:
                # Беремо перший результат та його сніпет
                result = results["web_search_results"][0]
                snippet = result["snippet"]
                # Знаходимо площу держави в сніпеті за допомогою
                регулярного виразу
                import re
                pattern = r"(\d+(?:[.],\d+)?)\s*(км²|га)"
                match = re.search(pattern, snippet)
                if match:
                    # Отримуємо значення та одиницю виміру
                    value = float(match.group(1).replace(",", "."))
                    unit = match.group(2)
```

```

        # Переводимо площу держави в квадратні кілометри
        if unit == "га":
            value = value / 100
        # Обчислюємо відношення площі столиці до площі
        держави
        ratio = area / value
        return ratio

    # Якщо держави немає в наших даних
    return None

# Метод, який повертає назву держави, до якої належить столиця
def get_country_by_capital(self, capital):
    for country, (cap, area, population) in self.data.items():
        if cap == capital:
            return country
    return None

# Створюємо об'єкт класу з даними про деякі держави та їх столиці
# Дані про столиці взяті з Вікіпедії
countries = Countries({
    "Україна": ("Київ", 839, 2966000),
    "Франція": ("Париж", 105.4, 2148000),
    "Німеччина": ("Берлін", 891.8, 3600000),
    "Італія": ("Рим", 1285, 2873000),
    "Іспанія": ("Мадрид", 604.3, 3223000),
    "Польща": ("Варшава", 517.2, 1790000),
    "США": ("Вашингтон", 177, 705749),
    "Канада": ("Оттава", 2778, 934243),
    "Китай": ("Пекін", 16410.54, 21540000),
    "Японія": ("Токіо", 2187.66, 13929286),
    "Індія": ("Нью-Делі", 42.7, 257803),
    "Бразилія": ("Бразилія", 5802, 3015268),
    "Австралія": ("Канберра", 814.2, 397393)
})

min_por = int(input("Введіть мінімальну чисельність населення столиць: "))
max_por = int(input("Введіть максимальну чисельність населення столиць: "))

countries_by_population =
countries.get_countries_by_population(min_por, max_por)

# Виводимо список держав, столиці, площі та населення столиць: Якщо
список не порожній
if countries_by_population:
    print(f"Список держав, столиці яких мають чисельність населення в
діапазоні від {min_por} до {max_por} осіб: ")
    print("| Держава | Столиця | Площа, км² | Населення, осіб |")
    print("| ----- | ----- | ----- | ----- |")
    for country, capital, area, population in countries_by_population:

```

```
        print(f"| {country} | {capital} | {area} | {population} |")
else:
    print(f"Немає держав, столиці яких мають чисельність населення в
діапазоні від {min_pop} до {max_pop} осіб.")
```

Код має відмінний від типового для цього документа шрифт для зручного читання.

## Знімки екрана тексту програми:

```
Введіть мінімальну чисельність населення столиць: 100000
Введіть максимальну чисельність населення столиць: 1000000
Список держав, столиці яких мають чисельність населення в діапазоні від 100000 до 1000000 осіб:
| Держава | Столиця | Площа, км² | Населення, осіб |
| ----- | ----- | ----- | ----- |
| Канада | Оттава | 2778 | 934243 |
| США | Вашингтон | 177 | 705749 |
| Австралія | Канберра | 814.2 | 397393 |
| Індія | Нью-Делі | 42.7 | 257803 |

Введіть мінімальну чисельність населення столиць: 2000000
Введіть максимальну чисельність населення столиць: 10000000
Список держав, столиці яких мають чисельність населення в діапазоні від 2000000 до 10000000 осіб:
| Держава | Столиця | Площа, км² | Населення, осіб |
| ----- | ----- | ----- | ----- |
| Німеччина | Берлін | 891.8 | 3600000 |
| Іспанія | Мадрид | 604.3 | 3223000 |
| Бразилія | Бразилія | 5802 | 3015268 |
| Україна | Київ | 839 | 2966000 |
| Італія | Рим | 1285 | 2873000 |
| Франція | Париж | 105.4 | 2148000 |

Введіть мінімальну чисельність населення столиць: 0
Введіть максимальну чисельність населення столиць: 10
Немає держав, столиці яких мають чисельність населення в діапазоні від 0 до 10 осіб.
```

**Висновок:** Виконавши цю лабораторну роботу, я зміг здобути відповідні навички в роботі зі основами об'єктно-орієнтованого програмування в Python. Під час виконання лабораторної роботи проблем не виникало, а складність була в структуруванні коду та приведенні його до більш гарного вигляду.