

Лабораторна робота № 2

Тема: Довгий список параметрів. Надлишкові елементарні типи

Мета: Аналіз та оптимізація методів з великою кількістю параметрів. Рефакторинг класів з надлишковими елементарними типами даних.

Завдання:

1. Надано код

```
def calculate_score(name, age, gender, height, weight, score1, score2,
score3, score4, score5):
    # Бізнес-логіка для розрахунку загального рейтингу
    total_score = score1 + score2 + score3 + score4 + score5

    # Перевірка чи користувач є повнолітнім
    is_adult = True if age >= 18 else False

    # Виведення результатів
    print("Name:", name)
    print("Age:", age)
    print("Gender:", gender)
    print("Height:", height)
    print("Weight:", weight)
    print("Total Score:", total_score)
    print("Adult:", is_adult)

    # Приклад виклику функції
    calculate_score("John", 25, "Male", 175, 70, 85, 90, 75, 88, 92)
```

2. Виконати рефакторинг цієї функції, зменшивши кількість параметрів та полегшивши розуміння та обслуговування коду

3. Надано код

```
class User:
    type_engineer = 1
    type_manager = 2

    def __init__(self, name, age, type, phone, phone_code):
        self.name = name
        self.age = age
        self.type = type
        self.phone = phone
        self.phone_code = phone_code

    def print_details(self):
        # Виведення інформації про користувача
        print("Name:", self.name)
        print("Age:", self.age)
        print("Type:", self.type)
        print("Phone:", self.phone_code + self.phone)

# Приклад використання класу
user = User("John", 25, User.type_engineer, "9379992", "050")
user.print_details()
```

4. Виконати рефакторинг використовуючи техніки надані в теоретичній частині.

Звіт:

Звіт має включати в себе чотири файла:

1. **output_1.py** - код після рефакторінгу з п.1
2. **output_2.py** - код після рефакторінгу з п.3
3. **README.md** - опис аналізу і прийнятих рішень що до рефакторінгу.

Робота може бути виконана на будь якій мові програмування або на псевдокодi.