

## Лабораторна робота № 7

**Тема:** Створення користувацьких функцій в Python.

**Мета:** Отримати практичні навички роботи з функціональним програмуванням в Python.

Васильєв О.М. Програмування мовою Python – Тернопіль: Навчальна книга – Богдан, 2019. – 504с. стор. 139-198

### Порядок виконання завдань.

#### Завдання 1.

Дано список цілих чисел. Написати програму подвоєння елементів списку з використанням лямбда-функції та функції `map()`.

#### Завдання 2.

Дано рядок з 20 слів. Використовуючи лямбда-функцію та функції `map()` всі слова записати з великої букви.

#### Завдання 3.

Напишіть функцію для перевірки того, чи введена літера є голосною чи приголосною.

#### Завдання 4.

Напишіть функцію, яка приймає два слова в якості вхідних даних, і надрукуйте найдовше слово. Якщо слова мають однакову довжину, то функція повинна надрукувати слова в окремих рядках.

#### Завдання 5.

Написати функцію для перевірки правильності введеної дати. Функція приймає 3 аргументи - день, місяць та рік і повертає `True`, якщо така дата є в календарі, і `False` в протилежному випадку.

#### Завдання 6.

На стадіоні є три категорії місць для сидіння: місця класу А коштують `a` грошових одиниць, місця класу В коштують `b` грошових одиниць, а місця класу С - `c` грошових одиниць. Напишіть першу функцію, яка запитує скільки продано квитків на кожний клас місць, і другу функцію, яка відображає суму отриманого доходу від продажу квитків на кожен клас окремо і загалом.

## Методичні рекомендації.

Вбудована в Python функція `map()` використовується для застосування функції до кожного елементу об'єкта (наприклад, списку або словника) і повернення нового ітератора для отримання результатів.

Замість використання циклу `for` функція `map()` дає можливість застосувати функцію до кожного елементу об'єкта. Це підвищує продуктивність, оскільки функція застосовується тільки до одного елементу за раз без створення копій елементів в іншому об'єкті. Це особливо корисно при обробці великих наборів даних. Також `map()` може приймати кілька ітеріруємих об'єктів в якості аргументів функції, відправляючи в функцію по одному елементу кожного ітеріруємого об'єкта за раз.

Перший аргумент `map()` - це функція, яку ми використовуємо для застосування до кожного елементу. Python викликає функцію один раз для кожного елементу об'єкта, який ми передаємо в `map()`, і повертає змінений елемент в об'єкт `map`. Як перший аргумент функції ми можемо передати певну користувачем функцію або використовувати функції `lambda`, особливо якщо вираз буде менш складним.

Синтаксис `map()` з функцією `lambda` виглядає наступним чином:

```
map(lambda item: item[] expression, iterable)
```

У такому випадку ми можемо використовувати лямбда-функцію з виразом, яке ми хочемо застосувати до кожного елементу в цьому списку. Щоб зробити це, ми можемо використовувати функції `map()` і `lambda` наступним чином:

```
mapped_numbers = list(map(lambda x: x + 3, numbers))
```

## Контрольні запитання.

1. Як створити функцію у мові Python?
2. Що таке анонімні функції та інструкція `lambda`?
3. Який синтаксис оголошення функції?
4. Що таке позиційні та непозиційні аргументи функції?
5. Як створити функцію зі значеннями за замовчуванням?
6. Який порядок передачі аргументів у функцію, якщо вона містить позиційні та непозиційні аргументи та аргументи із значеннями за замовчуванням?
7. Для чого використовується інструкція `return`? Чи обов'язково вона присутня у функції?
8. Що повертає функція, якщо в її тілі відсутня інструкція `return`?