НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №3

із дисципліни «Основи програмування»

на тему

“Списки, кортежі, множини”

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-11 | асистент Громова В. В. |
| Єсипенко Є.Ю. |  |

Київ — 2021

Зміст

[1 ПОСТАНОВА ЗАДАЧІ 2](#_gjdgxs)

[1.1 Мета роботи 3](#_30j0zll)

[1.2 Завдання до лабораторної роботи 3](#_1fob9te)

[2 ОПИС ПРОГРАМИ 3](#_3znysh7)

[2.1 Опис 1 4](#_2et92p0)

[2.2 Опис 2 4](#_tyjcwt)

[Висновки 6](#_3dy6vkm)

[Додаток А 8](#_1t3h5sf)

[Текст програми 1 8](#_4d34og8)

[Текст програми 2 9](#_2s8eyo1)

[Додаток Б 10](#_17dp8vu)

[Скріншоти тестувань програми 11](#_3rdcrjn)

[Відповіді на контрольні запитання 12](#_26in1rg)

# 1 ПОСТАНОВА ЗАДАЧІ

## **1.1 Мета роботи**

Вивчення структур даних (list, tuple, set), створюваних користувачем.

## **1.2 Завдання до лабораторної роботи**

1) Вивчити принципи створення списків і множин у мові Python.

2) Розробити програми відповідно до варіанта завдання.

3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.

4) Показати розроблену програму викладачеві.

5) Письмово відповісти на Питання для самоперевірки.

6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:

* робота зі списками;
* робота з множинами.

# 2 ОПИС ПРОГРАМИ

## 2.1 Опис 1. Задача на списки.

Перетворити список, у якому елементи представляють групи символів, у список з поодинокими символами, вибраними з цих груп. Необхідно:

* виконати це завдання 4-ма варіантами, визначити час виконання кожним і знайти самий швидкодіючий;
* 1-ий варіант: працювати з оператором *for* (це найпростіший варіант);
* 2-ий варіант: застосувати функціональне програмування, а саме, функції *reduce, map, lambda* (один рядок);
* 3-ий варіант: застосувати функцію *sum* (один рядок);
* 4-ий варіант: застосувати функцію *join* (один рядок).

Дії відбуваються зі списком, що заздалегідь був підготований. Завдяки цьому ми зможемо порівняти чотири варіанти виконання та зрівняти швидкість та зручність кожного з ним при виконанні такої задачі.

Першим варіантом є виконання умов за допомогою оператора for(). Цей оператор має бути визначений у декілька рядків, з написанням умови відбору елементів(символів) та виведенням результату (print()). За умовою, програма буде перебирати елементи списку (тут існують групи як чисел, так і булеві значення й строки) , та видаляти всі, що за порядковим номером діляться на 3 (з урахуванням першого (0) елементу, адже ділення на нуль не відбуватиметься. Результат буде виведений завдяки новому списку, до якого будуть додані вихідні елементи(символи). Написання швидке, а читання легке. Це найлегший варіант, адже найзрозуміліший.

Другий спосіб включає у себе функційне програмування (ФП). Метод ФП полягає у створенні програм, в яких єдиною дією є виклик функції, єдиним способом розбиття програми є створення нового імені функції та задання для цього імені виразу, що обчислює значення функції, а єдиним правилом композиції є оператор суперпозиції функцій. Жодних комірок пам'яті, операторів присвоєння, циклів, ні, тим більше, блок-схем чи передачі управління.

(1). Оператор Map приймає функцію і набір даних. Створює нову колекцію, виконує функцію на кожній позиції даних і додає значення, що повертається в нову колекцію. Повертає результатом нову колекцію. У програмі Map виконує свою дію, сортуючи слова з цитати та видаляючи кожен, що не відповідає умові, після чого додає всі слова, що відповідають ним, та виводить для користувача.

(2) Функція Reduce приймає функцію і набір пунктів. Повертає значення, що отримується комбінуванням всіх пунктів.

(3) Lambda - це інструмент для виклику анонімних функцій. Ідея була в тому, аби програма мала змогу перебрати кожен елемент зі списку та вибірково множити на другий список.

Третій спосіб — спосіб обробки інформації оператором sum. Вбудована функція sum () - це ефективний спосіб підсумовування списку числових значень. Додавання кількох чисел є звичайним проміжним кроком у багатьох обчисленнях, тому sum () - досить зручний інструмент для програміста Python.

Четвертий спосіб — спосіб оператора join. У Python він відповідає за об'єднання списку рядків з допомогою певного покажчика. Часто це використовується при конвертації списку в рядок. Наприклад, так можна конвертувати список букв алфавіту в розділену запитом рядок для збереження.

Метод приймає ітеріруемий об'єкт в якості аргументу, а оскільки список відповідає цим умовам, то його цілком можна використовувати і тут.

На мою думку найшвидшим та найлегшим для читання та написання спосіб буде спосіб оператором for().

## 2.2 Опис 2

Задача:

Першим кроком є введення даних множин А та В. Множина F знаходиться за один крок одним рядком за допомогою функцій A.symmetric\_difference(B), A.difference(B) та x.intersection.y. . Після цього результат виводиться на екран.

# Висновки

Вивчено структури даних, створені користувачем. Вивчено список, кортеж та множину. Опрацьовано операції зі списками, перетворення списку в інші дані.

# Додаток А

## Текст програми 1

from itertools import product

import sys

import os

import string

print("Лабораторна робота номер 3: СТРУКТУРИ ДАНИХ: СПИСКИ, КОРТЕЖИ, МНОЖИНИ", "Роботу виконала: Єсипенко Є.Ю. ,КМ-11", "Варіант номер 8. ", "", '', sep = "\n")

print('Перетворити список, у якому елементи представляють групи символів, у список з поодинокими символами, вибраними з цих груп.',

    '1-ий варіант: працювати з оператором for (це найпростіший варіант);', '2-ий варіант: застосувати функціональне програмування, а саме, функції reduce, map, lambda (один рядок);',

    '3-ий варіант: застосувати функцію sum (один рядок);', '4-ий варіант: застосувати функцію join (один рядок).', '',

     sep = "\n")

while True:

    print('1-ий варінт: ', 'list = [12,барвінок,їжак,2,43,47,True,колір,5,690,7,привіт,9,101, False]', 'програма вибере кожний елемент кратний 3-м зі списку та видалить його. (перший елемент також буде видалено, адже його номер - це 0)', '', sep = "\n")

    list = [12,'барвінок','їжак',2,43,47,True,'колір',5,690,7,'привіт',9,101, False] #тут існують групи як чисел, так і булеві значення й строки

    list\_1 = []

    for el, q in enumerate(list):

     if el % 3:

        list\_1.append(q)

    print(list\_1)

    # print('2-ий варіант:')

    # # list = [12,'барвінок','їжак',2,43,47,True,'колір',5,690,7,'привіт',9,101, False] #тут існують групи як чисел, так і булеві значення й строки

    # # list\_2 = []

    # text = """Some people, when confronted with a problem, think

    # ... "I know, I'll use regular expressions."

    # ... Now they have two problems. Jamie Zawinski"""

    # text1=text.split

    # list(map(remove\_punctuation, text1))

    print('Вибачте, я не змогла зробити саме те, що хотіла, хоча багато дізналася про цю функцію/ з якоїсть причини жодний з поераторів ФП в мене не працює, а функції до них не підключаються','' ,sep = "\n")

    # print(text1)

    print('2-ий варіант: 2 та 3 пункт: використання оператора lambda() та reduce()')

    # s = ['їжак їсть',

    #          'кількість їжаків',

    #          'їжак - Павло', 'їсти',

    #          'спати',

    #          'їж та ак']

    # s\_new = reduce(lambda a, x: a + x.count('Їжак'), sentences, 0)

    # print(s\_new)

    print('Прикро, але тут та сама ситуація, хоча код написаний', '', sep = "\n")

    print('3 варіант',  'list = [12,барвінок,їжак,2,43,47,True,колір,5,690,7,привіт,9,101, False]', '', sep = "\n")

    print('4 варіант',  'list = [12,барвінок,їжак,2,43,47,True,колір,5,690,7,привіт,9,101, False]','', sep = "\n")

    print('3 та 4 варіанти я не змогла адаптувати під себе, а код вийшов дуже заплутаним. Покищо я не розумію як правильно треба працювати з цими оперторами. Я обов''язково це виправлю, бо я хочу, коли розберуся, або коли ми це пройдемо на програмуванні', '', sep = "\n")

## Текст програми 2

from itertools import product

print("Лабораторна робота номер 3: СТРУКТУРИ ДАНИХ: СПИСКИ, КОРТЕЖИ, МНОЖИНИ", "Роботу виконала: Єсипенко Є.Ю. ,КМ-11", "Варіант номер 8. ", "", '', sep = "\n")

print('Завдання №2: Множини', 'Маємо множини A = {1, 2, ee, ww, a, d} та B = {1, a, c, tt, 4, 3}. Знайдемо множину F, якщо F = ((A \ B) & (A ^ B))',

 sep = "\n")

A = {1, 2, 'ee', 'ww', 'a', 'd'}

B = {1, 'a', 'c', 'tt', 4, 3}

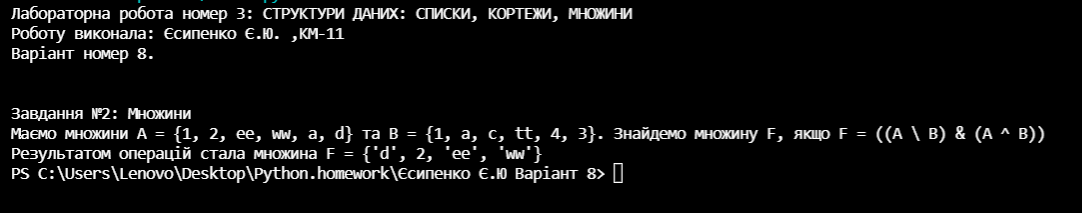
print("Множина F =", (A.difference(B)).intersection(A.symmetric\_difference(B)))

# Додаток Б

Скріншоти тестувань програм

Скріншот тестування програми 1

Скріншот тестування програми 2



# Відповіді на контрольні запитання

1. Що таке список, кортеж, множина.

Списки - впорядковані змінювані колекції об'єктів довільних типів. Майже як масив, але типи можуть відрізнятися.

Кортеж умовно можна назвати незмінним «списком», тому що до нього можна застосовувати багато функції для списків, крім зміни. Кортежі використовуються, коли ми хочемо бути впевнені, що елементи структури даних не будуть змінені в процесі роботи програми.

Множина – невпорядкована колекція незмінних, унікальних елементів. Множину можна створювати на основі списків.

1. Перерахуйте відмінності списку від кортежу

Їх відмінність полягає в тому, що списки можуть бути зміненими, на відміну від кортежу. За умови спроби змінити кортеж, автоматично створюється новий, до якого додаються зміни, що відносилися до першого.

1. Наведіть приклади створення списку, кортежу, множини.

a = [‘ ‘, ‘ ‘, ‘ ‘] --- список

b = (‘ ’, ‘ ‘, ‘ ‘) --- кортеж

c = {‘’, ‘ ’, ‘ ’} --- множина

1. Перерахуйте методи зміни списку

.pop() - видаляє елемент зі списку та повертає його значення

.append(x) - додає елемент у кінець списку

.extend() - додає список/ кортеж у кінець списку

.remove() - видаляє перший елемент із значенням у дужках зі списку

.reverse() - змінює порядок на протилежний

.sort() - розташовує елементи у порядку зростання

.clear() - очищує список

.insert(q , b) - вставляє елемент b перед q

1. Перерахуйте функції вищого порядку для роботи з послідовностями.

Mар(), zip(), filter(), reduce(), lambda()

1. Що повертає функція zip()?

Функція zip() на кожній ітерації повертає кортеж з елементыв послідовностей на однаковому розмыщенні.

1. Як кортеж перетворити у список.

Аби перетворити кортеж у список треба задати це рядком: list(a\_tuple).

1. Скільки разів елемент може входити до множини.

Один раз.

1. Чи може список бути елементом множини?

Не може. За умови введення списку до множини програма видасть помику

1. Як можна із множини зробити список.

Функцією list().

1. Чи можна у Python виконувати звичайні для математики операції над множинами?

Можна.