Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Програмування

Лабораторна робота №4

«Списки, кортежі, множини і діапазони в Python»

Виконав:

студент групи ІО-64

Андрійчук Д. А.

Залікова книжка №ІО 6401

Перевірив Новотарський М. А.

Київ 2016

**Мета**: вивчити способи створення списків, кортежів, множин та задавання діапазонів. Операції над списками, кортежами та діапазонами. Функції для перетворень списків, кортежів та множин.

**Завдання:**

1. Вивчити матеріал лекцій 11, 12, 13 та 14.

2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

**Короткі теоретичні відомості**:

Метод append () - додає один об'єкт у кінець списку. Метод змінює поточний список і нічого не повертає.

Оператор циклу:

while – дозволяє виконувати діюБ доки виконується умова.

for -дозволяє виконувати ті самі інструкції багаторазово.

Оператор розгалуження

– задають умови, за яких дія виконується

Функція len() – виводить кількість елементів.

Функція del – видаляє елементи по індексу

Функція randrange модуля randome – виводить довільний елемент діапазону.

Роздруківка фрагменту тексту програми

1. Випадковим чином створити два цілочисельних списки. Сформувати новий список, на парних місцях якого будуть елементи з непарними індексами з першого списку, а на непарних – з парними індексами з другого. Сформувати з нового списку вдвічі коротший список з випадковими елементами. Перетворити цей список на рядок.

**from** random **import** randrange # імпоруємо модуль рандом  
length\_of\_list = int(input(**"Ввведіть довжину списка: "**))  
list1 = [] #створюємо пустий список  
list2 = []   
list3 = []   
**while** len(list2) != length\_of\_list: #доки список не буде заповнено  
 list1.append(randrange(1000)) #до списків буде додаватись довільний  
 list2.append(randrange(100)) #елемент з діапазону від 0 до 999  
print(**"List #1:"**,list1) #вивід списків  
print(**"List #2:"**,list2)   
  
index = 0  
**while** index != length\_of\_list: #доки індекс не буде дорівнювати довжені спиків  
 **if** len(list3) == length\_of\_list: #якщо формування списку закінчено   
 **pass #** дія пропускається **else**: #якщо ні  
 list3.append(list2[index]) #до списка3 додається елемент списка2  
 index += 1 #індекс збільшується на 1  
  
 **if** len(list3) == length\_of\_list: #якщо формування списку закінчено  
 **pass #** дія пропускається **else**: #якщо ні  
 list3.append(list1[index]) #до списка3 додається елемент списка1  
 index += 1 #індекс збільшується на 1  
print(**"List #3:"**,list3) # вивід спискa  
  
**while** len(list3) != length\_of\_list//2: #доки список не скоротиться вдвічі  
 **del** list3[randrange(len(list3))] #буди видалятись довільний елемет  
print(**"List #4: {}"**.format(list3)) #Вивід списка

2. Згенерувати матрицю Amn ( , ) розмірністю m n × з випадкових елементів, що є цілими числами від 0 до 10. Видалити рядки, які містять елементи, що повторюються.

**from** random **import** randrange # імпоруємо модуль рандом  
m = int(input(**"Кількість рядків"**))  
n = int(input(**"Кількість стовпців"**))  
list1 = [] #створюємо пустий список  
**for** i **in** range(m): #для кожного числа по кількості рядків  
 list1.append([]) #створити піделемент  
 **for** elem **in** range(n): #для кожного числа по к-сті стовпців  
 list1[i].append([]) #створити піделемент піделемента  
 list1[i][elem] = randrange(11)#та дати йому значення довільного цілог  
**for** i **in** range(len(list1)): #для кількості піделементів  
 print(list1[i]) #вивести піделементи  
  
  
deleted\_list\_elem = [] # створюємо пустий список  
**for** i **in** range(m): #для кожного числа по кількості рядків  
 set\_list1 = set(list1[i]) #створити множину з піделемента списку  
 **if** len(set\_list1) != len(list1[i]):#якщо к-ть елем. множини не рівна  
 deleted\_list\_elem.append(i) #к-ті елем. підсписку то його індекс #додається до списку deleted\_list\_elem.reverse() #перевертаємо список що містить індекси #майбутніх видалених елементів  
**for** i **in** range(len(deleted\_list\_elem)):#для всіх елементів списку  
 **del** list1[deleted\_list\_elem[i]] #видаляємо елементи по індексу 2 списка  
  
print(**"\n"**) #для краси друкуємо 2 пустих рядка  
**for** i **in** range(len(list1)): # для кількості піделементів  
 print(**"%s"** % list1[i]) # вивести піделементи

Роздруківка результатів виконання програми з контрольним прикладом.

Кількість рядків4

Кількість стовпців4

[4, 2, 9, 0]

[2, 8, 10, 7]

[2, 5, 6, 4]

[0, 9, 2, 0]

[4, 2, 9, 0]

[2, 8, 10, 7]

[2, 5, 6, 4]

Висновок. За допомогою типів, що підтримують ітерацію, можна перебрати відразу величезну кількість елементів написавши лише декілька рядків коду. Однак потрібно мати гарне розуміння, як його можна реалізувати на практиці, адже воно зазвичай має складну структуру, а правила виконання дій потрібно записувати з розрахунком на зміну об’єкта, що обробляється.