

Лабораторная работа № 11

Работа со списками в Python

Цель работы: получение практических навыков программирования в задании переменных списочного типа и выполнении простейших операций над ними.

Оборудование: ПЭВМ.

Общие положения

Большинство программ работает не с отдельными переменными, а с набором переменных. Для хранения таких данных можно использовать структуру данных, называемую в Python *список* (в большинстве же языков программирования используется другой термин «массив»).

Списки в Python (list) - упорядоченные изменяемые коллекции объектов произвольных типов.

В отличие от массивов, включающих в себя лишь однотипные элементы, списки не привязаны к определенной разновидности данных, а также не имеют жестких ограничений, связанных с их размером.

Список представляет собой последовательность элементов, заключенных в квадратные скобки [],отделяющиеся друг от друга с помощью запятой, пронумерованных от 0, как символы в строке.

Создать список можно несколькими способами:

1. Получение списка через присваивание конкретных значений, используя конструкцию []

```
l = [] # это пустой список
l = [25, 755, -40, 57, -41] # список целых чисел
l = [1.13, 5.34, 12.63, 4.6, 34.0, 12.8] # список из дробных чисел
l = ["Sveta", "Sergei", "Ivan", "Dasha"] # список из строк
l = ["Москва", "Иванов", 12, 124] # смешанный список
l = [[0, 0, 0], [1, 0, 1], [1, 1, 0]] # список, состоящий из списков
l = ['s', 'p', ['isok'], 2] # список из значений и списка
```

2. Создание списка при помощи функции **List()**

```
l = list() # пустой список
l = list('spisok') # 'spisok' - строка
print(l) #['s', 'p', 'i', 's', 'o', 'k'] - результат - список
```

3. Создание списка при помощи функции **Split()**

```
stroka = "Hello, world" # stroka - строка
l = stroka.split(",") # l - список
print(l) # ['Hello', ' world']
```

4. Генераторы списков

```
# список из 10 элементов, заполненный единицами
l = [1]*10
# список l = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
l = [i for i in range(10)]
# список l = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
c = [c * 3 for c in 'list']
print(c) # ['lll', 'iii', 'sss', 'ttt']
```

```
from random import randint
l = [randint(10,80) for x in range(10)]
# 10 чисел, сгенерированных случайным образом в диапазоне (10,80)
```

Каждый элемент списка имеет присвоенный ему индекс. Важно отметить, в Python индекс первого элемента в списке - 0.

```
z = [3, 7, 4, 2] # создаем список
# обращение к первому элементу списка с индексом 0
print(z[0]) # 3
```

Python поддерживает отрицательную индексацию. Отрицательная индексация начинается с конца.

```
z = [3, 7, 4, 2] # создаем список
# выведите последний элемент списка
print(z[-1]) # 2
```

Для того, чтобы изменить значение определенного элемента, используют оператор присваивания, обращаясь к элементу по индексу.

```
z = [3, 7, 4, 2] # создаем список
z[1] = 5 # замена 2 элемента списка
print(z) # [3, 5, 4, 2]
```

Срезы(slice) списка. Срезы хороши для получения подмножества значений списка.

```
z = [3, 7, 4, 2] # создаем список
print(z[0:2]) # Вывод элементов с индексом от 0 до 2 (не включая 2)
[3, 7]
print(z[:3]) # Все, кроме индекса 3
[3, 7, 4]
print(z[1:]) # начиная с индекса 1 до конца списка
[7, 4, 2]
```

Для **ввода** элементов списка используется цикл **for**:

```
L = [ int(input()) for i in range(N) ]
```

Функция *int* здесь используется для того, чтобы строка, введенная пользователем, преобразовывалась в целые числа.

Список можно **выводить** целиком и поэлементно:

```
# вывод целого списка (массива)
print (L)
# поэлементный вывод списка (массива)
for i in range(N):
```

```
print ( L[i], end = " " )
```

К спискам применяются **операции соединения и повторения**:

1) Операция конкатенации:

```
l = [1, 3] + [4, 23] + [5] # l = [1, 3, 4, 23, 5]
```

или

```
a=[33, -12, 'may']
```

```
b=[21, 48.5, 33]
```

```
print(a+b) # [33, -12, 'may', 21, 48.5, 33]
```

2) Операция повторения:

```
[[0,0],[0,1],[1,1]] * 2 # [[0, 0], [0, 1], [1, 1], [0, 0], [0, 1], [1, 1]]
```

У списков Python есть разные методы, которые помогают работать со списками (таблица 1).

Таблица 1 - Методы списков

Операция	Описание	Пример
x in a	Проверка, что x содержится в a	5 in [2, 3, 5]
x not in a	Проверка, что x не содержится в a То же, что и not (x in a)	5 not in [2, 3, 6]
a + a2	Конкатенация списков, то есть новый список, в котором сначала идут все элементы a, а затем все элементы a2	[2, 4] + [5, 3] == [2, 4, 5, 3]
a * k	Список a, повторенный k раз	[2, 3] * 3 == [2, 3, 2, 3, 2, 3]
a[n]	n-й элемент списка, отрицательные n — для отсчёта с конца	[2, 3, 7][0] == 2 [2, 3, 7][-1] == 7
a[start:stop:step]	Срез списка	[2, 3, 7][:2] == [2, 3]
len(a)	Длина списка	len([2, 3, 7]) == 3
max(a)	Максимальный элемент списка	max([2, 3, 7]) == 7
min(a)	Минимальный элемент списка	min([2, 3, 7]) == 2
sum(a)	Сумма элементов списка	sum([2, 3, 7]) == 12
a.index(x)	Индекс первого вхождения x в a (вызовет ошибку, если x not in a, то есть если x отсутствует в a)	[2, 3, 7].index(7) == 2
a.count(x)	Количество вхождений x в a	[2, 7, 3, 7].count(7) == 2
a.append(x)	Добавить x в конец a	a = [2, 3, 7] a.append(8) a == [2, 3, 7, 8]
a.extend(a2)	Добавить элементы коллекции a2 в конец a	a = [2, 3, 7] a.extend([8, 4, 5]) a == [2, 3, 7, 8, 4, 5]
del a[n]	Удалить n-й элемент списка	a = [2, 3, 7] del a[1] a == [2, 7]
Del a[start:stop:step]	Удалить из a все элементы, попавшие в срез	a = [2, 3, 7] del a[:2] a == [7]
a.clear()	Удалить из a все элементы (то же, что del a[:])	a.clear()

a.copy()	Копия a (то же, что и полный срез a[:])	b = a.copy()
a += a2 a *= k	Заменить содержимое списка на a+a2 и a*k соответственно	
a.insert(n, x)	Вставить x в a на позицию n, подвинув последующую часть дальше	a = [2, 3, 7] a.insert(0, 8) a == [8, 2, 3, 7]
a.pop(n)	Получить n-й элемент списка и одновременно удалить его из списка. Вызов метода без аргументов равносителен удалению последнего элемента: a.pop() == a.pop(-1)	a = [2, 3, 7] a.pop(1) == 3 a == [2, 7]
a.remove(x)	Удалить первое вхождение x в a, в случае x not in a — ошибка	a = [2, 3, 7] a.remove(3) a == [2, 7]
a.reverse()	Изменить порядок элементов в a на обратный (перевернуть список)	a = [2, 3, 7] a.reverse() a == [7, 3, 2]
a.sort()	Отсортировать список по возрастанию	a = [3, 2, 7] a.sort() a == [2, 3, 7]
a.sort(reverse=True)	Отсортировать список по убыванию	a = [3, 2, 7] a.sort(reverse = True) a == [7, 3, 2]
bool(a)	Один из способов проверить список на пустоту (возвращает True, если список непустой, и False в противном случае)	

Сортировка списка «пузырьком»

```

n = int(input()) # количество элементов
a = []
for i in range(n): # считываем элементы списка
    a.append(int(input()))
# Сортировка пузырьком:
for i in range(n - 1):
    for j in range(n - 1 - i):
        if a[j] > a[j + 1]:
            a[j], a[j + 1] = a[j + 1], a[j]
print(a)

```

Порядок выполнения работы

Задание 1. Составить программу, которая считывает сначала количество оценок, потом по очереди сами эти оценки, затем выводит их же в том же порядке (используем список). Найдите среднюю оценку за урок.

Задание 2. Составить программу, согласно полученному варианту задания. Ввод данных сопровождать соответствующими запросами, а вывод - наименованиями выводимых переменных.

Вариант	Задание
1	Для списка A , состоящего из 20 элементов, вычислить количество отрицательных элементов списка.
2	Для списка B , состоящего из 20 элементов, вычислить сумму положительных элементов списка.
3	Для последовательности, состоящей из 16 элементов, создать список из значений под четными номерами этой последовательности.
4	Для списка X , состоящего из 15 элементов, найти наибольший элемент списка и его порядковый номер.
5	Для списка Y , состоящего из 15 элементов, найти наименьший элемент списка и его порядковый номер.
6	Для списка Z , состоящего из 20 элементов, найти наименьший из положительных элементов.
7	Для списка X , состоящего из 14 элементов, вывести на печать номера элементов, удовлетворяющих условию $0 < X(i) < 1$.
8	Информация о температуре воздуха за март задана в виде списка. Определить сколько раз температура опускалась ниже 0°C .
9	Дана последовательность целых чисел a_1, a_2, \dots, a_{18} . Создать список из четных чисел этой последовательности. Если таких чисел нет, то вывести сообщение об этом факте.
10	При поступлении в вуз абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. В списке A записаны оценки 25 экзаменуемых, полученные на первом экзамене. Подсчитать, сколько человек не допущено ко второму экзамену.

Задание 3. Составить программу, согласно полученному варианту задания. Ввод данных сопровождать соответствующими запросами, а вывод - наименованиями выводимых переменных.

Вариант	Задание
1	В списке из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.
2	В списке из 10 целых чисел найти наименьший элемент и поменять его местами с последним элементом.
3	В списке из 15 вещественных чисел найти наибольший элемент и поменять его местами с последним элементом.
4	В списке из 12 вещественных чисел найти наименьший элемент и поменять его местами с первым элементом.

5	Упорядочить по убыванию список, содержащий 10 вещественных чисел.
6	Упорядочить по возрастанию список, содержащий 15 целых чисел.
7	Дан список целых чисел, содержащий 10 элементов, записать в этот же список сначала все положительные числа, а затем все отрицательные, сохраняя порядок их следования.
8	Дан список вещественных чисел, содержащий 12 элементов, записать в этот же массив сначала все отрицательные числа, а затем все положительные, сохраняя порядок их следования.
9	Дан список целых чисел, содержащий 14 элементов. Переставить его элементы в обратном порядке.
10	Последовательность a_1, a_2, \dots, a_{12} состоит из нулей и единиц. Поставить в начало этой последовательности нули, а затем единицы.

Содержание отчета

1. Постановка задачи.
2. Текст программы.
3. Результаты выполнения программы.