

Тема 7 КОРТЕЖИ И СПИСКИ

Практическая работа №5

7.1 Цель практической работы

Приобретение практических навыков разработки и программирования вычислительного процесса с использованием списков.

7.2 Теоретические сведения

Списки в Python функционируют подобно кортежам, с той разницей, что являются изменяемой последовательностью. Таким образом, если такие действия, как добавление нового элемента, удаление элемента, сортировка и др., в кортеже невозможны, то методы работы со списком позволяют это сделать.

Точно так же, как и элементы кортежа, элементы списка содержат лишь ссылку на объект, поэтому в списки могут входить разные типы данных. Данный подход отличается от обработки традиционных массивов в других языках программирования, хотя сходство есть в методах обработки списков в Python с методами обработки массивов.

Рассмотрим синтаксис объявления списков.

Имя списка = [элемент 1, элемент 2, ...элемент N]

Например,

```
spisok=[1, 2, 3, 4, 5]
```

Типичными задачами при работе со списками являются:

- определение факта наличия в них заданного элемента;
- отбор элементов, удовлетворяющих определенным условиям.

В обоих случаях используется циклическое сравнение элементов списка с заданным образцом. Для определения факта наличия заданного образца в списке достаточно единственного совпадения, после чего дальнейший просмотр прекращается. Если условие отбора может выполняться для нескольких элементов списка, то необходим просмотр всего списка до конца.

Все рассмотренные выше приемы, позволяющие манипулировать с данными, представленными в кортежах, действительны и для списков, однако их набор может быть существенно расширен в силу изменяемости списков. В частности, можно удалять и добавлять элементы списка, сортировать их.

Пользователь может создавать список, размещая его элементы, которые необходимо обработать, в квадратных скобках (согласно синтаксису, показанному выше), а может генерировать случайным образом. Рассмотрим

эти способы подробнее на примере задачи «Найти сумму элементов списка». Ранее в этой теме мы рассмотрели решение точно такой же задачи, но для работы с кортежами. Здесь и далее будет полезно сравнить способы обработки последовательностей в Python.

```
sum=0
spisok=[1, 2, 3, 4, 5]
print(" Список ")
for i in spisok:
    print(i, end=" ")
for i in range(5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Рассмотрим возможность генерации списков случайным образом. Ее можно осуществить несколькими способами. Во-первых, в известной нам функции **range** можно указать начальное и конечное значения списка, а функция **list** возвратит список. Тогда код предыдущей программы будет выглядеть так:

```
sum=0
spisok=list(range(1, 21))
print(" Список ")
print(spisok)
for i in range(1, 20, 1):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Можно воспользоваться функцией **sample**, которая из исходной последовательности элементов списка будет возвращать указанное пользователем количество элементов. При этом необходимо воспользоваться модулем **random**, подключив его с помощью инструкции **import**. Как показано в нижеследующей программе, исходный список состоит из десяти элементов. Затем оператором **chislo=random.sample(spisok,5)** генерируются пять элементов из исходного списка.

```
import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
chislo=random.sample(spisok, 5)
print(chislo, end=" ")
for i in range(0, 5):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)
```

Скомбинируем предыдущие коды. Вместо заранее заданного списка сгенерируем его, воспользовавшись функцией **range**, а затем функцией **sample** (сделаем параметром число 7), тем самым ограничив количество элементов списка (до семи).

```
import random
sum=0
chislo=random.sample(range(100), 7)
print(chislo, end=" ")
for i in range(0, 7):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)
```

В модуле **random** есть функция **shuffle** с помощью, которой можно перемешать список случайным образом.

```
import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
random.shuffle(spisok)
print(spisok, end=" ")
for i in range(0, 5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Кроме рассмотренных способов создания списков в программе, можно применить прием динамического создания списка. Как показано в нижеприведенном коде, сначала объявляется пустой список оператором **sp=[]**. Затем у пользователя запрашивается количество элементов списка. В цикле пользователь начинает вводить элементы списка, а метод **append()** позволяет добавить их в список.

```
sp=[]
n=int(input("\n Введите количество элементов списка "))
for i in range(n):
    chislo=int(input("\n Введите число "))
    sp.append(chislo) # Добавляем элементы списка
for i in range(n): # Выводим элемента списка
    print(sp[i], end=" ")
```

Таким образом, рассмотрены основные способы создания списков в Python. Можно задавать списки, комбинируя рассмотренные выше основные способы создания списков. Для работы со списками в Python предусмотрены методы, некоторые из которых приведены в табл. 1.

Таблица 1 - Методы работы со списками

| Метод | Описание метода |
|--|--|
| <code>spisok.append(x)</code> | Добавляет значение x в конец списка |
| <code>spisok.insert(i, x)</code> | Вставляет значение x в позицию i |
| <code>spisok.extend(spisok1)</code> | Расширяет список, добавляя в конец все элементы списка <code>spisok1</code> |
| <code>spisok.remove(x)</code> | Удаляет первый элемент в списке, имеющий значение x |
| <code>spisok.pop(i)</code> | Возвращает значение в позиции номер i и одновременно удаляет его из списка. Если аргумент не передан, возвращается и удаляется последний элемент списка |
| <code>spisok.count(x)</code> | Возвращает количество элементов со значением x |
| <code>spisok.sort([reverse = True])</code> | Сортирует элементы по возрастанию. Параметр reverse является необязательным и принимает логические значения. Если передать значение True , то список сортируется по убыванию |
| <code>spisok.reverse()</code> | Возвращает обратный список |

7.3 Пример выполнения задания на практическую работу

Задача 1. В список, сгенерированный случайным образом, добавить введенный пользователем элемент.

Решение. Как видно из нижеприведенной программы, сначала происходит генерация списка, затем запрашивается число, и с помощью метода **append** происходит добавление числа в конец списка.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число "))
spisok.append(chislo) # Добавляем введенное число в конец списка
print(spisok, end=" ")
```

Задача 2. В список, сгенерированный случайным образом, добавить введенный пользователем элемент на указанную позицию.

Решение. В нижеприведенной программе, сначала происходит генерация списка, затем запрашивается число и номер позиции, на которую следует добавить число. Затем с помощью метода **insert()** с соответствующими параметрами **poz, chislo** выполняется вставка числа на указанную позицию.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число ")) # Вводим число
poz=int(input("\n Введите позицию ")) # Вводим позицию для добавления
введенного выше числа в список
spisok.insert(poz, chislo) # Добавляем введенное число
```

```
print(spisok, end=" ")
```

Задача 3. Имеются два списка, сгенерированные случайным образом. Добавьте в конец первого списка все элементы второго списка.

Решение. Для начала реализуем сгенерированные случайным образом два списка **spisok1** и **spisok2**. За счет использования метода **extend()**, примененного к первому списку, мы расширяем его элементами второго списка.

```
import random
spisok1=random.sample(range(100), 7)
print("\n Первый список ")
print(spisok1, end=" ")
spisok2=random.sample(range(100), 5)
print("\n Второй список ")
print(spisok2, end=" ")
spisok1.extend(spisok2) # Добавляем к первому списку элементы второго
списка
print("\n Объединенный список ")
print(spisok1, end=" ")
```

Задача 4. Из заранее сформированного списка следует удалить элемент, введенный пользователем.

Решение. Исходный список будет содержать пять элементов. Пользователь вводит значение элемента, подлежащего удалению. Метод **remove()**, примененный к списку, осуществляет указанные действия.

```
spisok=[8, 9, -7, 7, 0]
for i in spisok:
    print(i, end=" ")
chislo=int(input("\n Введите число "))
spisok.remove(chislo)
print(spisok, end=" ")
```

Задача 5. Список, сгенерированный случайным образом, отсортируйте по возрастанию.

Решение. Метод **sort()**, применяемый в Python для сортировки списка, имеет необязательный параметр **reverse**. В нижеприведенном коде он выставлен в положение **False**, что дает возможность отсортировать список по возрастанию.

```
import random
spisok=random.sample(range(100), 7)
print("\n Сгенерированный список ", spisok, end=" ")
```

```
spisok.sort(reverse=False)
print("\n Отсортированный список ", spisok, end=" ")
```

7.4 Задание на практическую работу

Реализовать программно одно из следующих заданий.

Варианты заданий

| № вар. | Задание |
|--------|--|
| 1 | Сформируйте список $x = [-1.5, 0.1, 12, 0, -2.2, 0.5, -1, 0.3]$. Вычислите и выведите сумму и количество элементов списка, принадлежащих отрезку $[0, 1]$. |
| 2 | Сформируйте список $a = [5, -2, 0, 3, 4, 12, 7]$. Вычислите и выведите среднее арифметическое значение положительных элементов списка. |
| 3 | Сформируйте список $x = [-1.5, 0, 0.8, 2.2, 3, 0.5, 0.1]$. Перепишите элементы списка, принадлежащие отрезку $[-1, 1]$, в список y и выведите его. |
| 4 | Сформируйте списки $a [5]$ и $b [5]$, состоящие из произвольных чисел и определите, в каком из списков больше положительных элементов. Выведите новый список c , состоящий из положительных элементов списков a и b |
| 5 | Сформируйте список $z = [-2, 0, 3.5, 7, -12, 5, -1, 3]$. Расположите в списке r сначала положительные, а затем неположительные элементы списка z . Выведите список r . |
| 6 | Сформируйте список $a = [2.35, -4.15, 0, -3.1, 7.8, 6.3, -3.05, 1.5]$. Найдите и выведите среднее геометрическое положительных элементов списка a и индекс элемента, наиболее близкого среднему геометрическому. |
| 7 | Сформируйте список $a [10]$, элементы которого должны быть сформированы по правилу $a[i+1] = a[i] + i^2$; $a[1] := 2$ и вычислите среднее арифметическое его элементов. |
| 8 | Сформируйте список $y = [2.5, 4.9, 10.2, -7.12, 3.1, -2, 6]$. Сформируйте из него новый список z , элементами которого будут являться положительные элементы списка y , и упорядочите по возрастанию список z . |

| | |
|----|--|
| 9 | Сформируйте список $y = [-6.3, 0.8, 12, -4, 13, 2.5, 7, 8, -9, 10]$. Найдите и выведите максимальный и минимальный элементы списка. Поменяйте их местами. Выведите полученный список y . Введите списки $x = [4.1, 16, 0, -3.2, 12]$ и $u = [4, 5.1, 6]$. |
| 10 | Введите списки $x = [4.1, 16, 0, -3.2, 12]$ и $u = [4, 5.1, 6]$. Объедините их в один список z , поместив элементы списка u между третьим и четвертым элементами списка x . Выведите список z . |
| 11 | Введите списки $z = [0, 1.6, 6.4, 3.8, -7, 1, -2]$ и $a = [5, 4, 6.4, 1]$. Найдите среди элементов списков a и z два одинаковых элемента с наименьшими индексами и выведите их значения и индексы. |
| 12 | Сформируйте список $n = [3, 5, 7, 9, 11, 13, 15]$. Переставьте элементы списка n в обратном порядке и выведите его. |
| 13 | Сформируйте список $m = [-1, 0, 10, -3, -5, 6, -2, 3, 4]$. Сформируйте и выведите список p , элементами которого являются индексы положительных элементов списка m . |
| 14 | Сформируйте список $l = [13, 4, -2, 6, 7, -1, -5, 2, -3, 4]$. Вычислите и выведите $m[0]n[0] + m[1]n[1] + \dots + m[k]n[k]$, где $m[0], m[1], \dots, m[p]$ - отрицательные элементы списка l , взятые в порядке их следования; $n[0], n[1], \dots, n[q]$ - положительные элементы списка l , взятые в обратном порядке их следования; $k = \min[p, q]$. |
| 15 | Сформируйте список $k = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. Переставьте пары элементов $k[i], k[i+1]$, где $i = 0, 2, 4, 6, 8$. Выведите полученный список. |
| 16 | В списке, состоящем из 15 вещественных элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите: 1) сумму отрицательных элементов списка; 2) произведение элементов списка, расположенных между максимальным и минимальным элементами. |
| 17 | Сформируйте список $x = [6, 3.8, -9.3, 2.87, 16, 5, 0.2, -3.1, 1, 10]$. Сформируйте и выведите список Y , вычислив его по правилу: $\begin{cases} Y_0 = \sqrt{X_0}, Y_1 = \sqrt{X_0 + X_1}, Y_2 = \sqrt{X_0 + X_1 + X_2}, \text{ если подкор. не отр.} \\ Y_i = 0, \text{ в противном случае} \end{cases}$ |
| 18 | Сформируйте список $m = [6, 10, 7, 14, 12, 12, -2, 3, -9, 6, -10]$. Вычислите и выведите количество и сумму тех элементов списка, которые делятся на 2 и не делятся на 3. |
| 19 | Сформируйте список $m = [14, 6, 3, 0, 7, 12, -3, 1, 5, 2]$. Вычислите и выведите произведение элементов списка, кратных 3. |

| | |
|----|--|
| 20 | Сформируйте список $l = [7, 6, 15, 17, 12, -12, 4, 0, -10, -22]$. Замените в списке нулями те элементы, модуль которых при делении на 5 дает в остатке 2. Выведите полученный список. |
| 21 | Сформируйте список $k = [1, 2, 3, 4, 6, 5, 8, 9, 10]$. Если элементы списка образуют возрастающую последовательность выведите сообщение "ДА"; в противном случае - сообщение "НЕТ". |
| 22 | Введите упорядоченный список $q = [1.5, 2, 3.1, 4.2, 6, 7.5, 8.3, 9]$. Удалите из списка элемент с индексом 4 и вставьте элемент с вводимым значением s так, чтобы не нарушилась упорядоченность. Выведите полученный список. |
| 23 | В списке, состоящем из 15 вещественных элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите: 1) сумму положительных элементов списка; 2) произведение элементов списка, расположенных между максимальным по модулю и минимальным по модулю элементами. |
| 24 | Сформируйте список $b = [15.1, 0.8, 32.3, 7.5, -1.5, 2.4, 6.3, 15.5]$. Подсчитайте и выведите среднее арифметическое значение элементов списка и количество элементов, больших среднего арифметического |
| 25 | В списке, состоящем из 20 вещественных элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите: 1) номер минимального элемента списка; 2) сумму элементов списка, расположенных между первым и вторым отрицательными элементами. Преобразуйте список таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом все остальные. |
| 26 | В списке, состоящем из 20 целых элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите: 1) номер максимального элемента списка; 2) произведение элементов списка, расположенных между первым и вторым нулевыми элементами. Преобразуйте список таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине - элементы, стоявшие в четных позициях. |
| 27 | Введите списки $x = [-6, 0.5, 0.12, 13, -10.1]$ и $y = [13, 2.1, 14, 6, -2]$. Создайте список r [10] такой, в котором элементы с нечетными номерами являются элементами списка x , с четными номерами - списка y . Выведите список r . |
| 28 | Введите списки $a = [-2, 0, 3.1, 4.6, 10]$, $b = [4, 7, 9.1, 12, 63]$. Сформируйте из элементов списков a и b список c . Упорядочите его по возрастанию и выведите список c . |

| | |
|----|--|
| 29 | <p>В списке, состоящем из 20 целых элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите:</p> <ol style="list-style-type: none"> 1) произведение элементов списка с четными номерами; 2) преобразуйте список таким образом, чтобы сначала располагались все положительные элементы списка, а потом - все отрицательные (элементы равные нулю, считать положительными). |
| 30 | <p>В списке, состоящем из 20 элементов целого типа (значения элементов списка задайте, используя генератор случайных чисел), вычислите:</p> <ol style="list-style-type: none"> 1) минимальный элемент списка; 2) сумму элементов списка, расположенных между первым и последним положительными элементами. <p>Преобразуйте список таким образом, чтобы сначала располагались все элемент равные нулю, а потом — все остальные.</p> |
| 31 | <p>Сформируйте список $x = [-1, 2, -3, 4, -5, 0, 6, 8, 9]$. Перепишите в список y подряд положительные элементы списка x. Выведите список y.</p> |
| 32 | <p>Сформируйте список $g = [-3.1, 2.8, 5, 7.7, -7.5, 7.6, 3, 0]$. Определите и выведите минимальный элемент списка и его номер.</p> |
| 33 | <p>Сформируйте список $b = [7.35, 0.12, -7, 3.12, 2.87, -4.12, 5.32, 0, 6.5]$. Определите и выведите максимальный элемент списка и его номер.</p> |
| 34 | <p>В списке, состоящем из 20 вещественных элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите:</p> <ol style="list-style-type: none"> 1) максимальный элемент списка; 2) сумму элементов списка, расположенных до последнего положительного элемента. |
| 35 | <p>В списке, состоящем из 20 вещественных элементов (значения элементов списка задайте, используя генератор случайных чисел), вычислите:</p> <ol style="list-style-type: none"> 1) сумму элементов списка с нечетными номерами; 2) сумму элементов списка, расположенных между первым и последним отрицательными элементами. |

7.5 Требования к оформлению отчета по практической работе

При оформлении отчета по практической работе рекомендуется следующая структура и последовательность элементов:

- титульный лист;
- название лабораторной работы;
- цель лабораторной работы;
- индивидуальное задание (по вариантам) на лабораторную работу;

- краткие комментарии по выполнению индивидуального задания и структурная схема алгоритма решения задачи;
- необходимый программный код индивидуального задания; – результаты работы программы; – выводы.

Индивидуальное задание на лабораторную работу содержит полный текст индивидуального задания, полученного у преподавателя, описание алгоритма выполнения индивидуального задания, структурную схему алгоритма решения задачи.

Необходимый программный код индивидуального задания содержит полный текст кода программы, разработанный студентом.

Результаты работы программы обычно содержат копии окон работы программы.

7.6 Контрольные вопросы для защиты лабораторной работы

- 7.6.1. Напишите синтаксис объявления списков.
- 7.6.2. Какие возможности языка Python используются для генерации списков?
- 7.6.3. Перечислите и поясните основные методы работы со списками.
- 7.6.4. Каким образом осуществляется доступ к каждому элементу списка при его обработке.
- 7.6.5. Поясните, каким образом осуществляется обмен значений элементов списка. Приведите примеры.