ЛАБОРАТОРНАЯ РАБОТА № 5
СПИСКИ В ЯЗЫКЕ PYTHON.

Время выполнения – 4 часа.

Цель работы: ознакомиться с понятием «список» в языке программирования Python.

Задачи работы:

1. Изучить:

– понятие «список» в языке программирования Python;

– способы задания списка;

– основные функции по работе со списками в языке программирования Python.

2. Научиться:

– создавать списки;

– выводить элементы списка на экран.

Перечень обеспечивающих средств:

Для выполнения работы необходимо иметь компьютер с установленной операционной системой семейства Windows, компьютер, проектор, интерактивная доска.

Общие теоретические сведения

Список в языке Python представляет собой структуру для хранения объектов различных типов. Элементы одного списка могут иметь одинаковый тип, но допускаются и смешанные типы. Элементы списка заключаются в квадратные скобки. Все элементы проиндексированы (пронумерованы), начиная с 0 (Таблица 1).

Таблица 1 – Примеры списков в Python

Список	Описание
L=[]	Пустой список
L=[5, 10, 15, 20]	Четыре элемента с индексами 0..3
L=[‘pqr’,[‘abc’,’xyz’]]	Вложенные списки

Списки в Python, с одной стороны, похожи на массивы в других языках программирования, например Pascal, но, с другой стороны, имеется ряд отличий. Например, Python размещает элементы списка в памяти, затем размещает указатели на эти элементы. Таким образом, список в Python – это массив указателей.

Опишем ряд существенных свойств списков в языке Python. Можно обращаться к элементу списка по его индексу, начиная с 0. В языке Python также используют отрицательную индексацию, которая начинается с индекса –1, при этом индекс –1 относится к последнему элементу списка. В таблице 2 представлены различные виды индексации элементов списка.

Таблица 2 – Варианты индексации списков в Python

Список L	8	– 2	6	1	0	2
Индекс	0	1	2	3	4	5
Отрицательный индекс	– 6	– 5	– 4	– 3	– 2	– 1

Например, элемент L[3] равен 1, а элемент L[–4] равен 6.

Список в языке Python может иметь переменную длину. Например, можно удалять элементы списка. Оператор del list[i] удаляет из списка list элемент с индексом i, del L[2] удаляет элемент с индексом 2.

Далее рассмотрим способы задания списка в языке Python.

Можно непосредственно задать список, перечислив его элементы в квадратных скобках через запятую:

```
l=[4,8,10]
```

Также можно использовать функцию range, описанную ранее:

```
a=list(range(20))
```

В результате сформируется список значений от 0 до 19. Для задания элементов списка случайным образом можно использовать функцию randint из модуля random. Например:

```
from random import randint
a=[randint(-10,10) for i in range(20)]
print(a)
```

В данном примере формируется список из 20 элементов, заданных случайным образом.

Ещё один вариант – задать список с клавиатуры:

```
a=[input() for i in range (10)]
```

Для формирования списка с клавиатуры можно использовать следующий вариант:

```
a=[]
n=int(input())
for i in range(n):
k=int(input())
a.append(k)
print(a)
```

Для получения доступа к элементам списка можно использовать следующие конструкции:

```
for x in ["abc", "cde", "efg"]:
print(x)
```

или

```
l=["abc", "cde", "efg"]
for x in l:
print(x)
```

При работе со списками в языке Python часто используют срезы. Эти конструкции нужны, чтобы обрезать список, взяв лишь те элементы, которые нам требуются. Срезы работают по следующей схеме:

```
list[начало: конец: шаг]
```

Здесь:

– начало указывает, с какого элемента нужно начать (по умолчанию равно 0);

– конец указывает, по какой элемент берутся элементы (по умолчанию равен длине списка);

– шаг определяет, с каким шагом берутся элементы, например каждый второй или третий (по умолчанию каждый первый).

Приведём примеры работы со срезами.

Пример выборки из списка каждого второго элемента, начиная со второго:

```
l=[2,4,6,8,10,12,14,16]
print(l[2::2])
```

Пример выборки из списка каждого третьего элемента:

```
l=[2,4,6,8,10,12,14,16]
print(l[::3])
```

Пример изменения существующего списка. В списке остаются только элементы со второго по шестой:

```
l = [2,4,6,8,10,12,14,16]
l = l[1:6:1]
    print(l)
```

В языке Python доступно достаточно много встроенных функций для обработки списков. Некоторые из них описаны в таблице 3. Квадратные скобки в синтаксисе функций означают необязательные элементы.

Таблица 3 – Основные функции (методы) обработки списков

Функция	Описание
append(x)	Добавляет элемент x в конец списка
clear()	Очищает список
copy()	Возвращает копию списка
count(x)	Возвращает количество элементов списка со значением x
extend(L)	Расширяет список путем добавления в него всех элементов списка L
index(x[,start[,end]])	Возвращает индекс первого элемента со значением x (при этом поиск ведётся от start до end; если start и end не указываются, то начиная с нулевой позиции)
insert(i,x)	Вставляет в список на i-ю позиция значение x

pop([i])	Удаляет из списка i-й элемент и возвращает его. Если индекс не указан, удаляется последний элемент
remove(x)	Удаляет первый элемент в списке, имеющий значение x. Возвращает ValueError, если такого элемента не существует
reverse()	Переворачивает список
sort()	Сортирует список

Поскольку данные функции являются методами класса «список», то обращение к ним имеет вид: <имя списка>. <имя метода>

Рассмотрим несколько примеров по работе с описанными функциями.

Пример вывода на экран индекса элемента, равного 4. Результатом работы будет 1:

```
l=[2,4,6,8,10,12,14,16]
print(l.index(4))
```

Пример вывода количества элементов в списке, равных 12. В данном случае это один элемент. В конец списка добавляется новый элемент – 0:

```
l=[2,4,6,8,10,12,14,16]
print(l.count(12))
l=[2,4,6,8,10,12,14,16]
l.append(0)
print(l)
```

Пример расширения списка l за счет добавления элементов списка z:

```
l=[2,4,6,8,10,12,14,16]
z=[3,7,10]
l.extend(z)
print(l)
```

Практические задания (задания разбираются на практических занятиях с преподавателем)

1. Задать числовой список случайным образом. Заменить значения элементов в списке на их квадраты;
2. Задан числовой список, вывести на экран те значения списка, которые встречаются в нём более одного раза.
3. Даны два числовых списка. Составить третий список, в который входят только те элементы второго списка, которые не входят в первый список.
4. Дан список некоторых целых чисел, поискать в нём значение 20 и, если оно присутствует, заменить его на 200.
5. Написать программу, которая выводит чётные числа из заданного списка и останавливается, если встречает число 15.
6. Дан список из 20 элементов. Сформировать новый список, поместив в него только те элементы из первого списка, которые не превосходят введённое с клавиатуры число.
7. Найти в списке наибольший и наименьший элементы. Вывести их индексы на экран.

Задания для самостоятельного выполнения (по вариантам)

Задание 1. Составить блок-схему и написать программу.

Вариант, №	Задание
1	Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вывести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания – по одному значению в строке. Используйте для сортировки либо метод sort, либо функцию sorted.
2	Напишите программу, которая, как и в предыдущем случае, будет запрашивать у пользователя целые числа и сохранять их в виде списка. Индикатором окончания ввода значений также должен служить ноль. На этот раз необходимо вывести на экран введенные значения в порядке убывания.
3	Во многих карточных играх после процедуры тасования колоды каждый игрок получает на руки определенное количество карт. Напишите программу, принимающую на вход три параметра: количество игроков, количество раздаваемых каждому из них карт и саму колоду. Программа должна возвращать список рук, которые были розданы игрокам. При этом каждая рука, в свою очередь, тоже является списком из входящих в нее карт. Во время раздачи карт игрокам функция параллельно должна удалять розданные карты из переданной ей третьим параметром колоды. Также принято раздавать карты каждому игроку по одной строго по очереди. Придерживайтесь этих принципов и при написании своей программы. Вам необходимо создать колоду карт, перетасовать ее и раздать четырем игрокам по пять карт. Выведите на экран карманные карты всех игроков, находящихся в раздаче, а также оставшиеся в колоде карты.
4	Разработать программу, в которой у пользователя будет запрошен список слов, пока он не оставит строку ввода пустой. После этого на экране должны быть показаны слова, введенные пользователем, но без повторов, – каждое по одному разу. При этом слова должны быть отображены в том же порядке, в каком их вводили с клавиатуры. Например, если пользователь на запрос программы введет следующий список слов: first second first third second программа должна вывести:

	first second third
5	Напишите программу, запрашивающую у пользователя целые числа, пока он не оставит строку ввода пустой. После окончания ввода на экран должны быть выведены сначала все отрицательные числа, которые были введены, затем нулевые и только после этого положительные. Внутри каждой группы числа должны отображаться в той последовательности, в которой были введены пользователем. Например, если он ввел следующие числа:3, -4, 1, 0, -1, 0 и -2, вывод должен оказаться таким: -4, -1, -2, 0, 0, 3 и 1. Каждое значение должно отображаться на новой строке.
6	Напишите программу, которая будет запрашивать у пользователя числа, пока он не пропустит ввод. Сначала на экран должно быть выведено среднее значение введенного ряда чисел, после этого друг за другом необходимо вывести список чисел ниже среднего, равных ему (если такие найдутся) и выше среднего. Каждый список должен предваряться соответствующим заголовком.
7	Обычно при написании перечислений и списков мы разделяем их элементы запятыми, а перед последним ставим союз «и», как показано ниже: яблоки яблоки и апельсины яблоки, апельсины и бананы яблоки, апельсины, бананы и лимоны Напишите программу, которая будет принимать на вход список из строк и возвращать собранную строку из его элементов в описанной выше манере. Хотя в представленном примере количество элементов списка ограничивается четырьмя, ваша программа должна уметь обрабатывать списки любой длины. В программе запросите у пользователя несколько элементов списка, отформатируйте их должным образом и выведите на экран.
8	Для выигрыша главного приза необходимо, чтобы шесть номеров на лотерейном билете совпали с шестью числами, выпавшими случайным образом в диапазоне от 1 до 49 во время очередного тиража. Напишите программу, которая будет случайным образом подбирать шесть номеров для вашего билета. Убедитесь в том, что среди этих чисел не будет дубликатов. Выведите номера билетов на экран по возрастанию.
9	Напишите программу, показывающую, отсортирован ли переданный ей список (по возрастанию или убыванию). Программа должна возвращать True, если список отсортирован, и False в противном случае. В программе запросите у пользователя последовательность чисел для списка, после чего выведите сообщение о том, является ли этот список отсортированным изначально.

10	При анализе собранных по результатам научных экспериментов данных зачастую возникает необходимость избавиться от экстремальных значений, прежде чем продолжать двигаться дальше. Напишите программу, создающую копию списка с исключенными из него n наибольшими и наименьшими значениями и возвращающую ее в качестве результата. Порядок следования элементов в измененном списке не обязательно должен в точности совпадать с источником. Для начала попросите пользователя ввести целые числа, затем соберите их в список. Выведите на экран измененную версию списка вместе с оригинальной. Если пользователь введет менее четырех чисел, должно быть отображено соответствующее сообщение об ошибке.
11	В стандартной библиотеке языка Python присутствует функция count, позволяющая подсчитать, сколько раз определенное значение встречается в списке. Напишите программу, которая будет подсчитывать количество элементов в списке, значения которых больше или равны заданному минимальному порогу и меньше максимального. Программа должна принимать три параметра: список, минимальную границу и максимальную границу. Возвращать она будет целочисленное значение, большее или равное нулю. В программе реализуйте работу для нескольких списков с разными минимальными и максимальными границами. Удостоверьтесь, что программа будет корректно работать со списками, содержащими как целочисленные значения, так и числа с плавающей запятой.
12	Напишите программу, которая требует на ввод списка. Список содержит следующие данные двух пользователей: логин, пароль, ФИО и e-mail. Организуйте хранение данных в много- мерном списке и выведите только фамилии пользователей по его логину, начинающемуся с определенной буквы.
13	«Поросячьей латынью» называют молодежный жаргонный язык, производный от английского. И хотя корни этого новообразованного языка неизвестны, упоминание о нем есть как минимум в двух документах, датированных XIX веком, а это значит, что ему уже больше сотни лет. Для перевода слова с английского на «поросячью латынь» нужно сделать следующее: – если слово начинается с согласной буквы (включая y), то все буквы с начала слова и до первой гласной (за исключением y) переносятся в конец слова и дополняются сочетанием букв ay. Например, слово computer будет преобразовано в omputercay, а слово think – в inkthay; – если слово начинается с гласной буквы (не включая y), к концу слова просто добавляется way. К примеру, слово algorithm превратится в algorithmway, а office – в officeway. Напишите программу, которая будет запрашивать у пользователя строку. После этого она должна переводить введенный текст на «поросячью

	латынь» и выводить его на экран. Вы можете сделать допуск о том, что все слова пользователь будет вводить в нижнем регистре и разделять их пробелами.
14	Напишите программу, которая принимает на вход строку, и выводит слово, которое встречается во фразе реже всего. Если редких слов несколько, нужно вывести то, которое меньше в лексикографическом порядке. Регистр слов не учитывается, знаки препинания в предложениях игнорируются.
15	Создайте словарь, связав его с переменной school, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т. п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе.

Контрольные вопросы

1. Дайте определение списка в языке программирования Python. Приведите примеры списков?
2. Как можно обратиться к элементу списка в языке программирования Python?
3. Для чего используется срез при работе со списками в языке Python?
4. Какие основные встроенные функции для работы со списками в языке Python вы можете назвать?
5. Какие основные операторы языка Python вы использовали при решении задач лабораторной работы?
6. Какой функцией надо воспользоваться для подсчёта количества элементов с данным значением в списке?

Содержание отчета

1. Титульный лист
2. Цель и задачи работы
3. Общие теоретические сведения
4. Задание
5. Составленный алгоритм в виде блок-схемы по каждому заданию.
6. Описание выполнения алгоритма

7. Исходный код программы

8. Тестирование

9. Результаты работы программы

10. Ответы на контрольные вопросы.

11. Общий вывод о проделанной работе.