***Практическая работа № 3***

## Основы работы с Python. Списки

### Задание 3.1. Работа со списками

1. Создание списка.
2. Метод split.
3. Индексация элементов списка. Обращение к элементу списка.
4. Работа со срезами. Границы срезов.
5. Срез с шагом.

### Задание 3.2. Методы для работы со списками

1. Метод Append.
2. Сортировка элементов списка. Метод sort.
3. Статистические показатели списка: функции min, max, метод count
4. Генераторы списков.
5. Написать отчет.
6. Написать программу.

## Отчет выполнения практической работы 3

|  |
| --- |
| **Что такое список. Примеры** |
| Список - это упорядоченное множество элементов, которые могут быть различных типов данных. В программировании, списки часто используются для хранения и обработки коллекций данных. |
| Примеры списков:  1. Список чисел  2. Список строк  3. Список смешанных типов данных  4. Список списков  5. Пустой список |
| **Создание списка. Синтаксис. Примеры** |
| my\_list = [1, 2, 3, 4, 5] |
|  |
| **Метод split Синтаксис. Примеры** |
| Метод split() в Python используется для разделения строки на подстроки на основе определенного разделителя. Вот его синтаксис:  string.split(separator, maxsplit) |
|  |
| **Индексация элементов списка. Обращение к элементу списка. Примеры** |
| Индексация элементов списка в Python начинается с 0. Это означает, что первый элемент списка имеет индекс 0, второй - индекс 1 и так далее. |
| Чтобы обратиться к элементу списка по его индексу, используйте квадратные скобки [] после имени списка, указав индекс элемента внутри скобок.  # Создание списка  fruits = ["apple", "banana", "cherry", "date"]  # Обращение к элементам списка по индексу  print(fruits[0]) # Выводит "apple"  print(fruits[2]) # Выводит "cherry" |
| **Работа со срезами. Границы срезов Примеры** |
| Срезы в Python позволяют получать подсписки из списка, задавая начальный и конечный индексы.  print(numbers[2:5]) # Выводит [2, 3, 4] |
|  |
|  |
|  |
| **Метод Append Синтаксис. Примеры** |
| Метод append() в Python используется для добавления элемента в конец списка.  # Создание пустого списка  my\_list = []  # Добавление элементов в список с помощью метода append()  my\_list.append(1) |
|  |
|  |
| **Метод sort Синтаксис. Примеры** |
| Метод sort() в Python используется для сортировки элементов списка.  # Создание списка  my\_list = [3, 1, 4, 1, 5, 9, 2, 6, 5]  # Сортировка списка по возрастанию  my\_list.sort()  print(my\_list) # Выводит [1, 1, 2, 3, 4, 5, 5, 6, 9] |
|  |
|  |
|  |
| **функции min, max, метод count. Синтаксис. Примеры** |
| Функции min(), max() и метод count() в Python используются для нахождения минимального значения, максимального значения и количества вхождений определенного элемента в списке соответственно.  my\_list = [3, 1, 4, 1, 5, 9, 2, 6, 5]  min\_value = min(my\_list)  print(min\_value) # Выводит 1 |
| max\_value = max(my\_list)  print(max\_value) # Выводит 9 |
| count\_of\_1 = my\_list.count(1)  print(count\_of\_1) # Выводит 2 |
|  |
|  |
|  |
|  |
| **Генераторы списков. Примеры** |
| Генераторы списков - это удобный способ создания списков в Python  squares = [x\*\*2 for x in range(1, 6)]  print(squares) # Вывод: [1, 4, 9, 16, 25] |
|  |
|  |
|  |
|  |
| Что делает данный код  my\_list = [x for x in range(10, 100) if x%3 == 0 and x%5 == 0] |
| Этот код создает список, содержащий все числа в диапазоне от 10 до 100, которые делятся и на 3, и на 5 без остатка. То есть, он создает список всех чисел, которые являются кратными 15 в этом диапазоне. |

### Варианты заданий

1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Сформировать новый список из убывающих нечётных чисел.
2. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор) и выводит на экран в одну строку значения, которые повторяются в нём более одного раза. Выводимые числа не дoлжны повторяться, порядок их вывода может быть произвольным. Например: 4 7 0 2 3 4 0 3 8. Результат: 4 0 3
3. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Поменять местами самый большой и самый маленький элементы списка.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Сформировать новый список из возрастающих чётных чисел.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Проверить, является ли введенный список симметричным (первый элемент равен последнему, второй - предпоследнему и т.д.).



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Подсчитать среднее арифметическое всех положительных элементов списка, введенного пользователем.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор) размера N и целые числа K и L (1 < K ≤ L ≤ N). Найти сумму всех элементов списка, кроме элементов с номерами от K до L включительно.



1. Напишите программу, которая принимает на вход список букв в одной строке (через генератор). Сформируйте список из гласных букв.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор). Сформируйте список из значений, которые больше последнего.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор), насчитывающий N элементов. Поменяйте местами первую и вторую половины списка



1. Дан произвольный текст. Найдите номер первого самого длинного слова в нем.



1. Дан произвольный текст, содержащий слова и цифры. Сформируйте список из всех имеющихся в нем цифр, определите их количество, сумму и найти максимальное.



1. Дан произвольный текст. Сформировать список из длин слов, которые входят в этот текст.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор) размера N и целые числа K и L. Числа, значения которых принадлежат [ K, L] необходимо удвоить.



1. Напишите программу, которая принимает на вход список чисел в одной строке (через генератор) размера N и целые числа K и L (1 < K ≤ L ≤ N). Сформировать список из чисел, кроме элементов с номерами от K до L включительно.