**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ))**

**Институт управления и цифровых технологий (ИУЦТ)**

**Кафедра Цифровые технологии управления транспортными процессами (ЦТУТП)**

**09.03.02 Информационные системы и технологии – Технологии искусственного интеллекта в транспортных системах**

**Программирование на языке Python**

**Лабораторная работа № 6**

**На тему: операции над списками в Python**

**Выполнил: Сергеев Егор Сергеевич**

**Группа: УИТ-111**

**Проверил: доцент к.т.н. Бесчастный Михаил Александрович**

Оглавление

[**1.** **Цель работы** 3](#_Toc210865652)

[**2.** **Порядок выполнения задания 1** 3](#_Toc210865653)

[2.1. Используемая среда и язык. 3](#_Toc210865654)

[2.2. Используемые библиотеки. 3](#_Toc210865655)

[2.3. Ход работы. 3](#_Toc210865656)

[2.4. Блок-схема алгоритма. 4](#_Toc210865657)

[2.5. Скриншот кода. 6](#_Toc210865658)

[**3.** **Результаты выполнения задания** 8](#_Toc210865659)

[**4.** **Выводы** 9](#_Toc210865660)

[Список литературы 9](#_Toc210865661)

1. **Цель работы**

Изучение одномерных массивов в Python

## **Порядок выполнения задания (Вариант 14)**

* 1. Используемая среда и язык. Для начала работы устанавливаем среду разработки Visual Studio Code, в ней устанавливаем расширения языка программирования Python от Microsoft (Python, Pylance, Python Debugger, Python Environments)
  2. Используемые библиотеки. Для написания программы не понадобилось использование сторонних библиотек

## Ход работы.

* + 1. Создадим свой индивидуальный список и проделаем с ним все операции, представленные в таблице лабораторной работы, при этом изначально напечатаем исходный массив и будем печатать этот массив, после выполнения каждой операции, чтобы показать как она влияет на этот массив.
    2. Запрашиваем у пользователя длину массива, пишем генератор списка по заданной длине, и показываем пользователю результат – введенный массив. Создаем две пары (два списка) для хранения [значение, индекс] максимума и минимума из массива, получаем их после прохода по этому массиву, меняем местами в этом массиве, выводим пользователю на экран.
    3. Принимаем с консоли 10 чисел в массив, создаем переменную sum = 0, проходим по массиву и прибавляем все его элементы к summ, после делим summ на длину массива и получаем среднее арифметическое элементов массива. Далее проходим по массиву и меняем все элементы большие среднего арифметического на 1, с массивами работаем с помощью генераторов массивов в Python.
    4. Реализуем программы для сортировок вставками, выбором (минимума и максимума), пузырьковую (модификация на проверку изменений после очередного прохода по массиву), подсчетом (реализация для целых чисел). Далее пишем функцию checker() и создаем тестовый массив, отправляем его в checker() на проверку написанных ранее алгоритмов сортировки.

## Блок-схема алгоритма.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 1 - Блок-схема алгоритма 1

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 2 – Блок-схема алгоритма 2

Изображение выглядит как снимок экрана, текст, диаграмма, линия

Автоматически созданное описание

Рисунок 3 – Блок-схема алгоритма 3

Изображение выглядит как текст, снимок экрана, Шрифт, визитная карточка

Автоматически созданное описание

Рисунок 4 – Блок-схема алгоритма 4

## Скриншот кода.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 5 – Программный код алгоритма 1

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 6 – Программный код алгоритма 2

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – Программный код алгоритма 3

Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

Рисунок 8 – Программный код алгоритма 4

## **Результаты выполнения задания**

* 1. В результате выполнения программ, правильно написаны методы работы с массивами в Python, их влияние на массив продемонстрировано с помощью функции print() (алгоритм 1). Успешно найдены максимум и минимум массива, успешно выполнена их смена в массиве (алгоритм 2). Найдено среднее арифметическое значение массива и заменен в нем ряд элементов на 1, то есть заменены элементы этого массива, которые больше среднего арифметического этого массива (алгоритм 3). Успешно выполнены алгоритмы сортировок (алгоритм 4)
  2. Скриншот результата

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 9 - Скриншот результата алгоритма 1

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 10 – Скриншот результата алгоритма 2

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 11 – Скриншот результата алгоритма 3

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 12 – Скриншот результата алгоритма 4

1. **Выводы**

Выполнение данной лабораторной работы помогает понять, как работать с одномерными массива в Python, учит взаимодействию программы и пользователя, посредством ввода данных пользователя в консоль и вывода ответных данных от программы пользователю в консоль.

# **Список литературы**

1. Оформление отчетов к Лабораторным работам РУТ (МИИТ) ЦТУТП 2025-2026: <https://www.dropbox.com/scl/fo/w6g4z0qdcymg5v67ou1ej/AGjQER9EQ9d3awkQsc1eYMc?dl=0&e=1&preview=Практические+работы+-+Оформление+отчета.pdf&rlkey=ne210ijmotnpa85164a8rxl1j>