|  |  |
| --- | --- |
| **К Г Э У** | МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

**Кафедра «Информатики и информационных управляющих систем»**

ОТЧЕТ

Дисциплина «Алгоритмизация и программирование»

Выполнил: Соловьёв Л.А.  
Группа: ПИ-1-22  
Преподаватель: Шорина Т.В.

Казань, 2024

***Постановка задачи***

1. Изучение работы функций библиотеки NumPy

NumPy — это открытая бесплатная Python-библиотека для работы с данными. Её чаще всего используют в анализе данных и обучении нейронных сетей — в каждой из этих областей нужно проводить много вычислений с такими матрицами.

**Основные команды модуля Numpy**

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| np.array() | Создает массив NumPy из обычного списка или кортежа данных. |
| np.arange() | Возвращает равномерно разнесенные значения в пределах заданного интервала. |
| np.linspace() | Возвращает равномерно разнесенные значения на интервале между двумя заданными точками. |
| np.zeros() | Создает массив, заполненный нулями. |
| np.ones() | Создает массив, заполненный единицами. |
| np.eye() | Создает единичную матрицу заданного размера. |
| np.random.rand() | Возвращает массив случайных чисел на отрезке [0, 1). |
| np.max() | Находит максимальное значение |
| np.min() | Находит минимальное значение |
| np.sum() | Вычисляет сумму элементов |

**Примеры обработки данных с использованием модуля NumPy**

**Задача 1.** Создать из списка чисел массив NumPy.

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, содержащий чередующиеся нули и единицы. После этого, используя функцию np.array() преобразуем список arr в массив NumPy c названием numpy\_arr. Затем, выводим массив с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив NumPy, содержащий элементы списка arr.

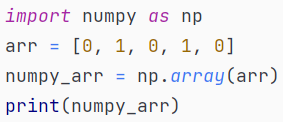


Рисунок 1.1 – Код к задаче №1



Рисунок 1.2 – Вывод к задаче №1

**Задача 2.** Создать массив вложенных массивов, состояющих из нулей.

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, используя функцию np.zeros(), которая принимает на вход число строк конечной матрицы. Затем, выводим массив с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив NumPy, содержащий три вложенных массива по два нуля в каждом.

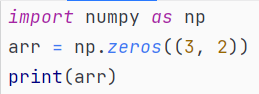


Рисунок 2.1 – Код к задаче №2



Рисунок 2.2 – Вывод к задаче №2

**Задача 3.** Создать массив единичной матрицы

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, используя функцию np.eye(), которая принимает на вход число строк конечной матрицы. Затем, выводим массив с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив NumPy размерностью 3, содержащий единицы на главной диагонали.

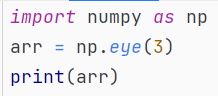


Рисунок 3.1 – Код к задаче №3

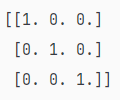


Рисунок 3.2 – Вывод к задаче №3

**Задача 4.** Создать массив, содержащий числа от 10 до 30 с шагом 6

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, используя функцию np.arange(), которая принимает на вход три числа: начальное число, конечное число, шаг. Затем, выводим массив с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив NumPy, состоящий из чисел от 10 до 30 с шагом 6([10, 10+6, 10+12, 10+18]).

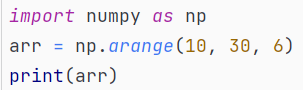


Рисунок 4.1 – Код к задаче №4



Рисунок 4.2 – Вывод к задаче №4

**Задача 5.** Создать матрицу

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, который содержит вложенные массивы с целыми числами. После этого преобразуем список arr в матрицу matrix с помощью функции np.mat(). Затем, выводим matrix с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив NumPy, состоящий из элементов массива arr.

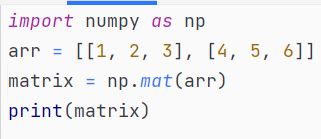


Рисунок 5.1 – Код к задаче №5

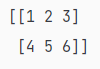


Рисунок 5.2 – Вывод к задаче №5

**Задача 6.** Транспонировать матрицу

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, который содержит вложенные массивы с целыми числами. После этого преобразуем список arr в матрицу matrix с помощью функции np.mat(). Затем, создаем матрицу transp\_matrix с помощью функции np.transpose() и выводим transp\_matrix с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим транспонированный массив NumPy, состоящий из элементов массива arr.

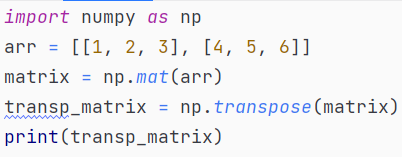


Рисунок 6.1 – Код к задаче №6



Рисунок 6.2 – Вывод к задаче №6

**Задача 7.** Произведение элементов матрицы

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, который содержит вложенные массивы с целыми числами. После этого преобразуем список arr в матрицу matrix с помощью функции np.mat(). Затем к полученной матрицы применяем функцию np.prod() и выводим результат этой функции.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим произведение элементов матрицы.

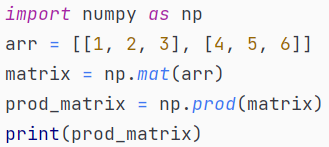


Рисунок 7.1 – Код к задаче №7



Рисунок 7.2 – Вывод к задаче №7

**Задача 8.** Найти максимум, минимум, сумму и среднее значение элементов в массиве

Алгоритм выполнения:

Для начала, импортируем библиотеку NumPy с помощью команды import numpy as np. Затем создаем список arr, содержащий целые числа от 1 до 6. После этого, используя функцию np.max(), np.min(), np.sum(), np.average() вычисляем максимум, минимум, сумму и среднее значение элементов списка arr. Затем, выводим результат с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим максимальное число, минимальное число, сумму элементов и среднее значение элементов матрицы.

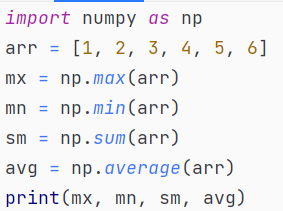


Рисунок 8.1 – Код к задаче №8



Рисунок 8.2 – Вывод к задаче №8

**Вывод**

В ходе работы были изучены основные возможности и функции данной библиотеки. NumPy предоставляет широкий набор инструментов для работы с многомерными массивами данных, а также множество функций для математических операций и обработки данных.

Были изучены основные методы работы с массивами. Также были рассмотрены способы выполнения различных математических операций.

Все изученные материалы были успешно применены в практических задачах, что позволило более глубоко понять возможности и преимущества использования библиотеки NumPy в разработке на языке программирования Python.

**Список использованных источников**

1. [**https://pythonworld.ru/numpy/1.html**](https://pythonworld.ru/numpy/1.html)
2. **dir(numpy)**