Практическое задание.

Тема: Двумерные массивы.

Цель: Изучить создание двумерных массивов и их отличие от массивов.

Задачи: Создать матрицы на основе двумерных массивов.

Теоретическая часть.

Двумерные массивы в Python представляют собой массивы, содержащие в себе другие массивы, создающие двумерную структуру данных. Они обычно используются для хранения и обработки таблиц или матриц, где данные организованы в виде строк и столбцов.

Пример создания двумерного массива:

```
matrix = [[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]]
```

В данном примере мы создали двумерный массив matrix, состоящий из трех строк и трех столбцов. Каждая строка представлена внутренним массивом.

Двумерные массивы в Python могут быть использованы для множества задач, включая обработку изображений, матричные вычисления и анализ данных.

Матрицы в Python - это двумерные массивы, которые содержат элементы, расположенные в строках и столбцах. Матрицы можно представить в виде списка списков, где каждый внутренний список представляет строку элементов. Матрицы могут быть использованы для хранения и обработки данных, таких как числа, символы или любые другие объекты. Они широко применяются в научных вычислениях, анализе данных, компьютерной графике и других областях, где требуется работа с двумерными структурами данных.

Матрицы в Python можно создавать с использованием списка списков. Например, вот простая матрица 2x3:

```
matrix = [[1, 2, 3], [4, 5, 6]]
```

В данном примере мы создали матрицу, состоящую из двух строк и трех столбцов. Каждый элемент матрицы представлен числом. Вы можете обращаться к элементам матрицы, используя индексы. Например, чтобы получить значение 4, нужно обратиться к элементу с индексом [1][0], так как первая цифра указывает на строку, а вторая - на столбец.

Матрицы могут быть использованы для решения различных задач, включая решение систем линейных уравнений, вычисление определителя, умножение матриц и многое другое.

Для обработки элементов матрицы в Python можно использовать циклы. Например, чтобы вывести все элементы матрицы на экран, мы можем использовать вложенные циклы:

```
matrix = [[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]]

for row in matrix:
    for element in row:
        print(element, end=' ')
    print()
```

В данном примере мы перебираем строки матрицы (внешний цикл) и для каждой строки перебираем элементы (внутренний цикл). Затем мы выводим каждый элемент на экран с помощью функции print(). Результат будет выглядеть так:

```
1 2 3
4 5 6
7 8 9
```

Это простой пример обработки элементов матрицы, и вы можете настраивать обработку в зависимости от своих потребностей.

Рассмотрим пример сложения двух матриц:

```
matrix1 = [[1, 2], [3, 4]]

matrix2 = [[5, 6], [7, 8]]
```

```
# Вывод элементов каждой матрицы
```

```
print('Матрица 1')

for elem1 in matrix1:

    print(elem1)
```

```
print('Матрица 2')
for elem2 in matrix2:
    print(elem2)
```

```
# Сложение элементов
```

```
print('Сложение:')
for i1 in range(len(matrix1)):
    for i2 in range(len(matrix2[0])):
        print(matrix1[i1][i2] + matrix2[i1][i2], end=' ')
```

Сложение двух матриц происходит путем сложения соответствующих элементов матриц. В данном примере есть две матрицы matrix1 и matrix2.

Сначала мы выводим содержимое каждой матрицы с помощью цикла for. Затем мы выполняем сложение матрицы1 и матрицы2, перебирая элементы каждой матрицы с помощью двух вложенных циклов.

Выводится результат сложения каждого элемента, полученного суммированием соответствующих элементов матрицы 1 и матрицы 2. Результат будет выглядеть следующим образом:

```
Матрица 1
[1, 2]
[3, 4]
Матрица 2
[5, 6]
[7, 8]
Сложение:
[6, 8]
[10, 12]
```

Таким образом, результатом сложения матрицы 1 и матрицы 2 будет новая матрица, где каждый элемент является суммой соответствующих элементов из исходных матриц.

Практическая часть.

Задание 1.

Дан двумерный массив:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Выведете все элементы, все нечётные числа и количество чётных чисел.

Выходные данные:

matrix:

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

нечётные числа matrix

1 3 5 7 9

кол-во чётных: 4

Задание 2.

Даны две матрицы:

```
matrix_1 = [[2, 4, 3, 6], [5, 7, 1, 5]]
matrix_2 = [[2, 9, 0, 2], [3, 4, 7, 6]]
```

1. Создайте аналогичную матрицу answer_matrix и присвойте ей нулевые значения (любые математические операции возможны только с одинаковыми матрицами).
2. Перемножьте между собой все числа из matrix_1 и matrix_2, а результат присвойте answer_matrix. Выведите результат в консоль.
3. Выведете сумму элементов для каждой строки answer_matrix.

Выходные данные:

[[4, 36, 0, 12], [15, 28, 7, 30]]

[4, 36, 0, 12] сумма строки: 52

[15, 28, 7, 30] сумма строки: 80

Задание 3.

Дан двумерный массив:

```
fruits = [['Banana', 'apple'], ['apricot', 'Avocado'],
['lime', 'lemon'], ['Mango', 'grapes']]
```

Выведете все элементы, которые написаны с заглавной буквы. Можно воспользоваться строчным методом isupper().

Задание 4.

Дан двумерный массив:

```
random_elements = [['toy', 'bee', 'cheese', 'ear'],
                   [False, 'word', '0110110', 10],
                   ['happiness', '(╯°□°)╯ ', 'luck', None],
                   ['car', '<- code ->', 4.7, True]]
```

Выведете каждый второй элемент из random_elements с помощью функции enumerate.

enumerate - это функция в Python, которая позволяет перебирать элементы массива вместе с их индексами. Она возвращает объект-перечислитель, который можно использовать в цикле for для доступа к индексу и значению элемента одновременно. Пример работы enumerate:

```
fruits = ['apple', 'banana', 'cherry']
for index, fruit in enumerate(fruits):
    print(f"Индекс: {index}, Фрукт: {fruit}")
```

В данном примере мы перебираем элементы массива fruits с помощью функции enumerate. В каждой итерации цикла for переменная index будет содержать индекс текущего элемента, а переменная fruit - значение текущего элемента. Результат вывода будет выглядеть следующим образом:

```
Индекс: 0, Фрукт: apple
Индекс: 1, Фрукт: banana
Индекс: 2, Фрукт: cherry
```

Задание 5.

Напишите программу, в которой пользователь сам сможет создать матрицу x на y и ввести туда свои числовые значения.

Пример входных данных:

```
Введите количество строк:
2
Введите количество столбцов:
3
Введите значение элемента [0][0]:
2
Введите значение элемента [0][1]:
1
Введите значение элемента [0][2]:
5
Введите значение элемента [1][0]:
67
Введите значение элемента [1][1]:
-1
Введите значение элемента [1][2]:
0
```

Выходные данные:

```
Ваш двумерный массив:
[2, 1, 5]
[67, -1, 0]
```

Алгоритм выполнения:

1. Ввод строк
2. Ввод столбцов
3. Пустой массив, для будущей матрицы
4. Цикл в диапазоне количества строк
5. Пустой массив для строк
6. Цикл в диапазоне количества столбцов
7. Ввод чисел
8. В массив для строк добавляются числа
9. Массивы (строки) с числами добавляются в массив матрицы
10. Вывод массива матрицы

Выставление оценки.

Оценка 5 – Вы мастерски справились с 5 заданиями.

Оценка 4 – Сделано 4 любых задания.

Оценка 3 – Сделано 3 любых задания.

Оценка 2 – Сделано меньше 3 заданий.