### **Лабораторная работа №8**

#### **Тема: Работа с многомерными массивами**

#### **Задача 1. Условие.**

Программа для нахождения номеров строк максимальных элементов каждого столбца матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Выходные данные | max\_rows | Список номеров строк максимальных элементов | список целых |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 35, 20], [30, 5, 40], ...] |
| Выходные данные | max\_rows | [2, 0, 1, 3, 4] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] max\_rows = [max(range(5), key=lambda i: matrix[i][j]) for j in range(5)]

print("Матрица:") for row in matrix: print(row) print("Номера строк максимальных элементов каждого столбца:", max\_rows)

#### **Задача 2. Условие.**

Программа для упорядочивания строк матрицы 5x5: строки с нечетными индексами по убыванию, с четными - по возрастанию.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Выходные данные | sorted\_matrix | Преобразованная матрица | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[1, 40, 35], [15, 5, 25], ...] |
| Выходные данные | sorted\_matrix | [[1, 20, 25], [40, 35, 10], ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for i in range(5): matrix[i].sort(reverse=(i % 2 != 0))

print("Преобразованная матрица:") for row in matrix: print(row)

#### **Задача 3. Условие.**

Программа для замены всех элементов матрицы 5x5 на нули и главной диагонали на ее номера.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Выходные данные | transformed\_matrix | Преобразованная матрица | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[1, 5, 10], [7, 15, 20], ...] |
| Выходные данные | transformed\_matrix | [[0, 0, 0], [0, 1, 0], ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for i in range(5): for j in range(5): matrix[i][j] = i if i == j else 0

print("Преобразованная матрица:") for row in matrix: print(row)

**Задача 4. Условие.**

Программа для нахождения суммы элементов матрицы 5x5, сумма индексов которых равна 4.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Выходные данные | sum\_indices\_4 | Сумма элементов, сумма индексов которых равна 4 | целое число |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [40, 50, 60], ...] |
| Выходные данные | sum\_indices\_4 | 100 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] sum\_indices\_4 = sum(matrix[i][j] for i in range(5) for j in range(5) if i + j == 4)

print("Матрица:") for row in matrix: print(row) print("Сумма элементов, сумма индексов которых равна 4:", sum\_indices\_4)

#### **Задача 5. Условие.**

Программа для нахождения среднего арифметического положительных элементов и количества нулей в матрице 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон -15 до 40 |
| Выходные данные | average\_positive | Среднее арифметическое положительных элементов | вещественное |  |
| Промежуточные данные | zeros\_count | Количество нулей | целое число |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 0, 30], [-10, 5, 0], ...] |
| Выходные данные | average\_positive | 15.00 |
| Промежуточные данные | zeros\_count | 3 |

**Листинг программы:**

import random

matrix = [[random.randint(-15, 40) for \_ in range(5)] for \_ in range(5)] positive\_elements = [matrix[i][j] for i in range(5) for j in range(5) if matrix[i][j] > 0] zeros\_count = sum(1 for i in range(5) for j in range(5) if matrix[i][j] == 0)

average\_positive = sum(positive\_elements) / len(positive\_elements) if positive\_elements else 0

print("Матрица:") for row in matrix: print(row) print("Среднее арифметическое положительных элементов:", average\_positive) print("Количество нулей:", zeros\_count)

#### **Задача 6. Условие.**

Программа для вычисления среднего арифметического элементов строки матрицы 5x5 в заданном диапазоне.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Входные данные | lower\_bound | Нижняя граница диапазона | целое |  |
| Входные данные | upper\_bound | Верхняя граница диапазона | целое |  |
| Выходные данные | averages | Средние арифметические для каждой строки | список вещественных |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Входные данные | lower\_bound | 10 |
| Входные данные | upper\_bound | 30 |
| Выходные данные | averages | [15.0, 25.0, ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] lower\_bound = int(input("Введите нижнюю границу диапазона: ")) upper\_bound = int(input("Введите верхнюю границу диапазона: "))

averages = [] for row in matrix: filtered\_elements = [x for x in row if lower\_bound < x < upper\_bound] avg = sum(filtered\_elements) / len(filtered\_elements) if filtered\_elements else 0 averages.append(avg)

print("Матрица:") for row in matrix: print(row) print("Средние арифметические для каждой строки:", averages)

#### **Задача 7. Условие.**

Программа для создания списков элементов выше и ниже главной диагонали матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-20 |
| Промежуточные данные | above\_diagonal | Список элементов выше главной диагонали | список |  |
| Промежуточные данные | below\_diagonal | Список элементов ниже главной диагонали | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | above\_diagonal | [20, 30, 40] |
| Промежуточные данные | below\_diagonal | [15, 25, 35] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 20) for \_ in range(5)] for \_ in range(5)] above\_diagonal = [matrix[i][j] for i in range(5) for j in range(i, 5)] below\_diagonal = [matrix[i][j] for i in range(1, 5) for j in range(i)]

print("Матрица:") for row in matrix: print(row) print("Элементы выше главной диагонали:", above\_diagonal) print("Элементы ниже главной диагонали:", below\_diagonal)

#### **Задача 8. Условие.**

Программа для вывода главной и побочной диагоналей матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-50 |
| Промежуточные данные | main\_diagonal | Главная диагональ | список |  |
| Промежуточные данные | secondary\_diagonal | Побочная диагональ | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | main\_diagonal | [10, 5, 35] |
| Промежуточные данные | secondary\_diagonal | [30, 20, 10] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 50) for \_ in range(5)] for \_ in range(5)] main\_diagonal = [matrix[i][i] for i in range(5)] secondary\_diagonal = [matrix[i][4 - i] for i in range(5)]

print("Матрица:") for row in matrix: print(row) print("Главная диагональ:", main\_diagonal) print("Побочная диагональ:", secondary\_diagonal)

#### **Задача 9. Условие.**

Программа для вывода половины матрицы 5x5 относительно главной диагонали.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | above\_half | Элементы выше главной диагонали | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | above\_half | [20, 30, 40] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Матрица:") for row in matrix: print(row)

print("Элементы выше главной диагонали:") for i in range(5): for j in range(i + 1, 5): print(matrix[i][j], end=" ") print()

#### **Задача 10. Условие.**

Программа для вычисления среднего арифметического элементов матрицы 5x5, больших числа 20.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-60 |
| Промежуточные данные | elements\_above\_20 | Список элементов больше 20 | список |  |
| Выходные данные | average\_above\_20 | Среднее арифметическое элементов больше 20 | вещественное |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 30, 40], [25, 5, 60], ...] |
| Выходные данные | average\_above\_20 | 35.00 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 60) for \_ in range(5)] for \_ in range(5)] elements\_above\_20 = [matrix[i][j] for i in range(5) for j in range(5) if matrix[i][j] > 20] average\_above\_20 = sum(elements\_above\_20) / len(elements\_above\_20) if elements\_above\_20 else 0

print("Матрица:") for row in matrix: print(row) print(f"Среднее арифметическое элементов больше 20: {average\_above\_20}")

#### **Задача 11. Условие.**

Программа для упорядочивания строк матрицы 5x5: строки с нечетными индексами по убыванию, с четными — по возрастанию.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | sorted\_matrix | Матрица после сортировки строк | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [40, 50, 60], ...] |
| Промежуточные данные | sorted\_matrix | [[10, 20, 30], [60, 50, 40], ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for i in range(5): matrix[i].sort(reverse=(i % 2 != 0))

print("Преобразованная матрица:") for row in matrix: print(row)

#### **Задача 12. Условие.**

Программа для вычисления суммы наибольших значений в столбцах матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | max\_elements | Список наибольших элементов в столбцах | список |  |
| Выходные данные | sum\_max\_elements | Сумма наибольших элементов в столбцах | целое |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | max\_elements | [20, 40, 50] |
| Выходные данные | sum\_max\_elements | 110 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] max\_elements = [max(matrix[i][j] for i in range(5)) for j in range(5)] sum\_max\_elements = sum(max\_elements)

print("Матрица:") for row in matrix: print(row) print("Список наибольших элементов в столбцах:", max\_elements) print("Сумма наибольших значений в столбцах:", sum\_max\_elements)

#### **Задача 13. Условие.**

Программа для замены всех элементов матрицы 5x5 на нули и главной диагонали на её номера.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | transformed\_matrix | Преобразованная матрица | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | transformed\_matrix | [[0, 0, 0], [0, 1, 0], ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for i in range(5): for j in range(5): matrix[i][j] = i if i == j else 0

print("Преобразованная матрица:") for row in matrix: print(row)

#### **Задача 14. Условие.**

Программа для обмена второй и четвёртой строки в матрице 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | swapped\_matrix | Матрица после обмена строк | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 5, 40], ...] |
| Промежуточные данные | swapped\_matrix | [[10, 20, 30], [40, 50, 60], ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

matrix[1], matrix[3] = matrix[3], matrix[1]

print("Матрица после обмена второй и четвёртой строки:") for row in matrix: print(row)



#### **Задача 15. Условие.**

Программа для перемножения двух матриц 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix\_A | Первая матрица 5x5, заполненная случайными числами | список списков | диапазон 1-10 |
| Входные данные | matrix\_B | Вторая матрица 5x5, заполненная случайными числами | список списков | диапазон 1-10 |
| Промежуточные данные | matrix\_C | Матрица 5x5, результат перемножения матриц | список списков |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix\_A | [[1, 2, 3], [4, 5, 6], ...] |
| Входные данные | matrix\_B | [[7, 8, 9], [10, 11, 12], ...] |
| Промежуточные данные | matrix\_C | [[58, 64, 60], [139, 154, 147], ...] |

**Листинг программы:**

import random

matrix\_A = [[random.randint(1, 10) for \_ in range(5)] for \_ in range(5)] matrix\_B = [[random.randint(1, 10) for \_ in range(5)] for \_ in range(5)] matrix\_C = [[sum(matrix\_A[i][k] \* matrix\_B[k][j] for k in range(5)) for j in range(5)] for i in range(5)]

print("Матрица A:") for row in matrix\_A: print(row)

print("Матрица B:") for row in matrix\_B: print(row)

print("Результат перемножения A и B (матрица C):") for row in matrix\_C: print(row)

#### **Задача 16. Условие.**

Программа для вычисления сумм элементов выше и ниже главной диагонали матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | sum\_above | Сумма элементов выше главной диагонали | целое |  |
| Промежуточные данные | sum\_below | Сумма элементов ниже главной диагонали | целое |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 25, 40], ...] |
| Промежуточные данные | sum\_above | 150 |
| Промежуточные данные | sum\_below | 130 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] sum\_above = sum(matrix[i][j] for i in range(5) for j in range(i)) sum\_below = sum(matrix[i][j] for i in range(5) for j in range(i + 1, 5))

print("Матрица:") for row in matrix: print(row) print("Сумма элементов выше главной диагонали:", sum\_above) print("Сумма элементов ниже главной диагонали:", sum\_below)

#### **Задача 17. Условие.**

Программа для замены первого элемента каждой строки матрицы 5x5 на среднее арифметическое элементов строки.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-20 |
| Промежуточные данные | avg\_values | Средние арифметические для каждой строки | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[5, 10, 15], [20, 25, 30], ...] |
| Промежуточные данные | avg\_values | [10.0, 25.0, ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 20) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for i in range(5): avg = sum(matrix[i]) / 5 matrix[i][0] = avg

print("Матрица после замены:") for row in matrix: print(row)

#### **Задача 18. Условие.**

Программа для определения номеров столбцов, среднее арифметическое которых меньше среднего арифметического всей матрицы.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | cols | Список номеров столбцов | список |  |
| Выходные данные | overall\_avg | Среднее арифметическое всей матрицы | вещественное |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 25, 35], ...] |
| Промежуточные данные | cols | [1, 3, 4] |
| Выходные данные | overall\_avg | 24.5 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

overall\_avg = sum(sum(row) for row in matrix) / 25 cols = [j for j in range(5) if sum(matrix[i][j] for i in range(5)) / 5 < overall\_avg]

print("Матрица:") for row in matrix: print(row) print("Номера столбцов с меньшим средним арифметическим:", cols)

#### **Задача 19. Условие.**

Программа для проверки, является ли введённая пользователем матрица 5x5 единичной.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, введённая пользователем | список списков | диапазон 0-1 |
| Промежуточные данные | is\_identity | Логическое значение проверки единичности | логическое |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[1, 0, 0], [0, 1, 0], ...] |
| Промежуточные данные | is\_identity | True |

**Листинг программы:**

matrix = [] print("Введите элементы матрицы 5x5:") for i in range(5): matrix.append([int(x) for x in input().split()])

is\_identity = all(matrix[i][j] == (1 if i == j else 0) for i in range(5) for j in range(5))

print("Матрица:") for row in matrix: print(row) print("Матрица является единичной" if is\_identity else "Матрица не является единичной")

#### **Задача 20. Условие.**

Программа для упорядочивания столбцов матрицы 5x5 по убыванию.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | column | Отсортированный по убыванию столбец | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 25, 40], ...] |
| Промежуточные данные | column | [40, 30, 20, 15, 10] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)]

print("Исходная матрица:") for row in matrix: print(row)

for j in range(5): column = sorted([matrix[i][j] for i in range(5)], reverse=True) for i in range(5): matrix[i][j] = column[i]

print("Матрица после упорядочивания столбцов по убыванию:") for row in matrix: print(row)

#### **Задача 21. Условие.**

Программа для вычисления количества положительных элементов на периметре и диагоналях матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон -20 - 20 |
| Промежуточные данные | positive\_count | Количество положительных элементов | целое |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[-10, 5, 15], [0, 20, -5], ...] |
| Промежуточные данные | positive\_count | 12 |

**Листинг программы:**

import random

matrix = [[random.randint(-20, 20) for \_ in range(5)] for \_ in range(5)] positive\_count = sum(matrix[i][j] > 0 for i in range(5) for j in range(5) if i == j or i + j == 4 or i == 0 or i == 4 or j == 0 or j == 4)

print("Матрица:") for row in matrix: print(row) print("Количество положительных элементов на периметре и диагоналях:", positive\_count)

#### **Задача 22. Условие.**

Программа для вычисления суммы элементов каждой строки матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Выходные данные | sums | Суммы элементов строк | список |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 25, 40], ...] |
| Выходные данные | sums | [60, 80, ...] |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] sums = [sum(row) for row in matrix]

print("Матрица:") for row in matrix: print(row) print("Суммы элементов строк:", sums)

#### **Задача 23. Условие.**

Программа для вычисления суммы максимальных значений в строках матрицы 5x5.

**Таблица данных:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Матрица 5x5, заполненная случайными числами | список списков | диапазон 1-40 |
| Промежуточные данные | max\_elements | Максимальные элементы в строках | список |  |
| Выходные данные | sum\_max\_elements | Сумма максимальных элементов | целое |  |

**Тесты:**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [[10, 20, 30], [15, 25, 40], ...] |
| Промежуточные данные | max\_elements | [30, 40, ...] |
| Выходные данные | sum\_max\_elements | 200 |

**Листинг программы:**

import random

matrix = [[random.randint(1, 40) for \_ in range(5)] for \_ in range(5)] max\_elements = [max(row) for row in matrix] sum\_max\_elements = sum(max\_elements)

print("Матрица:") for row in matrix: print(row) print("Список максимальных элементов в строках:", max\_elements) print("Сумма максимальных значений в строках:", sum\_max\_elements)

### **Задача 24. Условие.**

Программа для заполнения вложенной последовательности A[5,5] случайными целыми числами, находящимися в интервале от 1 до 50. Определите максимальный элемент среди элементов последовательности, расположенных выше главной диагонали, и минимальный элемент среди тех, которые находятся ниже главной диагонали.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от 1 до 50 |
| Промежуточные данные | max\_above | Максимальный элемент выше главной диагонали | целое число |  |
| Промежуточные данные | min\_below | Минимальный элемент ниже главной диагонали | целое число |  |
| Выходные данные | max\_above, min\_below | Максимальный и минимальный элементы | целое число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, 35, 12, 48, 7], [6, 19, 32, 11, 43], [41, 22, 16, 18, 27], [30, 24, 13, 31, 20], [5, 40, 33, 29, 17] ] |
| Выходные данные | max\_above | 48 |
| Выходные данные | min\_below | 5 |

#### **Листинг программы:**

Вначале создается функция для нахождения максимального и минимального элементов по условию. Далее, последовательность заполняется случайными числами, и для удобства выводится на экран. Максимальный элемент выше главной диагонали и минимальный элемент ниже главной диагонали выводятся в конце.

#### **Код программы:**

import random  
def find\_max\_above\_diagonal\_min\_below\_diagonal(matrix):  
max\_above = float("-inf")  
min\_below = float("inf")  
for i in range(5):  
for j in range(5):  
if i < j: # Элементы выше главной диагонали  
max\_above = max(max\_above, matrix[i][j])  
elif i > j: # Элементы ниже главной диагонали  
min\_below = min(min\_below, matrix[i][j])  
return max\_above, min\_below  
matrix = [[random.randint(1, 50) for \_ in range(5)] for \_ in range(5)]  
max\_above, min\_below = find\_max\_above\_diagonal\_min\_below\_diagonal(matrix)  
for row in matrix:  
print(row)  
print(f"Максимальный элемент выше главной диагонали: {max\_above}")  
print(f"Минимальный элемент ниже главной диагонали: {min\_below}")

### **Задача 25. Условие.**

Программа для заполнения вложенной последовательности A[5,5] случайными целыми числами, находящимися в интервале от –20 до 20. Найдите в строках самые правые наименьшие элементы и определите их местоположение, то есть выведите на экран номер столбца.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от -20 до 20 |
| Промежуточные данные | min\_val, min\_pos | Минимальные значения и их индексы | целое число |  |
| Выходные данные | rightmost\_min\_positions | Список номеров столбцов с минимальными элементами | список целых чисел |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [-1, -2, 3, -5, 10], [-10, 0, 5, 1, -3], [2, -4, 7, 0, 5], [4, 3, -1, -8, 9], [-10, -5, 0, 2, -3] ] |
| Выходные данные | rightmost\_min\_positions | [3, 4, 1, 3, 4] |

#### **Листинг программы:**

Программа создает вложенную последовательность и находит минимальные значения в каждой строке. Правое расположение минимального элемента определяется с помощью метода index() и выводится на экран.

#### **Код программы:**

import random  
def find\_rightmost\_min\_element(matrix):  
rightmost\_min\_positions = []  
for row in matrix:  
min\_val = min(row)  
min\_pos = len(row) - 1 - row[::-1].index(min\_val)  
rightmost\_min\_positions.append(min\_pos)  
return rightmost\_min\_positions  
matrix = [[random.randint(-20, 20) for \_ in range(5)] for \_ in range(5)]  
rightmost\_min\_positions = find\_rightmost\_min\_element(matrix)  
for row in matrix:  
print(row)  
for i, pos in enumerate(rightmost\_min\_positions):  
print(f"Строка {i+1}, номер столбца с самым правым наименьшим элементом: {pos}")

### **Задача 26. Условие.**

Программа для вычисления среднего арифметического элементов матрицы 5x5, больших числа 20.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от 1 до 60 |
| Промежуточные данные | elements\_above\_20 | Элементы больше 20 | список целых чисел |  |
| Выходные данные | average\_above\_20 | Среднее арифметическое | вещественное число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, 35, 18, 42, 7], [30, 12, 32, 21, 22], [9, 22, 37, 29, 40], [3, 27, 33, 19, 28], [10, 15, 38, 44, 13] ] |
| Выходные данные | average\_above\_20 | 31.58 |

#### **Листинг программы:**

Заполняется матрица случайными числами. Затем находятся все элементы больше 20, и вычисляется их среднее арифметическое.

#### **Код программы:**

import random  
matrix = [[random.randint(1, 60) for \_ in range(5)] for \_ in range(5)]  
elements\_above\_20 = [matrix[i][j] for i in range(5) for j in range(5) if matrix[i][j] > 20]  
average\_above\_20 = sum(elements\_above\_20) / len(elements\_above\_20) if elements\_above\_20 else 0  
for row in matrix:  
print(row)  
print(f"Среднее арифметическое элементов больше 20: {average\_above\_20}")

### **Задача 27. Условие.**

Программа для поворота матрицы 5x5 на 90 градусов по часовой стрелке.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от 1 до 30 |
| Выходные данные | rotated\_A | Повернутая матрица | двумерный список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, 12, 15, 23, 5], [30, 7, 14, 11, 19], [10, 4, 29, 28, 18], [16, 27, 31, 8, 32], [9, 26, 33, 13, 17] ] |
| Выходные данные | rotated\_A | [ [9, 16, 10, 30, 25], [26, 27, 4, 7, 12], [33, 31, 29, 14, 15], [13, 8, 28, 11, 23], [17, 32, 18, 19, 5] ] |

#### **Листинг программы:**

Матрица заполняется случайными числами и затем поворачивается на 90 градусов.

#### **Код программы:**

import random  
matrix = [[random.randint(1, 30) for \_ in range(5)] for \_ in range(5)]  
rotated\_A = [[matrix[4 - j][i] for j in range(5)] for i in range(5)]  
for row in matrix:  
print("Исходная матрица:", row)  
for row in rotated\_A:  
print("Повернутая матрица:", row)

### **Задача 28. Условие.**

Программа для проверки наличия отрицательных чисел ниже главной диагонали матрицы 5x5.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от -15 до 40 |
| Выходные данные | negatives\_below | Наличие отрицательных чисел ниже диагонали | логическое значение |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, 35, 12, 48, 7], [6, -10, 32, 11, 43], [-41, 22, 16, 18, 27], [30, 24, -13, 31, 20], [5, -40, 33, 29, 17] ] |
| Выходные данные | negatives\_below | Да |

#### **Листинг программы:**

Программа проверяет наличие отрицательных чисел ниже главной диагонали и выводит результат.

#### **Код программы:**

import random  
matrix = [[random.randint(1, 40) if i <= j else random.randint(-15, 0) for j in range(5)] for i in range(5)]  
negatives\_below = any(matrix[i][j] < 0 for i in range(1, 5) for j in range(i))  
for row in matrix:  
print(row)  
print("Есть ли отрицательные числа ниже главной диагонали?", "Да" if negatives\_below else "Нет")

### **Задача 29. Условие.**

Программа для замены строк с хотя бы одним отрицательным элементом на нули в матрице 5x5.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от -15 до 30 |
| Выходные данные | matrix | Обновленная матрица | двумерный список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, -12, 15, -23, 5], [30, 7, 14, 11, -19], [10, 4, 29, 28, 18], [16, 27, -31, 8, 32], [9, -26, 33, 13, 17] ] |
| Выходные данные | matrix | [ [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [10, 4, 29, 28, 18], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0] ] |

#### **Листинг программы:**

Программа заменяет строки с отрицательными элементами на строку из нулей.

#### **Код программы:**

import random  
matrix = [[random.randint(-15, 30) for \_ in range(5)] for \_ in range(5)]  
for row in matrix:  
print(row)  
for i in range(5):  
if any(matrix[i][j] < 0 for j in range(5)):  
matrix[i] = [0] \* 5  
for row in matrix:  
print("После замены:", row)

### **Задача 30. Условие.**

Программа для нахождения количества, суммы и среднего арифметического отрицательных чисел матрицы 5x5.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | matrix | Вложенная последовательность 5x5 | двумерный список | значения от -20 до 40 |
| Промежуточные данные | negative\_numbers | Отрицательные числа | список целых чисел |  |
| Выходные данные | negative\_count, negative\_sum, negative\_avg | Количество, сумма и среднее арифметическое | целое, вещественное |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | matrix | [ [25, 35, -12, -48, 7], [-6, -19, 32, -11, -43], [41, -22, 16, 18, -27], [30, 24, 13, 31, 20], [-5, 40, 33, 29, -17] ] |
| Выходные данные | negative\_count, negative\_sum, negative\_avg | 10, -253, -25.3 |

#### **Листинг программы:**

Программа ищет все отрицательные числа, подсчитывает их количество, сумму и среднее арифметическое.

#### **Код программы:**

import random  
matrix = [[random.randint(-20, 40) for \_ in range(5)] for \_ in range(5)]  
negative\_numbers = [matrix[i][j] for i in range(5) for j in range(5) if matrix[i][j] < 0]  
negative\_count = len(negative\_numbers)  
negative\_sum = sum(negative\_numbers)  
negative\_avg = negative\_sum / negative\_count if negative\_count > 0 else 0  
for row in matrix:  
print(row)  
print(f"Количество отрицательных чисел: {negative\_count}")  
print(f"Сумма отрицательных чисел: {negative\_sum}")  
print(f"Среднее арифметическое отрицательных чисел: {negative\_avg:.2f}")