Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии

Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**дисциплины**

**«Искусственный интеллект и машинное обучение»**

**Вариант 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Выполнил:  Борцов Богдан Михайлович  2 курс, группа ИТС-б-о-23-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль) «Инфокоммуникационные системы и сети», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |  |
|  | | Проверил:  Доцент департамента цифровых, робототехнических систем и электроники Воронкин Р. А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |  |
|  |  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2025 г.

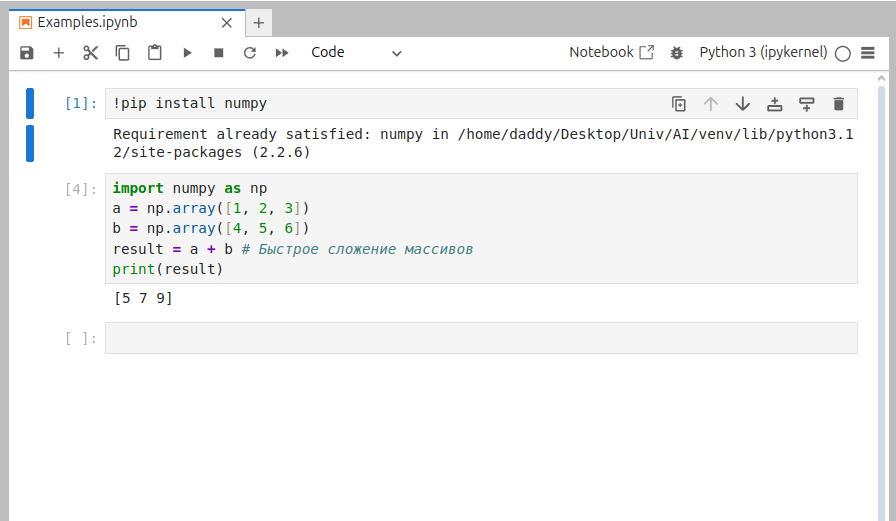
**Тема**: Основы работы с библиотекой NumPy

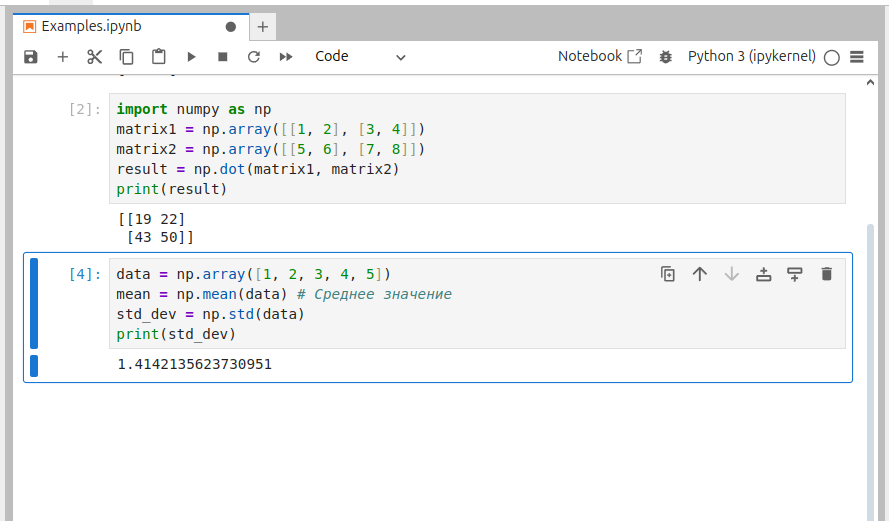
**Цель:** исследовать базовые возможности библиотеки NumPy языка программирования Python.

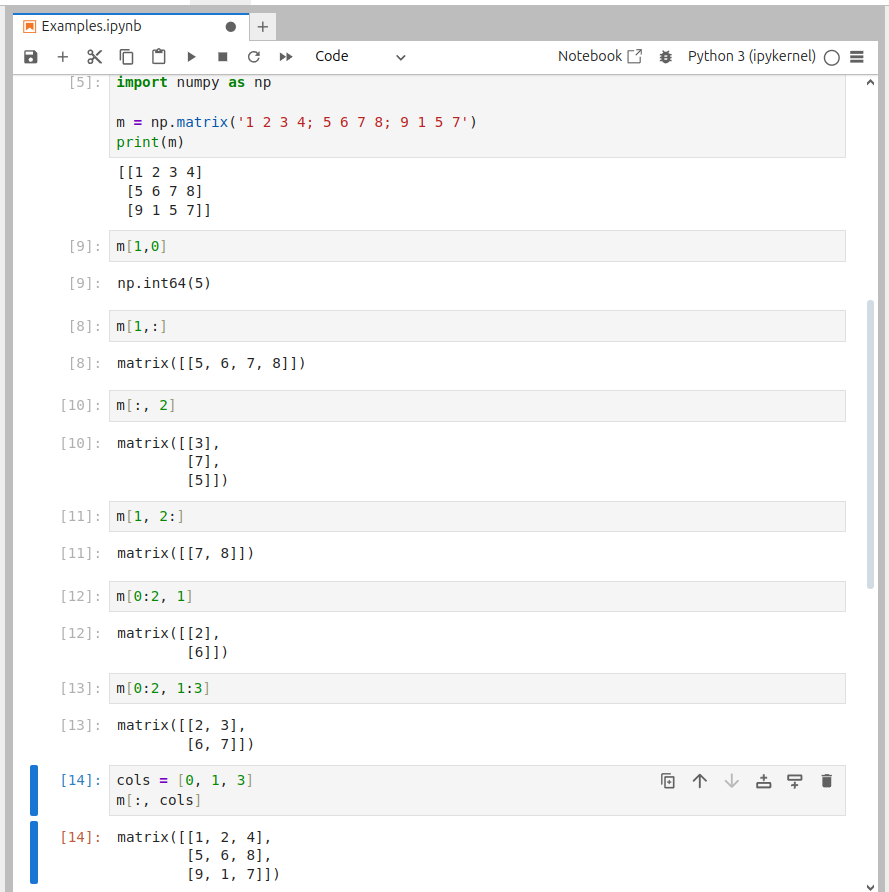
Ссылка на репозиторий: https://github.com/REPONCFU/ai-jlab2

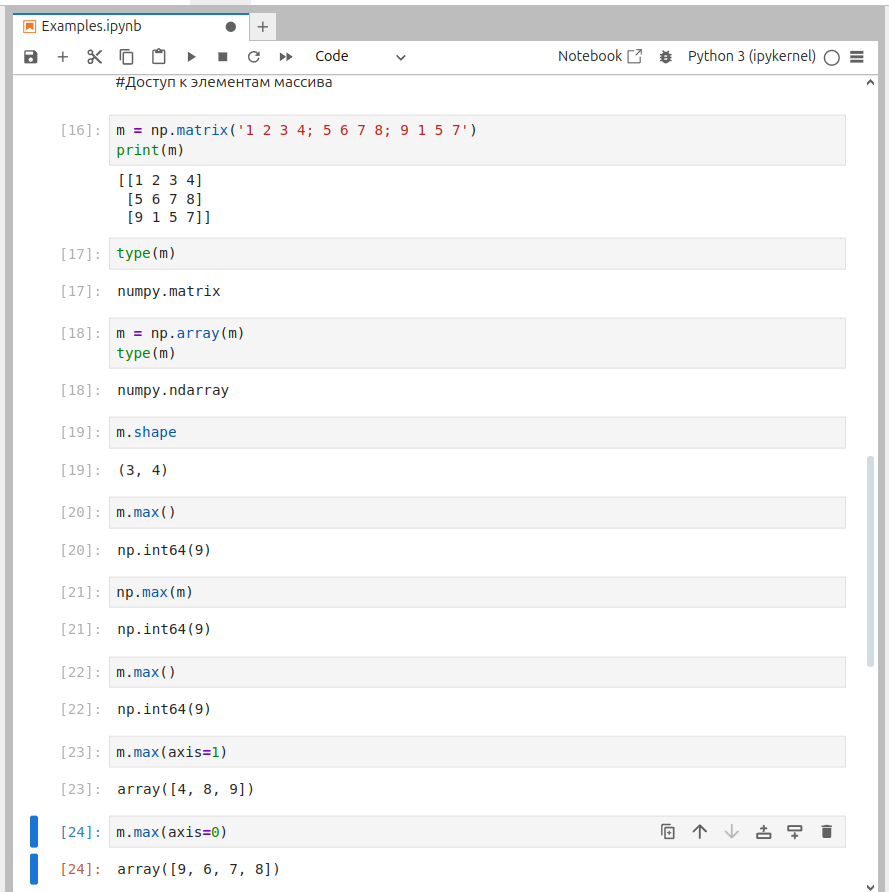
**Порядок выполнения работы:**

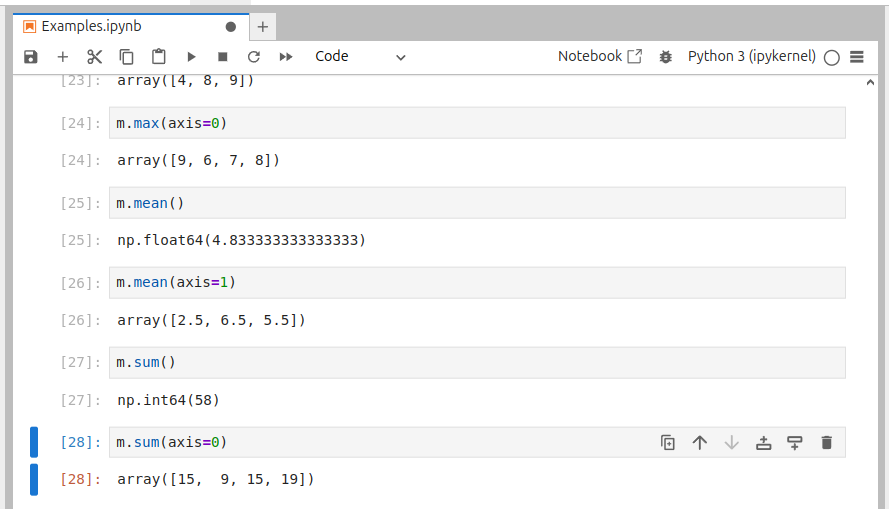
1. Ознакомление с методическими указаниями
2. Выполнение методических указаний

Рисунок 1. Векторизация NumPy

Рисунок 2. Матричные операции и статистические операции

Рисунок 3. Достаем из матрицы необходимые данные по координатам

Рисунок 4. Расчет статистик по данным в массиве

Рисунок 5. Методы расчета статистик

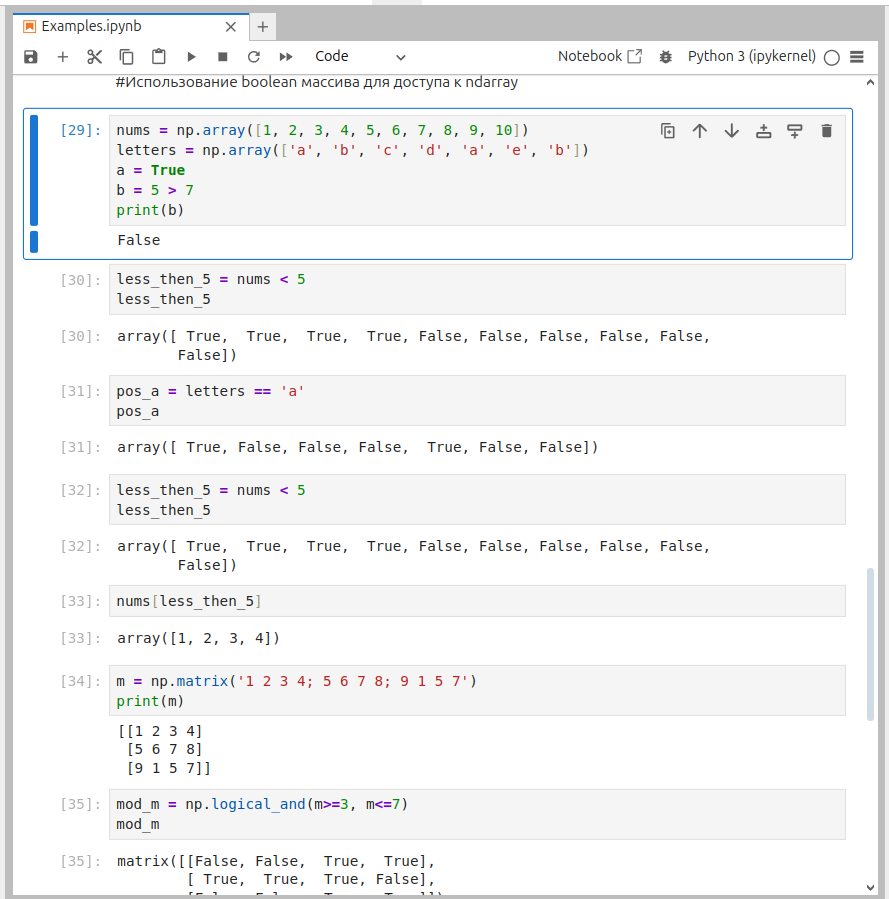


Рисунок 6. Использование boolean массива для доступа к ndarray

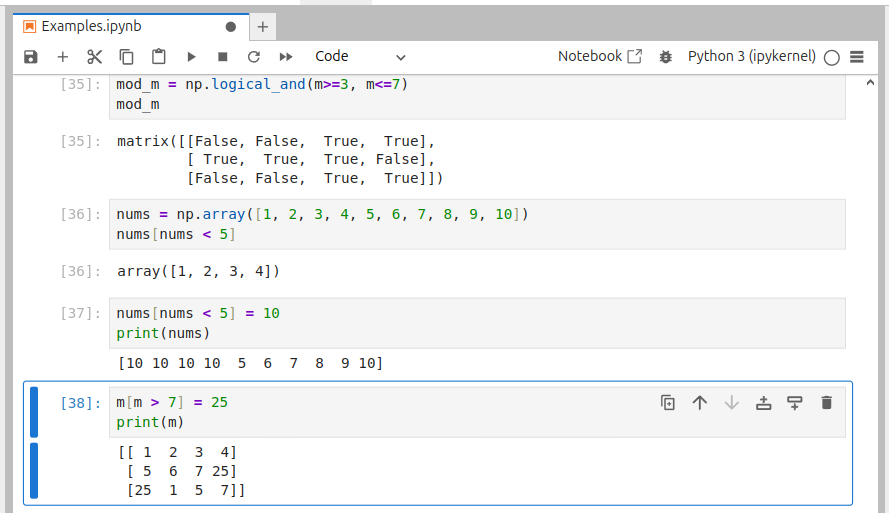


Рисунок 7. Дополнительные юзкейсы boolean

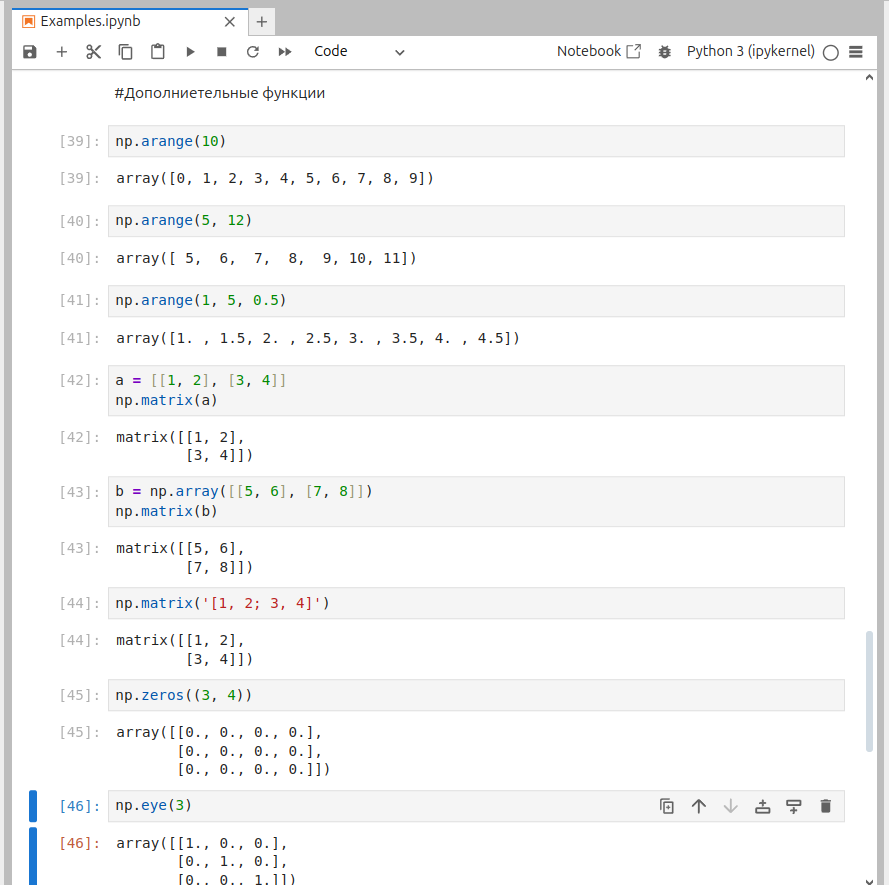


Рисунок 8. Дополнительные функции

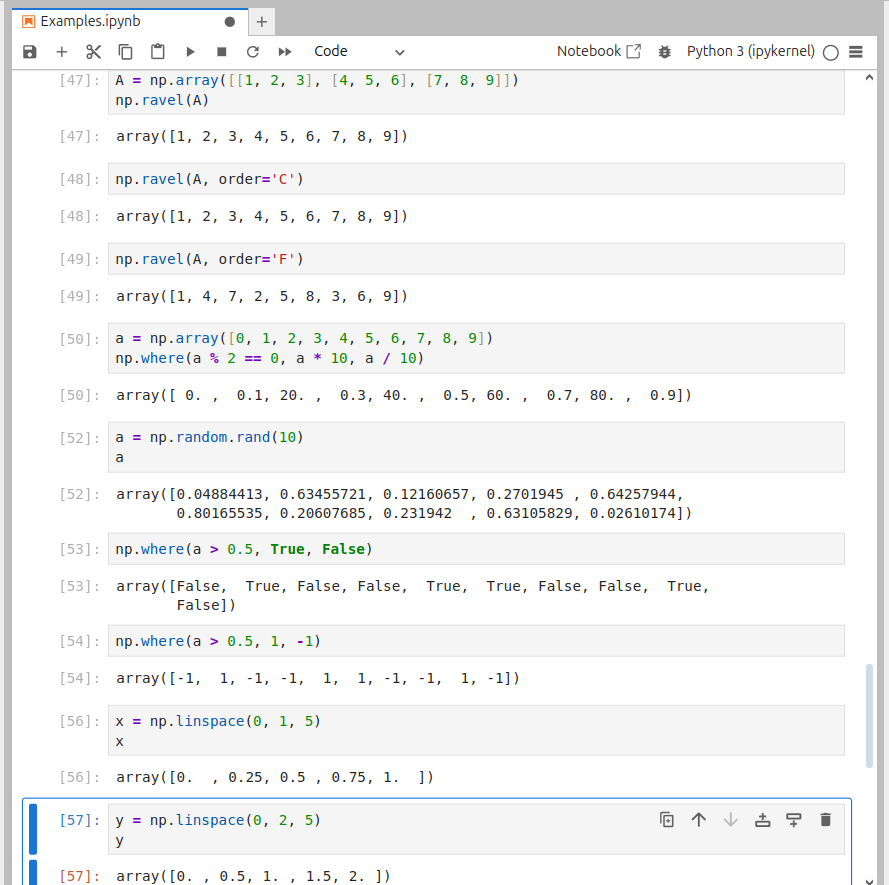


Рисунок 9. Дополнительные функции

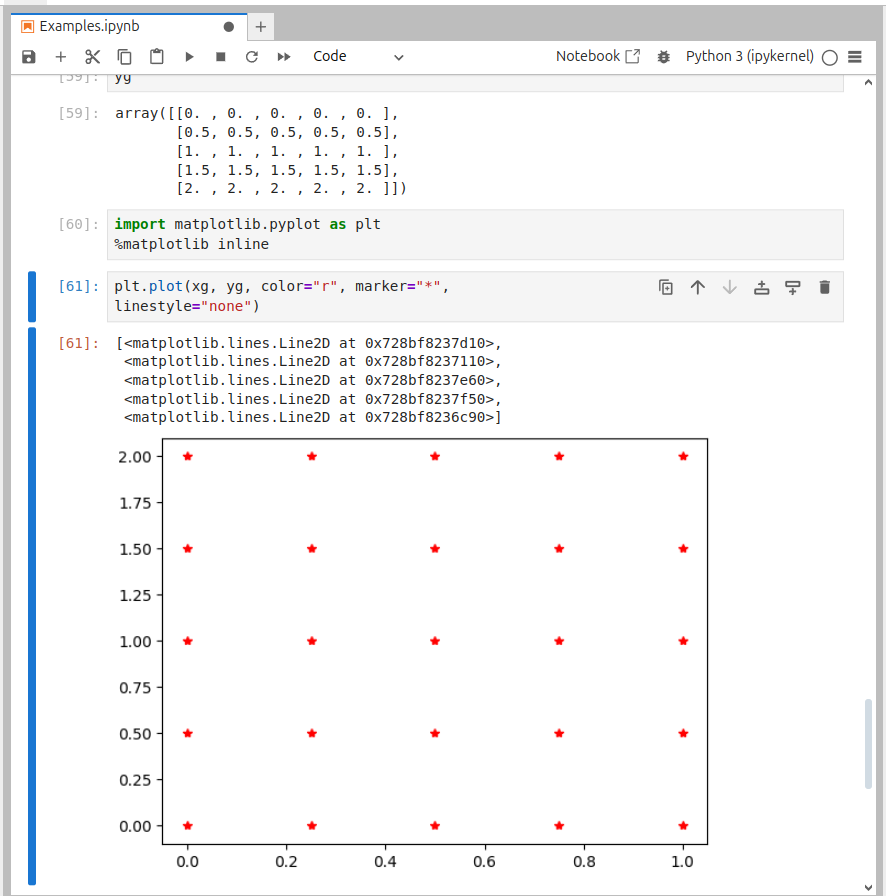


Рисунок 10. Построение графика mathplotlib

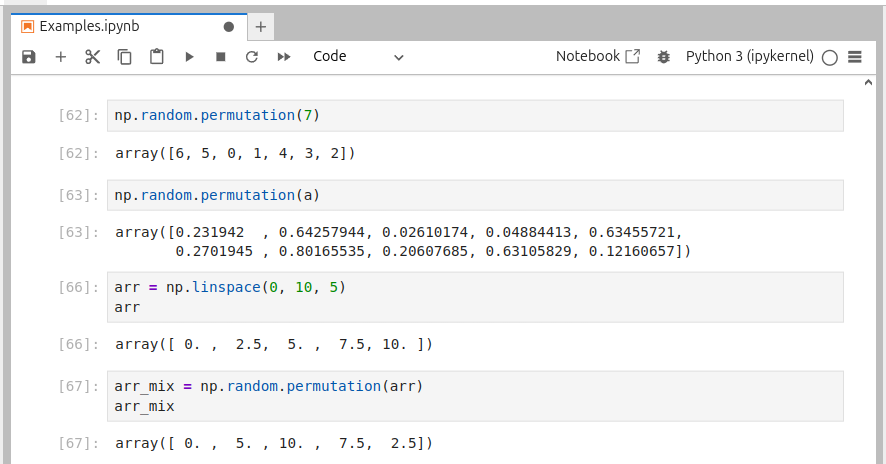


Рисунок 11. Работа с рандомом и миксингом

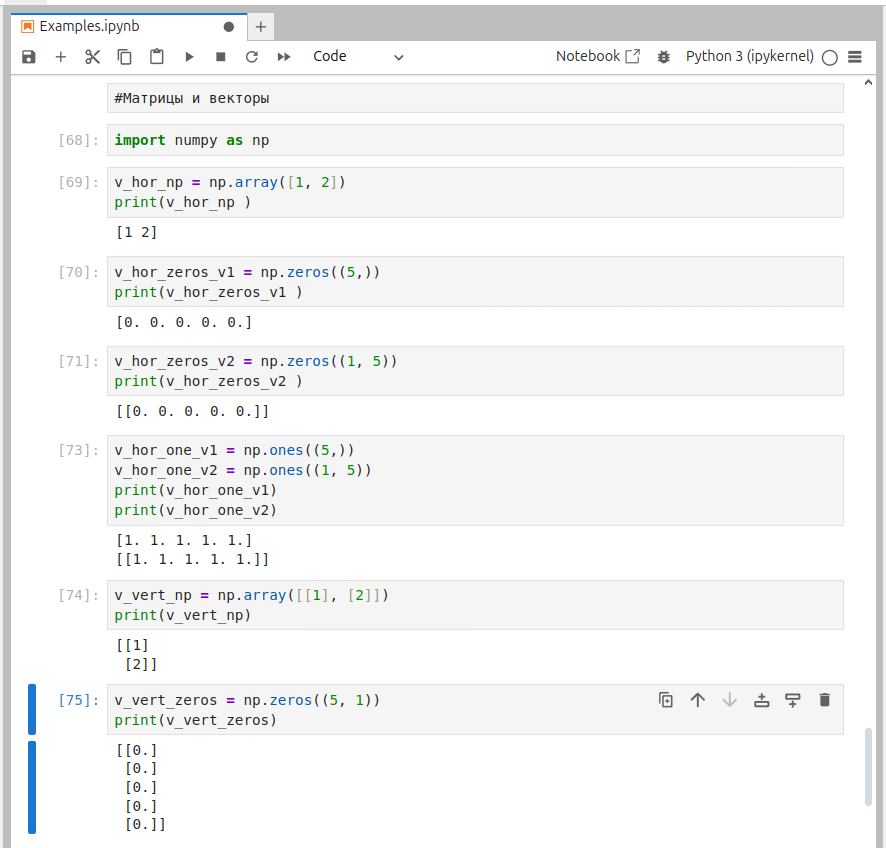


Рисунок 12. Работа с векторами

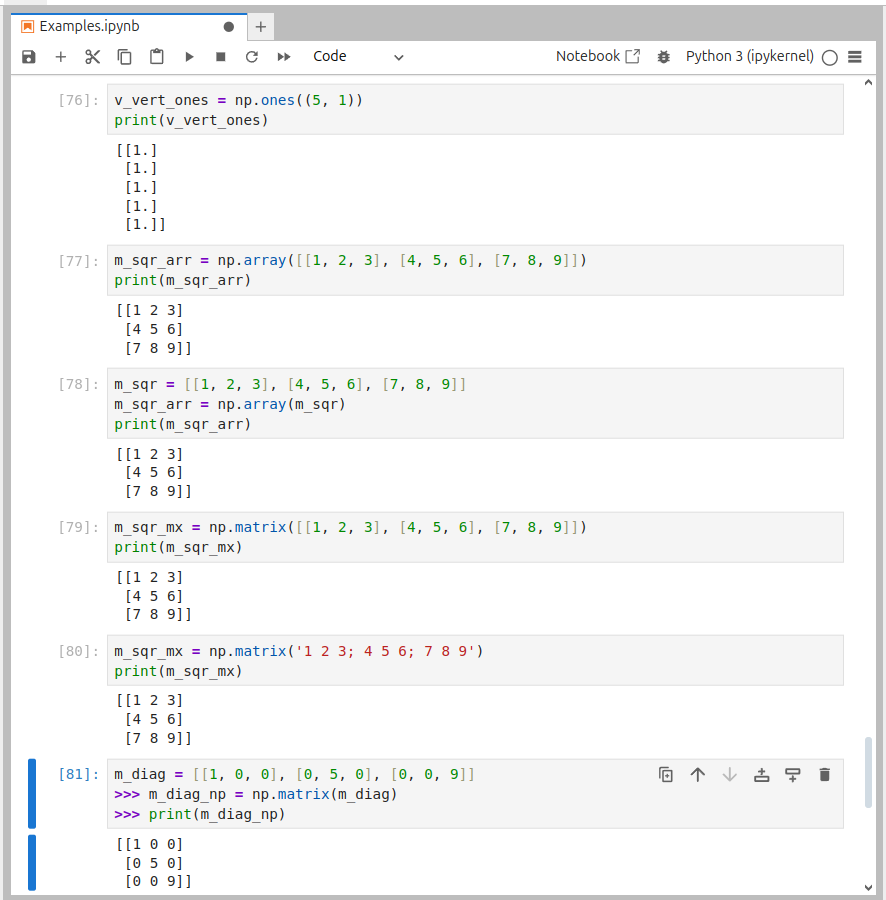


Рисунок 13. Работа с матрицами

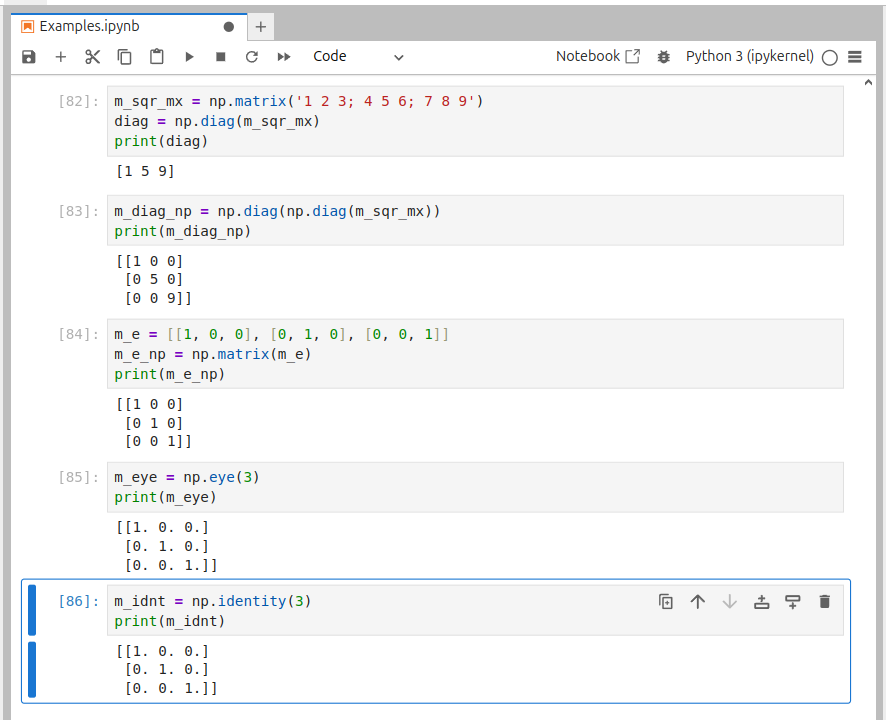


Рисунок 14. Единичные матрицы

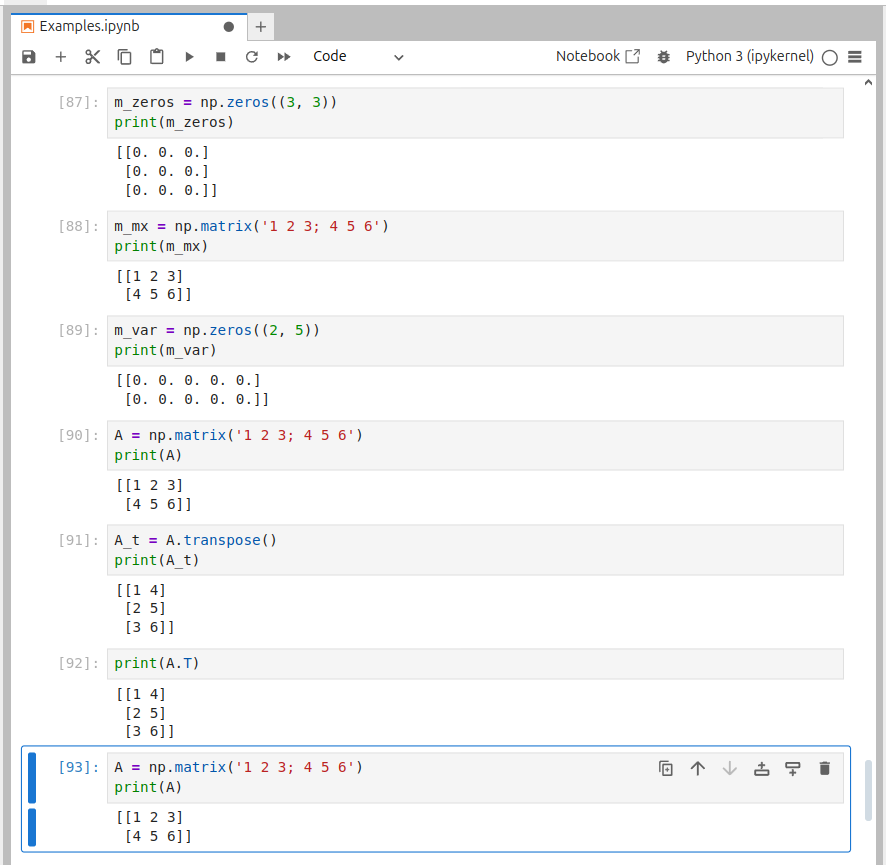


Рисунок 15. Операции над матрицами

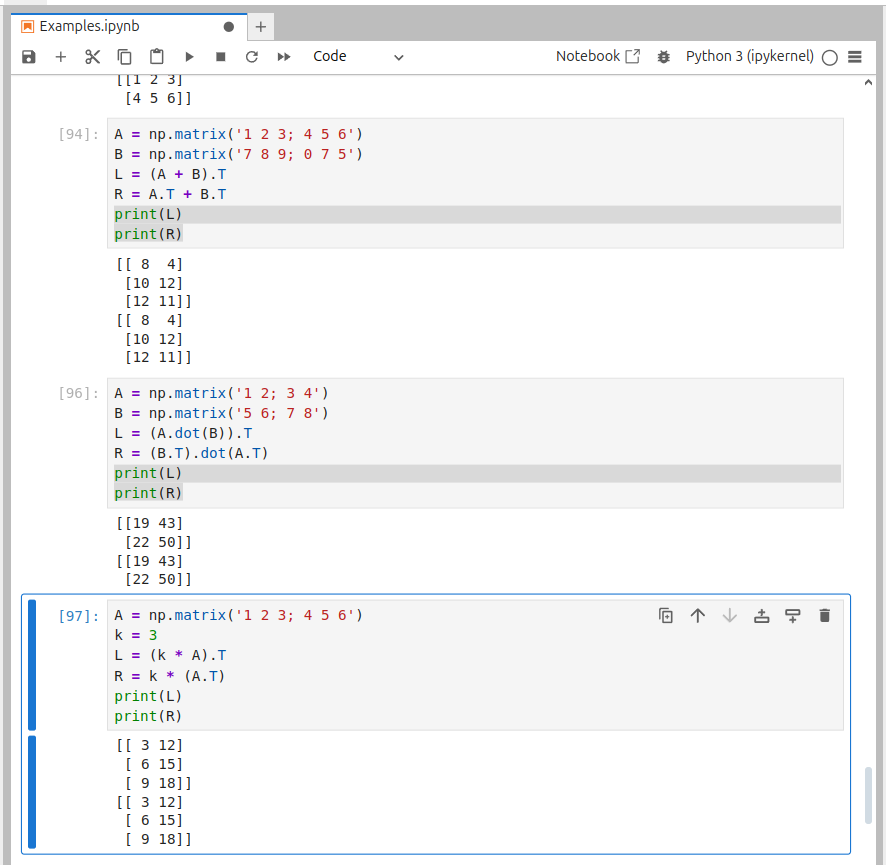


Рисунок 16. Транспонирование матриц

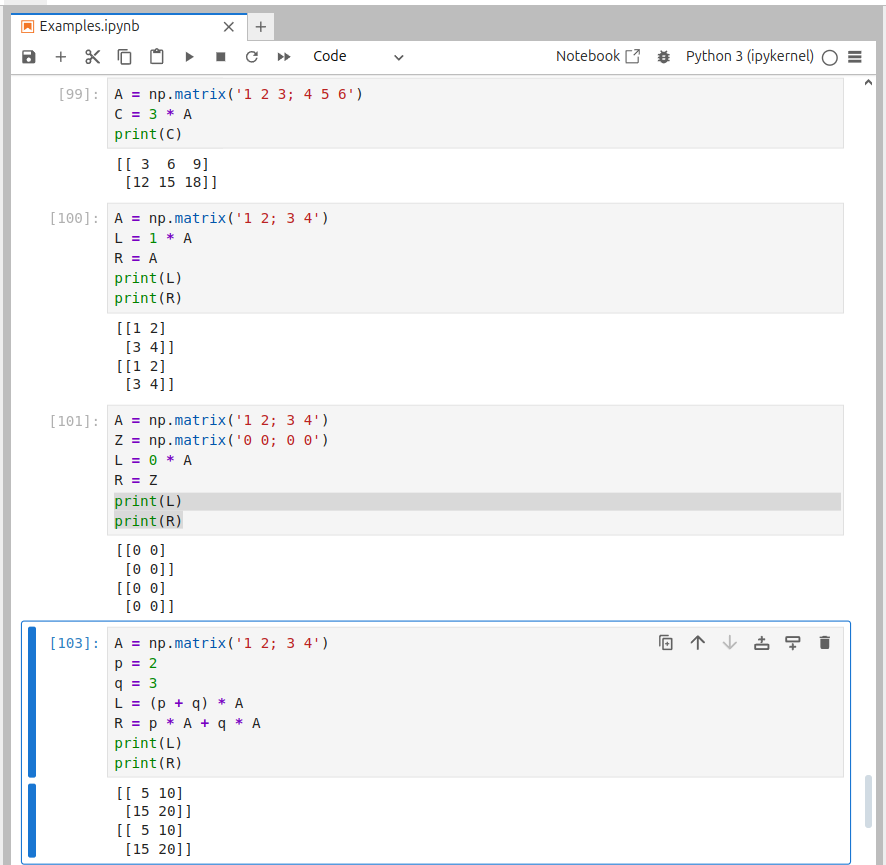


Рисунок 17. Свойства умножения матриц

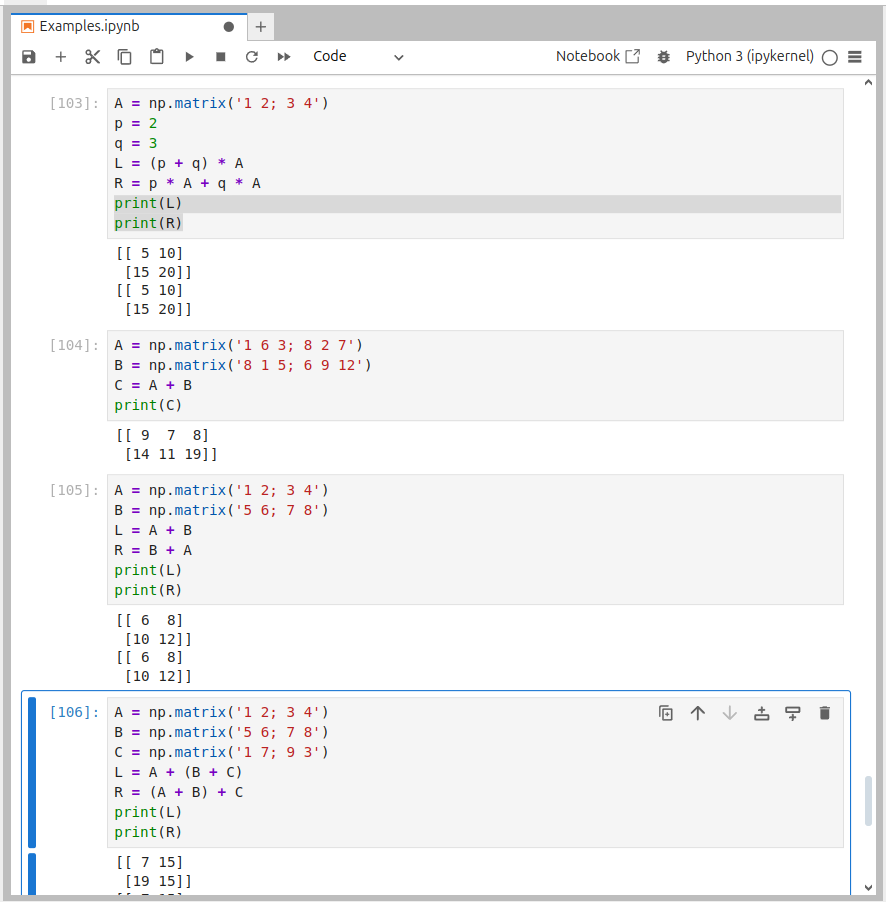


Рисунок 18. Свойства сложения матриц

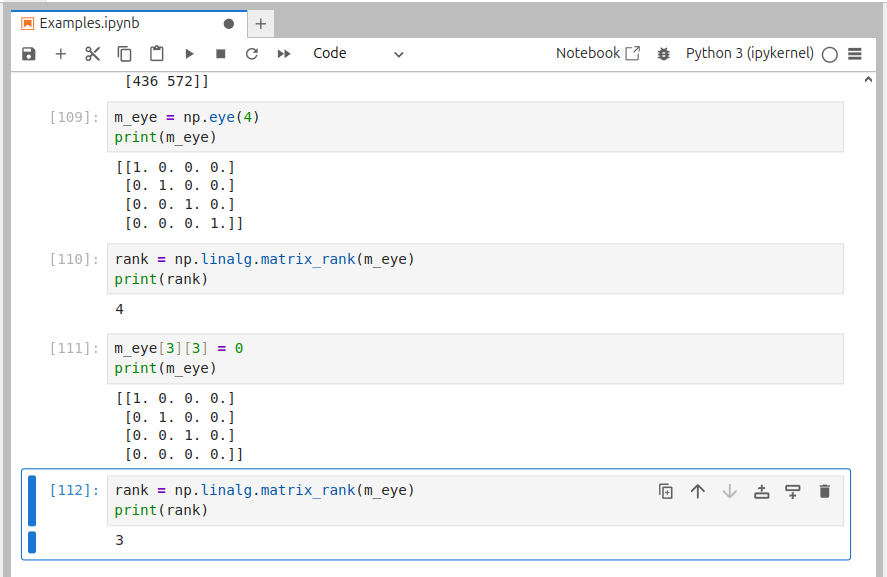


Рисунок 19. Ранг матрицы

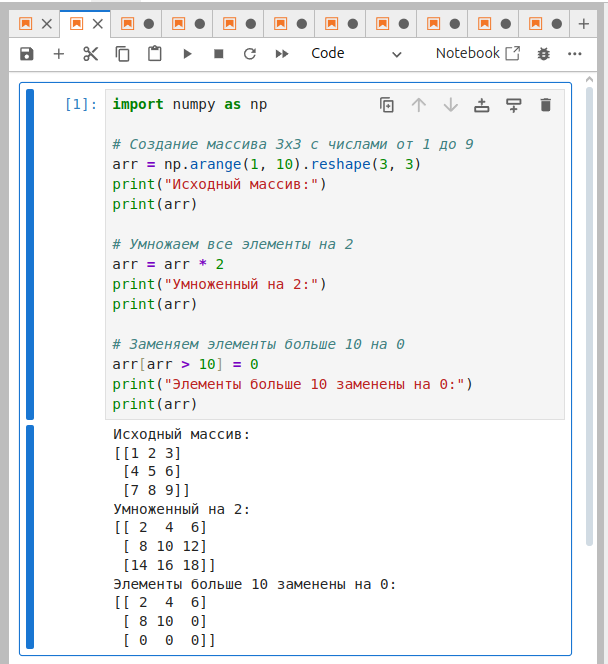


Рисунок 20. Практическое задание 1

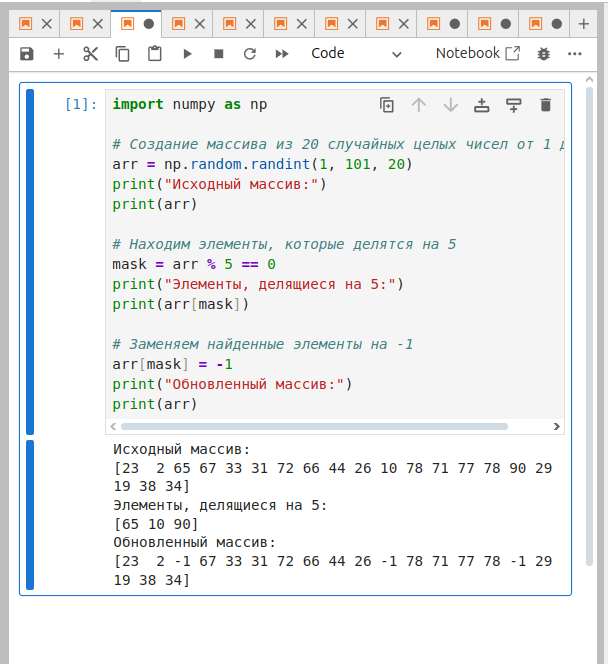


Рисунок 21. Практическое задание 2

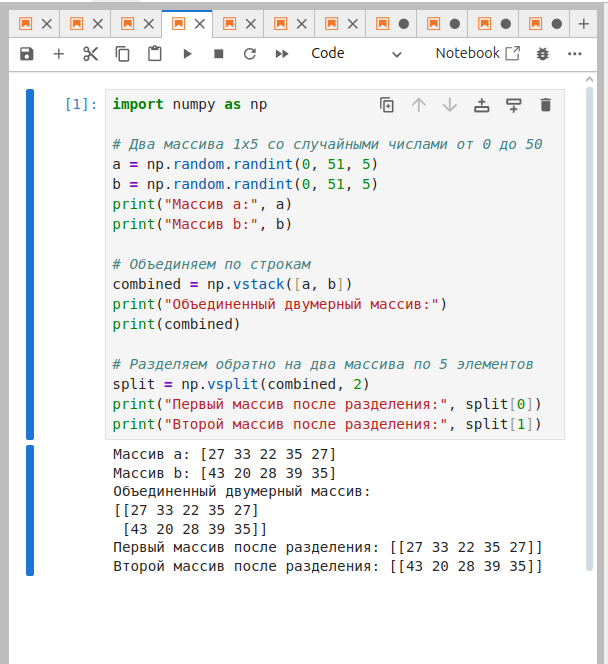


Рисунок 22. Практическое задание 3

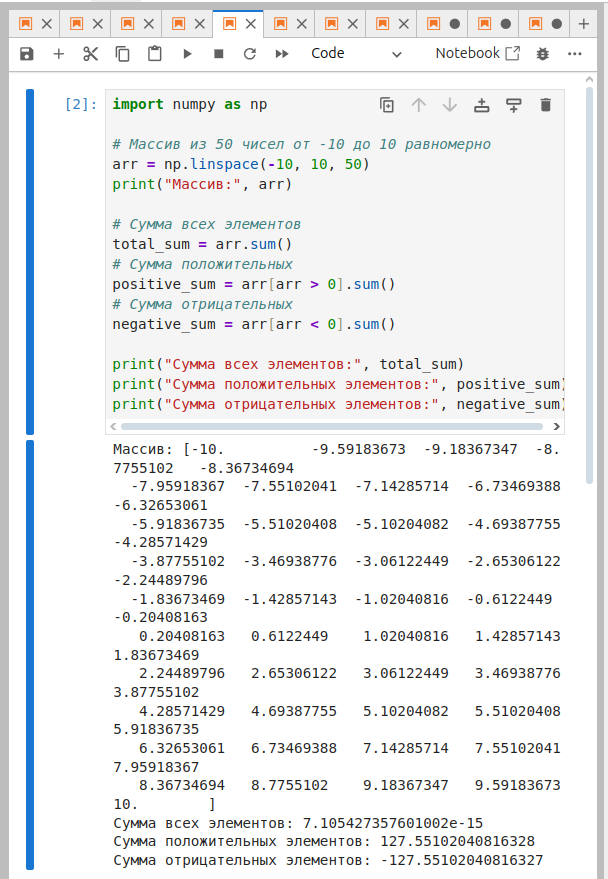


Рисунок 23. Практическое задание 4

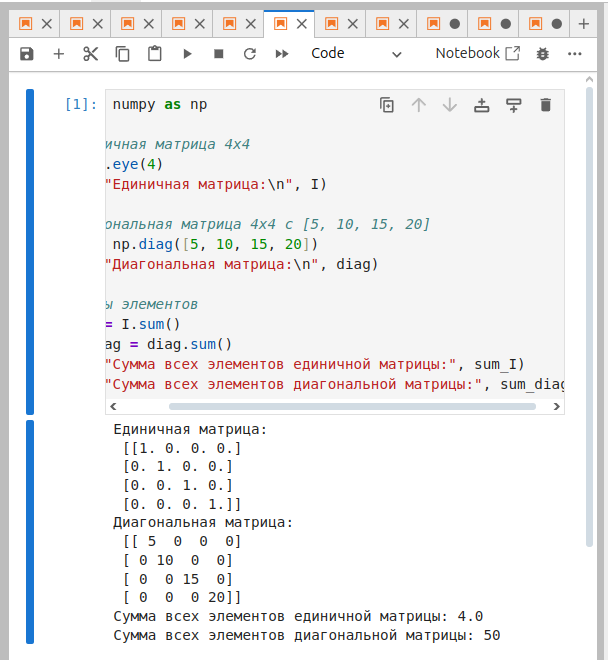


Рисунок 24. Практическое задание 5

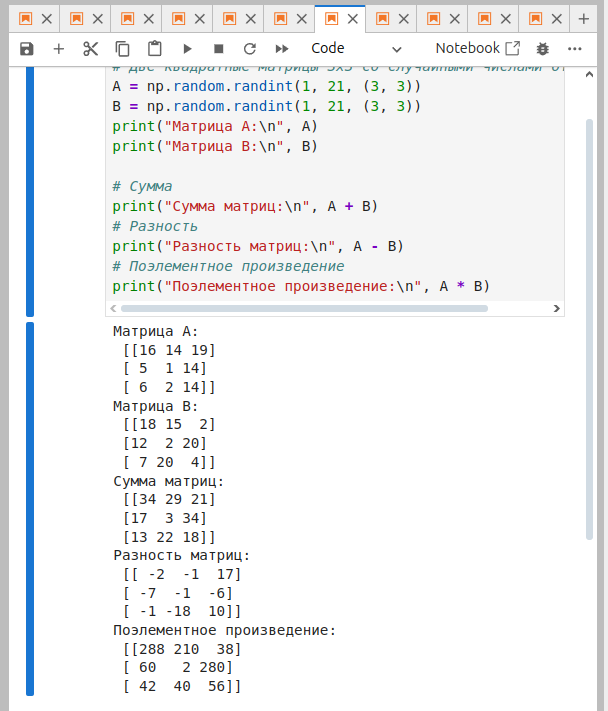


Рисунок 25. Практическое задание 6

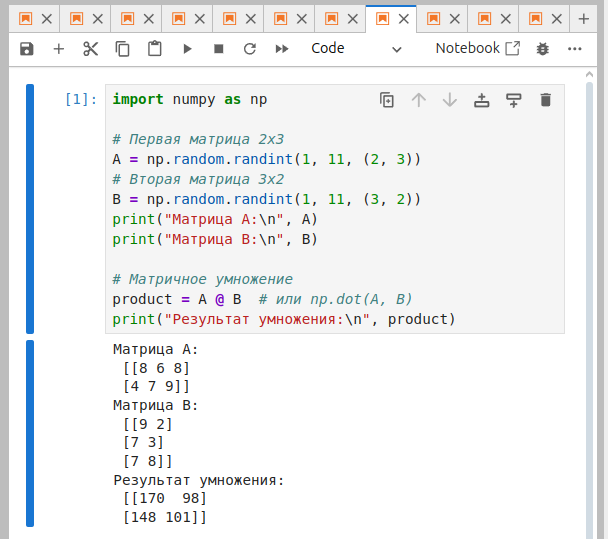


Рисунок 26. Практическое задание 7

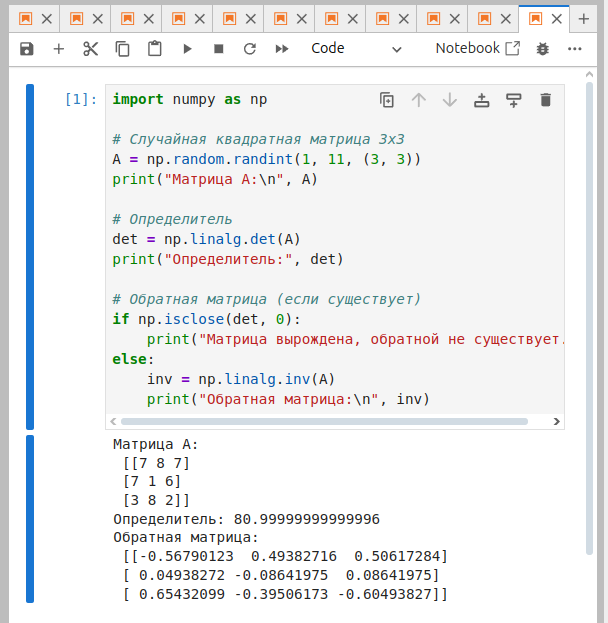


Рисунок 27. Практическое задание 8

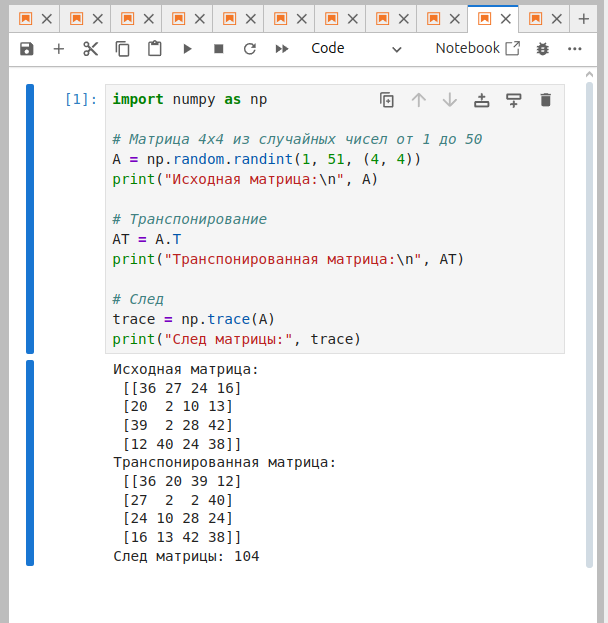
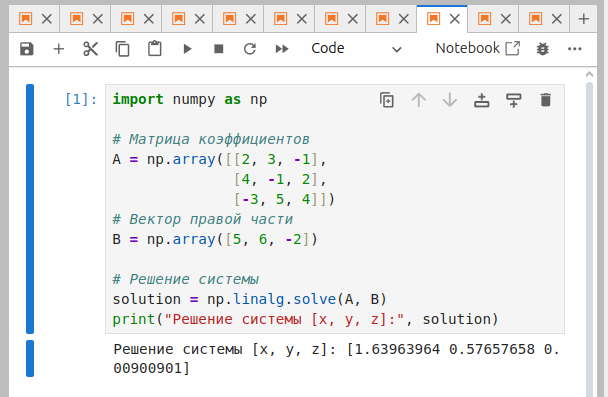


Рисунок 28. Практическое задание 9

Рисунок 29. Практическое задание 10

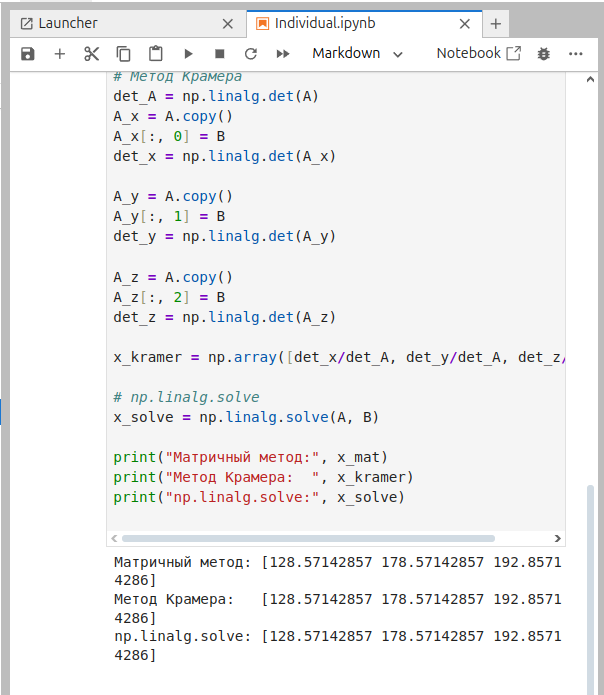


Рисунок 30. Индивидуальное задание

**Ответы на контрольные вопросы:**

**1. Каково назначение библиотеки NumPy?**  
 NumPy — это основная библиотека для научных вычислений в Python. Она предоставляет удобные структуры данных (главным образом массивы), высокоэффективные математические операции, средства для линейной алгебры, статистики, работы с большими наборами данных и интеграции с другими научными пакетами.

**2. Что такое массивы ndarray?**  
 ndarray — основной тип данных в NumPy, многомерный массив, который хранит элементы одного типа (например, только float или только int). Он похож на обычные списки Python, но поддерживает операции над всеми элементами сразу, быстрые вычисления, работу с матрицами и т.д.

**3. Как осуществляется доступ к частям многомерного массива?**  
 Доступ осуществляется с помощью индексов и срезов. Для двумерного массива: arr[0, 1] — элемент первой строки, второго столбца. Для выделения целого столбца или строки: arr[0, :] (вся первая строка), arr[:, 1] (весь второй столбец).

**4. Как осуществляется расчет статистик по данным?**  
NumPy содержит встроенные методы для подсчета статистик:

* arr.sum() — сумма,
* arr.mean() — среднее,
* arr.std() — стандартное отклонение,
* arr.min()/arr.max() — минимум/максимум,
* и др.  
  Часто есть параметр axis, указывающий направление подсчета.

**5. Как выполняется выборка данных из массивов ndarray?**  
С помощью индексов, срезов (arr[1:5], arr[:,0]), булевых масок (arr[arr > 5]), списка индексов (arr[[0,2,3]]).

**6. Приведите основные виды матриц и векторов. Опишите способы их создания в языке Python.**

* **Вектор:** одномерный массив (np.array([1,2,3]))
* **Матрица:** двумерный массив (np.array([[1,2],[3,4]]))
* **Нулевая матрица:** np.zeros((m, n))
* **Единичная матрица:** np.eye(n)
* **Диагональная:** np.diag([1,2,3])
* **Матрица с одинаковыми элементами:** np.full((m, n), val)

**7. Как выполняется транспонирование матриц?**  
С помощью свойства .T:

A = np.array([[1,2],[3,4]])

A\_T = A.T

**8. Приведите свойства операции транспонирования матриц.**

* Транспонирование транспонированной матрицы возвращает исходную: (A.T).T = A
* (A + B).T = A.T + B.T
* (kA).T = kA.T
* (AB).T = B.T @ A.T

**9. Какие имеются средства в библиотеке NumPy для выполнения транспонирования матриц?**

* A.T
* np.transpose(A)

**10. Какие существуют основные действия над матрицами?**

* Сложение, вычитание
* Умножение на число
* Поэлементное умножение
* Матричное умножение
* Транспонирование
* Определитель, обратная матрица

**11. Как осуществляется умножение матрицы на число?**  
Обычное умножение:

B = 3 \* A

**12. Какие свойства операции умножения матрицы на число?**

* k(A + B) = kA + kB
* (k + l)A = kA + lA
* k(lA) = (kl)A

**13. Как осуществляется операции сложения и вычитания матриц?**  
Элемент к элементу, если одинаковый размер:

C = A + B

D = A - B

**14. Каковы свойства операций сложения и вычитания матриц?**

* Переместительность: A + B = B + A
* Ассоциативность: (A + B) + C = A + (B + C)
* Нейтральность: A + 0 = A

**15. Какие имеются средства в библиотеке NumPy для выполнения операций сложения и вычитания матриц?**

* Операторы +, -
* Функции np.add(A, B), np.subtract(A, B)

**16. Как осуществляется операция умножения матриц?**  
С помощью @ или np.dot(A, B) для матричного (не поэлементного!) произведения.

**17. Каковы свойства операции умножения матриц?**

* Не переместительно (обычно): AB ≠ BA
* Ассоциативно: (AB)C = A(BC)
* Дистрибутивно относительно сложения: A(B + C) = AB + AC

**18. Какие имеются средства в библиотеке NumPy для выполнения операции умножения матриц?**

* A @ B
* np.dot(A, B)
* np.matmul(A, B)

**19. Что такое определитель матрицы? Каковы свойства определителя матрицы?**  
Определитель — числовая характеристика квадратной матрицы, определяющая её обратимость, объём преобразования, решение СЛУ.  
Свойства:

* det(A·B) = det(A)·det(B)
* det(A.T) = det(A)
* det(λA) = λ^n det(A), где n — размер
* Матрица обратима ⇔ det ≠ 0

**20. Какие имеются средства в библиотеке NumPy для нахождения значения определителя матрицы?**

* np.linalg.det(A)

**21. Что такое обратная матрица? Какой алгоритм нахождения обратной матрицы?**  
Обратная матрица A⁻¹ — такая, что A·A⁻¹ = I.  
Стандартный алгоритм: через элементарные преобразования или через присоединённую матрицу/определитель.

**22. Каковы свойства обратной матрицы?**

* (A⁻¹)⁻¹ = A
* (AB)⁻¹ = B⁻¹A⁻¹
* (A.T)⁻¹ = (A⁻¹).T

**23. Какие имеются средства в библиотеке NumPy для нахождения обратной матрицы?**

* np.linalg.inv(A)

**24. Самостоятельно изучите метод Крамера для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений методом Крамера средствами библиотеки NumPy.**  
**Метод Крамера:**  
Для Ax = b, где A — квадратная матрица,

* det\_A = np.linalg.det(A)
* Для каждого x\_i: заменить i-й столбец A на b, вычислить det\_i = np.linalg.det(A\_i), x\_i = det\_i / det\_A

**Алгоритм на Python:**

import numpy as np

def solve\_kramer(A, b):

n = len(b)

det\_A = np.linalg.det(A)

if np.isclose(det\_A, 0):

raise ValueError("Матрица вырождена")

result = []

for i in range(n):

Ai = A.copy()

Ai[:, i] = b

result.append(np.linalg.det(Ai) / det\_A)

return np.array(result)

**25. Самостоятельно изучите матричный метод для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений матричным методом средствами библиотеки NumPy.**  
**Матричный метод:**  
x = A⁻¹·b (если определитель ≠ 0)

**На Python:**

import numpy as np

A = np.array([[...]]) # матрица коэффициентов

b = np.array([...]) # вектор правой части

x = np.linalg.inv(A) @ b

**Вывод:** в результате практики освоены ключевые приёмы работы с массивами и матрицами в NumPy: создание, изменение, объединение, статистика, индексация, маски. Реализованы основные матричные операции — транспонирование, вычисление определителя, получение обратной матрицы, поэлементные и матричные произведения. Проведено сравнение решений систем линейных уравнений разными методами: матричным, Крамера и через стандартную функцию NumPy, все методы дали одинаковый результат. NumPy позволил быстро и точно решить задачи, продемонстрировав высокую эффективность для анализа и вычислений в Python. Освоенные инструменты являются прочной базой для дальнейшего развития в программном анализе и моделировании данных.