

КУРСОВАЯ РАБОТА — ЧАСТЬ 2: МАТРИЧНАЯ МАТЕМАТИКА И РАБОТА С ПАКЕТАМИ

Для первоначального знакомства с пакетом рекомендую прочитать небольшой ман¹ (англ.). Те, у кого плохо с английским, а это очень плохо для людей из мира информационных технологий, могут обратиться к Habr², где есть более-менее полный и грамотный перевод упомянутого мана³:

- **NumPy в Python. Часть 1**⁴. Введение, общая характеристика библиотеки. Установка. Массивы в Python/
- **NumPy в Python. Часть 2**⁵. Создание массивов и манипуляции с элементами.
- **NumPy в Python. Часть 3**⁶. Операторы сравнения и тестирования значений. Матричная и векторная математика.
- **NumPy в Python. Часть 4**⁷. Математика многочленов. Статистика. Случайные числа. Модули SciPy.

Для ускорения процесса написания курса молодого бойца нет смысла «толочь воду в ступе», используя собственное стилевое решение при редактировании еще одного варианта перевода известных основ. Частично материалы упомянутого выше перевода использованы в качестве базы фрагментов текста настоящих указаний, однако, в настолько переработанном виде, что это уже нельзя назвать цитированием

¹ <https://engineering.ucsb.edu/~shell/che210d/numpy.pdf>

² <https://habr.com/>

³ Ман (проф. диалект) — от *manual* (англ.), руководство

⁴ <https://habr.com/ru/post/352678/>

⁵ <https://habr.com/ru/post/353416/>

⁶ <https://habr.com/ru/post/413381/>

⁷ <https://habr.com/ru/post/415373/>

Итак,

КТО ТАКОЙ ЭТОТ NUMPY?

NumPy — это open-source модуль для python, который предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций. Они объединяются в высокоуровневые пакеты. Они обеспечивают функционал, который можно сравнить с функционалом MatLab. NumPy (Numeric Python) предоставляет базовые методы для манипуляции с большими массивами и матрицами. SciPy (Scientific Python) расширяет функционал numpy огромной коллекцией полезных алгоритмов, таких как минимизация, преобразование Фурье, регрессия и другие прикладные математические техники.

До сих пор, решая задачи «Разминки», мы не использовали пакеты расширений и особой настройки наша основная инструментальная среда, а это PyCharm, не требовала дополнительных настроек. Теперь нам потребуется, как минимум, установить нужные пакеты.

Можно воспользоваться оригинальной документацией, Имеется подробный «Путеводитель по NumPy»⁸, который занимает 371 страницу английского текста. Имеется официальный сайт библиотеки⁹, которым, при должном усердии, вы будете пользоваться постоянно. Но для PyCharm разберёмся с этим вопросом подробнее.

⁸ <http://web.mit.edu/dvp/Public/numpybook.pdf>

⁹ <http://www.numpy.org>

НАСТРОЙКА PYCHARM И ПОДКЛЮЧЕНИЕ БИБЛИОТЕК NUMPY И MATHPLOTLIB

Вспомним, как создаётся новый проект в PyCharm, см. Рис. 1-49.

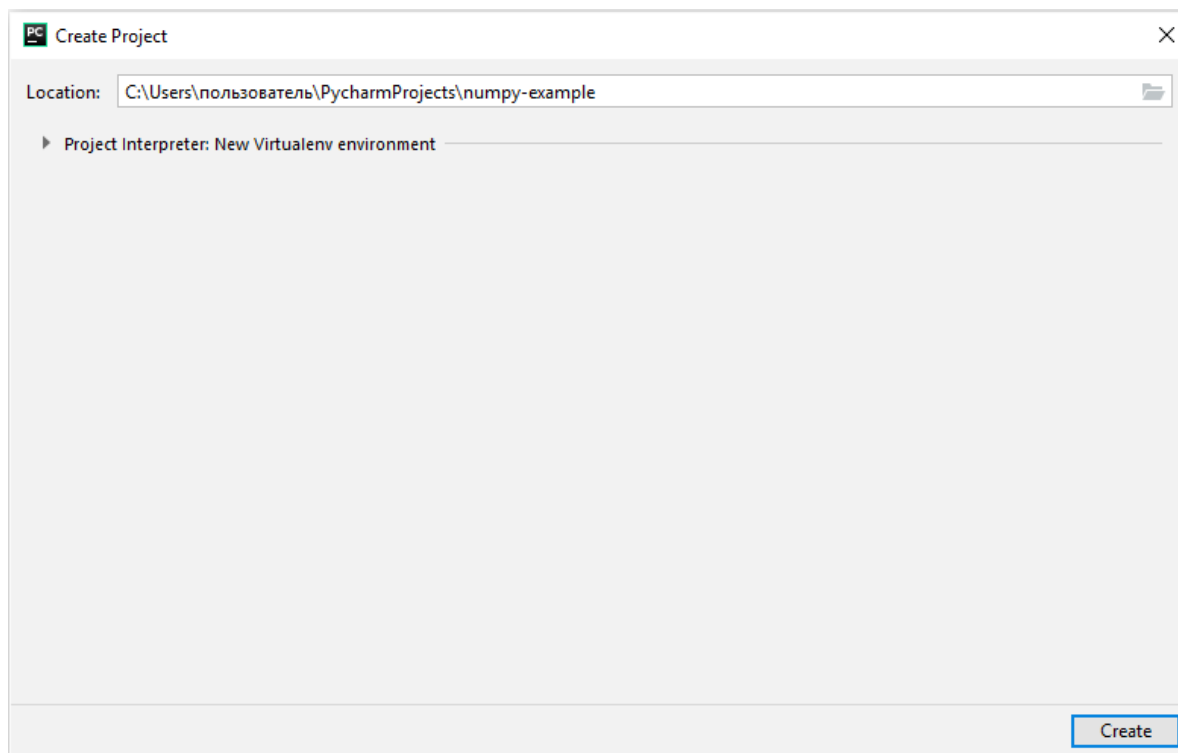


Рис. 2. Создаём и именуем новый проект

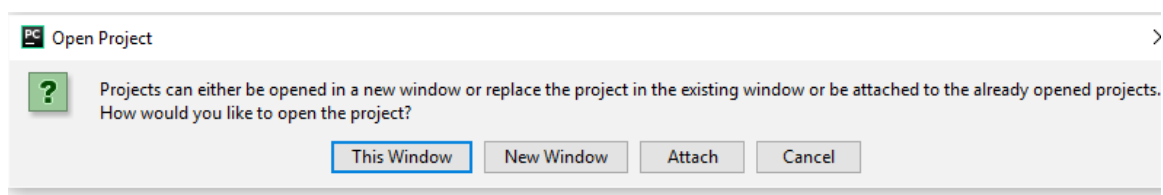


Рис. 3. Новый проект размещаем в текущем окне PyCharm. Не нравится — поступайте по-своему

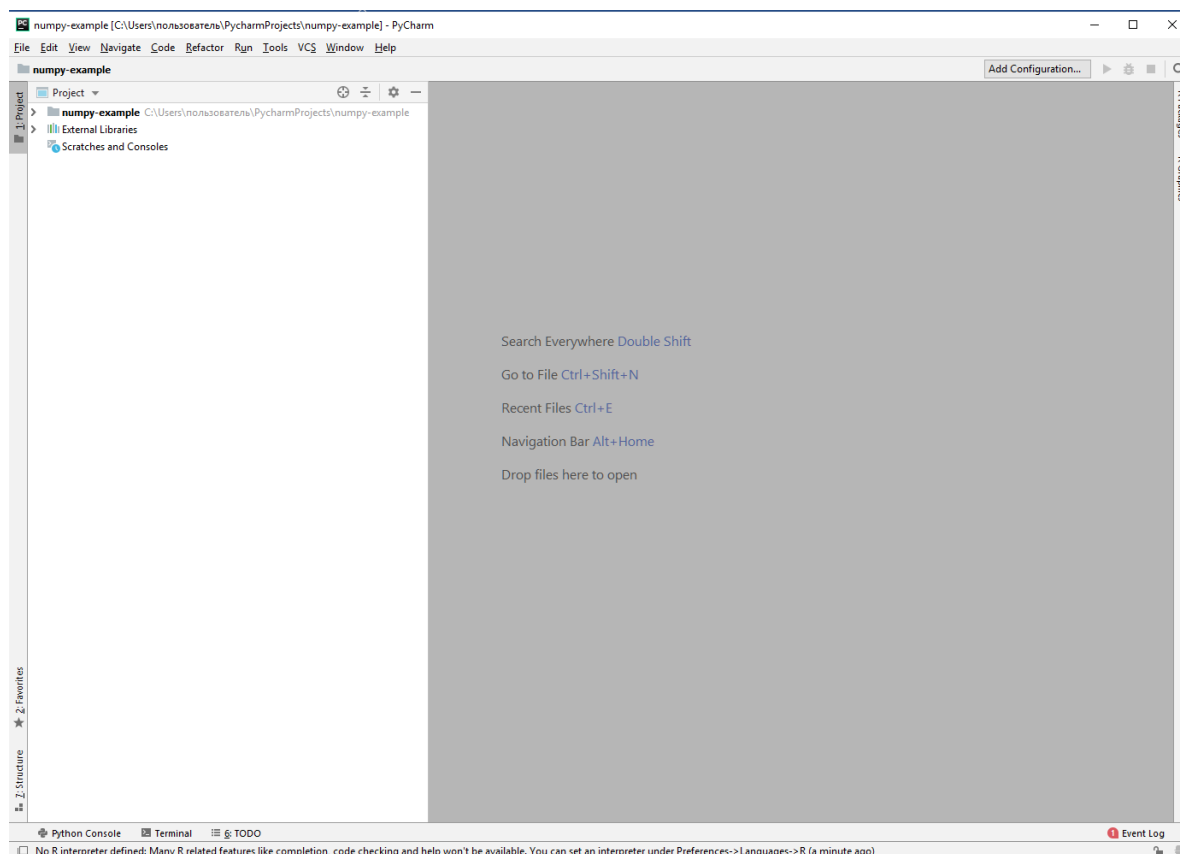


Рис. 4. Новый проект создан, приступаем к реализации

Первое, что мы сделаем, создадим новый проект, куда будем записывать все коды программ. Думаю, что особых трудностей это не вызовет см. рис. 47, 48 и 49.

Далее приступим к настройкам окружения для решения последующих задач, см. рис. 50, 51, 52, 53 и 54, как на путеводитель в интерфейсе — «делай раз, дела два, делай три ...».

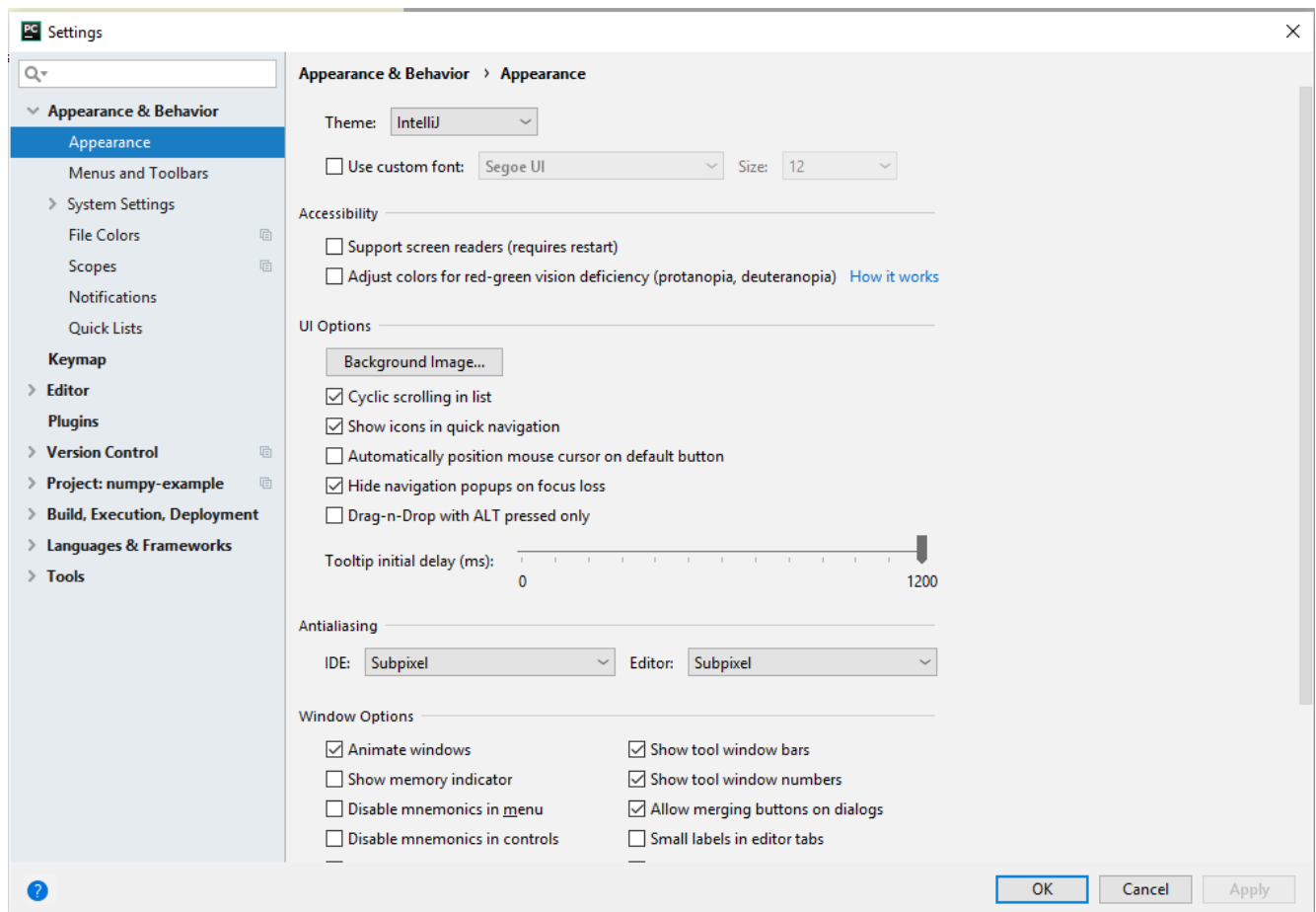


Рис. 5. Настройка окружения проекта

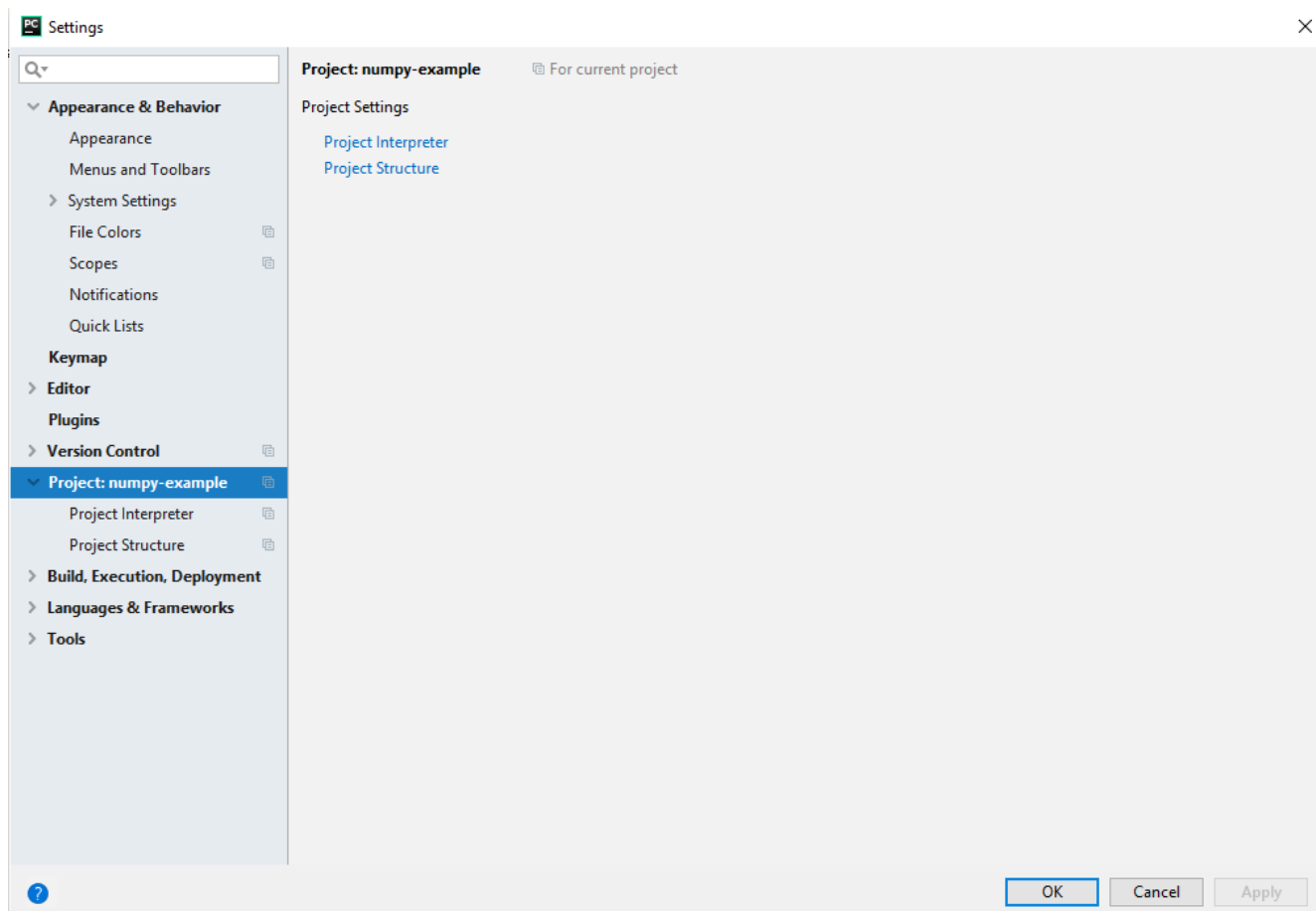


Рис. 6. Очень важно выбрать интерпретатор Project Interpreter

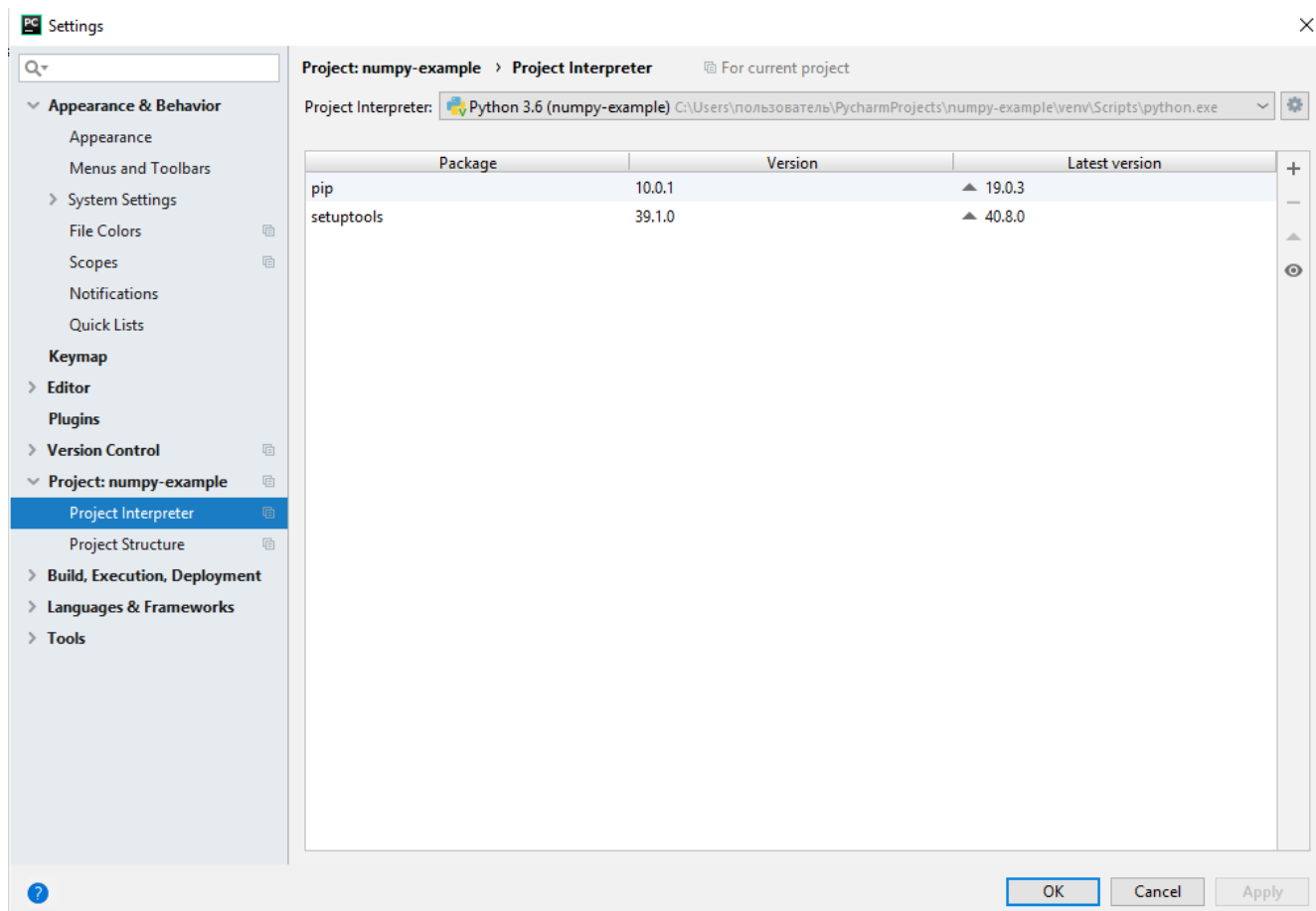


Рис. 7. Выбираем интерпретатор и добавляем пакеты — посмотрите на «+» справа вверху

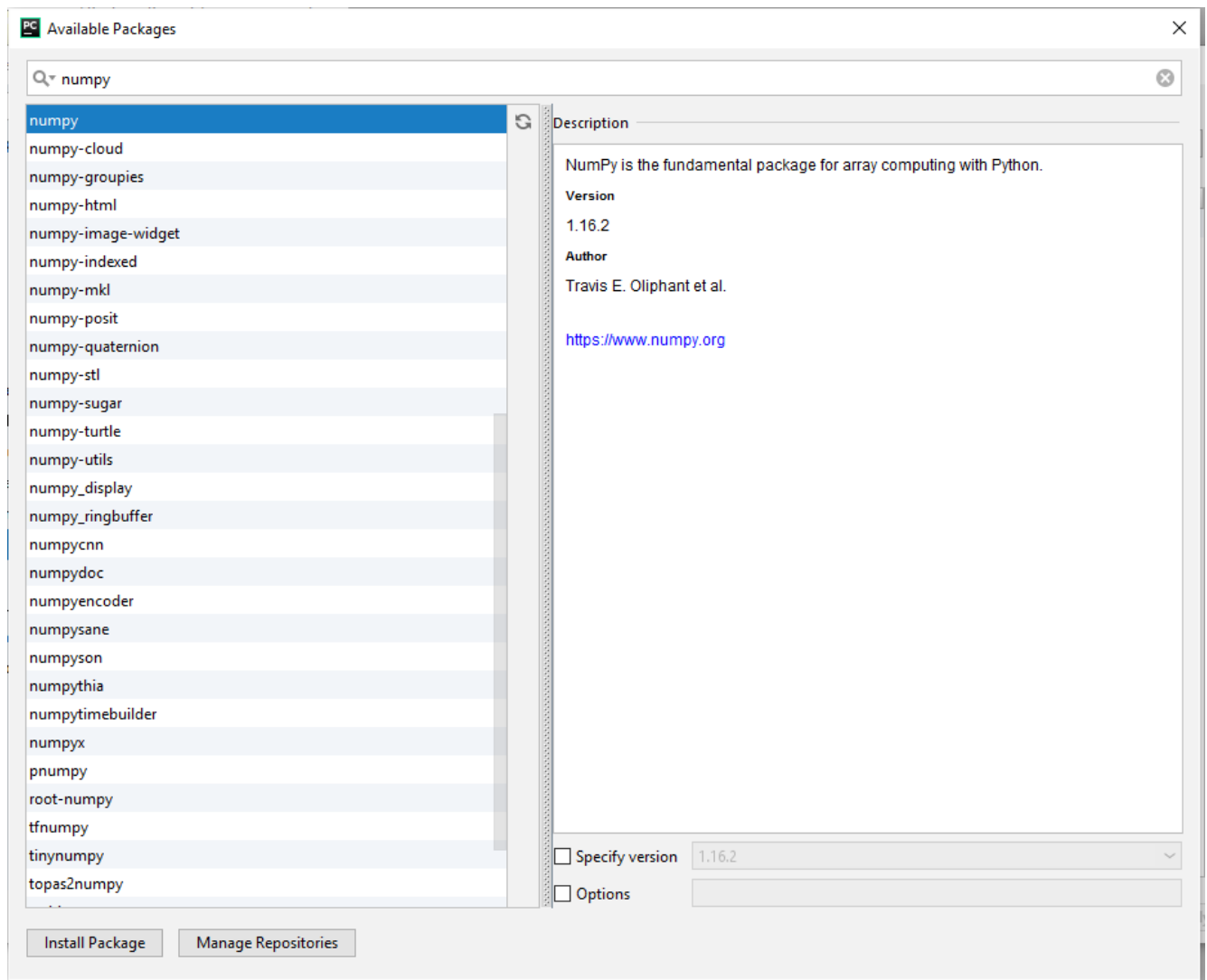


Рис. 8. Ищем numpy в репозитории и как только он будет найден, смело жмём кнопку «Install Package» для начала процесса установки пакета в окружение нашего проекта

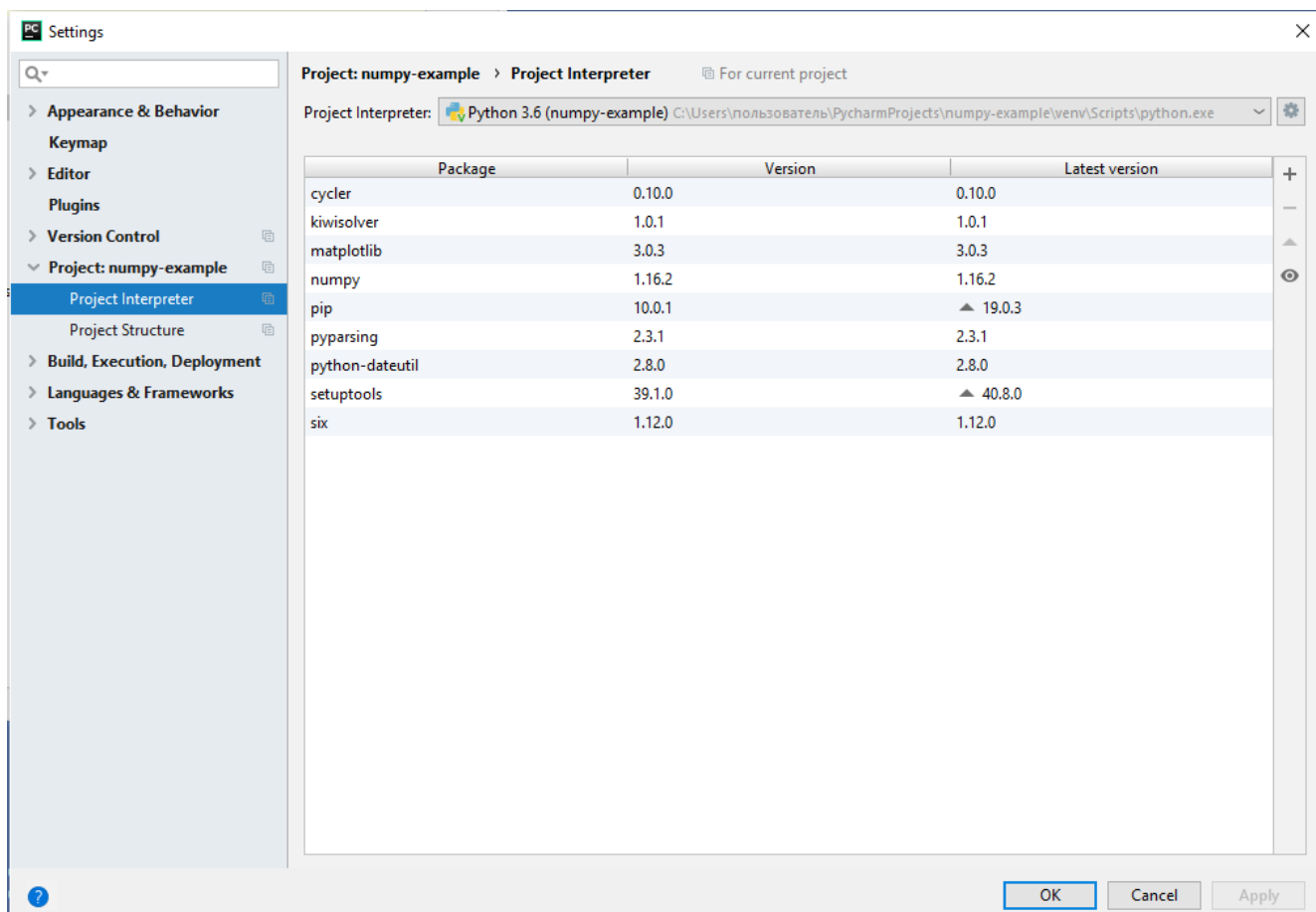


Рис. 9. Убедимся, что нужные пакеты есть в списке и установлены

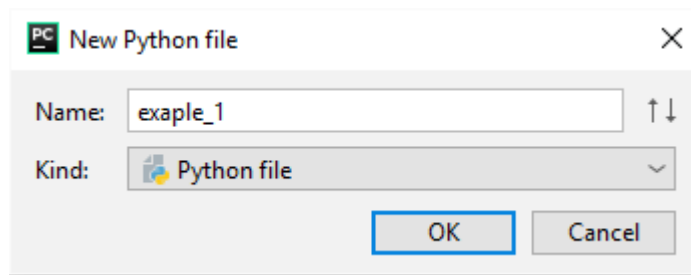
Если всё сделали правильно, то можно приступить к записи кода и проверке работы пакетов. Ниже приведены примеры, которые можно просто скопировать в отдельные файлы, запустить и посмотреть, как это работает.

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕК NUMPY И MATPLOTLIB

Код примеров, приведённых ниже, базируется на кодах, описанных на ресурсе «Записки программиста»¹⁰, но значительно переработаны, проверены и всё как положено. Исходные тексты программ можно получить по адресу

¹⁰ <https://eax.me/python-numpy/>

Итак, начинаем — создаем новый файл Python, в который записываем программный код первого примера (рис. 55).



*Рис. 10. Добавляем в проект питру-example новый файл example_1.py
«Визуализация функций»*

КОД ПРИМЕРА «ВИЗУАЛИЗАЦИЯ ФУНКЦИЙ»

#имя проекта: питру-example

#имя файла: example_1.py

#номер версии: 1.0¹¹

#автор и его учебная группа: Е. Волков, ЭУ-142

#дата создания: 20.03.2019

дата последней модификации: 25.03.2019

#связанные файлы: пакеты питру, matplotlib

описание: построение графика сигма-функции

#версия Python: 3.6

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(-5, 5, 100)
```

```
def sigmoid(alpha):
    return 1 / ( 1 + np.exp(- alpha * x))
```

```
dpi = 80
```

```
fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi))
```

¹¹ [Версия состоит из нескольких чисел \(как правило, трёх\), разделённых точкой: например, 1.5.2. Первое из них — старшая версия \(major\), второе — младшая \(minor\), третья — мелкие изменения \(maintenance, micro\). Для более подробной информации пройдите по этой ссылке и прочитайте в Wikipedia](#)

```
plt.plot(x, sigmoid(0.5), 'ro-')
plt.plot(x, sigmoid(1.0), 'go-')
plt.plot(x, sigmoid(2.0), 'bo-')

plt.legend(['A = 0.5', 'A = 1.0', 'A = 2.0'], loc = 'upper left')
plt.show()

fig.savefig('sigmoid.png')
```

РЕЗУЛЬТАТ ПРИМЕРА «ВИЗУАЛИЗАЦИЯ ФУНКЦИЙ»

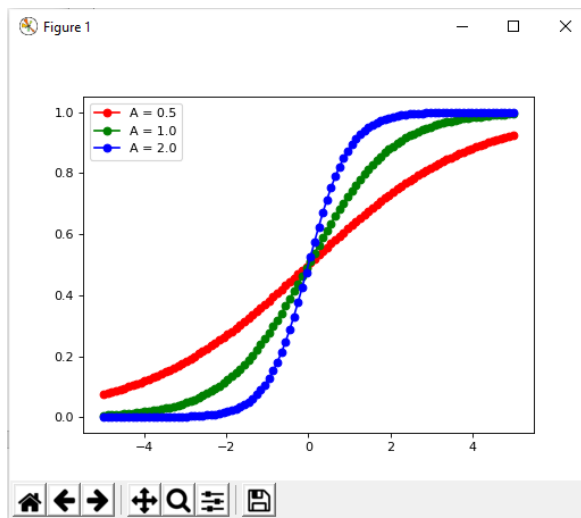


Рис. 11. Наблюдаем сигма-функцию

КОД ПРИМЕРА «СТАТИСТИКА»

```
#имя проекта: питру-example
#номер версии: 1.0
#имя файла: example_2.py
#автор и его учебная группа: Е. Волков, ЭУ-142
#дата создания: 20.03.2019
# дата последней модификации: 25.03.2019
#связанные файлы: пакеты питру, matplotlib
# описание: простейшие статистические вычисления
#версия Python: 3.6
```

```
ones = np.ones(50)
rnd = np.random.random(50) * 0.1
```

```

samples = ones + rnd
# Посчитаем среднее:
np.average(samples)
np.mean(samples)

# Медиану:

np.median(samples)

# Процентили:

np.percentile(samples, 50)
np.percentile(samples, 95)
np.percentile(samples, 99)

# Максимум, минимум, peak-to-peak:

samples.max()
samples.min()
samples.ptp()

# А заодно уж и стандартное отклонение с дисперсией:

np.std(samples)
np.var(samples)

# Используемая выше функция np.random.random генерирует случайные числа
с равномерным распределением. А если мы хотели бы использовать нормальное
распределение? Нет проблем:
#
import matplotlib.pyplot as plt
samples = np.random.normal(loc=0, scale=5, size=100000)
plt.hist(samples, 200)
plt.show()

```

РЕЗУЛЬТАТ ПРИМЕРА «СТАТИСТИКА»

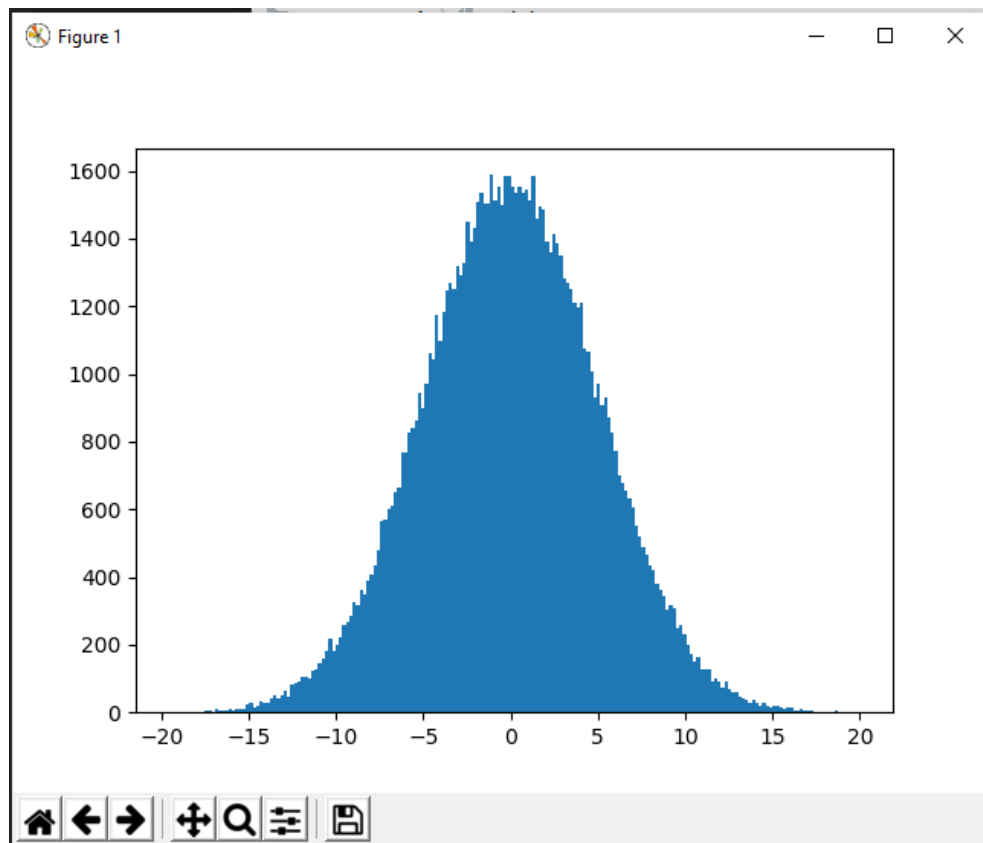


Рис. 12. Функция плотности вероятности случайной величины с нормальным законом распределения

КОД ПРИМЕРА «ЛИНЕЙНАЯ АЛГЕБРА»

#имя проекта: numpу-example

#номер версии: 1.0

#имя файла: example_3.py

#автор и его учебная группа: Е. Волков, ЭУ-142

#дата создания: 20.03.2019

дата последней модификации: 25.03.2019

#связанные файлы: пакеты numpу, matplotlib

описание: линейная алгебра

#версия Python: 3.6

```
import numpy as np
```

```
import numpy.matlib
```

```
import numpy.linalg
```

```

m1 = np.arange(1, 10).reshape(3,3)
print(m1)

m2 = np.identity(3)
print(m2)
# Транспонируем первую матрицу, а также посчитаем след и детерминант
второй:

m1.transpose()
print(m1)

m2.trace()
print(m2)

det = np.linalg.det(m2)
print(det)

# Матрицы можно складывать, умножать на число, умножать на вектор, а также
умножать на другую матрицу:

print(m1 + m2)

print(m1 * 3)

m1 + np.array([1,2,3])
print(m1 * m2)

# Посчитать матрицу, обратную к данной, можно функцией np.linalg.inv:
m3 = np.matlib.rand(3, 3)
(m3 * np.linalg.inv(m3))
print(m3)

print((m3 * np.linalg.inv(m3)).round())

```

РЕЗУЛЬТАТ ПРИМЕРА «ЛИНЕЙНАЯ АЛГЕБРА»

```

[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
1.0
[[ 2. 2. 3.]

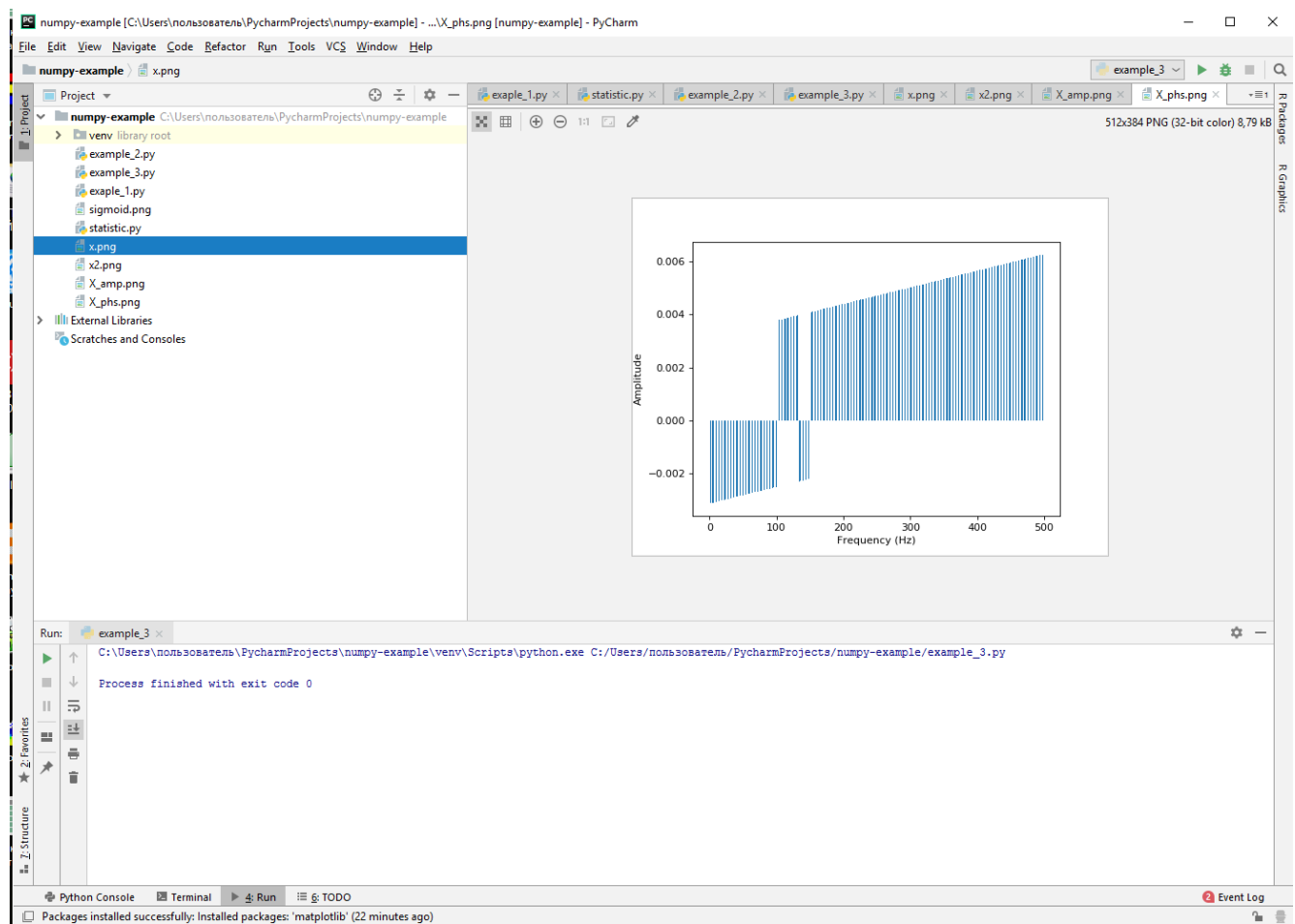
```

```

[ 4. 6. 6.]
[ 7. 8. 10.]]
[[ 3 6 9]
 [12 15 18]
 [21 24 27]]
[[1. 0. 0.]
 [0. 5. 0.]
 [0. 0. 9.]]
[[0.21960198 0.79353093 0.71379042]
 [0.79998886 0.24333293 0.4619364 ]
 [0.08058218 0.6665476 0.38367675]]
[[ 1. -0. -0.]
 [-0. 1. 0.]
 [ 0. 0. 1.]]

```

Process finished with exit code 0



СОДЕРЖАНИЕ ЧАСТИ 2 КУРСОВОЙ РАБОТЫ

Для начала понимания, как работать с матрицами решите следующие задачи, количеством 31. Их можно записать в одном проекте.

РАБОТА С ДВУМЕРНЫМИ МАССИВАМИ (МАТРИЦАМИ)

1. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти наибольший элемент столбца матрицы A , для которого сумма абсолютных значений элементов максимальна.
2. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти наибольшее значение среди средних значений для каждой строки матрицы.
3. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти наименьший элемент столбца матрицы A , для которого сумма абсолютных значений элементов максимальна.
4. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти наименьшее значение среди средних значений для каждой строки матрицы.
5. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Определить средние значения по всем строкам и столбцам матрицы. Результат оформить в виде матрицы из $N + 1$ строк и $M + 1$ столбцов.
6. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждого столбца. Результат оформить в виде матрицы из $N + 1$ строк и M столбцов.

7. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждой строки. Результат оформить в виде матрицы из N строк и $M+1$ столбцов.
8. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Определить, сколько отрицательных элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из $N + 1$ строк и $M + 1$ столбцов.
9. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Определить, сколько нулевых элементов содержится в верхних L строках матрицы и в левых K столбцах матрицы.
10. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Перемножить элементы каждого столбца матрицы с соответствующими элементами K -го столбца.
11. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Просуммировать элементы каждой строки матрицы с соответствующими элементами L -й строки.
12. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Разделить элементы каждой строки на элемент этой строки с наибольшим значением.
13. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Разделить элементы каждого столбца матрицы на элемент этого столбца с наибольшим значением.
14. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Разделить элементы матрицы на элемент матрицы с наибольшим значением.
15. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Все элементы имеют целый тип. Дано

целое число N . Определить, какие столбцы имеют хотя бы одно такое число, а какие не имеют.

16. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Исключить из матрицы строку с номером L . Сомкнуть строки матрицы.
17. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Добавить к матрице строку и вставить ее под номером L .
18. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Найти сумму элементов, стоящих на главной диагонали, и сумму элементов, стоящих на побочной диагонали (элементы главной диагонали имеют индексы от $[0,0]$ до $[N,N]$, а элементы побочной диагонали — от $[N,0]$ до $[0,N]$).
19. Создать квадратную матрицу A , имеющую N строк и N столбцов со случайными элементами. Определить сумму элементов, расположенных параллельно главной диагонали (ближайшие к главной). Элементы главной диагонали имеют индексы от $[0,0]$ до $[N,N]$.
20. Создать квадратную матрицу A , имеющую N строк и N столбцов со случайными элементами. Определить произведение элементов, расположенных параллельно побочной диагонали (ближайшие к побочной). Элементы побочной диагонали имеют индексы от $[N,0]$ до $[0,N]$.
21. Создать квадратную матрицу A , имеющую N строк и N столбцов со случайными элементами. Каждой паре элементов, симметричных относительно главной диагонали (ближайшие к главной), присвоить значения, равные полусумме этих симметричных значений (элементы главной диагонали имеют индексы от $[0,0]$ до $[N,N]$).
22. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Исходная матрица состоит из нулей и

единиц. Добавить к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.

23. Создать квадратную матрицу A , имеющую N строк и N столбцов со случайными элементами. Найти сумму элементов, расположенных выше главной диагонали, и произведение элементов, расположенных выше побочной диагонали (элементы главной диагонали имеют индексы от $[0,0]$ до $[N,N]$, а элементы побочной диагонали — от $[N,0]$ до $[0,N]$).
24. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Дан номер строки L и номер столбца K , при помощи которых исходная матрица разбивается на четыре части. Найти сумму элементов каждой части.
25. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Определить, сколько нулевых элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из $N + 1$ строк и $M + 1$ столбцов.
26. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Дан номер строки L и номер столбца K , при помощи которых исходная матрица разбивается на четыре части. Найти среднее арифметическое элементов каждой части.
27. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Все элементы имеют целый тип. Дано целое число H . Определить, какие строки имеют хотя бы одно такое число, а какие не имеют.
28. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Исключить из матрицы столбец с номером K . Сомкнуть столбцы матрицы.
29. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Добавить к матрице столбец чисел и вставить его под номером K .

30. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Добавить к элементам каждого столбца такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из $N + 1$ строк и M столбцов.
31. Создать прямоугольную матрицу A , имеющую N строк и M столбцов со случайными элементами. Добавить к элементам каждой строки такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из N строк и $M + 1$ столбцов.