

Создание графиков с помощью matplotlib

1 Установка библиотек

Для того чтобы использовать сторонние библиотеки их нужно установить. Нам понадобятся три библиотеки - **Numpy**, **Matplotlib** и **PyQt5**. **Numpy** предоставляет расширенный математический функционал, **Matplotlib** помогает рисовать графики, а **PyQt5** нужен библиотеке **Matplotlib** как зависимость для создания окон.

Перейдем к установке. Воспользуемся пакетным менеджером **pip** для установки пакетов в общую систему. В терминале выполним следующую команду:

- `pip install numpy matplotlib PyQt5 --break-system-packages`

2 Построение графиков

Библиотека **Matplotlib** рисует графики по точкам, то есть соединяет точки линиями в порядке их следования. Сначала попробуем нарисовать простую фигуру, а затем перейдем к графикам. Для рисования нам понадобится функция `plot()`. Она принимает массив с координатам точек по x, массив с координатами точек по y и дополнительные параметры графика, такие как цвет, ширина линии и способ зарисовки линии. Пока что сосредоточимся на координатах. Хочется нарисовать квадрат из четырех точек: (-1; -1), (-1; 1), (1; 1), (1; -1).

```
1 | import numpy as np
2 | import matplotlib.pyplot as plt
3 |
4 | plt.plot([-1, -1, 1, 1, 1], [-1, 1, 1, -1, -1])
5 |
6 | plt.show()
```

Теперь про функцию `show()`. Она нужна для показа полученного графика.

Теперь построим простой график. Выберем простую функцию, например: $y = x^2$.

Чтобы график был довольно плавным нужно чтобы было много соединяющихся точек. То есть нужно создать массив с координатами точек по x, посчитать значения функции для каждой x координаты и положить ее в отдельный массив, после чего отдать в функцию `plot` для рисования. Создать значения x поможет функция `linspace`. Она принимает три аргумента: начало отрезка a, конец отрезка b и количество чисел n, возвращает массив с ровно n числами, равномерно расположенными в порядке возрастания на отрезке от a до b. Теперь нужно посчитать значения функции, благо массивы `numpy` могут помочь нам и облегчить задачу. Массивы `numpy` могут участвовать в математических

выражениях как переменные и результатом такого выражения будет массив, к каждому элементу которого применено математическое выражение. Например: Пусть есть массив $a = [1, 2, 3, 4, 5]$ и он участвует в математическом выражении $a ** 2 + 5$, тогда вернется массив, каждое значение которого возведено в квадрат и увеличено на 5, то есть $[6, 9, 14, 21, 30]$. Теперь напишем код и построим график.

```
1 | import numpy as np
2 | import matplotlib.pyplot as plt
3 |
4 | x = np.linspace(-5, 5, 100)
5 | y = x ** 2
6 |
7 | plt.plot(x, y)
8 |
9 | plt.show()
```