

## Самостоятельная работа №2

### «Мой первый рисунок»

Выполнить самостоятельно. Задания по материалам лекции (тема 1.4)

### Визуализация с помощью библиотеки Matplotlib

#### Задание 1. Алгоритм работы с рисунками

Реализуйте описанный алгоритм работы с рисунками.

```
# подключение библиотеки numpy под псевдонимом np
import numpy as np

# подключение модуля pyplot из библиотеки matplotlib
# под псевдонимом plt
import matplotlib.pyplot as plt

# пользовательская переменная для хранения размера шрифта
fsize=12

# настройка типа шрифта на рисунке с помощью изменения
# записей в словаре rcParams из модуля pyplot
plt.rcParams['font.family'] = 'serif'
plt.rcParams['font.serif'] = 'Times New Roman'

# настройка размера шрифта в различных частях рисунка
# в заголовке:
plt.rcParams['axes.titlesize'] = fsize

# в подписях осей:
plt.rcParams['axes.labelsize'] = fsize

# в подписях меток на осях:
plt.rcParams['xtick.labelsize'] = fsize
plt.rcParams['ytick.labelsize'] = fsize

# в легенде рисунка:
plt.rcParams['legend.fontsize'] = fsize

# массив координат - 50 точек в диапазоне [0, 10]
x = np.linspace(0.0, 10.0, 50)

# создаём окно рисунка.
# Для дальнейшей работы рисунок ассоциируется с переменной
# fig
fig=plt.figure()

# добавляем панель (оси координат) с именем ax в окно fig.
# в дальнейшем настройка осей производится
# через обращение к переменной ax.
# аргументы 1, 1, 1 указывают, что на рисунке будет
# только одна панель для рисования графиков.
# Запятые могут быть опущены.
ax=fig.add_subplot(1,1,1)

# график синуса:
# кружки (o), соединённые сплошной (-) чёрной линией.
# графику присваивается строковый идентификатор <1>
# для дальнейшего отображения в легенде
ax.plot(x, np.sin(x), 'ko-', label='1')
```

```

# график косинуса:
# квадратики (s, размером 3), соединённые сплошной (-)
# оранжевой линией толщиной 1.
# графику присваивается строковый идентификатор <2>
# для отображения в легенде
ax.plot(x, np.cos(x), 'ks-', color='orange', linewidth=1,
markersize=3.0, label='2')

# график синуса в квадрате:
# треугольники (^), соединённые сплошной (-) лиловой
# линией толщиной 1.
# графику присваивается строковый идентификатор <3>
# для отображения в легенде
ax.plot(x, (np.sin(x))**2.0, 'k^-', color='magenta',
linewidth=1, label='3')

# график функции f(x)=x^0.15:
# чёрная штриховая линия толщиной 1.
# графику присваивается строковый идентификатор <x^2>
# для отображения в легенде.
# символ r и знаки доллара внутри строки позволяют
# вводить математические символы с помощью команд TeX
ax.plot(x, (x)**0.15, 'k--', linewidth=1, label=r'$x^2$')

# легенда
ax.legend(loc='best')

# диапазон отображаемых значений по оси x
ax.set_xlim(-1.0, 11.0)

# диапазон отображаемых значений по оси y
ax.set_ylim(-1.5, 1.5)

# подпись по оси x
ax.set_xlabel(r'$x$')

# подпись по оси y
ax.set_ylabel(r'$f(x)$')

# заголовок рисунка
ax.set_title('Мой первый рисунок')

# сетка на рисунке
ax.grid()

# сохраняем в файл с именем fig1 типа PNG с разрешением
# 300 точек на дюйм
# (dpi - dots per inch), с альбомной ориентацией
fig.savefig("fig1.png", orientation='landscape', dpi=300)

```

Если найдете ошибку – исправьте и сообщите. Если модифицируете программный код – укажите в пояснительной записке. Добавьте необходимые инструкции.

### ***Комментарии к программе для рисования графиков.***

Каждый график добавляется на текущую панель рисунка с помощью функции plot(). График функции синус изображён с помощью команды ax.plot(x, np.sin(x), 'ko-', label='1').

Тип точек и соединительной линии может быть указан с помощью дополнительного строкового аргумента в команде `plot()` после первых двух обязательных аргументов – отображаемых массивов.

Данная строка имеет следующий формат `'CDL'`, где

- `C` – буква-код, указывающая цвет линии.

Возможные варианты: `k` (чёрный), `b` (blue – синий), `c` (cyan – голубой), `g` (green – зелёный), `m` (magenta – пурпурный), `r` (red – красный), `y` (yellow – жёлтый), `w` (white – белый).

- `D` – символ, указывающий тип соединительной точки (маркера).

Варианты: `s` (square – квадратик), `d` (diamond – ромбик), `o` (кружок), `*` (звёздочка),

`^` (треугольник вверх), `v` (треугольник вниз).

- `L` – тип линии.

Варианты: `-` (сплошная линия), `--` (штриховая линия), `:` (пунктирная линия),

`-.` (штрихпунктирная линия).

Например, строка `'ko-'` в примере означает, что график нужно нарисовать чёрной сплошной линией, а соединительные точки отображать с помощью кружочков.

Каждый из указанных параметров можно настроить с помощью отдельного ключевого слова в списке аргументов функции `plot()`.

Например, график косинуса нарисован жёлтыми квадратиками размером 3 (`markersize = 3.0`), соединёнными сплошной оранжевой линией (`color = 'orange'`), толщиной 1 (`linewidth = 1`),

```
ax.plot(x, np.cos(x), 'ks-', color='orange', linewidth=1, markersize=3.0, label='2')
```

Аргумент `label` в команде `plot()` присваивает текущему графику имя-строку, которая будет отображена в легенде рисунка.

Настройки панели осуществляется посредством обращения к полям и методам объекта.

Основные свойства панели:

- `set_xlim(a, b)` – установка диапазона отображаемых по оси абсцисс значений от `a` до `b`;
- `set_ylim(a, b)` – установка диапазона отображаемых по оси ординат значений от `a` до `b`;
- `set_xlabel(str)` – установка строки `str` в качестве подписи оси абсцисс;
- `set_ylabel(str)` – установка строки `str` в качестве подписи оси ординат;
- `set_title(str)` – установка строки `str` в качестве заголовка панели;
- `set_xscale(str)` – установка масштаба по оси абсцисс, если `str="lin"`, то будет использован линейный масштаб, если `str="log"`, то логарифмический. По умолчанию все оси отображаются с линейным масштабом;
- `set_yscale(str)` – установка масштаба по оси ординат;
- `legend()` – отображение легенды на рисунке.

**Задание 2. Создайте рисунок с основными элементами изображения, похожий на приведенный ниже рисунок 1.**

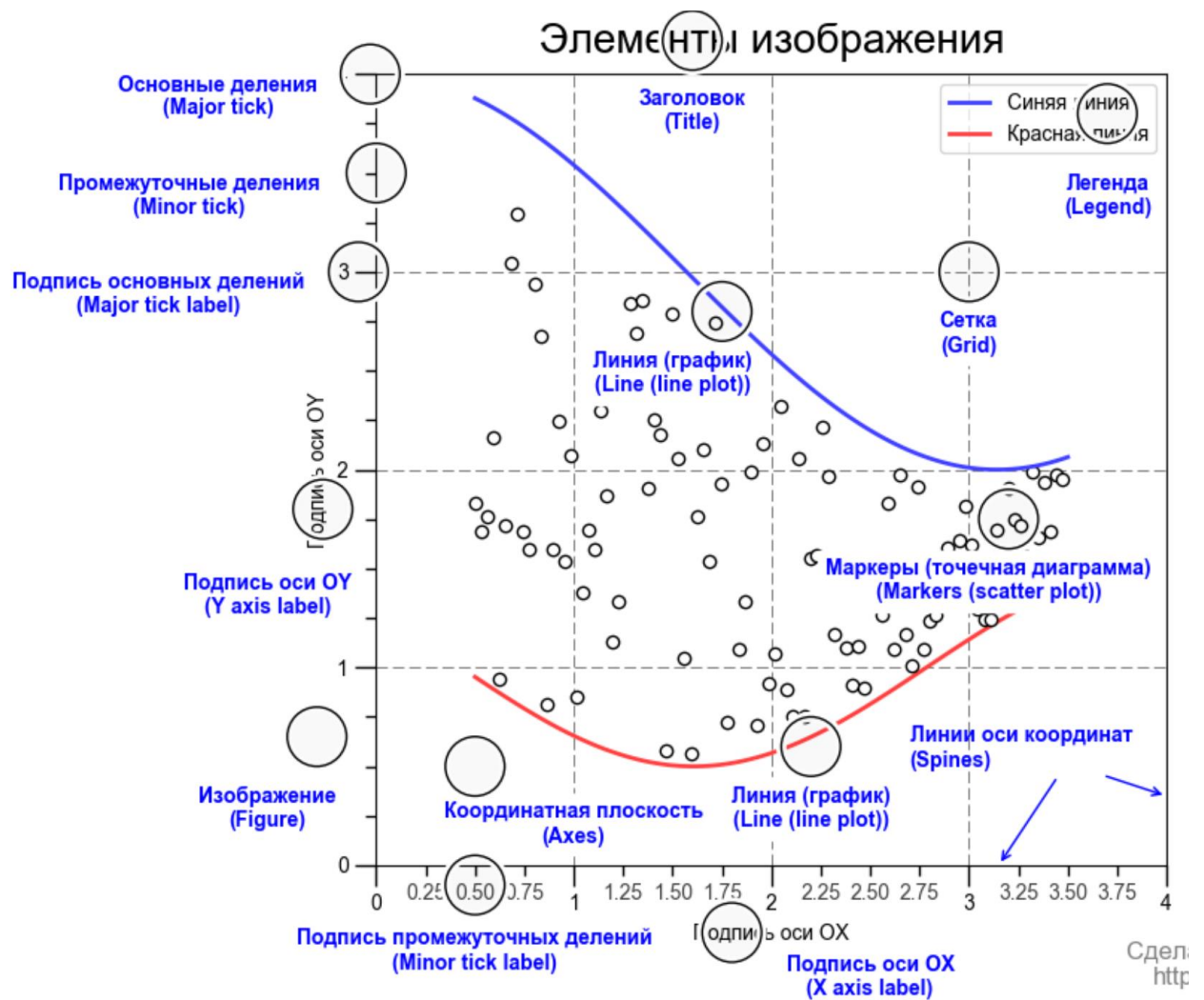


Рисунок 1 – Основные элементы изображения в matplotlib

Допускается изменение линий, цвета, толщины линий и т.п.