

2.6. Рисование плоских фигур

Плоские геометрические фигуры создаются с помощью библиотеки [matplotlib.patches](#), из которой можно импортировать шаблоны для различных геометрических фигур (все фигуры перечислить **на одной строке**):

```
from matplotlib.patches import Circle, Wedge, Polygon, Ellipse,
    Rectangle, Arc, Path, PathPatch
```

Для отображения фигур в области рисования используется библиотека [matplotlib.pyplot](#):

```
import matplotlib.pyplot as plt
```

2.6.1. Создание геометрической фигуры

Каждая геометрическая фигура создается методом библиотеки [matplotlib.patches](#), который имеет два типа параметров:

- положение фигуры;
- характеристики фигуры.

Каждая фигура имеет свой набор параметров, описывающих **положение фигуры** в области рисования. В зависимости от типа фигуры это могут быть координаты точек, радиусы, величины углов и пр. Рассмотрим методы создания наиболее часто используемых геометрических фигур.

`Rectangle((x, y), width, height)` – создает прямоугольник, левый верхний угол которого располагается в точке с координатами (x, y) , высота которого равна `width`, а длина – `height` (рисунок 15а).

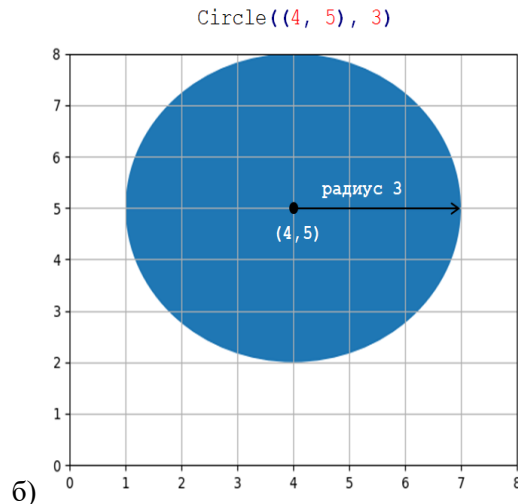
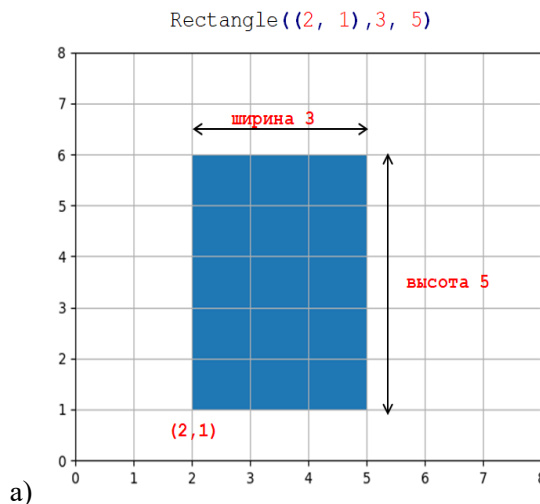


Рисунок 15. Фигуры: а) прямоугольник; б) окружность

`Circle((x, y), radius)` – создает круг с центром в точке (x, y) , радиуса `radius` (рисунок 15б).

`Ellipse((x, y), width, height)` – создает эллипс с центром в точке (x, y) , диаметр которого по горизонтальной оси равен `width`, а диаметр по вертикальной – `height` (рисунок 16а).

`Polygon([(x0, y0), (x1, y1), (x2, y2), ...], closed)` – создает ломаную линию по точкам $(x0, y0)$, $(x1, y1)$, $(x2, y2)$, ..., если параметр `closed` равен **True** (или он отсутствует), первая точка соединяется с последней, в противном случае (`closed= False`), ломаная линия остается незамкнутой (рисунок 16а).

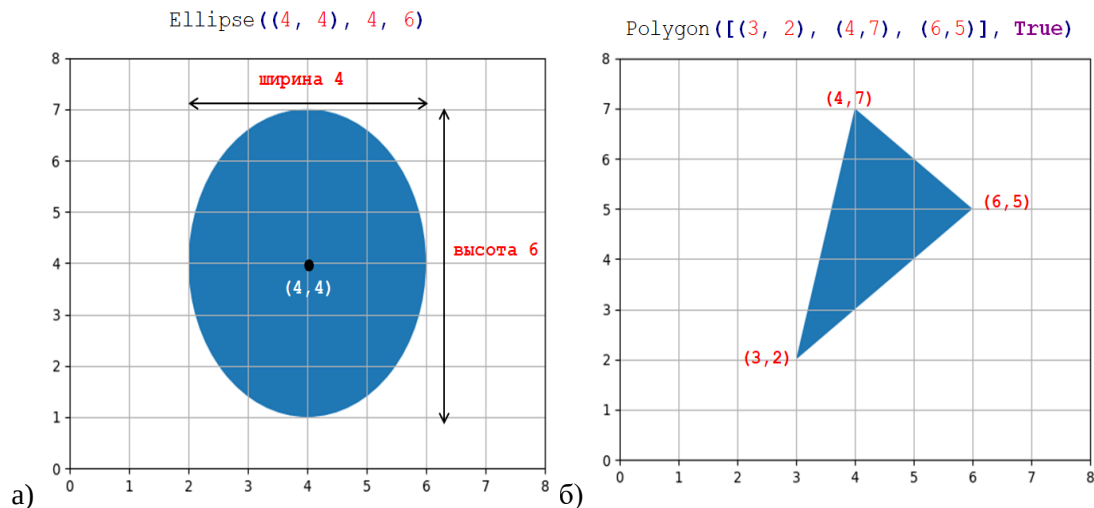


Рисунок 16. Фигуры: а) эллипс; б) полигон

`Wedge((x, y), radius, t1, t2)` – создает сектор с центром в точке (x, y) , радиуса `radius`, ограниченный линиями углов `t1` и `t2` (рисунок 17а).

`Arc((x, y), width, height, angle, t1, t2)` – создает дугу с центром в точке (x, y) , диаметр которой по горизонтальной оси равен `width`, а диаметр по вертикальной – `height`, угол поворота дуги относительно центра `angle`, дуга ограничена линиями углов `t1` и `t2` (рисунок 17б).

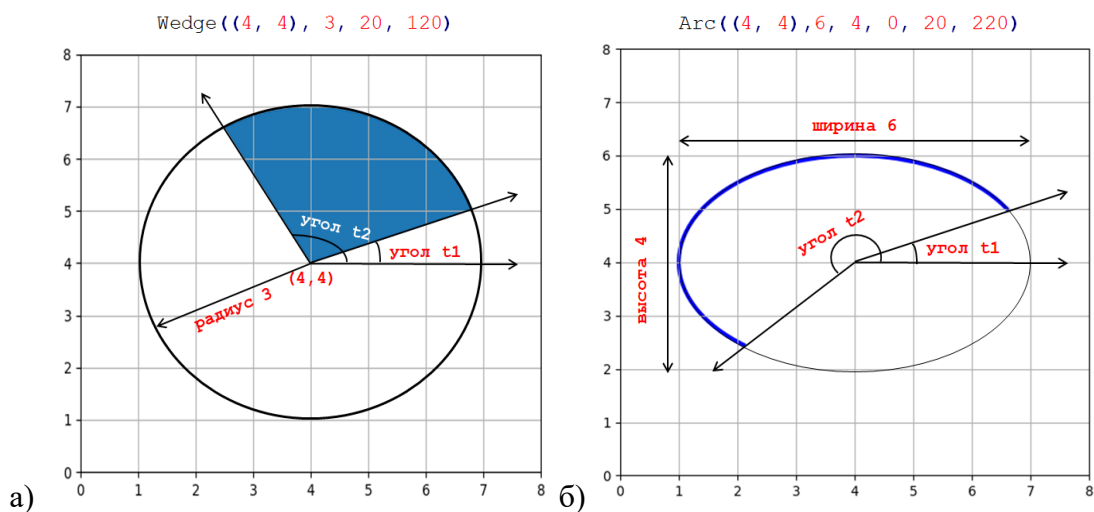


Рисунок 17. Фигуры: а) сектор; б) дуга

Совокупность линии можно рисовать с помощью так называемых путей (Path). Путь представляет собой список опорных точек (как при создании полигона) и список кодов, задающих, что с этими вершинами делать.

Коды – это числовые константы, определенные в модуле Path, которые могут принимать значения:

- MOVETO (переместиться в точку без рисования линии);
- LINETO (провести линию в заданную точку);
- другие.

Коды можно записывать двумя способами:

- полной ссылкой (`matplotlib.path.Path.MOVETO`, `matplotlib.path.Path.LINETO`);
- цифрой 1 или 2 соответственно.

Свойства, описывающие **стиль вывода** каждой фигуры, общие для всех фигур. Основные свойства представлены в таблице, большинство из них могут быть описаны в полной или сокращенной форме (таблица 16). Возможные значения характеристик приведены в Приложении В.

Таблица 16 – Стили вывода фигур

Характеристика	Описание
<code>facecolor="цвет"</code> <code>fc="цвет"</code>	Цвет заливки фигуры
<code>linewidth=значение</code> <code>lw=значение</code>	Толщина границы фигуры
<code>edgecolor="цвет"</code> <code>ec="цвет"</code>	Цвет границы фигуры
<code>fill=значение</code>	Фигура закрашенная (True, по умолчанию) или незакрашенная(False)

2.6.2. Алгоритм рисования

Все геометрические объекты рисуются в несколько шагов.

1. Импортируются все необходимые модули, функции, шаблоны:

```
from matplotlib.patches import Circle
import matplotlib.pyplot as plt
```

2. Определяются координаты окна, в котором будет создаваться изображение, например, следующие операторы создают окно, координаты которого изменяются от 0 до 12 по горизонтали и вертикали:

```
plt.xlim(0, 12)
plt.ylim(0, 12)
```

3. Создается область для рисования, связанная с осями координат с помощью метода `gca()`:

```
ax = plt.gca()
```

4. Создается геометрическая фигура на основе описания из модуля [matplotlib.patches](#). Например, следующий оператор создает круг с центром в точке (6, 7) и радиусом 5 и заносит результат в переменную **circle**:

```
circle = Circle((6, 7), 5)
```

5. Созданная фигура добавляется в область **ax** с помощью метода `add_patch()`:

```
ax.add_patch(circle)
```

6. Рисунок отображается в графическом окне:

```
plt.show()
```

Результат выполнения программы показан на рисунке 18.

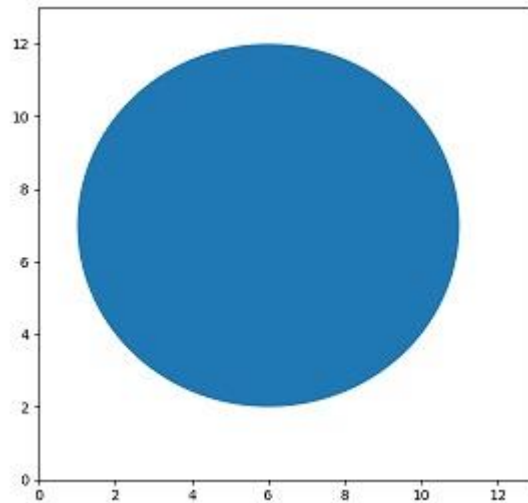


Рисунок 18. Результат выполнения программы

Пример. Создадим рисунок (рисунок 19) с помощью линий, объединенных в пути.

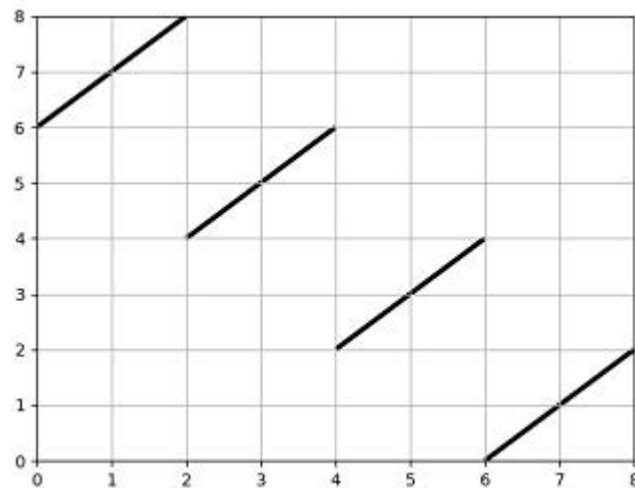


Рисунок 19. Шаблон рисунка

Для создания рисунка используется следующий код, пояснения даны в комментариях:

```
# Импортируем необходимые функции и библиотеки:
from matplotlib.patches import Path, PathPatch
import matplotlib.pyplot as plt

# Настроим область рисования:
n = 8
m = 8
plt.xlim(0, n)
plt.ylim(0, m)
ax = plt.gca()

# Создадим список опорных точек:
vertices = [(0, 6), (2, 8), (2, 4), (4, 6), (4, 2), (6, 4), (6, 0), (8, 2)]

# Создадим список кодов (установить курсор в первую точку (0, 6) - MOVETO,
# провести линию во вторую (2, 8) - LINETO,
# установить курсор в третью (2, 4) - MOVETO,
# провести линию в четвертую (4, 6) - LINETO и т.д.):
codes = [1, 2, 1, 2, 1, 2, 1, 2]

# Создадим переменную path с помощью метода matplotlib.patches.Path,
# указав список вершин (vertices) и список кодов (codes):
```

```

path = Path(vertices, codes)

# Создадим фигуру, представляющую собой совокупность линий, толщиной 3 px:
path_patch = PathPatch(path, lw=3)

#Добавим созданную фигуру в область ax, удалим оси и покажем рисунок
ax.add_patch(path_patch)
ax.axes.set_axis_off()
plt.show()

```

Результат выполнения программы показан на рисунке 20.

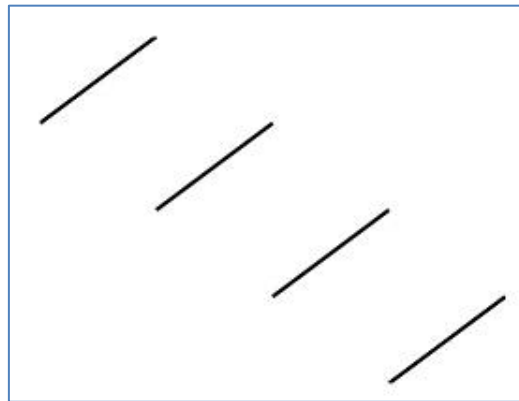


Рисунок 20. Результат выполнения программы

Задание для самопроверки

В программе создаются плоские фигуры с помощью методов библиотеки `matplotlib.patches`:

```
from matplotlib.patches import Circle, Wedge, Ellipse, Arc, Rectangle
```

Выберите **неверные** операторы:

- `figure = wedge((4, 4), 2, -90, 90)`
- `figure = Wedge((4, 4), 2, -90, 90)`
- `figure = Wedge((4, 4), 2, -90)`
- `figure = Rectangle((10, 12), 5, 8)`
- `figure = Ellipse(5, 1, 2, 3)`