

2.7. Задача, рисование фигуры из геометрических примитивов

2.7.1. Алгоритм рисования изображения из геометрических фигур

1. Нарисовать изображение на клетчатом листе бумаге, пронумеровать клетки от 0 до **n** по горизонтали, от 0 на **m** вертикали (в нашем случае от 0 до 13 и от 0 до 15), нумерация клеток может начинаться с любых значений, но удобнее взять 0 (рисунок 21).

2. Обозначить координаты опорных точек, образующих плоские фигуры, при необходимости указать другие характеристики (рисунок 21):

- полигон – координаты опорных точек (на рисунке туловище кота);
- круг – координаты центра и радиус (глаза, зрачки и нос);
- эллипс – координаты центра, диаметр по горизонтали и вертикали (передние лапы);
- сектор – координаты центра, радиус, угол начала сектора, угол его завершения (задние лапы);
- дуга – координаты центра, радиус по горизонтали, радиус по вертикали, угол поворота, угол начала дуги, угол ее завершения (улыбка);
- набор линий – координаты опорных точек (усы, линия между носом и улыбкой).

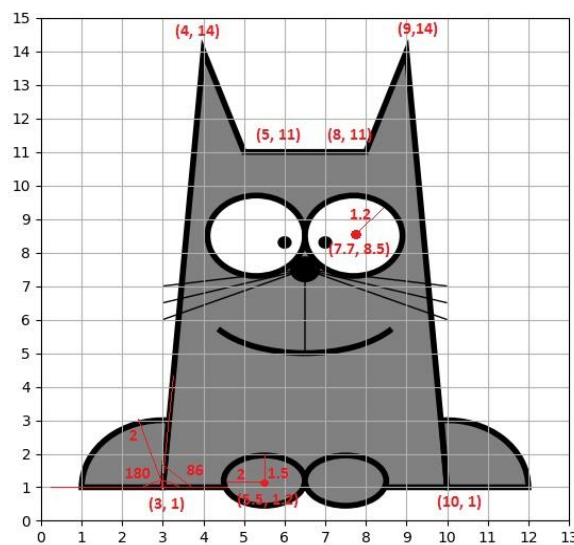


Рисунок 21. Рисунок с опорными точками

3. Определить стили вывода каждой фигуры:

- толщина границы набора линий – 1 пиксель, остальные – 4 пикселя;
- цвет границ всех фигур – черный;
- цвета закрашки – черный (для зрачков и носа), белый – для глаз, серый – для туловища и лап;
- все фигуры, кроме дуги и набора линий, – замкнутые.

4. Реализовать каждую фигуру с помощью шаблонов плоских фигур из matplotlib.patches.

2.7.2. Реализация

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

```
# Импортировать необходимые фигуры рисования,
# подключить модуль для рисования:
from matplotlib.patches import Circle, Wedge, Polygon, Ellipse, Arc,
    Path, PathPatch
import matplotlib.pyplot as plt

# Реализовать функцию для рисования фигуры,
# параметром передать область рисования:
def draw_cat(ax):
    # туловище
    poly = [(3, 1), (4, 14), (5, 11), (8, 11), (9, 14), (10, 1)]
    polygon = Polygon(poly, fc="grey", ec="black", lw=4)
    ax.add_patch (polygon)

    # глаза
    circle = Circle((5.3, 8.5), 1.2, fc="white", ec="black", lw=4)
    ax.add_patch (circle)
```

```

circle = Circle((7.7, 8.5), 1.2, fc="white", ec="black", lw=4)
ax.add_patch (circle)

# зрачки
circle = Circle((6, 8.3), 0.1, fc="black", ec="black", lw=4)
ax.add_patch (circle)
circle = Circle((7, 8.3), 0.1, fc="black", ec="black", lw=4)
ax.add_patch (circle)

# нос
circle = Circle((6.5, 7.5), 0.3, fc="black", ec="black", lw=4)
ax.add_patch (circle)

# задние лапы
wedge = Wedge((3, 1), 2, 86, 180, fc="grey", ec="black", lw=4)
ax.add_patch (wedge)

wedge = Wedge((10, 1), 2, 0, 94, fc="grey", ec="black", lw=4)
ax.add_patch (wedge)

# передние лапы
ellipse = Ellipse((5.5,1.2), 2, 1.5, fc="grey", ec="black", lw=4)
ax.add_patch (ellipse)

ellipse = Ellipse((7.5,1.2), 2, 1.5, fc = "grey", ec="black", lw=4)
ax.add_patch (ellipse)

# улыбка
arc = Arc((6.5, 6.5), 5, 3, 0, 200, 340, lw=4, fill=False)
ax.add_patch(arc)

# линия между носом и улыбкой, усы
vertices = [(6.5, 5), (6.5, 7.5), (10, 6), (6.5, 7.5), (10, 6.5),
            (6.5, 7.5), (10, 7), (6.5, 7.5), (3, 6), (6.5, 7.5),
            (3, 6.5), (6.5, 7.5), (3, 7)]

# число 1 соответствует команде matplotlib.path.Path.MOVETO
# число 2 соответствует команде matplotlib.path.Path.LINETO
codes = [1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2]

path = Path(vertices, codes)
path_patch = PathPatch(path, fill=False, lw=1)
ax.add_patch(path_patch)

# Установить размер и координаты углов для области рисования
# в соответствии с рисунком 21:
n = 13
m = 15
plt.xlim(0, n)
plt.ylim(0, m)

# Создать область, связанную с осями координат,
# куда будут выводиться плоские фигуры (ax):
ax = plt.gca()

# вызвать функцию рисования (draw_cat):
draw_cat(ax)

#Удалить оси координат и показать рисунок:
ax.axes.set_axis_off()
plt.show()

```

Результат выполнения программы показан на рисунке 22.

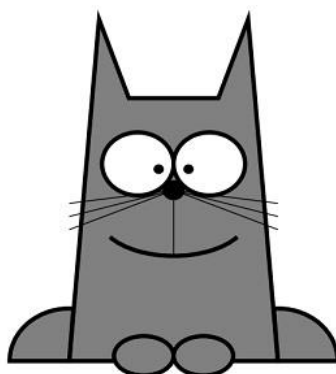


Рисунок 22. Результат выполнения программы

Задание для самопроверки

Допишите программу так, чтобы в результате ее выполнения получалось изображение, показанное на рисунке 23.

```
from matplotlib.patches import Path, PathPatch
import matplotlib.pyplot as plt

n = # указать размер области
m = # указать размер области

plt.xlim(0, n)
plt.ylim(0, m)
ax = plt.gca()

# создать массив точек
vertices = [(1, 3), (7, 2), (6, 1), (3, 1), (1, 3), (4, 2.5), (4, 6),
            (7, 3), (4, 2.5)]

# создать список кодов для последовательности рисования:
codes = # записать список

# создать объект path
path = Path(vertices, codes)

# создать фигуру
path_patch = PathPatch(path, lw=3)

# Добавить созданную фигуру в область ax:

plt.show()
```

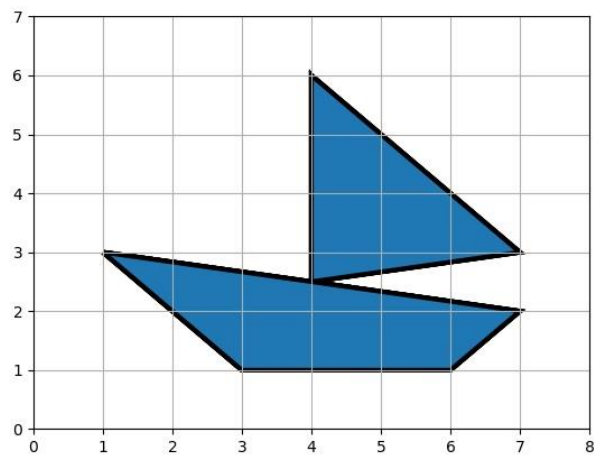


Рисунок 23. Изображение фигуры