

Лабораторна робота №3  
Тема: ОПУКЛА ОБОЛОНКА

Виконав:  
Студент групи КМ-11  
Пешков Антон

## ОПИС ПРОГРАМИ

Модуль: main.py

Для виконання задачі було використано бібліотеку PIL мови Python. Модуль складається з однієї функції, **convex\_hull**, що має наступну сигнатуру:

```
def convex_hull(  
    dataset_src: str, – шлях до датасету,  
    dataset_hull_name: str, – шлях для виводу файлу з оболонкою,  
    output_name: str – шлях для виводу зображення  
)
```

За заданим датасетом, що містить координати точок, знаходиться опукла оболонка, записується в файл за даним шляхом, та створюється зображення цієї оболонки.

## ІДЕЯ МЕТОДУ

Для знаходження опуклої оболонки у програмі застосовується алгоритм Алекса Ендрю, що наведено у його статті “Another efficient algorithm for convex hulls in two dimensions” (1979). Основна ідея методу полягає у проходженні по всіх точках 2 рази, спочатку для побудови верхньої частини оболонки, а потім нижньої; оскільки будь-які три послідовні точки оболонки, йдучи проти годинникової стрілки, розташовані також проти годинникової стрілки, у стек послідовно додаються і віднімаються точки до тих пір, доки не будуть пройдені усі надані та доки для усіх точок у стеку не буде виконуватися вищенаведена властивість. Важливо, щоб точки були відсортовані за зростанням абсцис, та за рівних абсцис за зростанням ординат.

## ОПИС ПРОГРАМИ

```
with open(dataset_src, 'r') as file:
    points = [
        tuple(map(int, line.split()))[::-1]
        for line in file.readlines()
    ]

length = len(points)
if length == 2:
    return draw_n_save(points)

points.sort()
```

Рисунок 1

Спочатку, точки считуються з файлу з датасетом (рис. 1). При наявних тільки двох точках, програма зберігає їх як результат і завершує роботу. Інакше, точки сортується за описаним у попередньому пункті порядком.

Виділяються два списки, для верхньої та нижньої оболонок відповідно (рис. 2). Імплементується алгоритм, наведений у попередньому пункті. Останній елемент одного списку є першим елементом іншого, тому вони видаляються. Після побудови, оболонка зберігається та створюється зображення заданого розміру, на яке наносяться лінії, що сполучають точки оболонок, та самі точки. Зображення зберігається за заданим розташуванням, підтримуються популярні растрові формати типу png, bmp, jpg (jpeg) (рис. 3).

Функція **counter\_clockwise** приймає список з 3 точок як аргумент та повертає **True**, якщо ці точки у такому порядку утворюють поворот проти годинникової стрілки, **False** інакше (рис. 4).

При запуску модуля як програми побудується зображення з датасету 9 та збережеться у розташування програми, у папку **result**, а датасет з випуклою оболонкою збережеться у кореневу папку (рис. 5).

```

upper_hull, lower_hull = [], []

for i in range(length):
    while len(lower_hull) >= 2 \
        and not counter_clockwise(lower_hull[-2:] + [points[i]]):
        lower_hull.pop()
    lower_hull.append(points[i])
lower_hull.pop()

for i in range(length - 1, -1, -1):
    while len(upper_hull) >= 2 \
        and not counter_clockwise(upper_hull[-2:] + [points[i]]):
        upper_hull.pop()
    upper_hull.append(points[i])
upper_hull.pop()

return draw_n_save(upper_hull + lower_hull)

```

Рисунок 2

```

def draw_n_save(points):
    with open(dataset_hull_name, 'w') as file:
        file.writelines(' '.join(map(str, p)) + '\n' for p in points)

    image = img.new('RGB', (960, 540), (255,) * 3)
    for i in range(len(points) - 1):
        Draw.Draw(image).line(
            (points[i], points[i + 1]),
            fill=(50, 150, 255),
            width=1
        )
    image.putpixel(points[i], 0)
    image.putpixel(points[-1], 0)
    image.save(output_name)

```

Рисунок 3

```
def counter_clockwise(pts: list[tuple[int, int]]) -> bool:
    '''Чи утворюють точки поворот проти годинникової стрілки'''
    a, b, c = pts
    if a[0] == b[0]:
        if b[1] > a[1]:
            return c[0] <= a[0]
        return c[0] >= a[0]

    q = (b[1] - a[1]) / (b[0] - a[0])
    return ((b[0] > a[0])*2 - 1)*(q*(c[0] - a[0]) + a[1] - c[1]) <= 0
```

Рисунок 4

```
if __name__ == '__main__':
    path = '\\'.join(__file__.split('\\')[:-1])
    convex_hull(
        f'{path}\\..\\DS9.txt',
        f'{path}\\..\\DS9_convex_hull.txt',
        f'{path}\\result\\output.png'
    )
```

Рисунок 5