

Санкт-Петербургский Национальный Исследовательский Университет  
ИТМО

Факультет Программной Инженерии и Компьютерной Техники



УНИВЕРСИТЕТ ИТМО

Лабораторная работа № 2

По дисциплине “ТФКП”

Вариант №24 (7 => 8)

Выполнил студент группы Р3208, поток 22.3:

Петров Вячеслав Маркович

Преподаватель:

Милюшин Александр Сергеевич

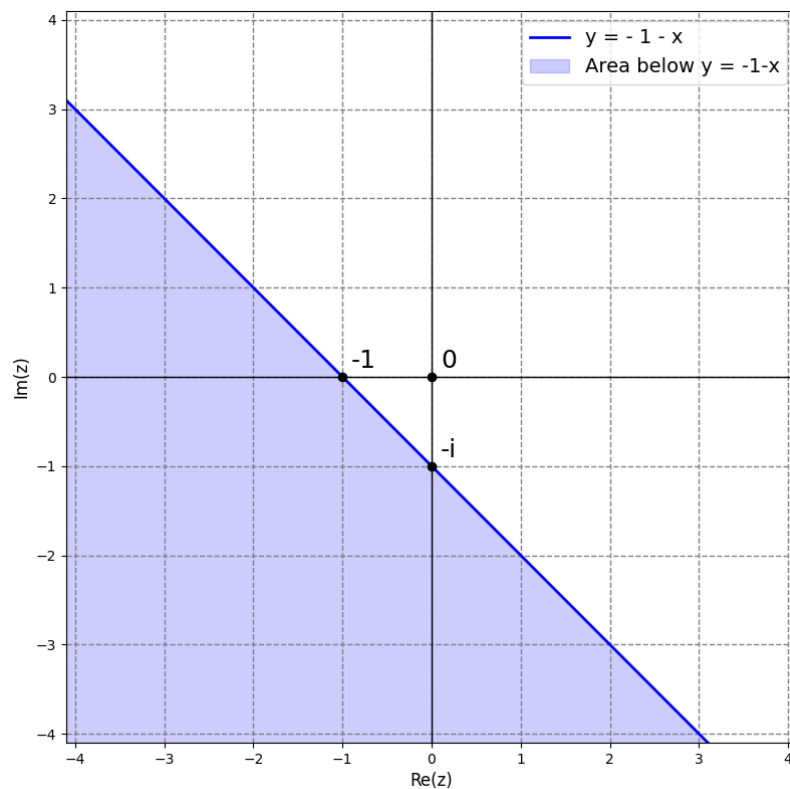
Санкт-Петербург 2025 г.

## Этапы работы

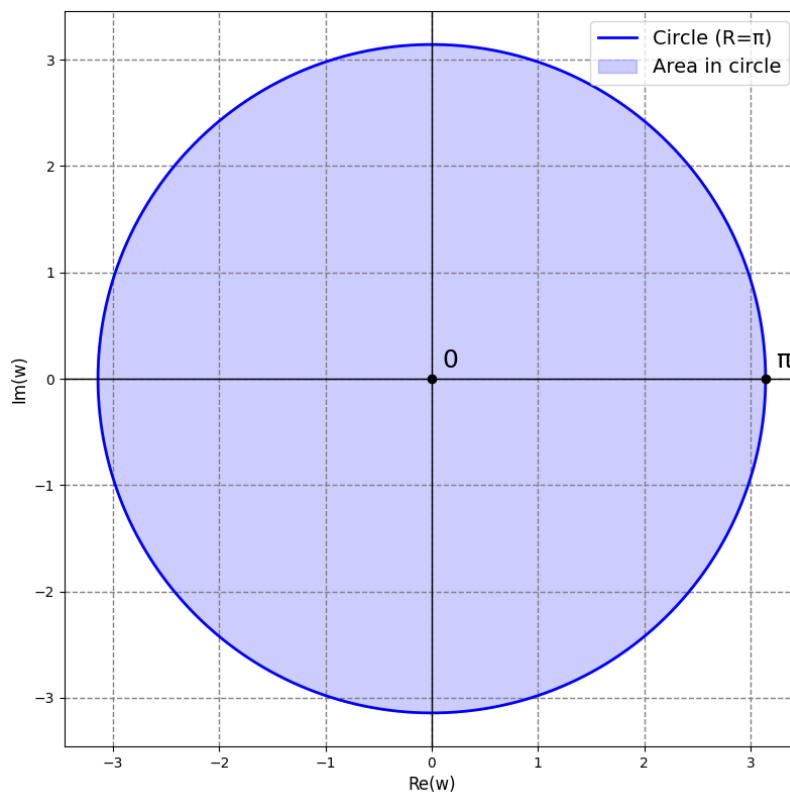
|                                 |   |
|---------------------------------|---|
| Задание 1. ....                 | 3 |
| Задание 2 .....                 | 4 |
| Задание 3 .....                 | 5 |
| Задание 4 .....                 | 6 |
| Программа на языке Python ..... | 6 |
| Вывод .....                     | 9 |

## Задание 1.

На первом графике изображено множество точек  $\operatorname{Im}(z) \leq -1 - \operatorname{Re}(z)$



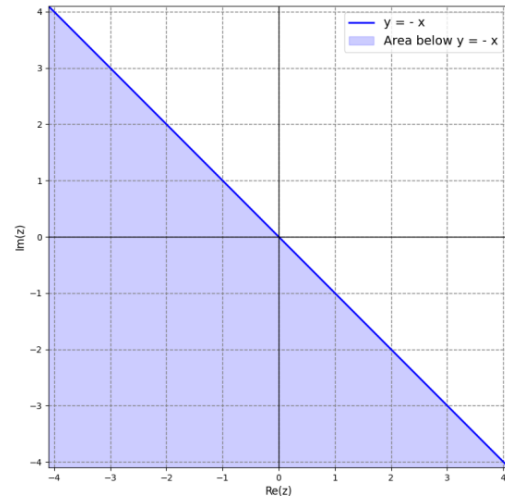
На втором графике изображено множество точек  $|w| \leq \pi$



## Задание 2

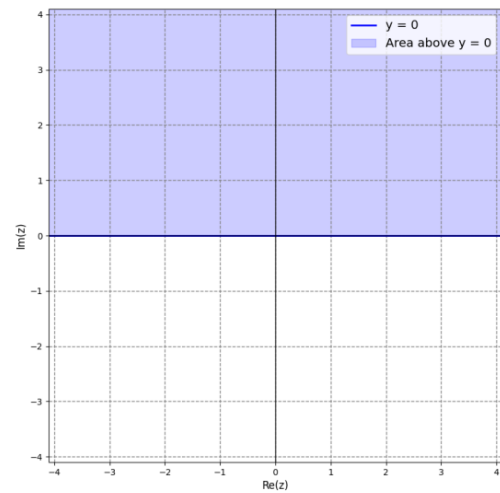
Конформное отображение, переводящее первое множество во второе:

1.  $z' = z + i$ . Поднимаем график на  $i$ .



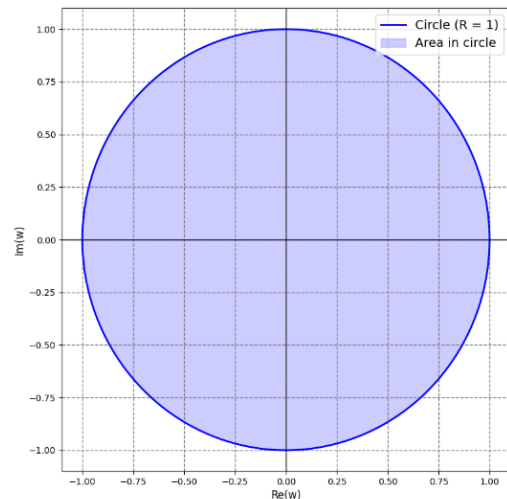
2.  $z'' = e^{-i\frac{3\pi}{4}} \cdot z' = e^{-i\frac{3\pi}{4}} \cdot (z + i)$ .

Поворачиваем график на  $\frac{3\pi}{4}$  по часовой.



3.  $z''' = \frac{z'' - i}{z'' + i} = \frac{e^{-i\frac{3\pi}{4}} \cdot (z+i) - i}{e^{-i\frac{3\pi}{4}} \cdot (z+i) + i}$ .

Преобразование в окружность, где точка  $(0; 1)$  переходит в центр окружности.

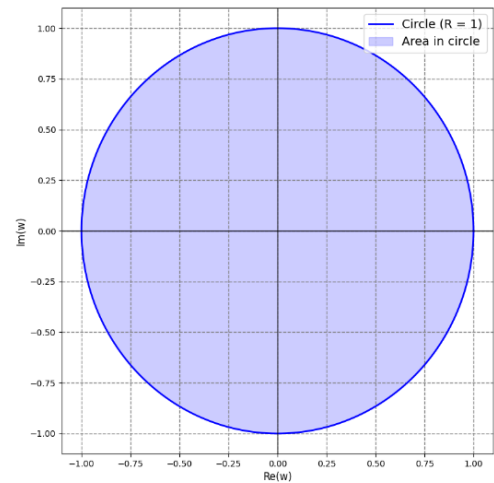


4.  $w = \pi \cdot z''' = \pi \cdot \frac{e^{-i\frac{3\pi}{4}} \cdot (z+i) - i}{e^{-i\frac{3\pi}{4}} \cdot (z+i) + i}$ . Увеличение радиуса окружности с 1 до  $\pi$

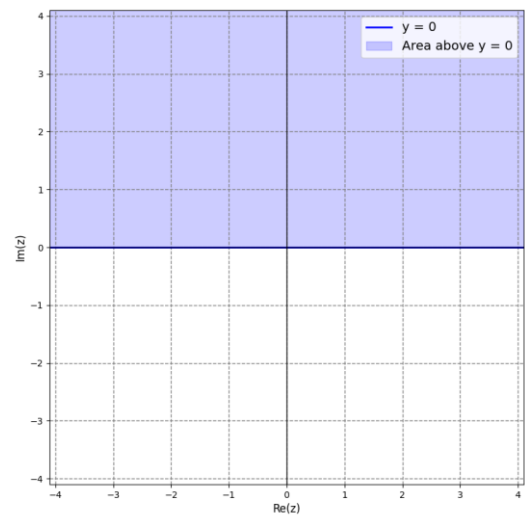
### Задание 3

Конформное отображение, переводящее второе множество в первое:

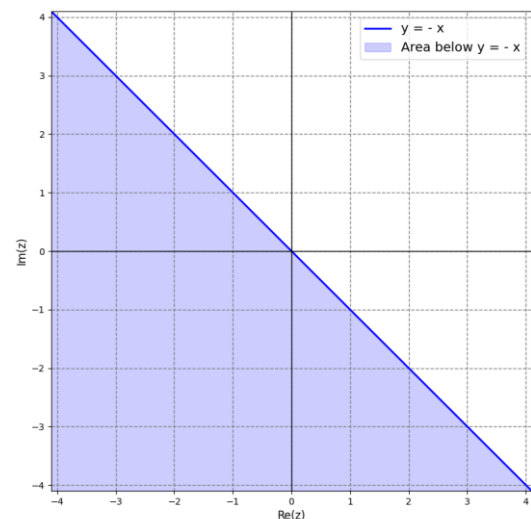
1.  $z' = \frac{z}{\pi}$ . Уменьшение радиуса с  $\pi$  до 1



2.  $z'' = i \frac{1+z'}{1-z'} = i \frac{1+\frac{z}{\pi}}{1-\frac{z}{\pi}} = i \frac{\pi+z}{\pi-z}$ . Переход от окружности к верхней полуплоскости, где центр окружности перешёл в точку (0; 1)



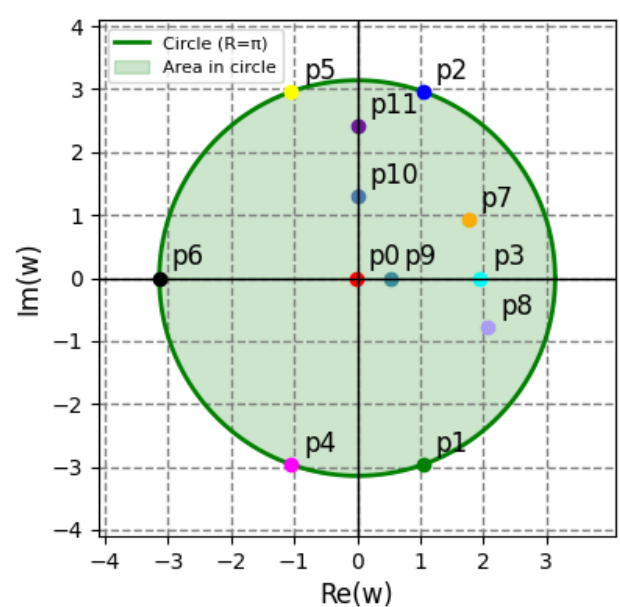
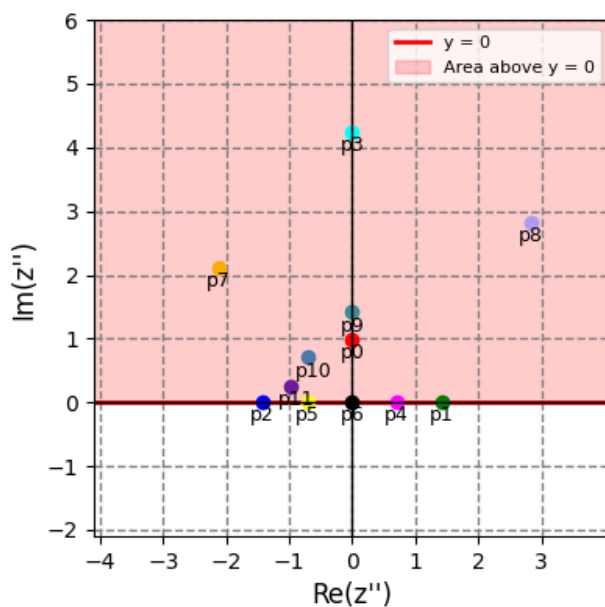
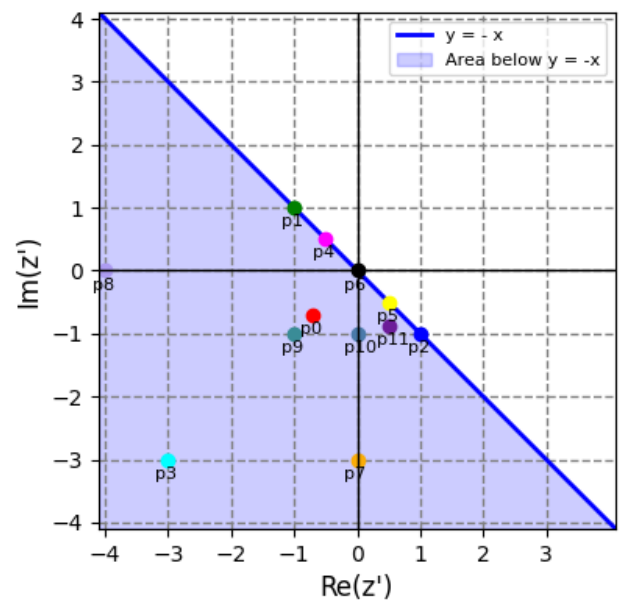
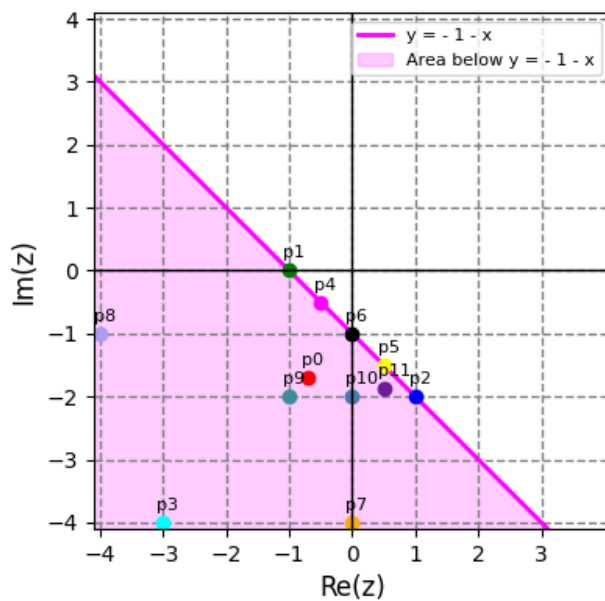
3.  $z''' = e^{i\frac{3\pi}{4}} \cdot z'' = e^{i\frac{3\pi}{4}} \cdot i \frac{\pi+z}{\pi-z}$ . Поворот полуплоскости против часовой на  $\frac{3\pi}{4}$



4.  $w = z''' - i = e^{i\frac{3\pi}{4}} \cdot i \frac{\pi+z}{\pi-z} - i$ . Опускаем на график на  $i$ .

## Задание 4

Преобразование первого множества во второе с дополнительными 11 точками, показывающими форму промежуточных и конечных множеств.



## Программа на языке Python

```
import math
```

```
import numpy as np
```

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

```
# Задаем значения x
```

```
x = np.linspace(-4.1, 4.1, 50) # Диапазон от -4 до 4, 50 точек
```

```
# Создаем график
```

```
plt.figure(figsize=(8, 8))
```

```
# фигуры
```

```

plt.subplot(2, 2, 1)
plt.plot(x, -1 - x, label='y = - 1 - x', color='magenta', linewidth=2)
plt.subplot(2, 2, 2)
plt.plot(x, -x, label='y = - x', color='blue', linewidth=2)
plt.subplot(2, 2, 3)
plt.plot(x, 0 * x, label='y = 0', color='red', linewidth=2)

plt.subplot(2, 2, 4)
theta = np.linspace(0, 2 * np.pi, 500) # Угол от 0 до 2π
radius = np.pi # 1
circle_x = radius * np.cos(theta) # x-координаты окружности
circle_y = radius * np.sin(theta) # y-координаты окружности
plt.plot(circle_x, circle_y, label='Circle (R=π)', color='green', linewidth=2)

# Закрашиваем
plt.subplot(2, 2, 1)
plt.fill_between(x, -1 - x, -4.1, color='magenta', alpha=0.2, label='Area below y = - 1 - x')

plt.subplot(2, 2, 2)
plt.fill_between(x, -x, -4.1, color='blue', alpha=0.2, label='Area below y = -x')

plt.subplot(2, 2, 3)
plt.fill_between(x, 0, 6, color='red', alpha=0.2, label='Area above y = 0')

plt.subplot(2, 2, 4)
plt.fill_between(circle_x, circle_y, color='green', alpha=0.2, label='Area in circle')

# точки
x_arr = [-0.7, -1, 1, -3, -0.5, 0.5, 0, 0, -4, -1, 0, 0.5]
y_arr = [-1.7, 0, -2, -4, -0.5, -1.5, -1, -4, -1, -2, -2, -1.87]
color_states = ["red", "green", "blue", "cyan", "magenta", "yellow", "black", "#FAAC07", "#AD9CF1",
"#3A8C97", "#4A7AAA", "#6C1C99"]
for i in range(len(x_arr)):
    x1 = x_arr[i]
    y1 = y_arr[i]
    color_state = color_states[i]

plt.subplot(2, 2, 1)
plt.plot(x1, y1, color=color_state, marker='o')
plt.annotate('p' + str(i), xy=(x1, y1), xytext=(x1 - 0.1, y1 + 0.2), fontsize=8, color='black')

plt.subplot(2, 2, 2)
plt.plot(x1, (y1 + 1), color=color_state, marker='o')
plt.annotate('p' + str(i), xy=(x1, (y1 + 1)), xytext=(x1 - 0.2, (y1 + 1) - 0.3), fontsize=8, color='black')

plt.subplot(2, 2, 3)
plt.plot(-(x1 - y1 - 1) / math.sqrt(2), -(x1 + y1 + 1) / math.sqrt(2), color=color_state, marker='o')
plt.annotate('p' + str(i),
            xy=(-(x1 - y1 - 1) / math.sqrt(2), -(x1 + y1 + 1) / math.sqrt(2)),
            xytext=(-(x1 - y1 - 1) / math.sqrt(2) - 0.2, -(x1 + y1 + 1) / math.sqrt(2) - 0.3),

```

```

        fontsize=9, color='black')

plt.subplot(2, 2, 4)
x_here = 0.5 * (x1 - y1 - 1) ** 2 + (0.5 * (x1 + y1 + 1) ** 2 - 1)
y_here = math.sqrt(2) * (x1 - y1 - 1)
znamenatel = (0.5 * (x1 - y1 - 1) ** 2 + ((-1 / math.sqrt(2)) * (x1 + y1 + 1) + 1) ** 2)
plt.plot(math.pi * x_here / znamenatel, math.pi * y_here / znamenatel, color=color_state, marker='o')
plt.annotate('p' + str(i),
            xy=(math.pi * x_here / znamenatel, math.pi * y_here / znamenatel),
            xytext=(math.pi * x_here / znamenatel + 0.2, math.pi * y_here / znamenatel + 0.2),
            fontsize=12, color='black')

# ОСНОВНОЕ
for i in range(4):
    plt.subplot(2, 2, i + 1)

    plt.xlim(-4.1, 4.1)
    if i == 2:
        plt.ylim(-2.1, 6)
    else:
        plt.ylim(-4.1, 4.1)

    plt.axhline(0, color='black', linewidth=1) # Горизонтальная ось
    plt.axvline(0, color='black', linewidth=1) # Вертикальная ось

    if i == 3:
        plt.xlabel('Re(w)', fontsize=12)
        plt.ylabel('Im(w)', fontsize=12)
    else:
        plt.xlabel('Re(z' + "" * i + ")", fontsize=12)
        plt.ylabel('Im(z' + "" * i + ")", fontsize=12)

    plt.xticks(np.arange(-4, 4, 1))
    plt.grid(color='gray', linestyle='--', linewidth=1)
    plt.legend(fontsize=8)

plt.subplots_adjust(left=0.08, right=0.99, bottom=0.08, top=0.99)
plt.show()

```



## Вывод

В ходе выполнения лабораторной работы я изучил построение конформных отображений и их построение при помощи языка программирования Python.