**Лабораторная работа №8**

**Обработка изображений с применением библиотеки PIL**

**Цель работы**: рассмотреть возможности работы библиотеки PIL, изучить её базовые инструменты для обработки изображений.

**Задание:** написать функцию, которая принимает текст и стиль шрифта в виде строки и создает изображение с полученным текстом и шрифтом.

**Выполнение задания:** Алгоритм функции create\_text\_image для создания изображения с текстом. Создаётся новое изображение заданного размера и цвета фона с помощью библиотеки PIL (Pillow). Тип изображения — RGB (красный, зелёный, синий). Библиотека PIL используется для вычисления ограничивающей рамки текста. Из этой рамки извлекаются ширина и высота текста. Этот алгоритм обеспечивает центрирование текста на изображении независимо от его длины. Обработка исключений гарантирует, что функция не завершится с ошибкой, если шрифт не найден.

**Листинг программы:**

*from PIL import Image, ImageDraw, ImageFont  
  
  
def create\_text\_image(text, font\_path, font\_size=40, image\_size=(500, 200), bg\_color=(255, 255, 255),  
 text\_color=(0, 0, 0)):  
  
 image = Image.new('RGB', image\_size, bg\_color)  
 draw = ImageDraw.Draw(image)  
  
 try:  
 font = ImageFont.truetype(font\_path, font\_size)  
 except IOError:  
 raise ValueError(f"Шрифт по пути {font\_path} не найден.")  
  
 text\_bbox = draw.textbbox((0, 0), text, font=font)  
 text\_width = text\_bbox[2] - text\_bbox[0]  
 text\_height = text\_bbox[3] - text\_bbox[1]  
  
 text\_x = (image\_size[0] - text\_width) // 2  
 text\_y = (image\_size[1] - text\_height) // 2  
  
 draw.text((text\_x, text\_y), text, fill=text\_color, font=font)  
  
 return image  
  
image = create\_text\_image("Привет, мир!", "arial.ttf", font\_size=50)  
image.show()  
image.save("text\_image.png")*

Результат работы кода:

