Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Лабораторная работа №8. Стеганография

Выполнил: cтудент гр. 853501

Яковлев А.Б.

Проверил:

Протько М.И.

Минск 2021

# Постановка задачи и описание алгоритма

**Цель:** реализовать программное средство, сокрытия (извлечения) текстового сообщения в (из) JPEG изображение(я) на основе метода сокрытия в частотной области изображения.

В отличие от криптографической защиты информации, предназначенной для сокрытия содержания информации, стеганографическая защита предназначена для сокрытия факта наличия (передачи) информации.

Методы и средства, с помощью которых можно скрыть факт наличия информации, изучает **стеганография** (от греч. – тайнопись).

Методы и способы внедрения скрытой информации в электронные объекты относятся к **компьютерной стеганографии**.

**Основные стеганографические понятия**

Основными стеганографическими понятиями являются **сообщение** и **контейнер.**

***Сообщением****m ∈ M*, называют секретную информацию, наличие которой необходимо скрыть, где *M*- множество всех сообщений, обычно *M = Z2n* для *n ∈ Z.*

***Контейнером*** *b ∈ B* называют несекретную информацию, которую используют для сокрытия сообщений, где *B* - множество всех контейнеров, обычно *B=Z2q*, при этом *q>>n*.

***Пустой контейнер*** (**контейнер-оригинал**) **-** это контейнер *b* , не содержащий сообщения, **заполненный контейнер** (**контейнер-результат**) *bm* - это контейнер *b*, содержащий сообщение *m*.

***Стеганографическим преобразованием*** принято называть зависимости

*F: M×B×K → B, F -1 : B×K → M,*

которые сопоставляют тройке (сообщение, пустой контейнер, ключ) контейнер-результат, и паре (заполненный контейнер, ключ) исходное сообщение, т.е.

*F(m,b,k) = bm,k , F-1 (b m,k ) = m, где m ∈ M, b, bm ∈ B, k∈K*

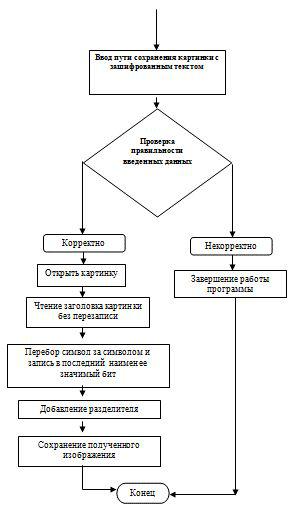
***Стеганографической системой*** называют (*F, F-1, M, B, K*) совокупность сообщений, контейнеров и связывающих их преобразований.

Среди методов и технологий, использующих стеганографическую защиту информации, наиболее развитыми представляются технологии защиты авторских прав на мультимедийную продукцию.

Предлагаемые на рынке программного обеспечения технологии и системы защиты авторских прав используют методы цифровой стеганографии *.* Системы защиты авторских прав сопровождают идентифицирующей информацией объекты, представляющие собой цифровое содержание: графические файлы, аудио- и видео файлы.

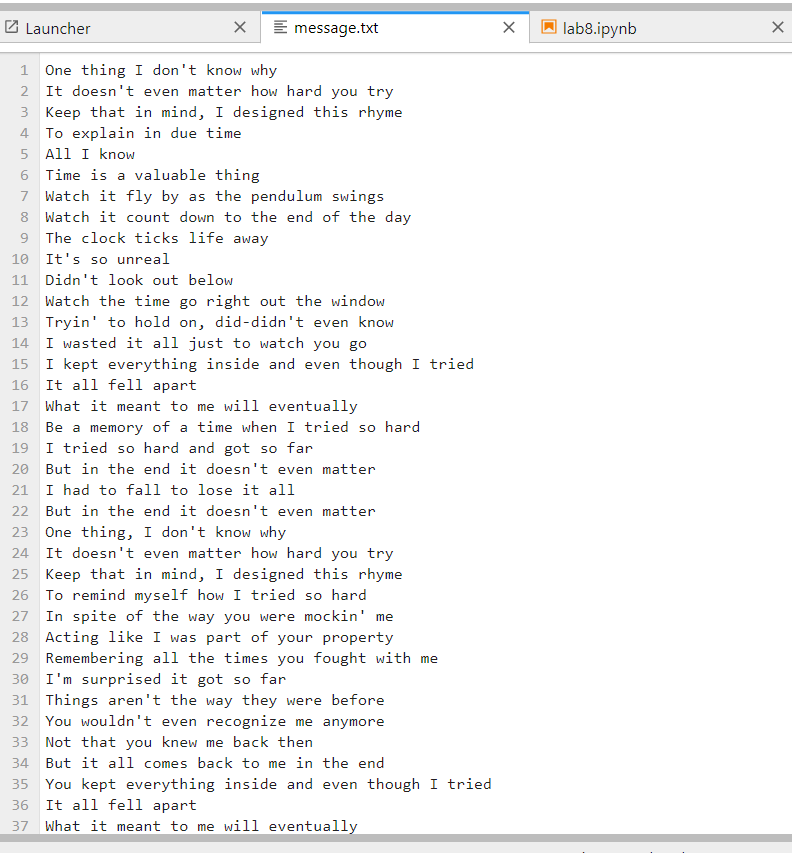
Самой известной технологией в области защиты прав автора на графическую информацию является технология Digital Water Marc (цифровой водяной знак) компании Digimarc Corporation (www.digimarc.com). Специальный программный продукт PictureMarc (ключевая часть технологии) позволяет внедрять в изображение цифровой идентификатор (метку) создателя. Для получения собственного идентификатора пользователь обязан зарегистрироваться в сервисном центре компании Digimarc (MarcCentre). Цифровая метка при внедрении в изображение кодируется величиной яркости пикселей, что определяет стойкость метки при различных трансформациях графического файла (редактирование, уменьшение/увеличение изображения, преобразование в другой формат, сжатие). Более того, цифровая метка, внедренная таким способом, не теряется даже после печати и последующего сканирования. Однако, цифровая метка не может быть ни изменена, ни удалена из маркированного изображения. Считывается цифровая метка с помощью программы ReadMarc. Специальный программный продукт MarcSpider просматривает изображения, доступные через Internet, и сообщает о незаконном использовании.

# Блок-схемы алгоритмов

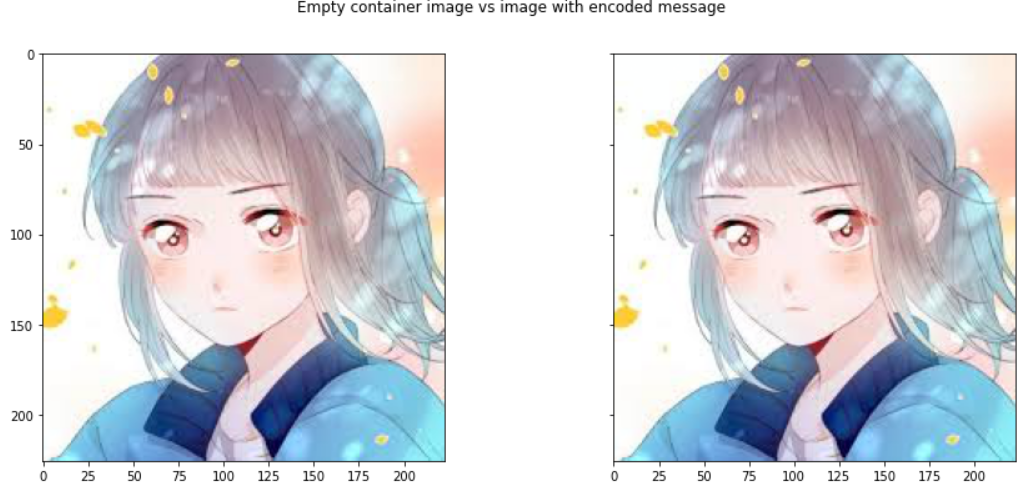


# Результаты выполнения программы

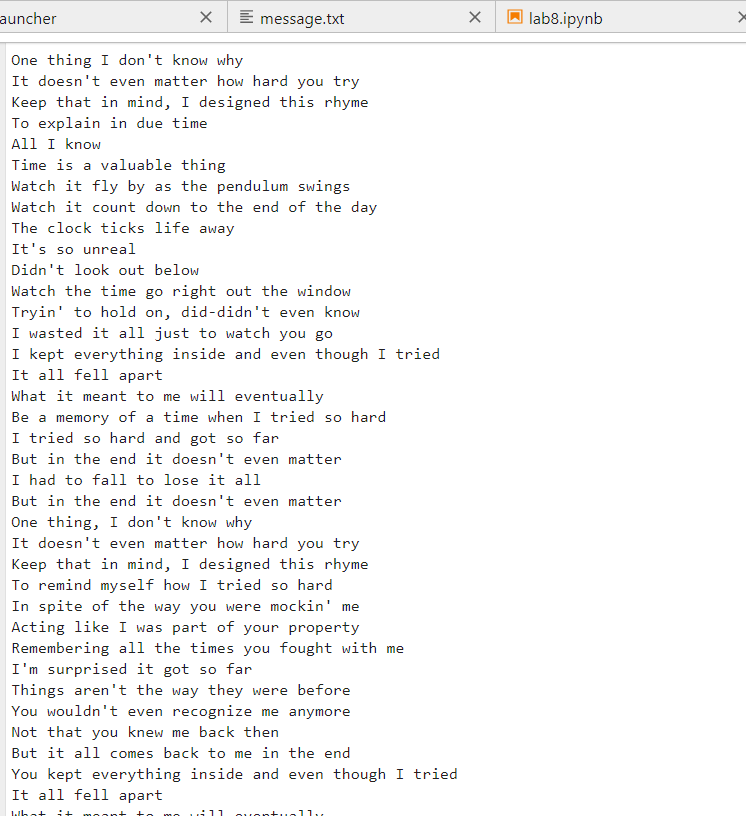
Исходное сообщение



Изображение с закодированным сообщением



Сообщение после декодирования



# Исходный код

import numpy as np  
import matplotlib.pyplot as plt  
  
from PIL import Image  
  
  
with Image.open('picture.jpg') as image:  
 plt.imshow(image)  
  
  
def encode(image: Image, message: str) -> Image:  
 image\_array = np.moveaxis(np.array(image, dtype='uint8'), -1, 0)  
 arr = image\_array.flatten()  
 encoded\_message = message.encode('UTF-8')  
 curr = 0  
 for byte in encoded\_message:  
 for i in range(8):  
 arr[curr] -= (arr[curr] & 1)  
 if (byte & (1 << i)) > 0:  
 arr[curr] += 1  
 curr += 1  
  
 return Image.fromarray(np.moveaxis(arr.reshape(image\_array.shape), 0, -1))  
  
  
def decode(image: Image, message\_length: int) -> str:  
 image\_array = np.moveaxis(np.array(image, dtype='uint8'), -1, 0)  
 arr = image\_array.flatten()  
 result = []  
 for i in range(message\_length):  
 curr = 0  
 for j in range(7, -1, -1):  
 curr = (curr << 1) + (arr[i \* 8 + j] & 1)  
 result.append(chr(curr))  
 return ''.join(result)  
  
  
with open('message.txt', 'r') as f:  
 message = f.read()  
  
  
with Image.open('picture.jpg') as image:  
 image\_with\_message = encode(image, message)  
 decoded\_message = decode(image\_with\_message, len(message))  
 fig, axes = plt.subplots(1, 2, figsize=(15, 6), sharey='all')  
 fig.suptitle('Empty container image vs image with encoded message')  
 axes[0].imshow(image)  
 axes[1].imshow(image\_with\_message)  
 plt.show()  
  
  
with open('decoded\_message.txt', 'w') as f:  
 f.writelines(decoded\_message)

# Вывод

В результате выполнения работы была реализована программа, позволяющая добавлять скрытые сообщения в изображения с помощью изменения последних битов пикселей.