

The purpose of the work: to study methods for detecting a hidden text message inside some text using changes in its formatting.

Assignment for laboratory work:

1) For a given file, it is necessary to programmatically determine the hidden message and the method used for its steganographic hiding (to implement the method, implement the program)

2) Character formatting methods used for secret messages (whole characters, zeros or ones):

- color of symbols;
- background color;
- font size;
- font scale;
- intercharacter interval.

3) Used binary character encodings:

- without encoding;
- Baudot code (MTK-2);
- KOI-8R;
- cp866;
- Windows 1251.

Baudot code (MTK-2) - The three-register telegraph code MTK-2 was adopted in the USSR in 1963. The code is 5-bit (a total of 32 different combinations), so 3 different registers are used (Russian, Latin, numbers), switched by control characters RUS, LAT, CIF. There are no letters Ъ and Ё; instead of the letter H, the number 4 was used.

KOI-8 (Information Interchange Code, 8 bits), KOI8 is an eight-bit ASCII compatible code page. Designed for coding letters of Cyrillic alphabets. It was widely used as the main Russian encoding in Unix-compatible operating systems and in e-mail, but closer to 2010, with the spread of Unicode, it is gradually falling into disuse.

Alternative encoding GOST (CP866) is a code page based on CP437, where all specific European characters in the second half are replaced by Cyrillic, and pseudo-graphic characters are left untouched. The disadvantage of this encoding is the gap in the order of small Cyrillic letters. Developed in 1984 at the IVND of the USSR Academy of Sciences

Windows-1251 is a character set and encoding that is the standard 8-bit encoding for Russian versions of Microsoft Windows prior to version 10.

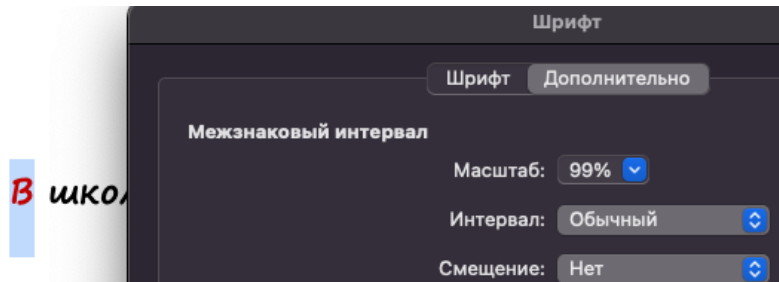
Text fragment (3rd option)

В школу

*Почему сегодня Петя
Просыпался десять раз?
Потому что он сегодня
Поступает в первый класс.*

*Он теперь не просто мальчик,
А теперь он новичок.
У него на новой куртке
Отложной воротничок.*

В школу



Screenshots of the program:

Menu demo

```
o (venv) nikita@Air ~/5 курс/Сафарьян/1 лаба$ python3 1.py
codes length 752
code - 100000011110101111100001111000101110000010101110001000001010110110100000101000111110000010
10010110100010101000001010010111100010111000011110111100100000010100100100000101000111101011111
0000111100010111000001010111000100000101011101110000111100010111010001010100000101001011110
00100010111000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
enter decoding number
1-MTK2
2-koi8_r
3-cp866
4-cp1251
5-all types
```

Choice 5 th paragraph

```

5
result with code Bodo / MTK2:
ЕМУКОАХЫАЕР
ЮЩ7
ЩЩ7Щ3((
5Щ/57Щ6ЮЩ
ЩЩ7
/
result with koi8-r: |КАБЮЩ Ъ=ёЮГГБАО ) ||КАБЮЩ ЪАБКГГБ.
result with cp866: Быстро нагревается ) быстро остывает.
result with cp1251: Ёлбва® ЖаГў Гвбп ) Ўлбва® ®бвлў Гв.

```

Conclusion: This laboratory work allowed to study methods for detecting a hidden text message inside some text using changes in its formatting.

<https://github.com/phantom-dreamer/Steganography>



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра Кибербезопасность информационных систем

ПРАКТИЧЕСКАЯ РАБОТА № 1

по дисциплине «Математические методы сокрытия и маскирования информации»

Выполнил обучающийся гр. ВКБ53

Егоров Никита Валентинович

Проверил

Доцент, Сафарьян Ольга Александровна

подпись

Ростов-на-Дону
2023

Цель работы: изучить методы обнаружения скрытого текстового сообщения внутри некоторого текста используя изменения в его форматировании.

Задание на выполнение лабораторной работы:]

1) Для заданного файла необходимо программно определить скрытое сообщение и использованный метод его стеганографического сокрытия (для реализации метода реализовать программу)

2) Способы форматирования символов, применяемые для секретных сообщений (символов целиком, нулей или единиц):

- цвет символов;
- цвет фона;
- размер шрифта;
- масштаб шрифта;
- межсимвольный интервал.

3) Применяемые двоичные кодировки символов:

- без кодировки;
- код Бодо (МТК-2);
- КОИ-8R;
- cp866;
- Windows 1251.

Код Бодо (МТК-2) - Телеграфный трёхрегистровый код МТК-2 был принят в СССР в 1963 году. Код 5-битовый (всего 32 разных комбинации), поэтому используются 3 разных регистра (русский, латинский, цифры), переключаемые управляющими символами РУС, ЛАТ, ЦИФ. Букв Ъ и Ё нет; вместо буквы Ч использовали цифру 4.

КОИ-8 (код обмена информацией, 8 бит), KOI8 — восьмибитовая кодовая страница, совместимая с ASCII. Разработана для кодирования букв кириллических алфавитов. Была широко распространена как основная русская кодировка в Unix-совместимых ОС и в электронной почте, однако ближе к 2010 году, с распространением Юникода, постепенно выходит из употребления.

«Альтернативная кодировка ГОСТ (CP866) — основанная на CP437 кодовая страница, где все специфические европейские символы во второй половине заменены на кириллицу, а псевдографические символы оставлены нетронутыми. Недостатком данной кодировки является разрыв в порядке малых кириллических букв. Разработана в 1984 году в ИВНД Академии наук СССР

Windows-1251 — набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии.

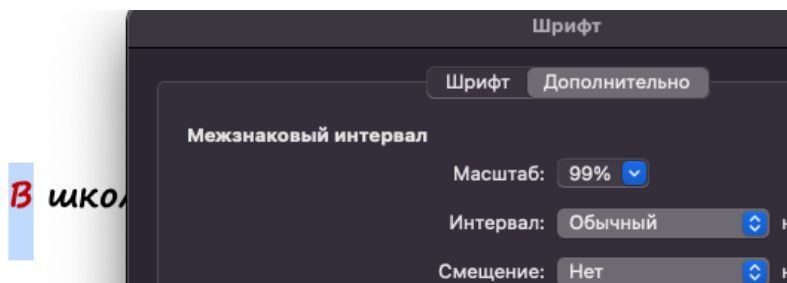
Фрагмент текста (3 вариант)

В школу

*Почему сегодня Петя
Просыпался десять раз?
Потому что он сегодня
Поступает в первый класс.*

*Он теперь не просто мальчик,
А теперь он новичок.
У него на новой куртке
Отложной воротничок.*

В школу



Скриншоты работы программы:

Демонстрация меню

```
○ (venv) nikita@Air ~/5 курс/Сафарьян/1 лаба$ python3 1.py
codes length 752
code - 100000011110101111100001111000101110000010101110001000001010110100000101000111110000010
1001011010001010100000101001011110001011100001111011110010000000101001001000001010000111101011111
0000111100010111000001010111000100000101011101110000111100010111010111010001010100000101001011110
0010001011100000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
enter decoding number
1-MTK2
2-koi8_r
3-cp866
4-cp1251
5-all types
```

Выбор 5-го пункта

```

5
result with code Bodo / MTK2:
ЕМУКОАХЫАЕР
ЮЩ7
ЩЩ7Щ3((
5Щ/57Щ6ЮЩ
ЩЩ7
/
result with koi8-r: |КАБЮ||=ёЮ||=БАО ) ||КАБЮ|| АБК||Б.
result with cp866: Быстро нагревается ) быстро остывает.
result with cp1251: Ёлва® Жагъ Гвбп ) Ылва® @бвлъ Гв.

```

Вывод: Данная лабораторная работа позволила изучить методы обнаружения скрытого текстового сообщения внутри некоторого текста используя изменения в его форматировании.

<https://github.com/phantom-dreamer/Steganography>

Листинг программы

```
import docx
import MTK2
from docx.shared import Pt, RGBColor
from docx.enum.text import WD_COLOR_INDEX

doc = docx.Document('variant03.docx')

def run_get_spacing(run):
    rPr = run._r.get_or_add_rPr()
    spacings = rPr.xpath("./w:spacing")
    return spacings

def run_get_scale(run):
    rPr = run._r.get_or_add_rPr()
    scale = rPr.xpath("./w:w")
    return scale

def main():
    code = ''
    for paragraph in doc.paragraphs:
        for run in paragraph.runs:
            font_color = run.font.color.rgb
            font_size = run.font.size
            font_highlight_color = run.font.highlight_color
            font_scale = run_get_scale(run)
            font_spacing = run_get_spacing(run)

            if (font_color != RGBColor(0, 0, 0) or
                font_size.pt != 12.0 or
                font_highlight_color != WD_COLOR_INDEX.WHITE or
                font_spacing or font_scale):
                for i in range(len(run.text)):
                    code += '1'
            else:
                for i in range(len(run.text)):
                    code += '0'

    print(f'codes length {len(code)}')
    if len(code) % 16 != 0:
        code += "0" * (16 - len(code) % 16)
        print(f'length after additional value - {len(code)}')
    print(f'code - {code}')

    choose = input('enter decoding number \n 1-MTK2 \n 2-koi8_r \n 3-cp866 \n 4-cp1251 \n 5-all\n types \n')
    match choose:
        case '1':
            plain_text = MTK2.MTK2_decode(code)
            print(f'result with code Bodo / MTK2:\n{plain_text}')
        case '2':
            plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="koi8_r")
            print(f'result with koi8-r: {plain_text}')
        case '3':
            plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="cp866")
            print(f'result with cp866: {plain_text}')
        case '4':
            plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="cp1251")
            print(f'result with cp1251: {plain_text}')
```



```
case '5':
    plain_text = MTK2.MTK2_decode(code)
    print(f'result with code Bodo / MTK2:\n{plain_text}')
    plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="koi8_r")
    print(f'result with koi8-r: {plain_text}')
    plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="cp866")
    print(f'result with cp866: {plain_text}')
    plain_text = bytes.fromhex(hex(int(code, 2))[2:]).decode(encoding="cp1251")
    print(f'result with cp1251: {plain_text}')

if __name__ == '__main__':
    main()
```