

*Додаток 1*

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни  
«Основи програмування - 2. Модульне програмування»

«Текстові файли»

Варіант 4

Виконав студент Бутов Даниїл Романович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

## Лабораторна робота 1

### Файли даних

**Мета** – вивчити особливості створення і обробки файлів даних.

### Варіант 4

4. Створити текстовий файл. Сформувати новий текстовий файл, що складається з рядків вхідного файлу, у яких вилучені всі односимвольні слова, а також зайві роздільники (пробіли, коми, крапки), що повторюються підряд. Визначити і дописати в кінець нового файлу кількість вилучених слів і зайвих символів кожного виду. Вивести вміст вихідного і створеного файлів.

### Особливості завдання.

Нам потрібно обробити заданий текст ( вводить користувачем ), потім його записати у новий текстовий файл. Вхідний файл буде мати назву "input.txt", вихідний - "output.txt".

Залежно від того, який розмір нашого контенту для файла буде, нам потрібно зберігати наш текст в деякій структурі, в нашому випадку ми будемо зберігати у векторі. Генерування вектора ми будемо робити окремо, записувати текст у файл ми будемо по-рядково в спеціальній функції. Надалі вся обробка тексту буде в векторі по-рядково.

Функція для створення нового тексту розділена на 3: Основна (Виклик двох інших функцій та ітерація по вектору, формування нового тексту та додавання інформації щодо видалених символів), Видалення односимвольних слів, Видалення роздільників, що повторюються підряд.

Останнім кроком стане запис по-рядково нового тексту до вихідного файлу та виведення його в консоль.

### Програма.

#### *Python*

*main.py*

```
import modules

# Variable with name of files
input_file = 'input.txt'
output_file = 'output.txt'

# Create input.txt
input_list = modules.create_input_text()
modules.create_file(input_file, input_list)
modules.read_txt_file(input_file)

# Change content of input file
output_list = modules.create_new_content(input_list)

# Save content to output file
modules.create_file(output_file, output_list)
modules.read_txt_file(output_file)
```

## modules.py

```
# Create new text for input.txt
def create_input_text():
    text = []
    flag = True
    print("Enter text (Press double Enter if u wanna end text)")
    while flag:
        temp = input()
        if temp == '':
            flag = False
        else:
            text.append(temp)
    return text

# Read and print txt file
def read_txt_file(name_of_file):
    print(f"\nYour {name_of_file} ->")
    with open(name_of_file, 'r') as file_txt:
        for line in file_txt:
            print(line, end='')

# Create output.txt
def create_file(name_of_file, content):
    with open(name_of_file, 'w+') as file:
        for i in range(0, len(content)):
            file.write(content[i])
            if i != len(content)-1:
                file.write('\n')

# Create new content of output.txt
def create_new_content(content):
    new_content = []

    chars_dict = {'comma': 0, 'point': 0, 'space': 0, 'words': 0}
    chars_string = "\nComma has been deleted {comma} times." \
                  "\nSpace has been deleted {space} times." \
                  "\nPoint has been deleted {point} times." \
                  "\nWords have been deleted {words} times."

    for string in content:
        new_string, chars_dict = delete_specific_chars(string, chars_dict)
        new_string, chars_dict = delete_words(new_string, chars_dict)
        new_content.append(new_string)
    new_content.append(chars_string.format(comma=chars_dict['comma'], space=chars_dict['space'],
                                          point=chars_dict['point'], words=chars_dict['words']))
    return new_content

# Delete specific chars like , . 'space'
def delete_specific_chars(string, chars_dict):
    new_string = ''
    specific_chars = [',', '.', ' ']
    for count in range(0, len(string) - 1):
        if string[count] == string[count + 1] and string[count] in specific_chars:
            if string[count] == ',':
                chars_dict['comma'] += 1
            elif string[count] == '.':
                chars_dict['point'] += 1
            else:
                chars_dict['space'] += 1
        else:
            new_string += string[count]
    new_string += string[-1]
    return new_string, chars_dict

# Delete specific words of content file
def delete_words(string, chars_dict):
    new_string_list = []
    string_list = string.split()
    for word in string_list:
        if len(word) != 1:
            new_string_list.append(word)
        elif not word.isalpha():
            new_string_list.append(word)
        else:
            chars_dict['words'] += 1
    new_string = ' '.join(new_string_list)
    return new_string, chars_dict
```

## C++

### *main.cpp*

```
#include "functions.h"

int main(){
    // Название файлов
    string NameOfInput= "input.txt";
    string NameOfOutput = "output.txt";

    // Создание переменных
    vector<string> newText;
    vector<string> text;

    // Обработка input.txt
    text = generateContent();
    writeFile( NameOfFile: NameOfInput, content: text);
    readFile( NameOfFile: NameOfInput);

    // Создание нового контента
    newText = createNewContent( content: text);

    // Обработка output.txt
    writeFile( NameOfFile: NameOfOutput, content: newText);
    readFile( NameOfFile: NameOfOutput);
}
```

### *functions.h*

```
//
// Created by Danya Butov on 14.02.2022.
//

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;

// Генерация Контента для изначального файла
vector<string> generateContent();

// Перенос контента в файл
void writeFile(const string& NameOfFile, vector<string> content);

// Обработка файла (ввод-вывод)
void readFile(const string& NameOfFile);

// Создание Нового выходного контента
vector<string> createNewContent(vector<string> content);
string newContent(string content, int& comma, int& space, int& point);
string deleteWords(string content, int& wordsCount);
```

## functions.cpp

```
//
// Created by Danya Butov on 14.02.2022.
//

#include "functions.h"

// Генерування контенту
vector<string> generateContent(){
    vector<string> content;
    string stringLine;
    int endCode = 126;

    cout << "Enter content of ur file (use - and Enter):\n";
    while(int(stringLine[0]) != endCode){
        getline(&cin, &stringLine);
        content.push_back(stringLine);
    }
    content.pop_back();

    return content;
}

// Перенос контенту в файл
void writeFile(const string& NameOfFile, vector<string> content){
    ofstream out;

    out.open(NameOfFile);
    for (int i = 0; i < content.size(); ++i) {
        out << content[i];
        if (i != content.size()-1)
            out << '\n';
    }
    out.close();
}

// Читання файла
void readFile(const string& NameOfFile){
    ifstream in;
    string line;

    cout << "Your " + NameOfFile + " file has content:" << endl;
    in.open(NameOfFile);
    while(getline(&in, &line))
        cout << line << endl;
    in.close();

    cout << endl;
}

// Створення нового контенту
vector<string> createNewContent(vector<string> content){
    vector<string> NewContent;
    string newString;

    string comma_count = "\nComma was deleted -> ";
    int comma = 0;
    string space_count = "Space was deleted -> ";
    int space = 0;
    string point_count = "Point was deleted -> ";
    int point = 0;
    string end_count = " times\n";
    string words_count = "Words were deleted -> ";
    int word = 0;

    for (int i = 0; i < content.size(); ++i) {
        newString = deleteWords(content[i], &word);
        newString = newContent(content[i], newString, comma, space, point);
        NewContent.push_back(newString);
    }

    NewContent.push_back("\n" + comma_count + to_string(val comma) + end_count +
        space_count + to_string(val space) + end_count +
        point_count + to_string(val point) + end_count +
        words_count + to_string(val word) + end_count);

    return NewContent;
}

// Вилучення символів і підрахунок кожного
string newContent(string content, int& comma, int& space, int& point){
    string new_line = "";

    for (int i = 0; i < content.size(); ++i) {
        if (content[i] == content[i+1]){
            if (content[i] == ',')
                comma++;
            else if (content[i] == '.')
                point++;
            else if (content[i] == ' ')
                space++;
            else
                new_line += content[i];
        }
        else
            new_line += content[i];
    }
    new_line = new_line.substr(pos 0, n: new_line.size()-1);

    return new_line;
}

// Вилучення односимвольного слова
string deleteWords(const string content, int& wordsCount){
    string FixedContent = "";
    for (int i = 0; i < content.size(); ++i) {
        if (content[i-1] == ' ' and content[i+1] == ' ' and content[i] != ' '){
            wordsCount++;
        }
        else if (i == 0 and content[i+1] == ' ' and content[i] != ' '){
            wordsCount++;
        }
        else if (i == content.size() and content[i-1] == ' ' and content[i] != ' '){
            wordsCount++;
        }
        else
            FixedContent += content[i];
    }
    return FixedContent;
}
```

### **Висновок.**

Ми вивчили особливості створення і обробки файлів даних. Склали програму, яка змінює вхідний текст та записує його в інший. Навчилися опрацьовувати великий масив тексту та використовувати файли через методи їх обробки. Визначили декі особливості для роботи з текстом.