

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.  
Методології програмування»

«Файли даних»

Варіант 10

Виконав студент ІП-11, Друзенко Олександра Юріївна  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна  
(прізвище, ім'я, по батькові)

Київ 2022

**Мета:** вивчити особливості створення і обробки текстових файлів даних.

**Постановка задачі:** Створити текстовий файл. Сформувати новий текстовий файл, що складається з рядків вихідного файлу, розміщених у порядку зростання кількості символів у рядку. Наприкінці кожного рядка нового файлу дописати кількість символів у рядку. Вивести вміст вихідного і створеного файлів.

**Виконання мовою C++:**

1)код:

lib.h:

```
#pragma once
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <Windows.h>
using namespace std;

void createFile(string opFile); //введення тексту в файл
void enterFile(string opFile, vector <string> text); //введення тексту в файл з вектора
void readFile(string opFile); //виведення тексту з файла
vector <string> getText(string from); //запис рядків у вектор
int numoSymb(string str); //визначення кількості символів у рядку
vector <string> quicksort(vector <string> array); //алгоритм швидкого сортування
vector <string> addSymb(vector <string> array); //додавання кількості символів у кінець рядка
```

source.cpp:

```
#include "lib.h"

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    vector <string> text;

    createFile("test.txt");
    readFile("test.txt");
    text = getText("test.txt");
    text = quicksort(text);
    text = addSymb(text);
    enterFile("result.txt", text);
    readFile("result.txt");
}
```

lib.cpp:

```
#include "lib.h"

//=====введення тексту в файл=====
void createFile(string opFile) {
    ofstream outFile;
    string str;
    outFile.open(opFile, ios::out);    //файл створюється за замовчуванням в
    папці проекту
    if (!outFile.is_open()) {          //перевірка чи відкрився файл
        cout << "Cannot open file\n";
        return;
    }
    cout << "введіть текст(для закінчення вводу ctrl+z -> enter):\n";
    getline(cin, str, char(26));
    outFile << str;
    outFile.close();
}

//=====виведення тексту з файлу=====
void readFile(string opFile) {
    string str;
    ifstream inFile;
    inFile.open(opFile);
    if (!inFile.is_open()) {
        cout << "Cannot open file\n";
        return;
    }
    cout << endl << "читання файлу " << opFile << " : " << endl;
    while (getline(inFile, str)) {
        cout << str << endl;
    }
    inFile.close();
}

//=====запис рядків у вектор=====
vector <string> getText(string from) {
    fstream file;
    vector <string> res;
    string temp;
    file.open(from);
    while (getline(file, temp)) {    //поки є рядки, додаємо їх в кінець
    вектора
        res.push_back(temp);
    }
    file.close();
    return res;
}

//=====визначення кількості символів у рядку=====
int numoSymb(string str) {
    int res = 0;
    for (int i = 0; i < str.length(); i++) {
        if (str[i] != ' ') {
            res++;
        }
    }
    return res;
}
```

```

//=====алгоритм швидкого сортування=====
vector <string> quicksort(vector <string> array) {
    if (array.size() < 2)    //базовий випадок
    {
        return array;
    }
    else    //рекурсивний випадок
    {
        string pivot = array[0];    //ініціалізуємо опорний елемент
        vector <string> less;
        vector <string> greater;
        for (int i = 1; i < array.size(); i++) {
            if (numOfSymb(array[i]) <= numOfSymb(pivot))
            {
                //якщо елемент вектору менший або
                //дорівнює
                less.push_back(array[i]);    //то записуємо його до вектору
                less
            }
            else { greater.push_back(array[i]); }    //інакше до greater
        }
        vector <string> result;
        vector <string> temp;
        temp = quicksort(less);
        result.insert(result.end(), temp.begin(), temp.end());    //об'єднуємо
        //отримані вектори
        result.push_back(pivot);
        temp = quicksort(greater);
        result.insert(result.end(), temp.begin(), temp.end());
        return result;
    }
}

//=====додавання кількості символів у кінець рядка=====
vector <string> addSymb(vector <string> array) {
    for (int i = 0; i < array.size(); i++) {
        array[i] += " [ " + to_string(numOfSymb(array[i]))+" ]";
    }
    return array;
}

//=====введення тексту в файл з вектора=====
void enterFile(string opFile, vector <string> text) {
    fstream file;
    file.open(opFile, ios::out);
    if (!file.is_open()) {
        cout << "Cannot open file\n";
        return;
    }
    for (string str : text) {
        file << str << "\n";
    }
    file.close();
}

```

## 2)Випробування коду на C++:

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (C
Microsoft Visual Studio Debug Console

введіть текст(для закінчення вводу ctrl+z -> enter):
hello world
my name is Sasha
cat^Z

читання файлу test.txt :
hello world
my name is Sasha
cat

читання файлу result.txt :
cat [ 3 ]
hello world [ 10 ]
my name is Sasha [ 13 ]

E:\1kpi\prog_tests\C++\2nd\lab_1.1(2)\Debug\lab_1.1(2).ex
```

## Виконання мовою Python

1) Код:

module1.py:

```
#запис у файл тексту з перевіркою на комбінацію клавіш
def catchC(str,file):
    i=0
    while i<len(str):
        if str[i]==chr(17):
            str=str[:i:] # Робимо зріз до символу "Ctrl" + "Q"
            if (len(str)!=0):
                file.write(str+'\n')
            return 0
        i+=1
    file.write(str+'\n')
    return 1

#введення тексту
def inputText(name):
    print("enter text (stop: ctrl+q):")
    bo=1
    file = open(name,"w")
    while (bo):
        str_text=input()
        bo = catchC(str_text,file)
    file.close()

#запис рядків в список
def getText(textList):
    file = open("text.txt","r")
    for i in file.readlines():
        textList.append(i)
    file.close()
```

```

    return textList

#підрахунок СИМВОЛІВ
def countSymb(text):
    res=0
    for i in text:
        if i!=' ' and i!='\n':
            res+=1
    return res

#сортування
def quicksort(array):
    if len(array) < 2:
        return array
    else:
        pivot = array[0]
        less = [i for i in array[1:] if countSymb(i) <= countSymb(pivot)]
        greater = [i for i in array[1:] if countSymb(i) > countSymb(pivot)]
        return quicksort(less)+[pivot]+quicksort(greater)

#файл з результатом
def resFile(textList,name):
    file=open(name,"w")
    for i in textList:
        i=i[:-1]+" [ "+str(countSymb(i))+" ]\n"
        file.write(i)
    file.close()

#читання файлу
def readFile(name):
    print("\nread file ",name," :")
    file=open(name,"r")
    for i in file.read():
        print(i,end='')

```

lab\_1.1.py:

```

from module1 import *

text_list=[]
inputText("text.txt")
readFile("text.txt")
text_list = getText(text_list)
text_list = quicksort(text_list)
resFile(text_list,"result.txt")
readFile("result.txt")

```

2)Випробування коду на Python:

```
C:\Python39\python.exe
enter text (stop: ctrl+q):
hello world
my name is sasha
python^Q

read file text.txt :
hello world
my name is sasha
python

read file result.txt :
python [ 6 ]
hello world [ 10 ]
my name is sasha [ 13 ]
Press any key to continue . . .
```

**Висновок.** Отже, на цій лабораторній роботі я вивчила особливості створення і обробки текстових файлів даних, надбала навички написання програм для роботи з текстовими файлами.