

Основи програмування 2. Модульне програмування

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»

«Файли даних»

Варіант 29

Виконав студент

ПІ-15 Рибалка Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірив

Всечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 1

Текстові файли

Мета - вивчити особливості створення і обробки текстових файлів даних.

Індивідуальне завдання

Вимоги до програми

- 1) Створення файлу програмним шляхом.
- 2) Організація введення інформації аналогічно текстовим редакторам (ознака кінця рядка – натиснення Enter, ознака кінця введення файла – будь-яка комбінація клавіш або клавіша, що генерують розширений код).
- 3) Можливість доповнення файла даних.
- 4) Виведення усіх вхідних, проміжних і вихідних даних.

Варіант 29

29. Створити два текстових файли. Переписати в новий текстовий файл рядки, які є в першому вихідному файлі, і відсутні у другому. Визначити кількість таких рядків. Вивести вміст вихідних і створеного файлів.

Розв’язання

Код

C++

```
#include "func.h"

int main()
{
    string path1 = "input1.txt", path2 = "input2.txt", path3 = "output.txt";
    inputfile(path1, 1);
    inputfile(path2, 2);
    croutfile(path1, path2, path3);
    int flen = filelen(path3);
    cout << "Кількість знайдених рядків - " << flen << endl;
    printfile(path1);
    printfile(path2);
    printfile(path3);
    return 0;
}
```

func.h

```
#pragma once
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;

void inputfile(string, int); // Введення файлу через консоль
void croutfile(string, string, string); // Створення вихідного файлу
vector<string> matchlines(string, string); // Пошук однакових рядків
vector<string> lstlines(string); // Створення вектора з файлу
int filelen(string); // Визначення кількості рядків у файлі
void printfile(string); // Вивід файлу в консоль
```

func.cpp

```
#include "func.h"

void inputfile(string path, int num)
{
    string str;
    ofstream file(path);
    cout << "Введіть текст " << num << "-го файлу:" << endl;
    getline(cin, str);
    while(str.find(1) == string::npos) //Ознака кінця введення файлу Ctrl+A
    {
        file << str << endl;
        getline(cin, str);
    }
    file.close();
}

void croutfile(string path1, string path2, string path3)
{
    ofstream outf(path3);
    vector<string> lst = matchlines (path1, path2);
    for (int i = 0; i < lst.size(); i++) outf << lst[i] << endl;
    outf.close();
}

vector<string> matchlines(string path1, string path2)
{
    vector<string> inpf1 = lstlines(path1);
    vector<string> inpf2 = lstlines(path2);
    vector<string> lst;
    for (int i = 0; i < inpf1.size(); i++)
    {
        if (find(inpf2.begin(), inpf2.end(), inpf1[i])!=inpf2.end()) lst.push_back(inpf1[i]);
    }
    return lst;
}

vector<string> lstlines(string path)
{
    ifstream file(path);
    string str;
    vector<string> lst;
    while(getline(file, str)) lst.push_back(str);
    file.close();
    return lst;
}

int filelen(string path)
{
    int res = 0;
    string temp;
    ifstream file(path);
    while(getline(file, temp)) res++;
    file.close();
    return res;
}

void printfile(string path)
{
    cout << "Зміст файлу " << path << endl;
    vector<string> lst = lstlines(path);
    if (lst.size()!=0)
    {
        for (int i = 0; i < lst.size(); i++) cout << lst[i] << endl;
    }
    else cout << "Файл пустий" << endl;
}
```

Python

```
from func import *

path1 = 'Lab1/Python/input1.txt'
path2 = 'Lab1/Python/input2.txt'
path3 = 'Lab1/Python/output.txt'
inputfile(path1, 1)
inputfile(path2, 2)
croutfile(path1, path2, path3)
flen = filelen(path3)
print("Кількість знайдених рядків -", flen)
printfile(path1)
printfile(path2)
printfile(path3)
```

func.py

```

# Введення файлу через консоль
def inputfile(path, num):
    file = open(path, "w")
    string = input("Введіть текст " + str(num) + "-го файлу:\n")
    while chr(1) not in string: #Ознака кінця введення файлу Ctrl+A
        print(string, file = file)
        string = input()
    file.close()

# Створення вихідного файлу
def croutfile(path1, path2, path3):
    outf = open(path3, 'w')
    for i in matchlines(path1, path2):
        outf.write(i)
    outf.close()

# Пошук однакових рядків
def matchlines(path1, path2):
    inpf1 = lstlines(path1)
    inpf2 = lstlines(path2)
    lst=[]
    for i in inpf1:
        if i not in inpf2:
            lst.append(i)
    return lst

# Створення списку з файлу
def lstlines(path):
    file = open(path)
    lst = file.readlines()
    file.close()
    return lst

# Визначення кількості рядків у файлі
def filelen(path):
    with open(path) as file:
        return len(file.readlines())

# Вивід файлу в консоль
def printfile(path):
    print("Зміст файлу", path.split("/")[-1:][0])
    lst = lstlines(path)
    if len(lst)!=0:
        for i in lst:
            print(i, end="")
    else:
        print("Файл пустий")

```

Тестування**C++**

```
Введіть текст 1-го файлу:
123
qwe
asd
zxc
456
rty
fgh
vbn
^A
Введіть текст 2-го файлу:
123
rty
asd
vbn
zaq
xsw
cde
^A
Кількість знайдених рядків - 4
Зміст файлу input1.txt
123
qwe
asd
zxc
456
rty
fgh
vbn
Зміст файлу input2.txt
123
rty
asd
vbn
zaq
xsw
cde
Зміст файлу output.txt
qwe
zxc
456
fgh
```

Python

```
Введіть текст 1-го файлу:
123
qwe
456
rty
^A
Введіть текст 2-го файлу:
^A
Кількість знайдених рядків – 4
Зміст файлу input1.txt
123
qwe
456
rty
Зміст файлу input2.txt
Файл пустий
Зміст файлу output.txt
123
qwe
456
rty
```

Висновок

Я дослідив особливості створення і обробки текстових файлів даних. В ході роботи на лабораторною роботою було створено програму для знаходження спільних рядків у 2 файлах.