

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»

«Текстові файли»

Варіант 15

Виконав студент ІП-11 Книш Дмитро Олегович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 1

Текстові файли

Мета – вивчити особливості створення і обробки текстових файлів даних.

Індивідуальне завдання:

Створити текстовий файл. Переписати його компоненти до нового текстового файлу, вставляючи в кінець кожного рядка літеру, що є останньою в першому слові рядка. Вивести вміст вихідного та створеного файлів.

Математична модель:

Складемо таблицю імен змінних:

Змінна	Тип	Ім'я
Шлях вихідного файлу	const char *	original_file_path
Шлях створеного файлу	const char *	output_file_path
Об'єкт вихідного файлу	std::fstream	original_file
Об'єкт створеного файлу	std::fstream	output_file

Постановка задачі:

Створюємо текстовий файл файловим потоком `fstream` та заповнюємо його з консолі. Далі створюємо новий текстовий файл і проходимо стандартним **getline**'ом по рядках вихідного файлу, копіюючи (записуючи) їх у потік нового. В кінці кожного рядка додаємо останній символ першого слова. Виводимо у консоль вміст обох файлів.

Розв'язання:

Код алгоритму

Нижче наведені скріншоти коду на C++ / Python та результат виконання програми в терміналі (з метою економії місця коментарі були видалені):

:

main.cpp:

```
#include "../headers/dependencies.h"

#include "../headers/touch_file.h"
#include "../headers/fill_original_file.h"
#include "../headers/copy_original_file_data.h"
#include "../headers/display_files.h"

int main()
{
    const char *original_file_path = "../assets/original.txt";
    const char *output_file_path = "../assets/output.txt";

    std::fstream original_file = touch_file(original_file_path);
    fill_original_file(original_file);

    std::fstream output_file = touch_file(output_file_path);

    copy_original_file_data(original_file, output_file);

    display_files(original_file, output_file);

    original_file.close();
    output_file.close();

    std::cin.get();
    return 0;
}
```

dependencies.h:

```
#pragma once

#include <fstream>

std::fstream create_file(const char *);
void fill_original_file(std::fstream &);
void copy_original_file_data(std::fstream &, std::fstream &);
void display_files(std::fstream &, std::fstream &);
```

touch_file.h:

```
#pragma once

#include <iostream>
#include <fstream>

std::fstream touch_file(const char *filepath)
{
    std::fstream file(filepath, std::fstream::in | std::fstream::out |
std::fstream::trunc);

    if (!file.is_open())
    {
        std::cerr << "Could not open/read file " << filepath <<
std::endl;
        exit(1);
    }

    file.clear();

    return file;
}
```

fill_original_file.h:

```
#pragma once

#include <iostream>
#include <fstream>
#include <string>

void fill_original_file(std::fstream &original_file)
{
    std::string line;

    std::cout << "Enter text:" << std::endl;
    while (std::getline(std::cin, line))
    {
        original_file << line << std::endl;
    }

    original_file.clear();
}
```

copy_original_file_data.h:

```
#pragma once

#include <fstream>
#include <string>

void copy_original_file_data(std::fstream &original_file, std::fstream
&output_file)
{
    std::string line;

    original_file.seekg(0, std::ios::beg);

    while (std::getline(original_file, line) && !original_file.eof())
    {
        char last_letter = line.substr(0, line.find(" ")).back();
        output_file << line << last_letter << std::endl;
    }

    original_file.clear();
    output_file.clear();
}
```

display_files.h:

```
#pragma once

#include <iostream>
#include <fstream>

void display_files(std::fstream &original_file, std::fstream
&output_file)
{
    original_file.seekp(0, std::ios::beg);
    output_file.seekp(0, std::ios::beg);

    std::cout << "\n[ Original file ]\n"
        << original_file.rdbuf() << std::endl;
    std::cout << "[ Output file ]\n"
        << output_file.rdbuf() << std::endl;
}
```

Код на Python:

main.py:

```
from os import system

from modules.fill_original_file import fill_original_file
from modules.touch_file import touch_file
from modules.copy_original_file_data import copy_original_file_data
from modules.display_files import display_files

def main():
    original_file_path = "../../../assets/original.txt"
    output_file_path = "../../../assets/output.txt"

    original_file = fill_original_file(touch_file(original_file_path,
"w+"))

    output_file = touch_file(output_file_path, "w+")

    copy_original_file_data(original_file, output_file)

    display_files(original_file, output_file)

    output_file.close()
    original_file.close()

    system("pause")

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

touch_file.py:

```
import os.path
from io import TextIOWrapper

def touch_file(filepath: str, mode: str) -> TextIOWrapper:
    try:
        f = open(os.path.dirname(__file__) + filepath, mode)
        return f

    except (OSError, IOError) as e:
        print(e)
        exit(1)
```

fill_original_file.py:

```
from io import TextIOWrapper

def fill_original_file(file: TextIOWrapper) -> TextIOWrapper:
    print("Enter the text:")
    while True:
        try:
            line = input()
        except EOFError:
            break

        file.write(line)
        file.write("\n")

    return file
```

copy_original_file_data.py:

```
from io import TextIOWrapper

def copy_original_file_data(original_file: TextIOWrapper, output_file:
TextIOWrapper):
    original_file.seek(0)

    for line in original_file:
        last_char = line.split(" ", 1)[0][-1]
        output_file.write(f"{line[:-1]}{last_char}\n")
```

display_files.py:

```
from io import TextIOWrapper

def display_files(original_file: TextIOWrapper, output_file:
TextIOWrapper):
    original_file.seek(0)
    output_file.seek(0)

    print(f"\n[ Original file ]\n{original_file.read()}")
    print(f"\n[ Output file ]\n{output_file.read()}")
```

Випробування алгоритму:

```
→ build git:(main) x ./out.exe
Enter text:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...
^Z


[ Original file ]
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...

[ New (output) file ]
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod temporm
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,t
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.s
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...s
```

 original.txt – Блокнот

Файл Правка Формат Вид Справка

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...

 output.txt – Блокнот

Файл Правка Формат Вид Справка

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod temporm
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,t
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.s
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...s

Висновок:

Під час лабораторної роботи я набув практичних навичок у створенні та обробці текстових файлів даних. У результаті виконання лабораторної роботи я програмним шляхом створив та заповнив файл, скопіював його вміст у новий файл, додатково модифікуючи його.