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 № 2 з дисципліни

«Основи програмування-2.

Методології програмування»

«Файли даних. Бінарні файли»

Варіант 31

Виконав студент ІП-11 Трикош Іван Володимирович

Перевірила Вітковська Ірина Іванівна

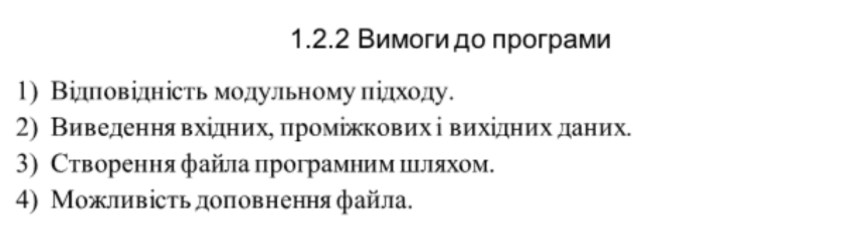
Київ 2022

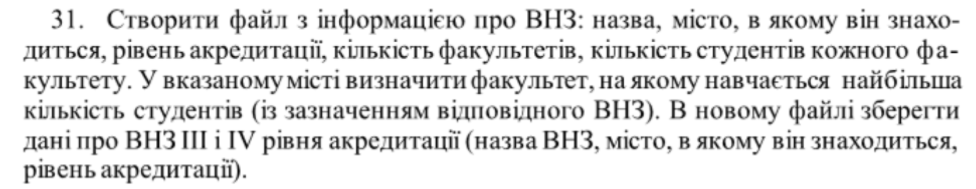
**Лабораторна робота №2**

**Файли даних. Бінарні файли**

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

Варіант 31.





**Постановка задачі.** Просимо користувача ввести інформацію про ВНЗ, для завершення вводу треба ввести “end”. Записуємо її у перший файл. Далі простимо ввести місто, і в цьому місті шукаємо факультет, на якому навчається найбільша кількість студентів. Виводимо отримані дані в консоль. Далі знаходимо університети з III та IV рівнями акредитації, виводимо їх в консоль та записуємо у нових файл.

**Код та результат програми на С++:**

**Func.h**

#pragma once

#include <iostream>

#include <Windows.h>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

struct Faculty // Факультет

{

string nameF; // Назва факультету

int amount; // кількість студентів факультету

};

struct University // Університет

{

string name; // Назва університету

string city; // Місто

string level; // Рівень акредитації

int amountF; // Кількість факультетів

vector<Faculty> amountf; // Факультети

};

void inputfile1(string&);

void outputfile(string&);

void universities(vector<University>&, string&);

void findfaculty(vector<University>&);

void inputfile2(vector<University>&, string&);

**Laba-C++.cpp**

#include "Func.h"

int main()

{

SetConsoleCP(1251); // Для вводу кирилиці

SetConsoleOutputCP(1251); // Для виводу кирилиці

vector<University> Universities; // Список університетів

// Шлях до файлів та їх назва

string namefile1 = "C:/Ivan/OP-Laba2/firstfile.dat";

string namefile2 = "C:/Ivan/OP-Laba2/secondfile.dat";

inputfile1(namefile1); // Вводимо список університетів

outputfile(namefile1); // Виводимо перший файл в консоль

universities(Universities, namefile1); // Зчитуємо університети з файлу

findfaculty(Universities); // Знаходимо факультет з найб. кількістю студентів в заданому місті

inputfile2(Universities, namefile2); // Записуємо інформацію в другий файл

outputfile(namefile2); // Виводимо другий файл в консоль

return 0;

}

**Func.cpp**

#include "Func.h"

void inputfile1(string& namefile1)

{

cout << "Input universities to the file " << namefile1 << " . Press \"Ctrl\" + \"Q\" to exit" << endl;

cout << "Example: Київський політехнічний інститут, Київ, IV, 2, ФІОТ - 1500, ІАТ - 500" << endl;

cout << "You wanna delete the previous contents of the file " << namefile1 << "?(Y/N):";

char choice; // Символ, що ввів користувач

cin >> choice;

ofstream outfile;

if (choice == 'Y' || choice == 'y')

{

outfile.open(namefile1, ios::binary); // Вікриваємо файл, попередньо очистивши його

}

else

{

outfile.open(namefile1, ios::binary | ios::app); // Відкриваємо файл на дозапис

}

string value; // Поточний рядок

bool flag = true; // Для виходу із циклу

while (flag)

{

getline(cin, value);

for (int i = 0; i < value.size(); i++) // Перевіряємо, чи є в рядку символ з кодом 17 ("Ctrl" + "Q")

{

if (value[i] == 17)

{

flag = false; // Для виходу із циклу

value.erase(i); // Видаляємо символ "Ctrl" + "Q" і все, що іде за ним

}

}

if (!value.empty()) // Перевіряємо, чи не пустий рядок

{

// Видаляємо пробіли на початку та в кінці рядка

while (value[0] == ' ')

{

value.erase(0, 1);

}

while (value[value.size() - 1] == ' ')

{

value.erase(value.size() - 1, 1);

}

value += "\n";

for (int i = 0; i < value.size(); i++)

{

outfile.write((char\*)(&value[i]), sizeof(value[i])); // Посимвольно записуємо інформацію у файл

}

}

}

outfile.close();

}

void outputfile(string& namefile)

{

cout << "\nFile " << namefile << endl;

ifstream file(namefile, ios::binary); // Відкриваємо файл як бінарний

char s; // Поточний символ

do

{

string value; // Поточний рядок

int i = 0; // Лічильник циклу

do

{

file.read((char\*)&s, sizeof(s)); // Зчитуємо символ

value += s;

i++;

} while (value[i - 1] != '\n'); // Зчитуємо, поки не буде наступного рядка

if (value != "")

{

cout << value; // Виводимо рядок в консоль

}

} while (!file.eof()); // Зчитуємо до кінця файлу

file.close();

}

void universities(vector<University>& universities, string& namefile1)

{

ifstream file1(namefile1, ios::binary); // Відкриваємо для читання файл

do

{

int i = 0; // Лічильник циклу

char s; // Поточний символ

string value; // Рядок з даними про університет

do

{

file1.read((char\*)&s, sizeof(s)); // Посимвольно зчитуємо інформацію

value += s;

i++;

} while (value[i - 1] != '\n'); // Зчитуємо до кінця рядка

value.erase(value.size() - 1); // Видаляємо останній символ "\n"

int pos = 0; // Позиція першої коми

int counter = 0; // Номер інформації

University univer; // Університет

while (value != "")

{

pos = value.find(",", 0); // Знаходимо позицію першої коми

switch (counter)

{

case 0: univer.name = value.substr(0, pos); // Знаходимо назву університету

break;

case 1: univer.city = value.substr(0, pos); // Знаходимо місто

break;

case 2: univer.level = value.substr(0, pos); // Знаходимо акредитацію

break;

case 3: univer.amountF = stoi(value.substr(0, pos)); // Знаходимо кількість факультетів

break;

}

value.erase(0, pos + 2); // Виаляємо поточну інформацію

counter++; // Переходимо до наступних даних

if (counter == 4) // Якщо дійшли до кількості факультетів, то зчитуємо вже їх

{

int position; // Позиція коми та тире

for (int i = 0; i < univer.amountF; i++)

{

Faculty faculty; // Факультет

position = value.find("-", 0); // Позиція першого тире

faculty.nameF = value.substr(0, position - 1); // Назва факультету

value.erase(0, position + 2); // Видалення назви з рядка

position = value.find(",", 0); // Знаходимо першу кому

faculty.amount = stoi(value.substr(0, position)); // Кількість студентів

if (i != univer.amountF - 1) // Перевіряємо, чи це останній факультет

{

value.erase(0, position + 2); // Видаляємо кількість студентів

}

else

{

value.erase(0); // Видаляємо рядок

}

univer.amountf.push\_back(faculty); // Додаємо факультет в факультети університету

}

}

}

universities.push\_back(univer); // Додаємо у список університетів інформацію про університет

} while (!file1.eof());

file1.close();

}

void findfaculty(vector<University>& list)

{

string town; // Введене місто

cout << "\nInput city: ";

getline(cin, town);

// Видаляємо пробіли на початку на в кінці рядка

while (town[0] == ' ')

{

town.erase(0, 1);

}

while (town[town.size() - 1] == ' ')

{

town.erase(town.size() - 1, 1);

}

string facultet = "error"; // Факультет

string univer = "error"; // Назва університету

int maxamount = 0; // Максимальна кількість студентів

for (int i = 0; i < list.size(); i++)

{

if (town == list[i].city) // Перевіряємо, чи поточне місто є шуканим

{

for (int j = 0; j < list[i].amountF; j++)

{

if (maxamount < list[i].amountf[j].amount) // Перевіряємо кількість студентів на кожному факультеті

{

maxamount = list[i].amountf[j].amount; // Максимальна кількість студентів

univer = list[i].name; // Назва університету

facultet = list[i].amountf[j].nameF; // Назва факультету

}

}

}

}

cout << "Maxamount:\n" << univer << ", " << facultet << " - " << maxamount << endl;

}

void inputfile2(vector<University>& list, string& namefile2)

{

cout << "\nYou wanna delete the previous contents of the file " << namefile2 << "?(Y/N):";

char choice; // Символ, що ввів користувач

cin >> choice;

ofstream outfile;

if (choice == 'Y' || choice == 'y')

{

outfile.open(namefile2, ios::binary); // Вікриваємо файл, поперредньо очистивши його

}

else

{

outfile.open(namefile2, ios::binary | ios::app); // Відкриваємо файл на дозапис

}

for (int i = 0; i < list.size(); i++)

{

if (list[i].level == "III" || list[i].level == "IV") // Перевіряємо рівень акредитації

{

string value = list[i].name + ", " + list[i].city + ", " + list[i].level + "\n"; // Записуємо інформацію у файл

for (int j = 0; j < value.size(); j++)

{

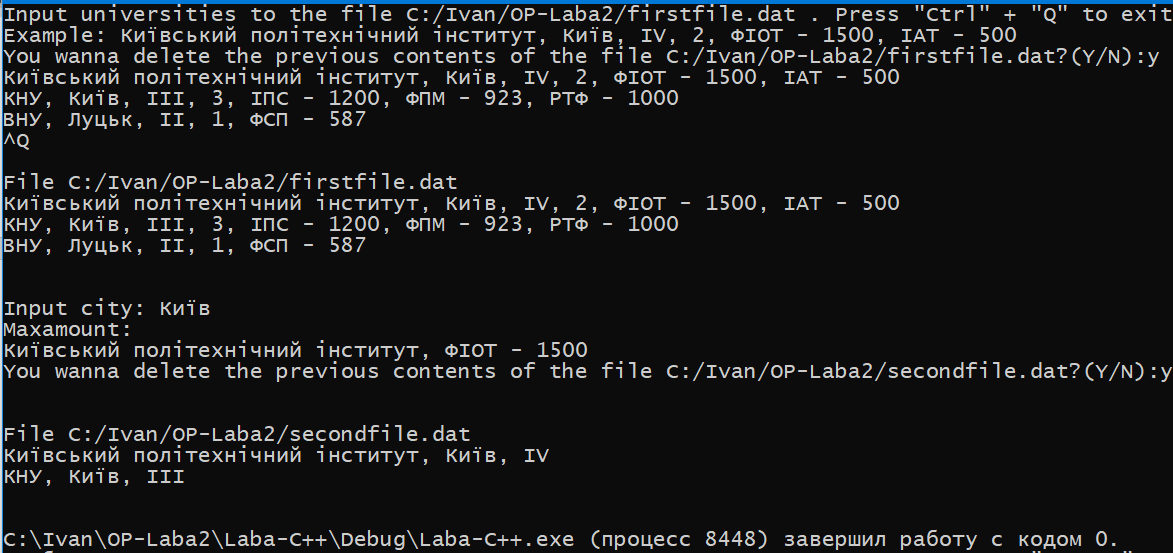
outfile.write((char\*)(&value[j]), sizeof(value[j])); // Записуємо посивольно рядок у файл

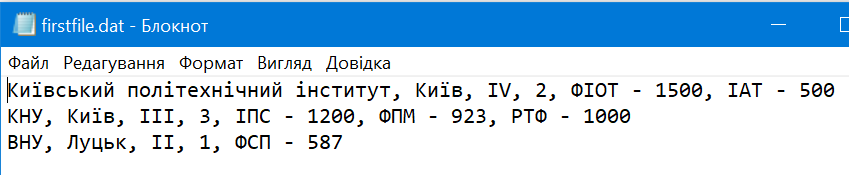
}

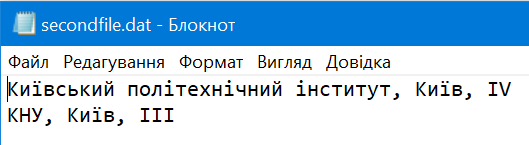
}

}

outfile.close(); // Закриваємо файл

} ****



****

**Код та результат програми на Python:**

**func.py**

def inputfile1(namefile1):

print("Input universities to the file", namefile1, ". Press \"Ctrl\" + \"Q\" to exit")

print("Example: Київський політехнічний інститут, Київ, IV, 2, ФІОТ - 1500, ІАТ - 500")

choice = input("You wanna delete the previous contents of the file " + namefile1 + " ? (Y / N): ")

if (choice == 'Y' or choice == 'y'):

outfile = open(namefile1, "wb") # Вікриваємо файл, попередньо очистивши його

else:

outfile = open(namefile1, "ab") # Відкриваємо файл на дозапис

flag = True # Для виходу із циклу

while (flag):

value = input()

i = 0

while(i < len(value)): # Перевіряємо, чи є в рядку символ з кодом 17 ("Ctrl" + "Q")

if (value[i] == chr(17)):

flag = False # Для виходу із циклу

value = value[:i:] # Видаляємо символ "Ctrl" + "Q" і все, що іде за ним

i += 1

if (not(len(value) == 0)): # Перевіряємо, чи не пустий рядок

# Видаляємо пробіли на початку та в кінці рядка

value.lstrip()

value.rstrip()

value += "\n"

# Записуємо рядок у файл

outfile.write(value.encode())

outfile.close()

def outputfile(namefile):

print("\nFile", namefile)

file = open(namefile, "rb") # Відкриваємо файл як бінарний

for value in file:

print(value.decode(encoding = 'UTF-8'), end = "")

print()

file.close()

def universities(namefile1):

file = open(namefile1, "rb") # Відкриваємо файл як бінарний

l = []

number = 0

for value in file:

if (len(value) > 0):

value = value.decode(encoding = 'UTF-8')

l += [value.split(", ")] # Розбиваємо рядок на список

for i in range(int(l[number][3])): # Розбиваємо елементи списку на факультети та кількість студентів

faculty = l[number][4 + i]

l[number][4 + i] = faculty.split(" - ")

number += 1 # Збільшуємо порядковий номер факультету

file.close()

return l

def findfaculty(l):

# Вводимо місто та видаляємо пробіли на початку і в кінці рядка

city = input("\nInput city: ")

city.lstrip()

city.rstrip()

facultet = "error"

university = "error"

maxnumber = 0

# Знаходимо максимальну кількість студентів у заданому місті

for i in range(len(l)):

if l[i][1] == city:

for j in range(int(l[i][3])):

if maxnumber < int(l[i][4 + j][1]):

maxnumber = int(l[i][4 + j][1])

university = l[i][0]

facultet = l[i][4 + j][0]

print("Maxamount:\n", university, ", ", facultet, " - ", maxnumber, sep = "")

def inputfile2(l, namefile2):

choice = input("\nYou wanna delete the previous contents of the file " + namefile2 + " ? (Y / N): ")

if (choice == 'Y' or choice == 'y'):

outfile = open(namefile2, "wb") # Вікриваємо файл, попередньо очистивши його

else:

outfile = open(namefile2, "ab") # Відкриваємо файл на дозапис

# Записуємо інформацію про університети з III та IV рівнем акредитації у файл та виводимо її в консоль

for i in range(len(l)):

if l[i][2] == "III" or l[i][2] == "IV":

value = l[i][0] + ", " + l[i][1] + ", " + l[i][2] + "\n"

outfile.write(value.encode())

outfile.close() # Закриваємо файл

**Laba\_Python.py**

import func

def main():

namefile1 = "C:/Ivan/OP-Laba2/firstfile.dat" # Шлях до першого файлу

namefile2 = "C:/Ivan/OP-Laba2/secondfile.dat" # Шлях до другого файлу

func.inputfile1(namefile1) # Вводимо університети

func.outputfile(namefile1) # Вводимо університети

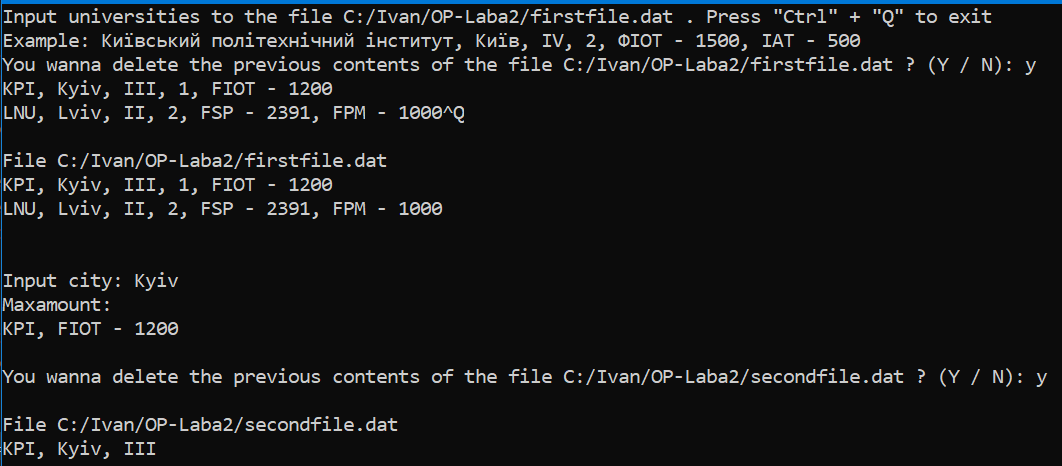
list\_universities = func.universities(namefile1) # Зчитуємо університети з файлу

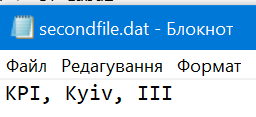
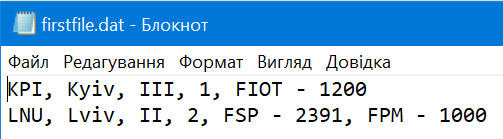
func.findfaculty(list\_universities) # Знаходимо факультет з найб. кількістю студентів в заданому місті

func.inputfile2(list\_universities, namefile2) # Записуємо інформацію в другий файл

func.outputfile(namefile2) # Виводимо другий файл в консоль

main()

****

****

**Висновок –** я вивчив особливості створення і обробки бінарних файлів даних, покращив свої навички у роботі з рядками та векторами та структурами. Я створив програму, яка введену інформацію про університети записує у файл, шукає в заданому місті факультет, в якому навчається найбільша кількість студентів, у новий файл записує університети, які мають III та IV рівень акредитації.