Звіт

Автор: Чугунов В.Ю

КІТ-119а

Лабораторна робота №3

РОБОТА З ПОТОКАМИ: ПОТОКОВЕ ВВЕДЕННЯ / ВИВЕДЕННЯ НА КОНСОЛЬ ТА У ФАЙЛ, РЯДКИ ТИПУ *STRING*, *STRINGSTREAM*

**Мета:** Отримати знання про основи роботи з потоковим введенням / виведенням на мові С++, роботу з файлами та рядками типу string.

1 ЗАВДАННЯ ДО РОБОТИ

**Загальне завдання.** Поширити попередню лабораторну роботу наступним чином:  
 використання функцій *printf* /*scanf* замінити на використання *cin*/*cout*;  
 усі конкатенації рядків замінити на використання *stringstream*;  
  замінити метод виводу інформації про об’єкт на метод, що повертає рядок-інформацію про об’єкт, який далі можна виводити на екран;

 замінити метод вводу інформації на метод, що приймає рядок з інформацією про об’єкт, обробляє його та створює об’єкт на базі цієї інформації;

поширити клас-список, шляхом реалізації методів роботи з файлами за допомогою файлових потоків.

2 РОЗРОБКА АЛГОРИТМІВ ФУНКЦІЙ

Змінений клас Lessons.  
 **private поля:**

string audienceID – ідентифікатор аудиторії

string lecturerName – ім’я лектора

string lessonType – назва предмета

int numOfWorks – кількість виконаних студентом робіт

int lessonStartTime – година початку занять

**Метод generateStudent ()**

Призначення: створення об’єкта класу (студента).

Зміни: функція відкриває файл, зчитує з нього інформацію та заносить її у об’єкт, закриває файл.

Схема алгоритму подана на рис. 1.

Змінений клас Works.

**Метод showAll ()**

Призначення: виведення інформації на екран.

Зміни: функція створює поток stringstream, заносить у нього інформацію, потім із цього потоку передає інформацію у рядок, який згодом повертає.

Схема алгоритму подана на рис. 2.

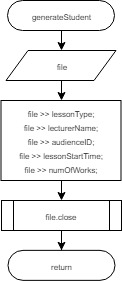


Рисунок 1 – Схема алгоритму функції generateStudent

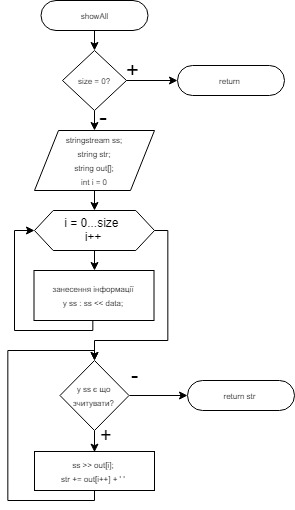


Рисунок 2 – Схема алгоритму функції showAll

3 ТЕКСТ ПРОГРАМИ

**Текст файлу Lessons.cpp**

#include <fstream>  
using std::ifstream;  
  
void Lessons::generateLesson(string filename)

{

filename = "input.txt";

ifstream file(filename);

file >> lessonType;

file >> lecturerName;

file >> audienceID;

file >> lessonStartTime;

file >> numOfWorks;

file.close();

}

**Текст файлу Works.cpp**

string Works::showAll()

{

if (size == 0)

cout << "\nError (the list is empty)\n";

else {

std::stringstream ss; //рядковий поток для занесення інформації

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

{

ss << "Info about lesson #" << i+1 << ": "

<< "Subject: " << lessons[i].get\_lessonType() << endl

<< "Lecturer: " << lessons[i].get\_lecturerName() << endl

<< "Audience ID: " << lessons[i].get\_audienceID() << endl

<< "Lesson start time: " << lessons[i].get\_lessonStartTime() << endl

<< "The number of works student wrote for the semester: " << lessons[i].get\_numOfWorks() << endl;

}

string str; // рядок, що буде повертати функція

string out[100]; // масив, у який буде занесена інформація із потоку

int i = 0;

while (ss)

{

ss >> out[i];

str += out[i++] + ' ';

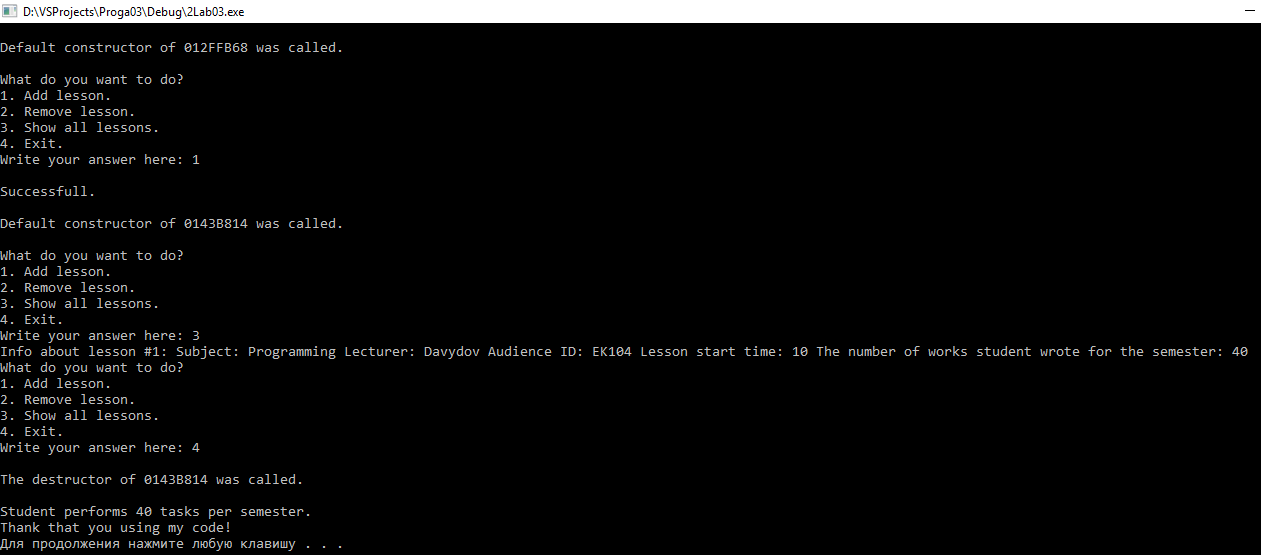
}

return str;

}

}

4 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

  
Рисунок 3 – Результат роботи програми

ВИСНОВКИ

При виконанні лабораторної роботи було перероблено попереднє завдання (Класи. Конструктори та деструктори) так, що:   
 1) метод виводу інформації про об’єкт замінений на метод, що повертає рядок-інформацію про об’єкт, який далі можна виводити на екран;

2) метод вводу інформації замінений на метод, що приймає рядок з інформацією про об’єкт, обробляє його та створює об’єкт на базі цієї інформації.