Звіт

Автор: Чугунов В.Ю

КІТ-119а

Лабораторна робота №4

РЕГУЛЯРНІ ВИРАЗИ

**Мета:** Отримати знання про базові регулярні вирази та досвід роботи по застосуванню їх на практиці.

1 ЗАВДАННЯ ДО РОБОТИ

**Загальне завдання.** Поширити попередню лабораторну роботу наступним чином:  
 - при введенні інформації про базовий клас, організувати перевірку відповідності наступним критеріям з використанням регулярних виразів:  
 - можна вводити тільки кириличні символи, латинські символи, цифри, пропуски, розділові знаки;  
 - не повинно бути пропусків та розділових знаків, які повторюються;  
 - перше слово не повинно починатися з маленького символу.

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 створює предмет, отримуючи дані з клавіатури, також використовує регулярні вирази ([A-Z])""(\\w)\*(\\s)?(\.)? для перевірки відповідності введених значень з умовами, зазначеними у завданні використовує функцію regex\_match. regex\_match порівнює вхідний рядок з умовами регулярного виразу.

Схема алгоритму подана на рис. 1.

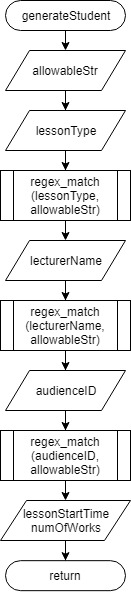


Рисунок 1 – Схема алгоритму функції generateStudent

3 ТЕКСТ ПРОГРАМИ

**Текст файлу Lessons.cpp**

#include <iostream>

#include <fstream>

#include <regex>

#include "Lessons.h"

using std::ifstream;

using std::regex;

using std::regex\_match;

using std::getline;

void Lessons::generateLesson()

{

regex allowableStr("([A-Z]|[А-Я])""([a-z]|[а-я])+[0-9]\*(\\w)?");

cout << "\nEnter the subject name: ";

cin.ignore(20, '\n');

do {

getline(cin, lessonType);

if (!(regex\_match(lessonType, allowableStr)))

cout << "Error (invalid syntax)!\nRepeat entering: ";

} while (!(regex\_match(lessonType, allowableStr)));

cout << "Enter lecturer name: ";

do {

getline(cin, lecturerName);

if (!(regex\_match(lecturerName, allowableStr)))

cout << "Error (invalid syntax)!\nRepeat entering: ";

} while (!(regex\_match(lecturerName, allowableStr)));

cout << "Enter an audience ID: ";

do {

getline(cin, audienceID);

if (!(regex\_match(audienceID, allowableStr)))

cout << "Error (invalid syntax)!\nRepeat entering: ";

} while (!(regex\_match(audienceID, allowableStr)));

cout << "Enter lesson start time: ";

cin >> lessonStartTime;

cout << "Enter a number of works student has completed per semester: ";

cin >> numOfWorks;

}

4 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

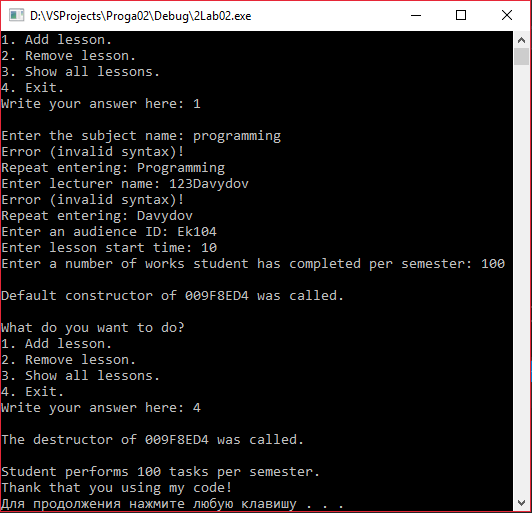


Рисунок 2 – Результат роботи програми

ВИСНОВКИ

При виконанні лабораторної роботи було перероблено попередню роботу так, що до методу створення об’єкта додано перевірку на коректність вводу, а саме: можна вводити тільки кириличні символи, латинські символи, цифри, пропуски, розділові знаки; не повинно бути пропусків та розділових знаків, які повторюються; перше слово не повинно починатися з маленького символу. Набуто практичних навичок щодо використання регулярних виразів.