ЗВІТ

ЛАБОРАТОРНА РОБОТА №1. Класи

**Тема.** Класи та специфікатори доступу. Інкапсуляція. Константи.

**Мета.** Отримати базові знання про класи. Дослідити механізм інкапсуляції.

1 ВИМОГИ

**1.1 Розробник**

- Котенко Сергій Миколайович;

- Студент групи КІТ 102.8(а);

- 24-012-2019р..

**1.2 Загальне завдання**

Для предметної галузі з табл. 1.2 розробити два класи:

− клас, що відображає сутність «базового класу». При цьому, в даному класі повинно бути мінімум три числових поля (бажано, щоб одне з цих полів було унікальним ідентифікатором об’єкту);

− клас, що має в собі динамічний масив об’єктів базового класу та має в собі методи додавання, видалення елементу, отримання елементу по індексу (або ідентифікатору), вивід усіх елементів на екран. Рекомендовані сигнатури методів:

− додавання: void CList::addPhone(Phone& phone);

− видалення: void CList::removePhone(int index);

− отримання по індексу: CPhone& CList::getPhone(int index);

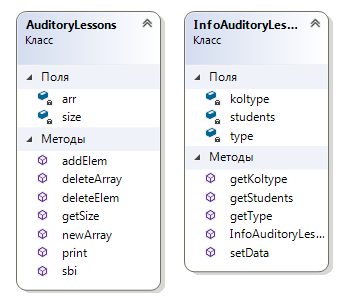
− вивід усіх елементів: void CList::showAll(); при цьому цей метод повинен викликати метод getPhone(index), щоб не було дублювання коду.

**1.3 Індивідуальне завдання**

В табл. 1.2 обрати прикладну галузь по варіанту у відповідності до номера у журналі групи. (10 | Роботи студентів | Аудиторні заняття)

2 ОПИС ПРОГРАМИ

**2.1 Опис логічної структури**



Діаграма класу AuditoryLessons Діаграма класу InfoAuditoryLessons

**2.2 Загальний вигляд коду**

InfoAuditoryLessons.h

#pragma once

#include <iostream>

#include <ctime>

class InfoAuditoryLessons {

private:

int type;

int koltype;

int students;

public:

InfoAuditoryLessons();

int getType();

int getKoltype();

int getStudents();

void setData(int x, int y, int z);

};

AuditoryLessons.h

#pragma once

#include "InfoAuditoryLessons.h"

class AuditoryLessons {

private:

int size;

InfoAuditoryLessons \*arr;

public:

void getSize(int s);

void newArray();

void print();

void addElem();

void deleteElem(int l);

void getByIndex(int index);

void deleteArray();

};

Main.cpp

/\*

\* @mainpage

\* <b> Лабораторна робота № 1. <br/> Класи. </b>

\* <br/><b><i >Мета роботи </i></b>:Отримати базові знання про класи. Дослідити механізм інкапсуляції.<br/>

\* <b>1 Завдання до роботи <br/> Загальне завдання <br/> </b>

\* Розробити програму рішення прикладної задачі мовою С++, використовуючи інтегроване середовище розробки

\* Visual Studio. Продемонструвати роботу програми. <br/>

\* <b>1.2 Основне завдання <br/> </b>

\* Реалізувати програму відповідно до індивідуального завдання. <br/>

\* <b><i> Індивідуальне завдання </i></b><br/>

\* Для предметної галузі з табл. 1.2 розробити два класи:

\* - клас, що відображає сутність «базового класу».

\* - клас, що має в собі динамічний масив об’єктів базового класу та має в собі методи додавання,

\* видалення елементу, отримання елементу по індексу (або ідентифікатору), вивід усіх елементів

\* на екран. <br/>

\*

\*

\* @author Сергій Котенко

\* @date 24.02.19

\* @version 2.0

\*/

#include "InfoAuditoryLessons.h"

#include "AuditoryLessons.h"

int main(void) {

system("color A");

AuditoryLessons Work;

int i = 0;

std::cout << "Enter size : " ;

std::cin >> i;

Work.getSize(i);

system("cls");

Work.newArray();

Work.print();

char option = 0;

do {

std::cout << "Choose option:" << std::endl << "1 - Exit " << std::endl << "2 - Add element" << std::endl << "3 - Delete element" << std::endl << "4 - Search by index" << std::endl;

std::cout << std::endl;

std::cin >> option ;

switch (option) {

case'1': {

system("cls");

Work.deleteArray();

exit(0);

}

case'2': {

std::cout << std::endl;

Work.addElem();

break;

}

case '3': {

int j = 0;

std::cout << std::endl << "Enter index by delete element : " ;

std::cin >> j;

std::cout << std::endl;

Work.deleteElem(j);

break;

}

case '4': {

int z = 0;

std::cout << std::endl << "Enter index : " ;

std::cin >> z;

Work.getByIndex(z);

break;

}

}

} while (option != 0);

\_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDERR);

\_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDERR);

\_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDERR);

\_CrtDumpMemoryLeaks();

}

InfoAuditoryLessons.cpp

#include "InfoAuditoryLessons.h"

InfoAuditoryLessons::InfoAuditoryLessons() {

type = rand() % 3;

koltype = rand() % 16 + 1;

students = rand() % 18;

}

int InfoAuditoryLessons :: getType() {

return InfoAuditoryLessons :: type;

}

int InfoAuditoryLessons :: getKoltype() {

return InfoAuditoryLessons :: koltype;

}

int InfoAuditoryLessons :: getStudents() {

return InfoAuditoryLessons :: students;

}

void InfoAuditoryLessons :: setData(int x, int y, int z) {

InfoAuditoryLessons :: type = x;

InfoAuditoryLessons :: koltype = y;

InfoAuditoryLessons :: students = z;

}

AuditoryLessons.cpp

#include "AuditoryLessons.h"

void AuditoryLessons :: getSize(int s) {

AuditoryLessons :: size = s;

}

void AuditoryLessons :: newArray() {

arr = new InfoAuditoryLessons[size];

}

void AuditoryLessons :: print() {

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

int type = arr[i].getType();

int koltype = arr[i].getKoltype();

int students = arr[i].getStudents();

if (type == 1) {

std::cout << " LectureLessons" << std::endl;

}

else if (type == 2) {

std::cout << " Workshops" << std::endl;

}

else {

std::cout << " LaboratoryWorks" << std::endl;

}

std::cout << "Number of classes held: " << koltype << std::endl;

std::cout << "Number of students (average): " << students << std::endl << std::endl ;

}

}

void AuditoryLessons :: addElem() {

int x, y, z;

std::cout << "Enter your type (1) - LectureLessons, (2) - Workshops, (3) - LaboratoryWorks : ";

std::cin >> x;

std::cout << "Enter number of classes held : ";

std::cin >> y;

std::cout << "Enter number of students (average) : ";

std::cin >> z;

std::cout << std::endl;

InfoAuditoryLessons \*mas = new InfoAuditoryLessons[size + 1];

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

mas[i] = AuditoryLessons::arr[i];

}

size++;

delete[] arr;

mas[size - 1].setData(x, y, z);

arr = new InfoAuditoryLessons[size];

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

AuditoryLessons::arr[i] = mas[i];

}

delete[] mas;

print();

}

void AuditoryLessons :: deleteElem(int l) {

InfoAuditoryLessons\* mas = new InfoAuditoryLessons[size];

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

mas[i] = AuditoryLessons::arr[i];

}

delete[] arr;

size--;

arr = new InfoAuditoryLessons[size];

int j = 0;

for (int i = 0; i < l; i++) {

AuditoryLessons::arr[i] = mas[j];

j++;

}

j++;

for (int i = l; i < size; i++) {

AuditoryLessons::arr[i] = mas[j];

j++;

}

delete[] mas;

print();

}

void AuditoryLessons :: getByIndex(int index) {

if (index >= size) {

std::cout << std::endl << "Error" << std::endl << std::endl;

return;

}

int type = arr[index].getType();

int koltype = arr[index].getKoltype();

int students = arr[index].getStudents();

if (type == 1) {

std::cout << std::endl << " LectureLessons" << std::endl; // Лекция

}

else if (type == 2) {

std::cout << std::endl << " Workshops" << std::endl; // Практика

}

else {

std::cout << std::endl << " LaboratoryWorks" << std::endl; // Лабы

}

std::cout << "Number of classes held: " << koltype << std::endl;

std::cout << "Number of students (average): " << students << std::endl << std::endl;

}

void AuditoryLessons::deleteArray() {

delete[] AuditoryLessons :: arr;

}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

**3.1 Опис поведінки програми**

Програма працює наступнім чином:

1) Ввід користувачем кількості вивідних даних, створення масиву даних та виведення на екран

2) Вивід на екран можливих опцій програми , обирання користувачем опції:

2.1) Вихід з програми

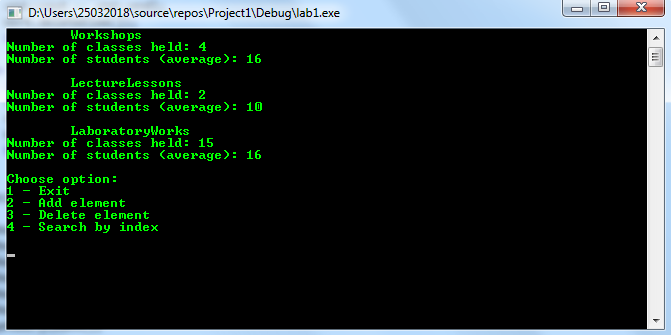
2.2) Додавання нового елементу

2.3) Видалення певного елементу

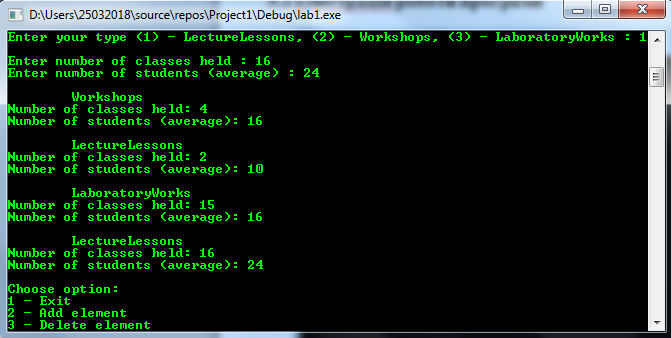
2.4) Пошук за індексом

3) Перевірка на витоки пам’яті

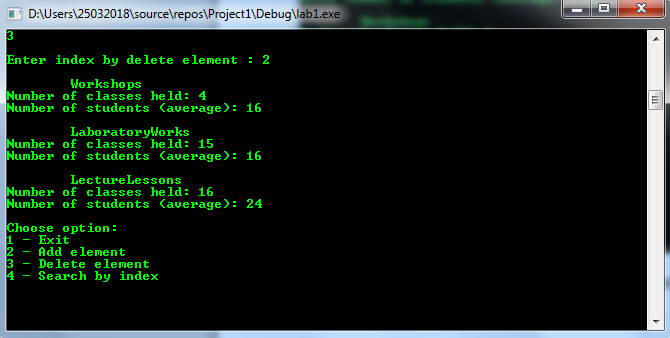
**3.2 Ілюстрація роботи програми**



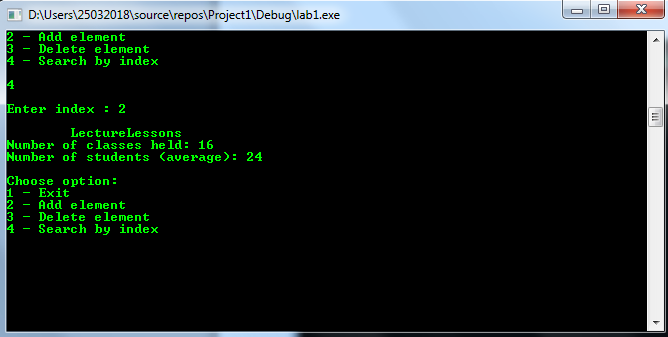
**Рисунок 3.1 - Створені дані та можливі опції роботи з програмою**



**Рисунок 3.2 - Додавання елементу**



**Рисунок 3.3 - Видалення елементу**



**Рисунок 3.4 - Пошук за індексом**

**ВИСНОВОК**

В інтегрованому середовищі *Visual Studio* розроблена програма мовою С++. Виконання програми дозволяє продемонструвати коректність роботи програм для створення класів та їх використання.