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);

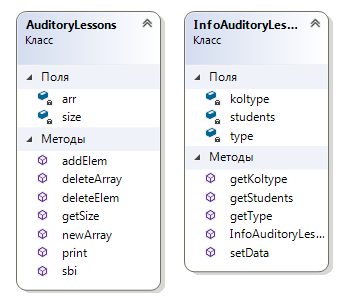
− вивід усіх елементів: void CList::showAll(); при цьому цей метод повинен викликати метод getPhone(index), щоб не було дублювання коду.

**1.3 Індивідуальне завдання**

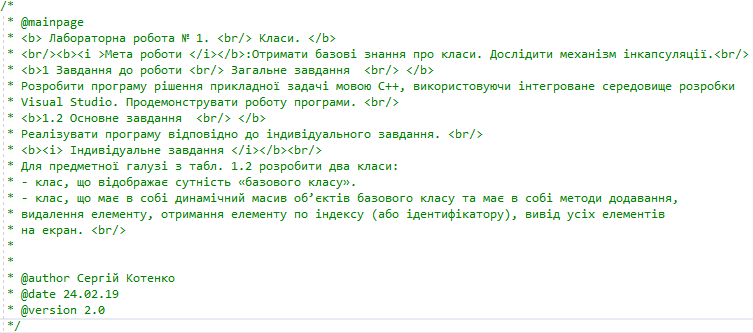
В табл. 1.2 обрати прикладну галузь по варіанту у відповідності до номера у журналі групи. (10 | Роботи студентів | Аудиторні заняття)

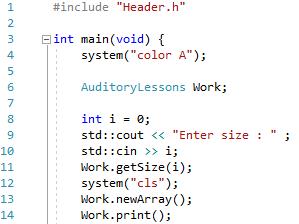
2 ОПИС ПРОГРАМИ

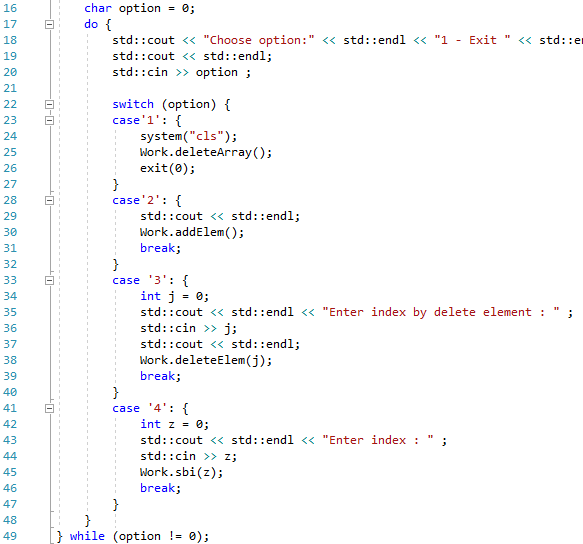
**2.1 Алгоритм програми**

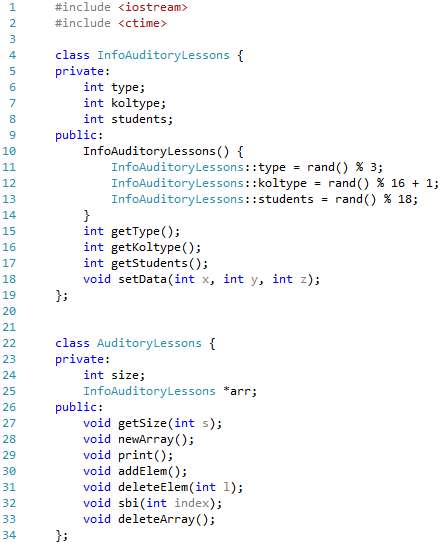


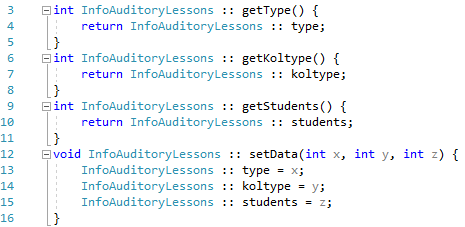
**2.2 Загальний вигляд коду**

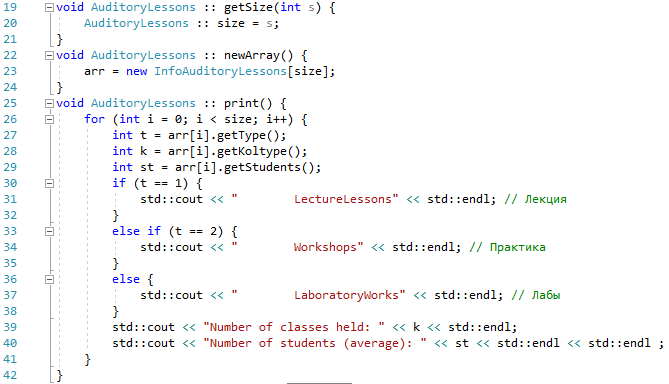


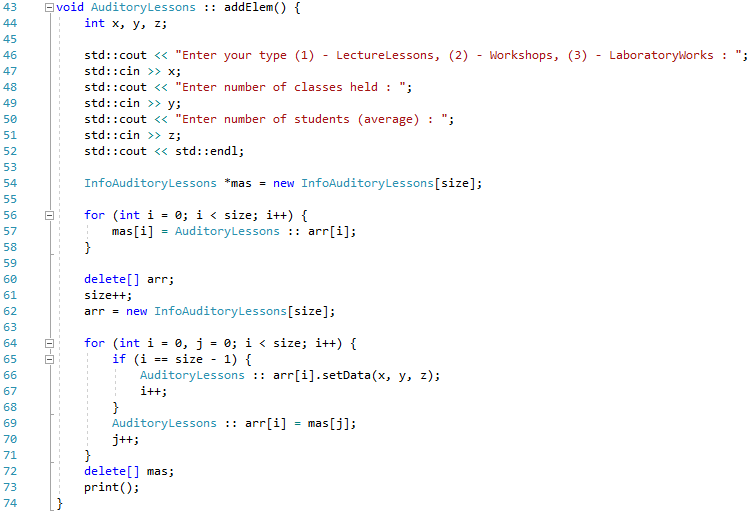


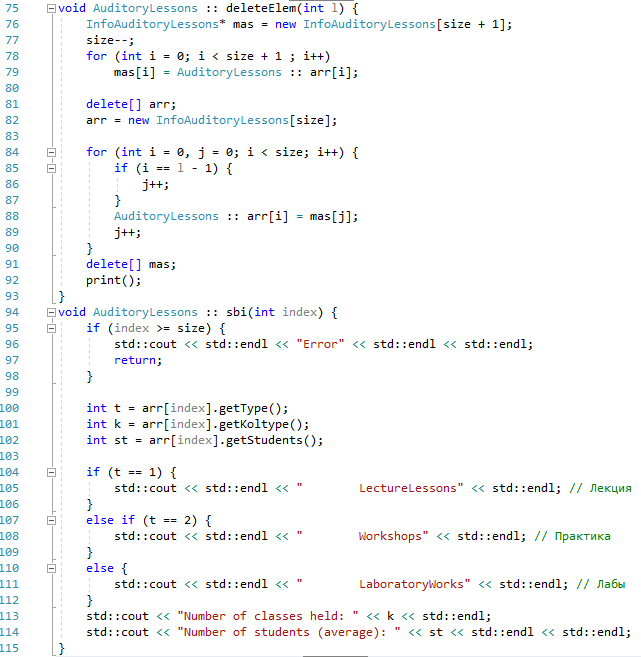












**3 ВАРІАНТИ ВИКОРИСТАННЯ**

**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

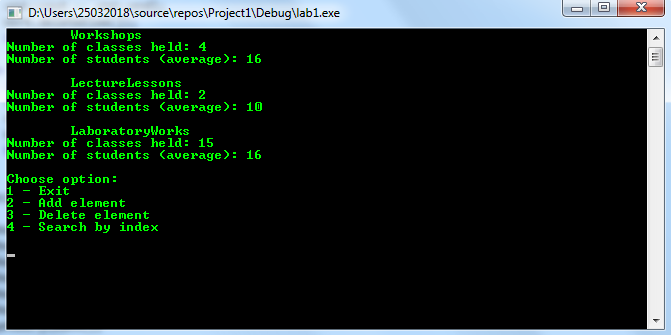
2.2) Додавання нового елементу

2.3) Видалення певного елементу

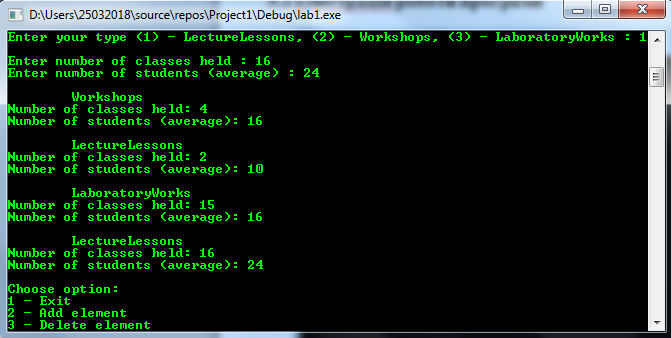
2.4) Пошук за індексом

3) Перевірка на витоки пам’яті

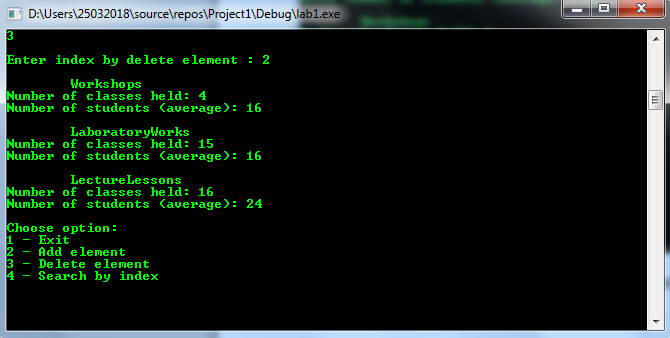
**3.2 Ілюстрація роботи програми**



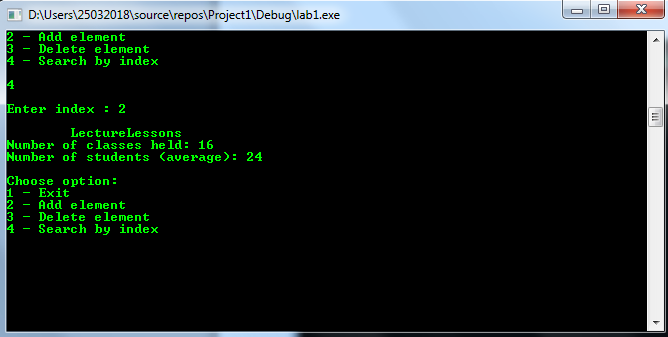
* **Створені дані та можливі опції роботи з програмою**



* **Додавання елементу**



* **Видалення елементу**



* **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.

Header.h

#include <iostream>

#include <ctime>

class InfoAuditoryLessons {

private:

int type;

int koltype;

int students;

public:

InfoAuditoryLessons() {

InfoAuditoryLessons::type = rand() % 3;

InfoAuditoryLessons::koltype = rand() % 16 + 1;

InfoAuditoryLessons::students = rand() % 18;

}

int getType();

int getKoltype();

int getStudents();

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

};

class AuditoryLessons {

private:

int size;

InfoAuditoryLessons \*arr;

public:

void getSize(int s);

void newArray();

void print();

void addElem();

void deleteElem(int l);

void sbi(int index);

void deleteArray();

};

Main.cpp

#include "Header.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.sbi(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();

}

Function.cpp

#include "Header.h"

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;

}

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 t = arr[i].getType();

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

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

if (t == 1) {

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

}

else if (t == 2) {

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

}

else {

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

}

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

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

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

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

}

delete[] arr;

size++;

arr = new InfoAuditoryLessons[size];

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

if (i == size - 1) {

AuditoryLessons :: arr[i].setData(x, y, z);

i++;

}

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

j++;

}

delete[] mas;

print();

}

void AuditoryLessons :: deleteElem(int l) {

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

size--;

for (int i = 0; i < size + 1 ; i++)

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

delete[] arr;

arr = new InfoAuditoryLessons[size];

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

if (i == l - 1) {

j++;

}

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

j++;

}

delete[] mas;

print();

}

void AuditoryLessons :: sbi(int index) {

if (index >= size) {

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

return;

}

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

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

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

if (t == 1) {

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

}

else if (t == 2) {

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

}

else {

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

}

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

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

}

void AuditoryLessons::deleteArray() {

delete[] AuditoryLessons :: arr;

}