

## 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:

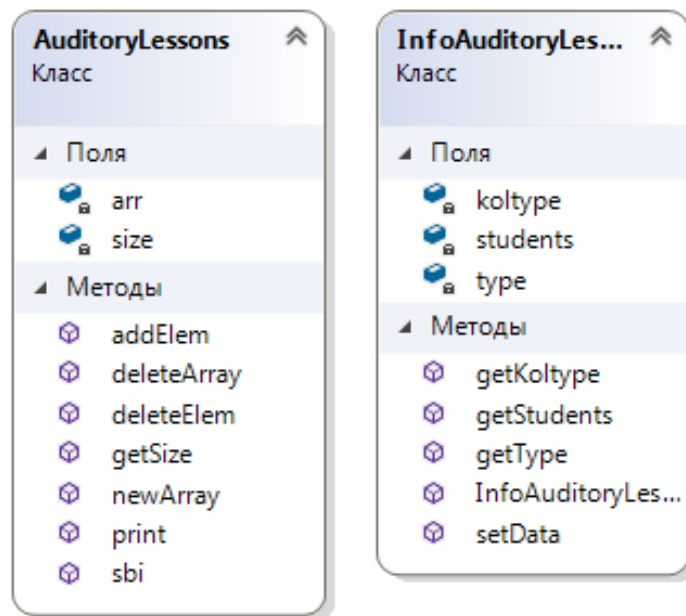
- клас, що відображає сутність «базового класу». При цьому, в даному класі повинно бути мінімум три числових поля (бажано, щоб одне з цих полів було унікальним ідентифікатором об'єкту);
- клас, що має в собі динамічний масив об'єктів базового класу та має в собі методи додавання, видалення елементу, 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

```
/*
 * @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 C++, 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 елемента, отримання елемента по індексу (або ідентифікатору), вивід усіх елементів
 * на екран. <br/>
 *
 *
 * @author Сергій Котенко
 * @date 24.02.19
 * @version 2.0
 */
```

```
1  #include "Header.h"
2
3  int main(void) {
4      system("color A");
5
6      AuditoryLessons Work;
7
8      int i = 0;
9      std::cout << "Enter size : " ;
10     std::cin >> i;
11     Work.getSize(i);
12     system("cls");
13     Work.newArray();
14     Work.print();
```

```

16 char option = 0;
17 do {
18     std::cout << "Choose option:" << std::endl << "1 - Exit " << std::endl;
19     std::cout << std::endl;
20     std::cin >> option ;
21
22     switch (option) {
23     case '1': {
24         system("cls");
25         Work.deleteArray();
26         exit(0);
27     }
28     case '2': {
29         std::cout << std::endl;
30         Work.addElem();
31         break;
32     }
33     case '3': {
34         int j = 0;
35         std::cout << std::endl << "Enter index by delete element : " ;
36         std::cin >> j;
37         std::cout << std::endl;
38         Work.deleteElem(j);
39         break;
40     }
41     case '4': {
42         int z = 0;
43         std::cout << std::endl << "Enter index : " ;
44         std::cin >> z;
45         Work.sbi(z);
46         break;
47     }
48     }
49 } while (option != 0);

```

```

1  #include <iostream>
2  #include <ctime>
3
4  class InfoAuditoryLessons {
5  private:
6      int type;
7      int koltype;
8      int students;
9  public:
10     InfoAuditoryLessons() {
11         InfoAuditoryLessons::type = rand() % 3;
12         InfoAuditoryLessons::koltype = rand() % 16 + 1;
13         InfoAuditoryLessons::students = rand() % 18;
14     }
15     int getType();
16     int getKoltype();
17     int getStudents();
18     void setData(int x, int y, int z);
19 };
20
21
22 class AuditoryLessons {
23 private:
24     int size;
25     InfoAuditoryLessons *arr;
26 public:
27     void getSize(int s);
28     void newArray();
29     void print();
30     void addElem();
31     void deleteElem(int l);
32     void sbi(int index);
33     void deleteArray();
34 };

```

```

3   int InfoAuditoryLessons :: getType() {
4       return InfoAuditoryLessons :: type;
5   }
6   int InfoAuditoryLessons :: getKoltype() {
7       return InfoAuditoryLessons :: koltype;
8   }
9   int InfoAuditoryLessons :: getStudents() {
10      return InfoAuditoryLessons :: students;
11  }
12  void InfoAuditoryLessons :: setData(int x, int y, int z) {
13      InfoAuditoryLessons :: type = x;
14      InfoAuditoryLessons :: koltype = y;
15      InfoAuditoryLessons :: students = z;
16  }

```

```

19 void AuditoryLessons :: getSize(int s) {
20     AuditoryLessons :: size = s;
21 }
22 void AuditoryLessons :: newArray() {
23     arr = new InfoAuditoryLessons[size];
24 }
25 void AuditoryLessons :: print() {
26     for (int i = 0; i < size; i++) {
27         int t = arr[i].getType();
28         int k = arr[i].getKoltype();
29         int st = arr[i].getStudents();
30         if (t == 1) {
31             std::cout << "          LectureLessons" << std::endl; // Лекция
32         }
33         else if (t == 2) {
34             std::cout << "          Workshops" << std::endl; // Практика
35         }
36         else {
37             std::cout << "          LaboratoryWorks" << std::endl; // Лабы
38         }
39         std::cout << "Number of classes held: " << k << std::endl;
40         std::cout << "Number of students (average): " << st << std::endl << std::endl ;
41     }
42 }

```

```

43 void AuditoryLessons :: addElem() {
44     int x, y, z;
45
46     std::cout << "Enter your type (1) - LectureLessons, (2) - Workshops, (3) - LaboratoryWorks : ";
47     std::cin >> x;
48     std::cout << "Enter number of classes held : ";
49     std::cin >> y;
50     std::cout << "Enter number of students (average) : ";
51     std::cin >> z;
52     std::cout << std::endl;
53
54     InfoAuditoryLessons *mas = new InfoAuditoryLessons[size];
55
56     for (int i = 0; i < size; i++) {
57         mas[i] = AuditoryLessons :: arr[i];
58     }
59
60     delete[] arr;
61     size++;
62     arr = new InfoAuditoryLessons[size];
63
64     for (int i = 0, j = 0; i < size; i++) {
65         if (i == size - 1) {
66             AuditoryLessons :: arr[i].setData(x, y, z);
67             i++;
68         }
69         AuditoryLessons :: arr[i] = mas[j];
70         j++;
71     }
72     delete[] mas;
73     print();
74 }

```

```

75 void AuditoryLessons :: deleteElem(int l) {
76     InfoAuditoryLessons* mas = new InfoAuditoryLessons[size + 1];
77     size--;
78     for (int i = 0; i < size + 1 ; i++)
79         mas[i] = AuditoryLessons :: arr[i];
80
81     delete[] arr;
82     arr = new InfoAuditoryLessons[size];
83
84     for (int i = 0, j = 0; i < size; i++) {
85         if (i == l - 1) {
86             j++;
87         }
88         AuditoryLessons :: arr[i] = mas[j];
89         j++;
90     }
91     delete[] mas;
92     print();
93 }
94 void AuditoryLessons :: sbi(int index) {
95     if (index >= size) {
96         std::cout << std::endl << "Error" << std::endl << std::endl;
97         return;
98     }
99
100     int t = arr[index].getType();
101     int k = arr[index].getKoltype();
102     int st = arr[index].getStudents();
103
104     if (t == 1) {
105         std::cout << std::endl << "          LectureLessons" << std::endl; // Лекция
106     }
107     else if (t == 2) {
108         std::cout << std::endl << "          Workshops" << std::endl; // Практика
109     }
110     else {
111         std::cout << std::endl << "          LaboratoryWorks" << std::endl; // Лабы
112     }
113     std::cout << "Number of classes held: " << k << std::endl;
114     std::cout << "Number of students (average): " << st << std::endl << std::endl;
115 }

```

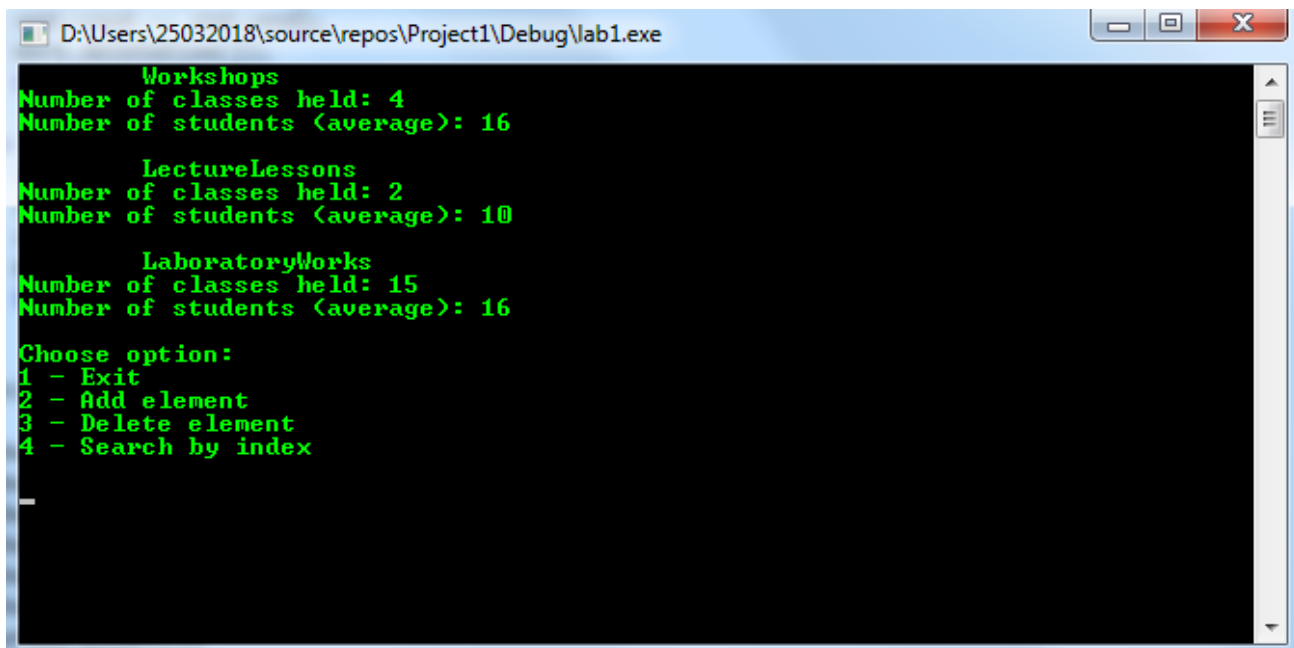
## 3 ВАРІАНТИ ВИКОРИСТАННЯ

### 3.1 Опис поведінки програми

Програма працює наступним 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



```
D:\Users\25032018\source\repos\Project1\Debug\lab1.exe

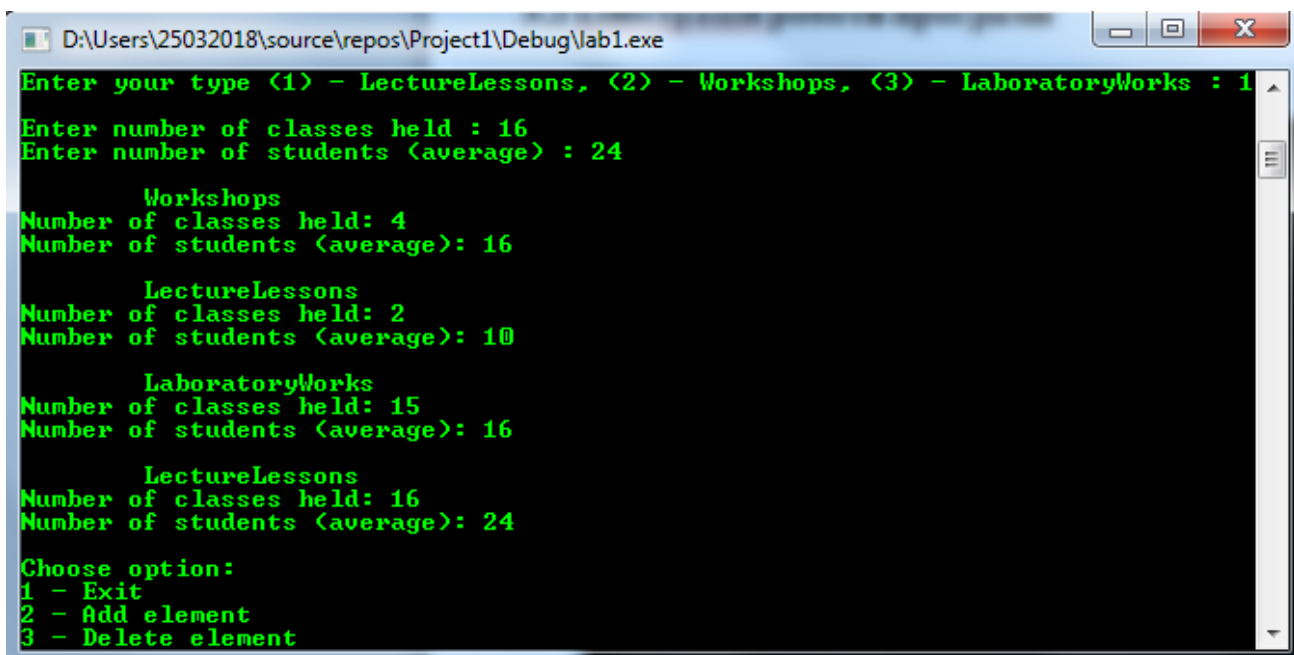
Workshops
Number of classes held: 4
Number of students (average): 16

LectureLessons
Number of classes held: 2
Number of students (average): 10

LaboratoryWorks
Number of classes held: 15
Number of students (average): 16

Choose option:
1 - Exit
2 - Add element
3 - Delete element
4 - Search by index
-
```

✓ Створені дані та можливі опції роботи з програмою



```
D:\Users\25032018\source\repos\Project1\Debug\lab1.exe

Enter your type <1> - LectureLessons, <2> - Workshops, <3> - LaboratoryWorks : 1
Enter number of classes held : 16
Enter number of students (average) : 24

Workshops
Number of classes held: 4
Number of students (average): 16

LectureLessons
Number of classes held: 2
Number of students (average): 10

LaboratoryWorks
Number of classes held: 15
Number of students (average): 16

LectureLessons
Number of classes held: 16
Number of students (average): 24

Choose option:
1 - Exit
2 - Add element
3 - Delete element
```

✓ Додавання елементу

```
D:\Users\25032018\source\repos\Project1\Debug\lab1.exe

3
Enter index by delete element : 2

      Workshops
Number of classes held: 4
Number of students (average): 16

      LaboratoryWorks
Number of classes held: 15
Number of students (average): 16

      LectureLessons
Number of classes held: 16
Number of students (average): 24

Choose option:
1 - Exit
2 - Add element
3 - Delete element
4 - Search by index
```

✓ Видалення елементу

```
D:\Users\25032018\source\repos\Project1\Debug\lab1.exe

2 - Add element
3 - Delete element
4 - Search by index
4
Enter index : 2

      LectureLessons
Number of classes held: 16
Number of students (average): 24

Choose option:
1 - Exit
2 - Add element
3 - Delete element
4 - Search by index
```

✓ Пошук за індексом

## ВИСНОВОК

В інтегрованому середовищі *Visual Studio* розроблена програма мовою C++. 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();
            }
        }
    } while (option != '1');
```



```

        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;
}

```