

Лабораторна робота № 1

Класи та специфікатори доступу. Інкапсуляція. Константи.

Мета: Отримати базові знання про класи, конструктори та деструктори. Дослідити механізм створення та видалення об'єктів.

1 ВИМОГИ

1.1 Розробник Інформація про розробника:

Куйдин Михаил Андреевич

НТУ “ХП”,

КІТ 102.8а

1.2 Завдання

Поширити попередню лабораторну роботу наступним чином:

1. в базовому класі необхідно додати: ☐ мінімум одне поле типу `char*`;
2. конструктор за замовчуванням, копіювання та конструктор з аргументами;
3. деструктор;
4. в клас-список потрібно додати метод обходу масиву для виконання індивідуального завдання.

Приклад сигнатури такого методу:

```
CPhone& findCheapestPhone(float diagonal);
```

В наведеному прикладі реалізоване завдання пошуку самого дешевого телефону з заданою діагоналлю (повертається один телефон).

1.3 Індивідуальне завдання.

В табл. 2.1 оберіть завдання для обходу колекції по варіанту у відповідності до номера у журналі групи.

11	Школа	Визначити кількість навчальних закладів з кількістю працівників більше за середній показник
----	-------	---

2. Діаграма класів

На рис.1 зображені діаграми класів School і InfoStorge

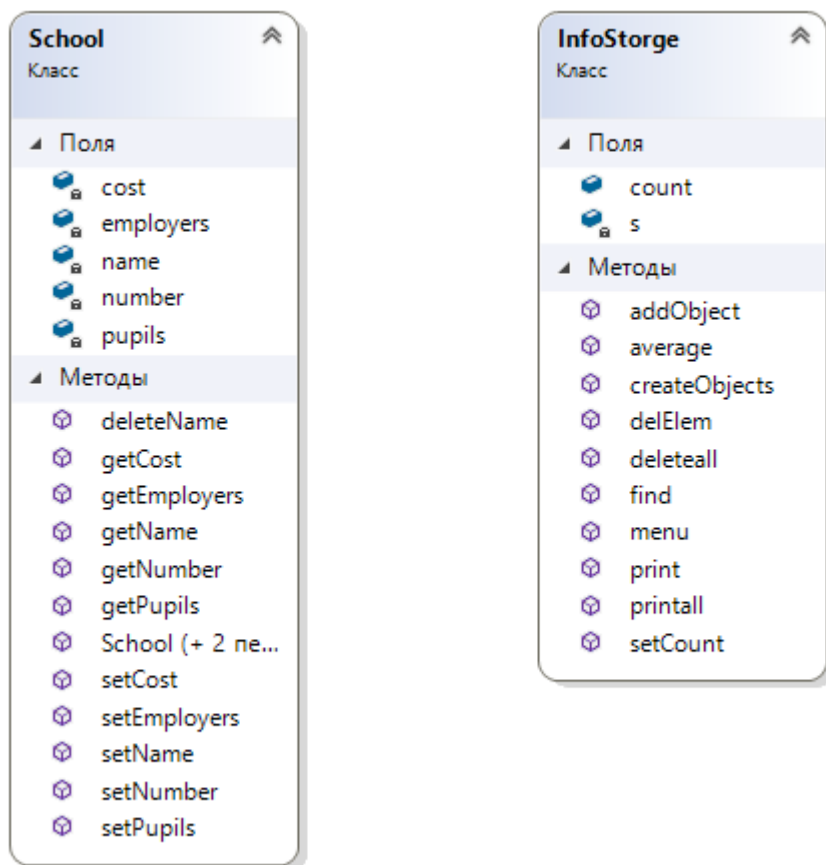


Рисунок 1

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма складається з 6 файлів:

1. menu.cpp
2. main.cpp
3. School.h
4. School.cpp
5. InfoStorge.h
6. InfoStorge.cpp

1.menu.cpp – файл у якому знаходиться функція меню , у якій описано меню програми. Код файлу зображено на рис.2

```

#include "School.h"
#include "InfoStorage.h"
void InfoStorage::menu() {
    InfoStorage tmp;
    int coun, num, cos, pup, emp, index;
    char p[30];
    cout << "Hi! How many School do u want to create:\n";
    cin >> coun;

    // поменять имена переменных, доделать метод createObjects
    tmp.setCount(coun);
    tmp.createObjects();
    int i = 0;
    int a = 100;
    while (a != 0) {

        cout << "Menu: \n 1 - Add new objects \n 2 - Delete object \n 3 - Find by index \n 4 - Print all schools \n 5 - Print schools with employers more than average \n 0 - End program \n Your choose: ";
        cin >> a;
        switch (a)
        {
            case 1:
                system("cls");

                cout << "In which position u want add new school? Last position:" << tmp.count << endl;
                cin >> index;

                while (tmp.count <= index || index < 1) {
                    cout << "You can't put in this position, choose another! Last position: " << tmp.count << endl;
                    cin >> index;
                }

                cout << "Enter cost per month: ";
                cin >> cos;

                cout << "Enter type (gymnasium, high, secondary, primary) if school: ";
                cin >> p;

                cout << "Enter number of school: ";
                cin >> num;

                cout << "Enter number of pupils: ";
                cin >> pup;

                cout << "Enter number of employers: ";
                cin >> emp;
                tmp.addObject((index-1), num, cos, pup, emp, p);
                break;

            case 2:
                system("cls");

                cout << "Enter index ";
                cin >> i;
                tmp.delElem(i);
                break;
            case 3:
                system("cls");

                cout << "Enter index ";
                cin >> i;
                tmp.find(i);
                break;
            case 4:
                system("cls");
                tmp.printall();
                break;
            case 5:
                system("cls");
                tmp.average();
                break;
            case 0:
                a = 0;
                break;
        }
    }
    tmp.deleteall();
}

```

Рисунок 2

2. main.cpp – файл, у якому знаходиться функція main. Код файлу зображено на рис.3

```

1  #include "School.h"
2  #include "InfoStorage.h"
3
4
5  int main() {
6      InfoStorage mp;
7
8      mp.menu();
9
10     _CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_FILE);
11     _CrtSetReportFile(_CRT_WARN, _CRTDBG_FILE_STDERR);
12     _CrtSetReportMode(_CRT_ERROR, _CRTDBG_MODE_FILE);
13     _CrtSetReportFile(_CRT_ERROR, _CRTDBG_FILE_STDERR);
14     _CrtSetReportMode(_CRT_ASSERT, _CRTDBG_MODE_FILE);
15     _CrtSetReportFile(_CRT_ASSERT, _CRTDBG_FILE_STDERR);
16
17     system("pause");
18
19     return _CrtDumpMemoryLeaks();
20
21 }

```

Рисунок 3

3. School.h – файл, у якому знаходиться клас School. Код зображено на рис.4

```
1  #pragma once
2  #pragma once
3  #define _CRT_SECURE_NO_WARNINGS
4  #include <iostream>
5  #include <string>
6
7  using std::cin;
8  using std::cout;
9  using std::endl;
10 using namespace std;
11
12 class School {
13 private:
14     unsigned int number;
15     unsigned int pupils;
16     unsigned int cost;
17     unsigned int employers;
18     char *name;
19 public:
20
21     School() {
22         number = 0;
23         pupils = 0;
24         cost = 0;
25         employers = 0;
26         strcpy(name, 0);
27     }
28
29
30     School(int num, int cos, int pup, int emp, char *p) {
31         number = num;
32         cost = cos;
33         pupils = pup;
34         employers = emp;
35         strcpy(name, p);
36     }
37     School(const School &obj){
38         number = obj.number;
39         cost = obj.cost;
40         pupils = obj.pupils;
41         employers = obj.employers;
42         strcpy(name, obj.name);
43     }
44
45     int getNumber();
46     int getCost();
47     int getPupils();
48     int getEmployers();
49     void getName(char *ptr);
50     void setCost(int c);
51     void setPupils(int a);
52     void setNumber(int n);
53     void setEmployers(int d);
54     void setName(char *p);
55     void deleteName();
56 };
57
58
```

Рисунок 4

4.School.cpp – файл , у якому описані методи класу School. Код файлу зображено на рис.5

```

#include "School.h"
#include "InfoStorge.h"

void School::setNumber(int n) {
    number = n;
}

void School::setPupils(int a) {
    pupils = a;
}

void School::setCost(int c) {
    cost = c;
}

void School::setEmployers(int d) {
    employers = d;
}

void InfoStorge::setCount(int c) {
    count = c;
}

int School::getNumber() {
    return number;
}

int School::getCost() {
    return cost;
}

int School::getPupils() {
    return pupils;
}

int School::getEmployers() {
    return employers;
}

void School::setName(char *p) {
    name = (char*) operator new (sizeof(char) * 256);
    strcpy_s(name, 30, p);
}

void School::getName(char *ptr) {
    strcpy(ptr, name);
}

void School::deleteName() {
    delete[] name;
}

```

Рисунок 5

5.InfoStorge.h – файл , у якому знаходиться клас InfoStorge. Код зображено на рис.6

```

#pragma once
#include "School.h"
class InfoStorge {
private:
    School *s;
public:

    int count = 0;
    void menu();
    void printall();
    void addObject(int index, int num, int cos, int pup, int emp, char *p);
    void delElem(int index);
    void find(int index);
    void print(int i);
    void createObjects();
    void deleteall();
    void setCount(int c);
    void average();
};

```

Рисунок 6

6. InfoStorge.cpp – файл , у якому описані методи класу InfoStorge.

1) delElem – метод який видаляє об'єкт з масиву об'єктів (рис.7)

```
void InfoStorge::delElem(int index) {
    int edit = 0;
    char ptr[20];

    School *deltmp = (School*)operator new(sizeof(School)*(count-1));
    int j = 0;
    for (int i = 0; i < index; i++) {
        edit = s[i].getCost();
        deltmp[j].setCost(edit);
        edit = s[i].getNumber();
        deltmp[j].setNumber(edit);
        edit = s[i].getPupils();
        deltmp[j].setPupils(edit);
        edit = s[i].getEmployers();
        deltmp[j].setEmployers(edit);
        s[i].getName(ptr);
        deltmp[j].setName(ptr);
        s[i].deleteName();
        j++;
    }

    for (int i = (index + 1); i < count; i++) {
        edit = s[i].getCost();
        deltmp[j].setCost(edit);
        edit = s[i].getNumber();
        deltmp[j].setNumber(edit);
        edit = s[i].getPupils();
        deltmp[j].setPupils(edit);
        edit = s[i].getEmployers();
        deltmp[j].setEmployers(edit);
        s[i].getName(ptr);
        deltmp[j].setName(ptr);
        s[i].deleteName();
        j++;
    }
    count--;
    s[index].deleteName();
    delete[] s;
    s = deltmp;
}
```

Рисунок 7

2) print – метод , який виводить у вікно консолі зміст об'єкта (рис.8)

```
void InfoStorge::print(int i) {
    char *name_ = (char*) operator new (sizeof(char) * 20);
    s[i].getName(name_);
    cout << "Number of school: " << s[i].getNumber();
    printf("\nType:%s", name_);
    cout << endl << "Pupils: " << s[i].getPupils() << endl << "Cost per month: " << s[i].getCost() << "\nEmployers: " << s[i].getEmployers() << endl << endl;
    delete[] name_;
}
```

Рисунок 8

3) addObject – метод , який додає новий об'єкт (рис.9)

```

void InfoStorage::addObject(int index, int num, int cos, int pup, int emp, char *p) {
    int edit = 0;
    char ptr[20];

    int j = 0;
    School *tmp = (School*) operator new (sizeof(School)*(count+1));

    for (int i = 0; i < index; i++) {
        edit = s[i].getCost();
        tmp[j].setCost(edit);
        edit = s[i].getNumber();
        tmp[j].setNumber(edit);
        edit = s[i].getPupils();
        tmp[j].setPupils(edit);
        edit = s[i].getEmployers();
        tmp[j].setEmployers(edit);
        s[i].getName(ptr);
        tmp[j].setName(ptr);
        //tmp[j] = s[index];
        // доделать передачу полей между объектами
        s[i].deleteName();
        j++;
    }
    tmp[index].setCost(cos);
    tmp[index].setNumber(num);
    tmp[index].setEmployers(emp);
    tmp[index].setName(p);
    tmp[index].setPupils(pup);
    j++;
    for (int i = index; i < count; i++) {
        edit = s[i].getCost();
        tmp[j].setCost(edit);
        edit = s[i].getNumber();
        tmp[j].setNumber(edit);
        edit = s[i].getPupils();
        tmp[j].setPupils(edit);
        edit = s[i].getEmployers();
        tmp[j].setEmployers(edit);
        s[i].getName(ptr);
        tmp[j].setName(ptr);
        s[i].deleteName();
        j++;
    }
    count++;
    delete[] s;
    s = tmp;
}

```

Рисунок 9

4) createObjects – метод , який створює «початкові» об'єкти (рис.10)

```

void InfoStorage::createObjects() {
    s = (School *)operator new(sizeof(School) * count);

    for (int i = 0; i < count; i++) {
        char p[25];
        unsigned int tmp = 0;
        cout << "\n\nSchool number: " << i + 1 << endl;
        cout << "Enter cost per month: ";
        cin >> tmp;
        s[i].setCost(tmp);

        cout << "Enter type (gymnasium, high, secondary, primary) if school: ";
        cin >> p;
        s[i].setName(p);

        cout << "Enter number of school: ";
        cin >> tmp;
        s[i].setNumber(tmp);

        cout << "Enter number of pupils: ";
        cin >> tmp;
        s[i].setPupils(tmp);

        cout << "Enter number of employers: ";
        cin >> tmp;
        s[i].setEmployers(tmp);
    }
}

```

Рисунок 10

5) average – метод, у якому визначається якій з об'єктів має більше значення в полі employers ніж середнє за всіх (рис.11)

```

void InfoStorage::average() {
    float average = 0;
    int sum = 0;
    int tmp = 0;
    for (int i = 0; i < count; i++) {
        tmp = s[i].getEmployers();
        sum = sum + tmp;
    }
    average = (sum / count);
    for (int i = 0; i < count; i++) {
        tmp = s[i].getEmployers();
        if (tmp > average) {
            cout << "\nThis school has more employer than avarege of all schools: " << endl;
            print(i);
        }
    }
}

```

Рисунок 11

На рисунках 12 та 13 наведені приклади роботи програми.


```
All schools:-----
Number of school: 500
Type:secondary
Pupils: 400
Cost per month: 400
Employers: 300

Number of school: 300
Type:high
Pupils: 200
Cost per month: 100
Employers: 1

Menu:
1 - Add new objects
2 - Delete object
3 - Find by index
4 - Print all schools
5 - Print schools with employers more than average
0 - End program
Your choose:
```

Рисунок 12

```
This school has more employer than avarege of all schools:
Number of school: 500
Type:secondary
Pupils: 400
Cost per month: 400
Employers: 300

Menu:
1 - Add new objects
2 - Delete object
3 - Find by index
4 - Print all schools
5 - Print schools with employers more than average
0 - End program
Your choose:
```

Рисунок 13

ВИСНОВОК

На даній лабораторній роботі я отримав базові знання про класи, конструктори та деструктори. Дослідити механізм створення та видалення об'єктів.