

Лабораторна робота № 1

Класи та специфікатори доступу. Інкапсуляція. Константи.

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

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Куйдин Михаил Андреевич
- НТУ “ХП”,
- КІТ 102.8а

1.2 Завдання

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.

Додаткові умови виконання завдання:

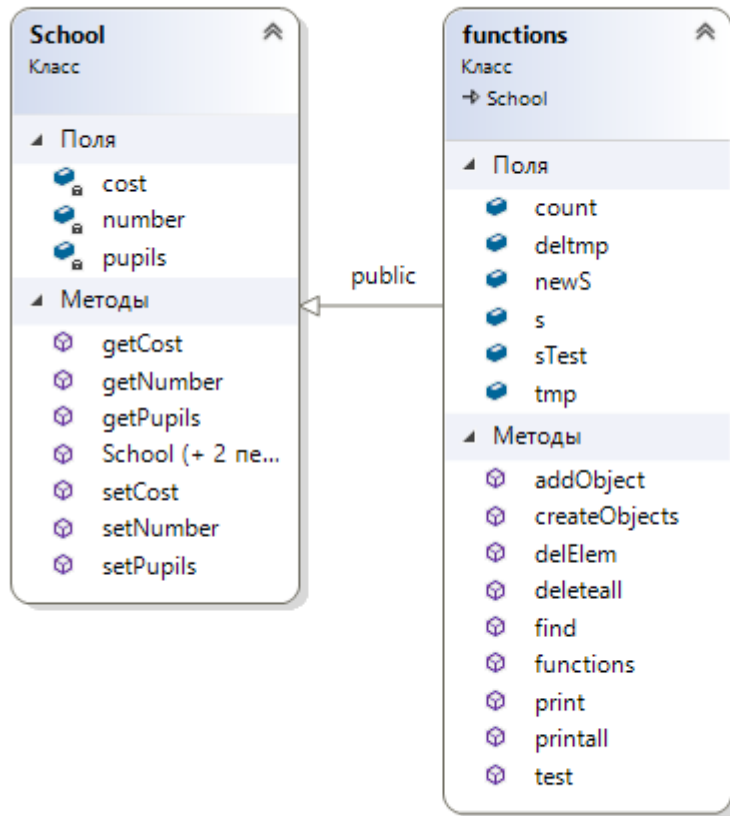
- усі поля «базового класу» повинні бути приватними, та мати публічні гетери та сетери (модифікатори доступу), використовувати механізм інкапсуляції);
- усі функції, що не повинні змінювати поля поточного об'єкту, повинні бути константними;
- усі аргументи функцій, що не змінюються, по можливості також повинні бути константними. Якщо їх не можна зробити константними, в такому разі повинно бути обґрунтування цього;
- в класі-списку метод додавання елемента не повинен вводити дані з клавіатури або файлу, а повинен приймати вже готовий об'єкт для додавання. Метод вводу даних має бути відокремленим;
- продемонструвати відсутність витоків пам'яті;
- продемонструвати роботу розроблених методів класу-списку за допомогою модульних тестів.

2. Описание программы

2.1 Функціональне призначення

Програма створена для створення об'єктів, за допомогою класів та методів.

2.2 Опис логічної структури



3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма складається з 5 файлів:

1.Header.h

2.ConsoleApplication15.cpp

3.Test.cpp

4.SetGet.cpp

5.functions.cpp

Header.h – в хедер-файлі знаходяться прототипи класів та методів:

Class School (рис.2)

```
class School {  
private:  
    unsigned int number;  
    unsigned int pupils;  
    unsigned int cost;  
public:  
    School(int n, int p, int c) {  
        number = n;  
        pupils = p;  
        cost = c;  
    }  
    School()  
    {  
        number = 0;  
        pupils = 0;  
        cost = 0;  
    }  
    School(const School &obj);  
    int getNumber();  
    int getCost();  
    int getPupils();  
    void setCost(int c);  
    void setPupils(int a);  
    void setNumber(int n);  
};
```

Рисунок 2

Class functions (рис.3)

```
class functions: public School {
private:
public:
    int count;
    functions *s;
    functions *tmp;
    functions *newS;
    functions *deltmp;
    functions *sTest;
    functions() :School() {
        ...
    }

    void printall();
    void addObject(int index);
    void delElem(int index);
    void find(int index);
    void print();
    void createObjects();
    void deleteall();
    void test();
};
```

Рисунок 3

ConsoleApplication15.cpp – c++ файл у якому знаходиться main (рис.4)

```

int main() {
    functions tmp;

    tmp.createObjects();
    tmp.addObject(2);
    tmp.addObject(3);
    tmp.delElem(2);
    tmp.find(2);
    tmp.printall();
    tmp.deleteall();

    _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();

    return 0;
}

```

Рисунок 4

SetGet.cpp – файл у якому знаходяться методи setter і getter (рис.5).

```

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

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

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

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

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

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

```

Рисунок 5

functions.cpp – файл у якому знаходяться інші функції програми.

find – функція, яка знаходить об'єкт за індексом та виводить його у вікно консолі (рис.6)

```
void functions::find(int index) {  
    int i = 0;  
    while (i != index) {  
        i++;  
    }  
    cout << "\nInformation about school that u find:\n";  
    s[i].print();  
}
```

Рисунок 6

delElem – функція , яка видаляє об'єкт за індексом (рис.7)

```
void functions::delElem(int index) {  
    count--;  
    deltmp = (functions*)operator new(sizeof(functions)*count);  
    int j = 0;  
    for (int i = 0; i < count; i++) {  
        if (i == index) {  
            j++;  
        }  
        deltmp[i] = s[j];  
        j++;  
    }  
    delete[] s;  
    s = (functions*) operator new (sizeof(functions)*count);  
    for (int i = 0; i < count; i++) {  
        s[i] = deltmp[i];  
    }  
    delete [] deltmp;  
}
```

Рисунок 7

addObject – функція , яка додає новий об'єкт (рис.8)

```

void functions::addObject(int index) {
    tmp = (functions*)operator new(sizeof(functions)*200);
    for (int i = 0; i < count; i++) {
        tmp[i] = s[i];
    }

    delete [] s;

    int j = 0;
    count++; // Потому что добавляем новый элемент
    newS = (functions*)operator new(sizeof(functions) * count);
    s = (functions*)operator new (sizeof(functions)*count);
    for (int i = 0; i < count; i++) {
        if (i != index) {
            newS[i] = tmp[j];
            j++;
        }
        else {
            int a = 0;
            cout << "\nEnter number of school: " << endl;
            cin >> a;
            newS[i].setNumber(a);
            cout << "How many pupils study in this school: " << endl;
            cin >> a;
            newS[i].setPupils(a);
            cout << "Enter cost per month: " << endl;
            cin >> a;
            newS[i].setCost(a);
        }
    }
    for (int i = 0; i < count; i++) {
        s[i] = newS[i];
    }

    delete [] newS;
    delete [] tmp;
}

```

Рисунок 8

createObjects – функція, яка створює “перші” об’єкти та заповнює їх випадковими значеннями (рис.9)

```

void functions::createObjects() {
    functions t;

    cout << " How many objects you want to create: ";
    cin >> count;

    s = (functions*)operator new(sizeof(functions) * count);

    for (int i = 0; i < count; i++) {
        new(&s[i])functions();
    }

    for (int i = 0; i < count; i++) {
        s[i].setCost(rand()%100);
        s[i].setNumber(rand()%100);
        s[i].setPupils(rand()%100);
    }
}

```

Рисунок 9

test – функція, яка запускає модульні тести для функції addObject (рис.10)

```

void functions::test(){
    int snth = 0;
    sTest = (functions*) operator new (sizeof(functions) * 20);

    sTest[0].setCost(41);
    sTest[0].setNumber(67);
    sTest[0].setPupils(34);

    sTest[1].setCost(400);
    sTest[1].setNumber(47);
    sTest[1].setPupils(600);

    for (int i = 0; i < 2; i++) {
        int sNum = s[i].getNumber();
        int sCost = s[i].getCost();
        int sPup = s[i].getPupils();

        int tNum = sTest[i].getNumber();
        int tCost = sTest[i].getCost();
        int tPup = sTest[i].getPupils();

        if (sNum == tNum && sCost == tCost && sPup == tPup) {
            cout << "Test " << i+1 << " passed!" << endl;
        }
        else cout << "Test" << i+1 << " failed" << endl;
    }
}

```

Рисунок 10

У консольному вікні (рис.11)


```
Enter number of school:
179
How many pupils study in this school:
69
Enter cost per month:
63

Enter number of school:
15
How many pupils study in this school:
89
Enter cost per month:
63

Information about school that u find:
Number of school: 15
Pupils: 89
Cost per month: 63

All schools:-----
Number of school: 67
Pupils: 34
Cost per month: 41

Number of school: 69
Pupils: 24
Cost per month: 0

Number of school: 15
Pupils: 89
Cost per month: 63
```

Рисунок 11

У консольному вікні (рис.12) зображено модульні тести.

```
How many objects you want to create: 1

Enter number of school:
47
How many pupils study in this school:
600
Enter cost per month:
400

All schools:-----
Number of school: 67
Pupils: 34
Cost per month: 41

Number of school: 47
Pupils: 600
Cost per month: 400

Test 1 passed!
Test 2 passed!
Для продовження натисніть будь-яку клавішу . . .
```

Рисунок 12

ВИСНОВОК

На данній лабороторній роботі я отримав базові знання про класи та дослідив механізм інкапсуляції.