

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:

- Кононенко Дмитро Олексійович
- 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 елемента, отримання елемента по індексу (або ідентифікатору), вивід усіх елементів на екран.

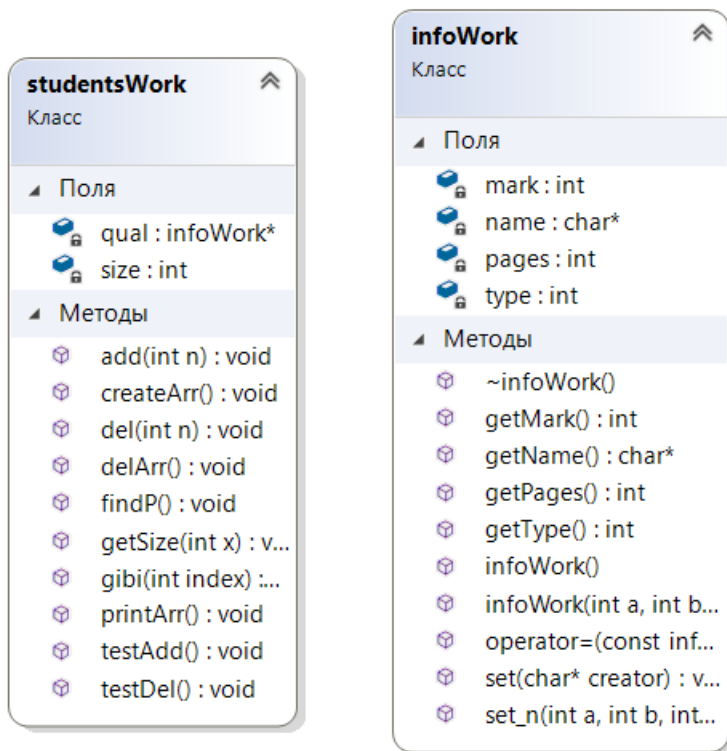
- Додаткові умови виконання завдання:
 - усі поля «базового класу» повинні бути приватними, та мати публічні гетери та сетери (модифікатори доступу), використовувати механізм інкапсуляції);
 - усі функції, що не повинні змінювати поля поточного об'єкту, повинні бути константними;
 - усі аргументи функцій, 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 одульних тестів.

2. Опис програми

2.1 Призначення

Програма створює динамічний масив об'єктів класу за допомогою іншого класу

2.2 Опис логічної структури



3.Варіанти використання

Програма складається з 2 файлів:

1.Laba1.h

2.Laba1.cpp

ваы

ыва

вап

ываы

Рисунок 1 — Базовий клас та методи

```
class infoWork {
private:
    int pages;
    int mark;
    int type;
    char *name;
public:
    infoWork(int a, int b, int c, char* creator){ ... }
    infoWork(){ ... }
    infoWork &operator=(const infoWork &obj){ ... }
    ~infoWork(){ ... }
    void set(char *creator);
    void set_n(int a, int b, int c, char *creator);
    int getPages();
    int getMark();
    int getType();
    char* getName();
};
```

Рисунок 2 — додатковий клас

```
class studentsWork {
private:
    int size;
    infoWork *qual;
public:
    void getSize(int x);
    void createArr();
    void gibi(int index);
    void printArr();
    void findP();
    void add(int n);
    void del(int n);
    void testDel();
    void testAdd();
    void delArr();
};
```

Рисунок 3 — сетери та гетери

```
void infoWork::set(char *creator) {
    this->mark = rand() % 5 + 1;
    this->pages = rand() % 336 + 100;
    this->type = rand() % 2 + 1;
    strcpy_s(this->name, 30, creator);
}

void infoWork::set_n(int a, int b, int c, char *creator) {
    this->mark = a;
    this->pages = b;
    this->type = c;
    strcpy_s(this->name, 30, creator);
}

int infoWork::getMark() {
    return this->mark;
}

int infoWork::getPages() {
    return this->pages;
}

int infoWork::getType() {
    return this->type;
}

char* infoWork::getName() {
    return this->name;
}
```

Рисунок 4 — головна частина програми

```
int main() {
    system("color 70");
    studentsWork qualWork;
    int j, x;
    int choose = 0;

    do {
        cout << "\n";
        cout << "0 - Exit" << endl;
        cout << "1 - Create vector" << endl;
        cout << "2 - Print vector" << endl;
        cout << "3 - Find element by index" << endl;
        cout << "4 - Add element" << endl;
        cout << "5 - Delete element" << endl;
        cout << "6 - Find percent" << endl;
        cout << "Choose: ";
        cin >> choose;
        switch (choose)
        {
            case 1:
                cout << "Input size: ";
                cin >> x;
                qualWork.getSize(x);
                qualWork.createArr();
                break;
            case 2:
                qualWork.printArr();
                break;
            case 3:
                cout << "input index: ";
                cin >> j;
                qualWork.gibi(j);
                break;
            case 4:
                cout << "input insert point: ";
                cin >> j;
                qualWork.add(j);
                break;
            case 5:
                cout << "input delete point: ";
                cin >> j;
                qualWork.del(j);
                break;
            case 6:
                qualWork.findP();
                break;
            default:

```

Рисунок 5 — результат роботи програми

```
0 - Exit
1 - Create vector
2 - Print vector
3 - Find element by index
4 - Add element
5 - Delete element
6 - Find percent
Choose: 2
Name: Ivanova Mark= 2 Size= 423 BACALAVR
Name: Chelak Mark= 1 Size= 117 BACALAVR
Name: Kabak Mark= 4 Size= 226 BACALAVR
```

Висновок: отримав базові знання про класи та дослідив механізм інкапсуляції.