

Лабораторна робота №2. Перевантаження методів

Перевантаження методів

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

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кононенко Дмитро Олексійович
- НТУ “ХПІ”,
- КІТ 102.8а

1.2 Завдання

Загальне завдання. Поширити попередню лабораторну роботу наступним чином:

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

Індивідуальне завдання:

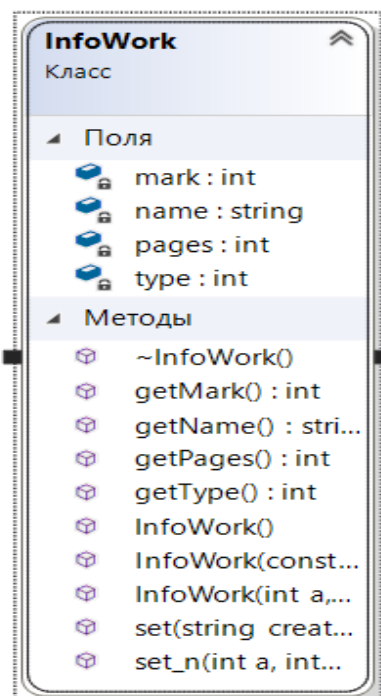
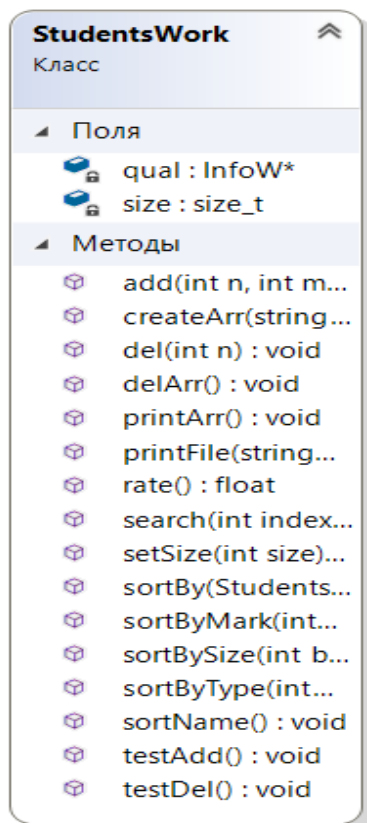
Визначити % магістерських робіт у порівнянні з бакалаврськими роботами

2. Опис програми

2.1 Призначення

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

2.2 Опис логічної структури



Class InfoWork містить такі поля як:

1. int mark – оцінка за кваліфікаційну роботу
- 2.name* char – ім'я студента
- 3.int pages – кількість сторінок в роботі
- 4.int type – тип роботи(магістр\бакалавр)

Class StudentsWork створює масив об'єктів класу InfoWork, а також має методи для обробки цього масиву, а саме:

1. void add(int n) — метод приймає індекс(місце розташування) нового елементу та вставляє його в це місце.
 2. void del(int n) — метод приймає індекс(місце розташування) елементу та видаляє його.
 3. float rate() - метод який знаходить відсоток магістрів у масиві.
 - 4.void createArr() - метод який створює масив об'єктів класу InfoWork.
 5. void printArr() - виводить на екран масив об'єктів класу InfoWork
 6. void search(int index) – виводить об'єкт з індексом який приймає функція
 7. void setSiz(int size) – встановлює розмір масиву об'єктів InfoWork
- *інші методи не входять у завдання лабораторної роботи(за значення функції звертатися до розробника)

3.Варіанти використання

Програма складається з 2 файлів:

1.StudentsWork.cpp

2.StudentsWork.h

3.InfoWork.h

4.InfoWork.cpp

5.Lab1.cpp

```
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 1 — Базовий клас та методи

Рисунок 2 — конструктори та деструктор

```
InfoWork(int a, int b, int c, string creator) {
    pages = a;
    mark = b;
    type = c;
    name = creator;
}
InfoWork() {
    cout << "konstructor " << endl;
    pages = 0;
    mark = 0;
    type = 0;
    name;
}
InfoWork &operator=(const InfoWork &obj) {
    cout << "copy operator" << endl;
    this->pages = obj.pages;
    this->name = obj.name;
    this->mark = obj.mark;
    this->type = obj.type;
    return *this;
}
```

Рисунок 3 — 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.gibl(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 4 — 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: дослідити механізм створення та видалення об'єктів.