

Міністерство освіти і науки України  
Національний технічний університет  
«Харківський політехнічний інститут»  
Кафедра «Обчислювальна техніка та програмування»

### **ЗВІТ**

Про виконання лабораторної роботи № 28  
«ООП. Обробка виключних ситуацій»

Керівник: викладач  
Бульба С. С.

Виконавець: студент гр. КІТ-120в  
Бабенко А. П.

Харків 2021

# Лабораторна робота № 28. ООП. Обробка виключних ситуацій

## 1. Вимоги

### 1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 15 травня 2021;

### 1.2 Загальне завдання

У файлі розміщена інформація про N масивів.

У першому рядку міститься інформація про кількість масивів, у кожній наступній – інформація про кількість елементів у кожному масиві та власне дані масиву.

Необхідно реалізувати програму, що виконує перераховані нижче дії, причому кожна з них в окремій функції, поки користувач не введе замість назви файлу рядок `\exit`

Дії, що має виконувати програма, такі:

- введення з клавіатури назви вхідного файлу з даними;
- читання даних з файлу;
- виконання індивідуального завдання;
- введення з клавіатури імені вихідного файлу;
- запис результату операції у файл;

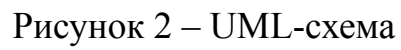
## 2 Виконання роботи

2.1 Написання коду, що відповідає функціональному призначенню..  
Зображено на рис.1.

```
try
{
    while (getline(file, line))
    {
        if (j >= size)
        {
            throw "В файле больше строк чем указано";
        }
        istringstream iss(line);
        std::string token;
        int i = 0;
        int *arr;
        while (getline(iss, token, ' '))
        {
            if (i == 0)
            {
                arr = CreateArray(atoi(token.c_str()));
                arr[i] = atoi(token.c_str());
            }
            else
            {
                arr[i] = atoi(token.c_str()); //заполнение
                i++;
            }
            arrays[j] = arr; //присваивание указателя на массив
            j++;
        }
    }
}
catch (const char *ex)
{
    cout << ex << endl;
}
```

Рисунок 1 – код програми

### 2.3 Створення UML-схеми класу масиву. Зображено на рис.3.



### 2.3 Створення UML-схеми класу масиву. Зображено на рис.3.

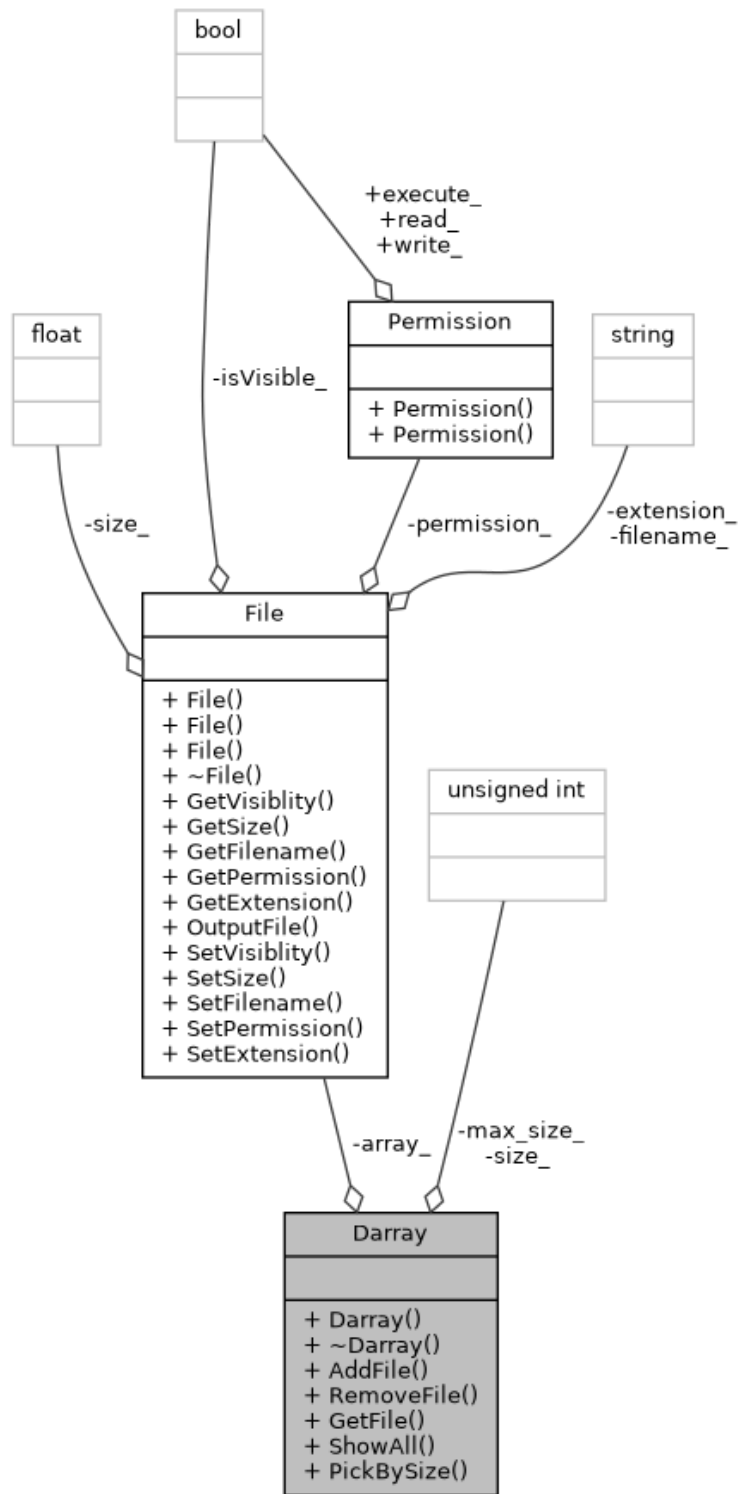


Рисунок 3 – UML-схеми

2.4 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 4.

	<b>File ()</b> Конструктор за замовчуванням <a href="#">Детальніше...</a>
	<b>File (const File &amp;file)</b> Конструктор копіювання <a href="#">Детальніше...</a>
	<b>File (bool isVisible, float size, string filename, Permission *permission, string extension)</b> Конструктор з параметрами <a href="#">Детальніше...</a>
	<b>~File ()</b> Деструктор <a href="#">Детальніше...</a>
bool	<b>GetVisibility () const</b> Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
float	<b>GetSize () const</b> Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
string	<b>GetFilename () const</b> Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
struct Permission *	<b>GetPermission () const</b> Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
string	<b>GetExtension () const</b> Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
void	<b>SetVisibility (bool isVisible)</b> Запис даних в поле екземпляру класу. <a href="#">Детальніше...</a>
void	<b>SetSize (float size)</b> Запис даних в поле екземпляру класу. <a href="#">Детальніше...</a>
void	<b>SetFilename (string filename)</b> Запис даних в поле екземпляру класу. <a href="#">Детальніше...</a>
void	<b>SetPermission (struct Permission *permission)</b> Запис даних в поле екземпляру класу. <a href="#">Детальніше...</a>
void	<b>SetExtension (string extension)</b> Запис даних в поле екземпляру класу. <a href="#">Детальніше...</a>
string	<b>toString ()</b> Виведення файлу колекції. <a href="#">Детальніше...</a>
void	<b>fromString (string str)</b> Введення файлу колекції. <a href="#">Детальніше...</a>

	<b>Darray</b> (unsigned int max_size=4) Конструктор с параметрами <a href="#">Детальніше...</a>
	<b>~Darray</b> () Деструктор <a href="#">Детальніше...</a>
void	<b>AddFile</b> (string str) Додавання екземпляру класу у колекцію. <a href="#">Детальніше...</a>
void	<b>RemoveFile</b> (unsigned int index) Видалення екземпляру класу з колекцію. <a href="#">Детальніше...</a>
<b>File *</b>	<b>GetFile</b> (unsigned int index) const Повернення файлу з колекції. <a href="#">Детальніше...</a>
void	<b>ShowAll</b> () const Виведення всіх файлів колекції. <a href="#">Детальніше...</a>
void	<b>PickBySize</b> () const Виведення файлів колекції > 50kb. <a href="#">Детальніше...</a>
unsigned int	<b>GetSize</b> () const Повернення даних з поля екземпляру класу. <a href="#">Детальніше...</a>
void	<b>readFromFile</b> (string fileName) Читання з файлу. <a href="#">Детальніше...</a>
void	<b>writeToFile</b> (string filename) Запис в файл. <a href="#">Детальніше...</a>

Рисунок 4 – опис методів

## Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм з використанням базових принципів ООП, а саме обробки виключних ситуацій.