

# Лабораторна робота № 3. Потоки

**Тема.** Робота з потоками: потокове введення / виведення на консоль та у файл, рядки типу *string*, *stringstream*.

**Мета роботи.** Отримати знання про основи роботи з потоковим введенням / виведенням на мові C++, роботу з файлами та рядками типу *string*.**1**

## ВИМОГИ

### 1.1 Розробник

Інформація про розробника:

- Малюга Андрій Володимирович,
- НТУ “ХПІ” КІТ 102.8а
- Варіант 13

### 1.2 Загальне завдання

Поширити попередню лабораторну роботу наступним чином:

- ☐ використання функцій *printf/scanf* замінити на використання *cin/cout*;
  - ☐ усі конкатенації рядків замінити на використання *stringstream*;
  - ☐ замінити метод виводу інформації про об’єкт на метод, що повертає рядок-інформацію про об’єкт, який далі можна виводити на екран;
  - ☐ замінити метод вводу інформації про об’єкт на метод, що приймає рядок з інформацією про об’єкт, обробляє його та створює об’єкт на базі цієї інформації;
- поширити клас-список, шляхом реалізації методів роботи з файлами за допомогою файлових потоків (*fstream*) (якщо використовувалися функції *fprintf/fscanf* – замінити їх на класи *ifstream/ofstream*), при цьому сигнатури методів повинні виглядати наступним чином: ☐ читання:

```
void CList::readFromFile(string fileName);
```

де *CList* – клас-список об’єктів, при цьому слід пам’ятати, що при повторному читанні з файлу, попередні дані списку повинні бути очищені;

☐ запис: 

```
void CList::writeToFile(string fileName);
```

*Додаткові умови виконання завдання:*

- ☐ продемонструвати відсутність витоків пам’яті;
- ☐ продемонструвати роботу розроблених методів за допомогою модульних тестів;
- ☐ не використовувати конструкцію «*using namespace std;*», замість цього слід роботи «*using*» кожного необхідного класу: *using std::string*, *using std::cout*;
- ☐ в проекті не повинні використовуватися бібліотеки введення / виведення мови C, а також не повинні використовуватися рядки типу *char\**.

## 2 ОПИС ПРОГРАМИ

### 2.1 Функціональне призначення

За допомогою цієї програми можна створити масив об'єктів, додавати та видаляти об'єкти, виводити вміст масиву на екран та вивід об'єкта по індексу, пошук по імені, читання з файлу даних про об'єкти та запис значень полів об'єктів масиву у файл. Також у цій програмі реалізоване зручне меню спілкування з користувачем.

### 2.2 Важливі фрагменти програми

На рисунку №1 зображено діаграму класів

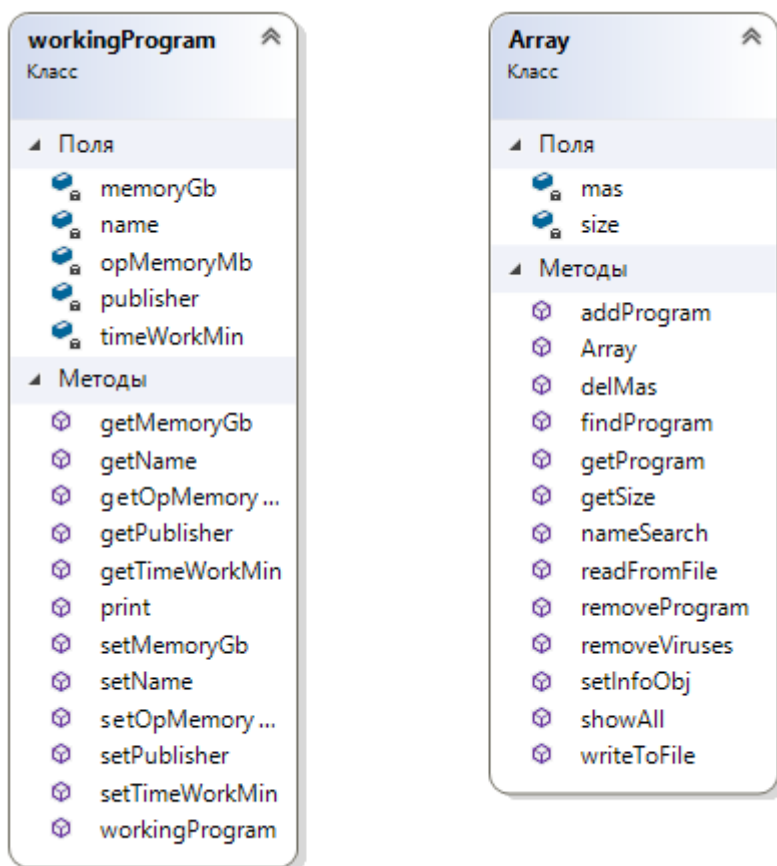


Рисунок №1 – діаграма класів

Методи класу `workingProgram` :

- `workingProgram()` - Конструктор за замовчуванням
- `stringstream print()` - метод виводу значень полів об'єкту
- `void setName(string name)` - заповнення поля `workingProgram::name` (інші методи `set` роблять теж саме але з іншими полями)
- `string getName()` - читання значення поля `name` (інші методи `get` роблять теж саме але з іншими полями)

Методи класу **Array**:

- **Array()** - конструктор за замовчуванням
- **void writeToFile()** - метод для запису значень полів в файл.
- **void readFromFile(int &sizeMas, workingProgram &newObj)** - метод для читання інформації про об'єкти з файлу
- **void setInfoObj(workingProgram& obj)** - метод створений для читання з клавіатури інформації про об'єкт
- **void addProgram(workingProgram &newObj, int ind)** - метод створений, щоб додати об'єкт в масив
- **void showAll()** - метод створений для виведення всіх елементів масиву на екран
- **void removeProgram(int ind)** - метод створений для видалення елемента з масиву
- **void getProgram(int ind)** - метод створений для виведення одного елемента за індексом з масиву
- **void nameSearch(string n)** - метод створений для пошуку об'єкта масиву по імені
- **void findProgram(float memoryGB)** - метод створений для виведення на екран об'єктів з масиву в заданому діапазоні значення поля memoryGB
- **void removeViruses()** - метод створений для видалення підозрілих програм з масиву
- **void delMas()** - метод створений для очищення виділеної пам'яті для масиву об'єктів
- **int getSize()** - метод створений для читання значення поля size

### 3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма може бути використана для створення масиву об'єктів. Програма має методи додавання, видалення об'єктів, пошук об'єктів по імені та вивід по індексу з масиву, читання з файлу даних про об'єкти та запис значень полів об'єктів масиву у файл. Меню робить роботу з цією програмою зручною.

Меню спілкування з користувачем зображено на рисунку №2

```
Quantity of objects in array: 0
What function do you want to cause the list?
<0>exit from program
<1>output array on display
<2>name search
<3>to delete object from array
<4>to add new object to array
<5>index output on display
<6>show programs, that take up more memory of a given size
<7>to delete suspicious programs from array
<8>to read information of objects from file
<9>to write array to file
```

Рисунок №2 – меню спілкування з користувачем

Результат виводу об'єктів масиву на екран зображено на рисунку №3

```
Name of program: Smart Defrag
Publisher: IObit
Amount of consumed RAM(Mb): 356
Occupied amount of hard disk memory(Gb): 0.749
Time of work (in minutes): 20

Name of program: Internet Security
Publisher: ESET
Amount of consumed RAM(Mb): 200
Occupied amount of hard disk memory(Gb): 0.5
Time of work (in minutes): 49

Для продовження натисніть будь-яку клавішу . . .
```

Рисунок №3 - результат виводу об'єктів масиву на екран

## ВИСНОВКИ

В інтегрованому середовищі Visual Studio розроблена програма мовою C. Засоби налагодження дозволяють за допомогою меню спілкування створити масив об'єктів, змінювати його за допомогою методів класів.