

Лабораторна робота №28.

Обробка виключних ситуацій.

- **Вимоги**
- **Розробник**
 - Гладков Костянтин Сергійович
 - Студент групи КІТ-320;
 - 01-march-2021.

- **Основне завдання**

Загальне завдання. (Задача не пов'язана з попередніми роботами).

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;
- доступ до елемента за індексом слід винести в окрему функцію, що виконує перевірку на можливість виходу за межі масиву.

Зверніть увагу. Слід окремо звернути увагу, що при обробці виключення цикл не повинен перериватись.

- **Опис роботи**

Програма працює за принципом читання з файлу інформації, її записи в масив, а потім вже подальшої роботи з цими масивами і файлами.

- **Функціональне призначення**

Дана програма може бути використана для маніпуляцій з нашим динамічним масивом, який має в собі класи. Таким чином ми отримуємо

великий функціонал на увазі більше кількості можливих методів для роботи з нашим масивом.

- **Опис логічної структури**

int main містить в собі виклик функцій, а також корисних властивостей для функцій

int amount_of_arr - функція зчитує з файлу скільки масивів існує

void amounts_in_file - функція зчитує з файлу скільки елементів в кожному масиві

void fill_arrays - функція заповнює масив

void test_indexing - функція перевіряє індексування

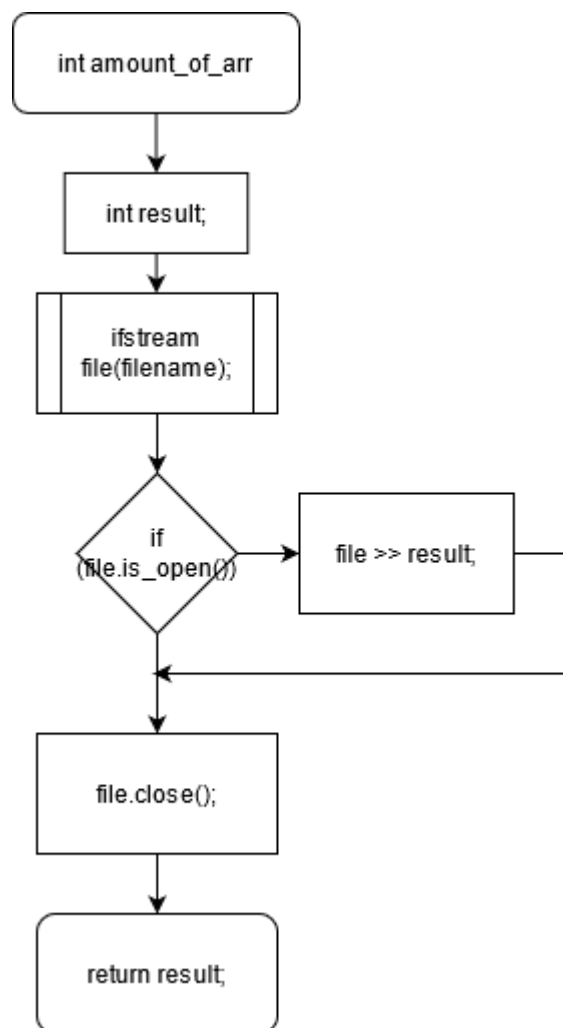


рис. 1 - amount_of_arr

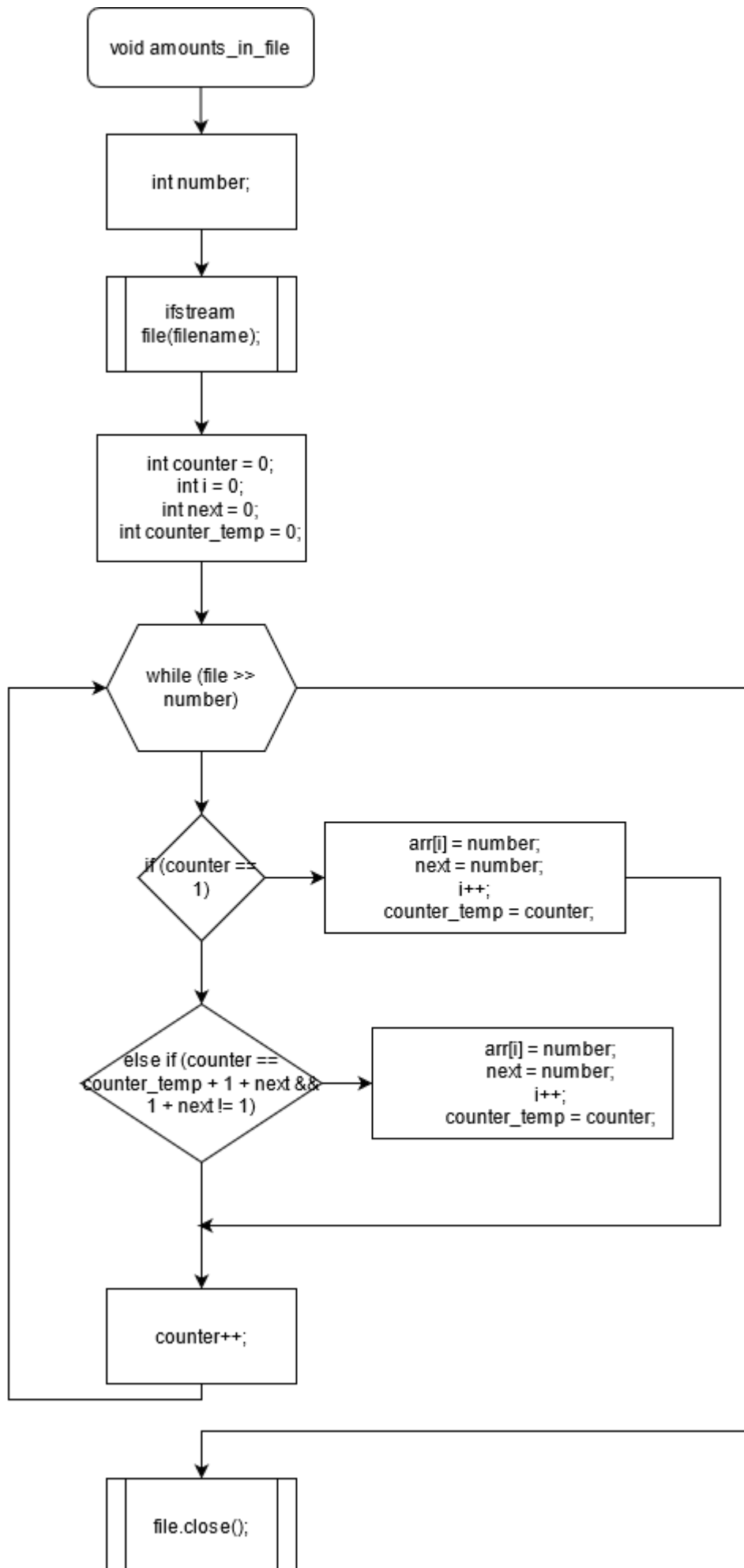


рис. 2 - amounts_in_file

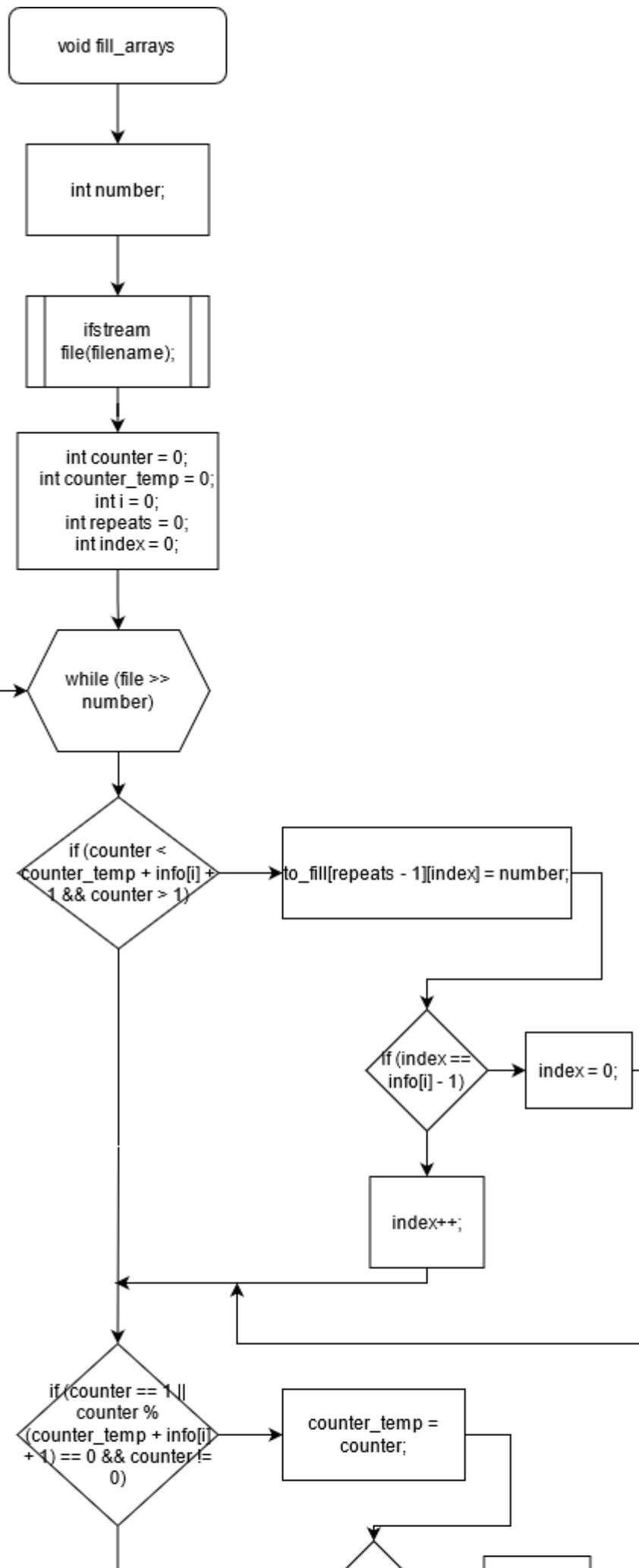


рис. 3 - fill_arrays

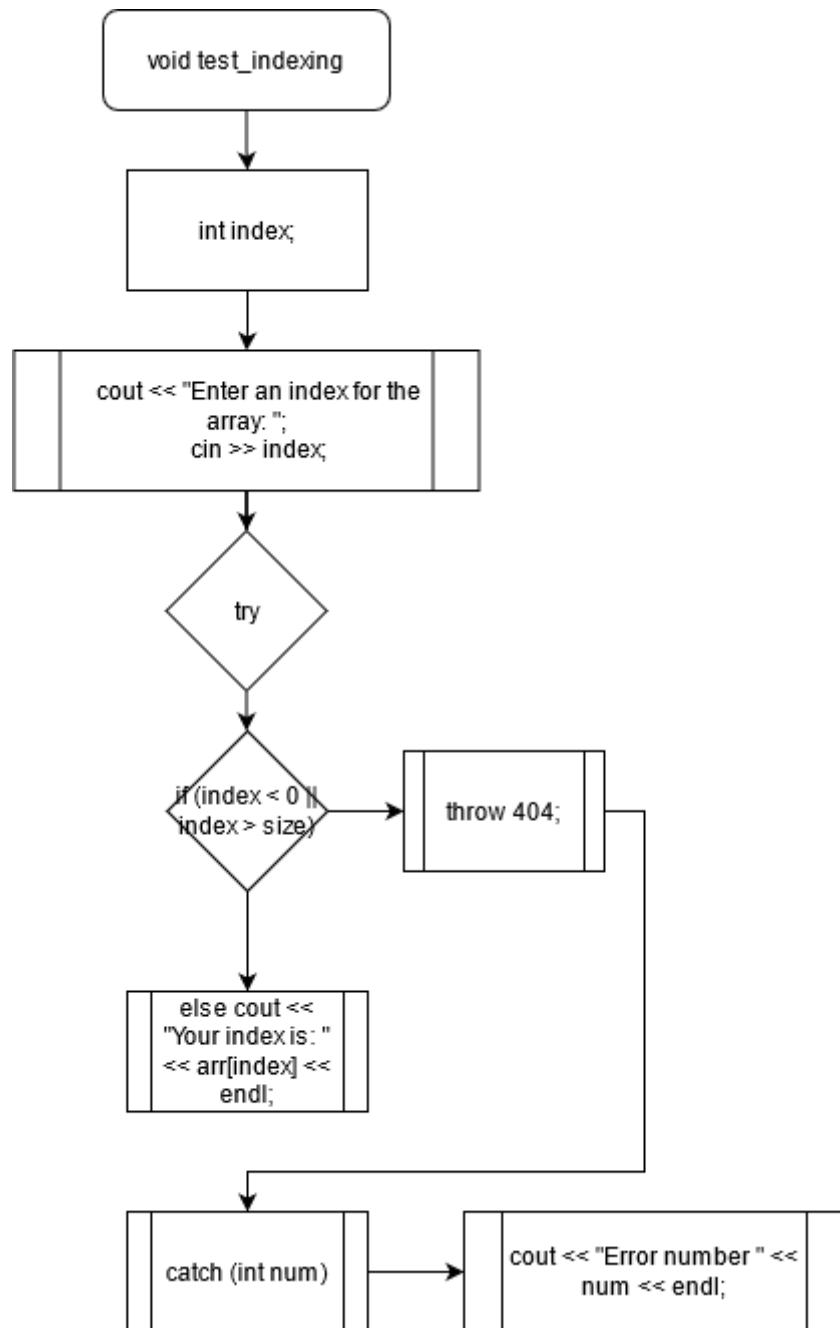


рис. 4 - test_indexing

- **Важливі елементи програми**


```

int result;

ifstream file(filename);

if (file.is_open()){
    file >> result;
}

file.close();

return result;


int number;

ifstream file(filename);

int counter = 0;
int i = 0;
int next = 0;
int counter_temp = 0;

while (file >> number) {
    if (counter == 1){
        arr[i] = number;
        next = number;
        i++;
        counter_temp = counter;
    }
    else if (counter == counter_temp + 1 + next && 1 + next != 1){
        arr[i] = number;
        next = number;
        i++;
        counter_temp = counter;
    }
    counter++;
}

file.close();


int index;
cout << "Enter an index for the array: ";
cin >> index;

```

```
try{
    if (index < 0 || index > size){
        throw 404;
    }
    else cout << "Your index is: " << arr[index] << endl;

}
catch (int num){
    cout << "Error number " << num << endl;
}
```

- **Варіанти використання**

Дана програма може бути використана для аналізу масиву, який зберігається в файлі.

Висновки

У даній лабораторній був придбаний досвід роботи з винятками, а також був повторений досвід роботи з файлами.