

Лабораторна робота №9.

ВИКЛЮЧЕННЯ

Тема. Виключення.

Мета – навчитись розробляти програми з реалізацією виключень.

1 Завдання до роботи

Індивідуальне завдання 19.

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

9. Знайти у масиві номер першого елемента з максимальним значенням.

Результат операції – масив з номерів максимальних елементів кожного із вхідних масивів

2 Розробка алгоритму розв'язання задачі.

2.1 Опис змінних

Arr stud_array; class Student; class Arr;

Класи, методи, функції, конструктори

3 Код програми

arr.h

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS
#define CRTDBG_MAP_ALLOC
#include <crtdbg.h>
#define DEBUG_NEW new(_NORMAL_BLOCK, FILE, __LINE)

#include <iostream>
#include <string>
#include <locale>
#include <fstream>
#include <sstream>
#include <algorithm>

using std::ifstream;
using std::ofstream;
using std::ostream;
using std::string;
using std::cout;
using std::cin;
using std::endl;
using std::exception;
using std::stringstream;

class Array
{
public:
    int* array;

public:
    Array();
    ~Array();
};
```

arrList.h

```
#pragma once
#include "arr.h"

class ArrayList
{
private:
    int listSize;
    Array* arrayList;
    int* task;

public:
    void ReadFile(string);
    int LinesInFile(string) const;
    void OutputArr() const;
    void Task();
    void SaveToFile(string) const;
    void DeleteArr();
    void CheckArr(int, int);

    ~ArrayList();
};
```

arr.cpp

```
#include "arr.h"

Array::Array() : array(0)
{

}

Array::~~Array()
{

}
```

arrList.cpp

```
#include "arrList.h"
#include <cassert>

void ArrayList::ReadFile(string filename)
{
    ifstream file;
    file.exceptions(ifstream::badbit | ifstream::failbit);
    string line;
    int arrAmount; //количество массивов в файле
    int arrLength; //длина массива
    int intTemp;
    int count = 0;
    size_t i, j;

    try
    {
        file.open(filename);

        getline(file, line);
        arrAmount = std::stoi(line);
        if (arrAmount <= 0 || arrAmount != (LinesInFile(filename) - 1)) //проверка
на правильное кол-во массивов
        {
            throw "Неверное количество массивов в файле.";
        }

        for (i = 0; i < arrAmount; i++) //проверка правильности массивов в файле
        {
            getline(file, line);
            istringstream temp(line);
            temp >> arrLength;

            if (std::count(line.begin(), line.end(), ' ') != arrLength ||
arrLength == 0)
            {
                arrAmount--;
                i--;
            }
        }
        arrayList = new Array[arrAmount]; //выделяем память под массив массивов

        file.clear(); //очищаем поток
        file.seekg(0, std::ios_base::beg); //возвращаем указатель в начало файла
        getline(file, line);
        arrAmount = std::stoi(line);

        for (i = 0; i < arrAmount; i++)
```

```

        {
            getline(file, line);
            istringstream temp(line);
            temp >> arrLength;

            try
            {
                if (std::count(line.begin(), line.end(), ' ') != arrLength)
                {
                    throw exception("Ошибка в данных файла. Переменная
размера и фактический размер одного из массивов не совпадают. ");
                }
                if (arrLength == 0)
                {
                    throw exception("Пустой массив данных.");
                }

                arrayList[i].array = new int[arrLength + 1];
                arrayList[i].array[0] = arrLength;

                for (j = 1; j <= arrLength; j++)
                {
                    temp >> intTemp;
                    arrayList[i].array[j] = intTemp;
                }
            }
            catch (const exception & ex)
            {
                cout << ex.what() << endl;
                arrAmount--;
                i--;
            }
        }
    }
    catch (const char* error)
    {
        cout << error << endl;
        file.close();
        return;
    }

    file.close();
    listSize = arrAmount;
}

int ArrayList::LinesInFile(string filename) const
{
    ifstream fin(filename);
    string line;
    int size = 0;

    while (getline(fin, line))
    {
        size++;
    }

    fin.close();
    return size;
}

void ArrayList::OutputArr() const
{
    cout << endl << "Элементы в массиве: " << endl;
    for (size_t i = 0; i < listSize; i++)
    {
        cout << arrayList[i].array[0];
    }
}

```

```

        cout << " ";
        for (size_t j = 1; j <= arrayList[i].array[0]; j++)
        {
            cout << arrayList[i].array[j] << " ";
        }
        cout << endl;
    }
    cout << endl;
}

void ArrayList::Task()
{
    task = new int[listSize];
    int max, numb;

    cout << "Номера наибольших значений массивов: " << endl;
    for (size_t i = 0; i < listSize; i++)
    {
        max = arrayList[i].array[1];
        numb = 1;
        for (size_t j = 2; j <= arrayList[i].array[0]; j++)
        {
            if (arrayList[i].array[j] > max)
            {
                max = arrayList[i].array[j];
                numb = j;
            }
        }
        task[i] = numb;
        cout << task[i] << endl;
    }
}

void ArrayList::SaveToFile(string filename) const
{
    try
    {
        ofstream fout(filename);
        fout.exceptions(ifstream::badbit | ifstream::failbit);

        for (size_t i = 0; i < listSize; i++)
        {
            fout << task[i] << endl;
        }

        fout.close();

        cout << "Данный в файл записаны." << endl;
    }
    catch (const char* error)
    {
        cout << error << endl;
        return;
    }
}

void ArrayList::DeleteArr()
{
    for (size_t i = 0; i < listSize; i++)
    {
        delete arrayList[i].array;
    }
    delete[] arrayList;
    delete[] task;
}

```

```

}

void ArrayList::CheckArr(int Arr, int Element)
{
    try
    {
        if (Arr > listSize || Arr <= 0)
        {
            throw exception("Неверный номер массива.");
        }
        else if (Element > arrayList[Arr - 1].array[0])
        {
            throw exception("Неверный номер элемента.");
        }
        else
        {
            cout << "Элемент: " << arrayList[Arr - 1].array[Element] << endl;
        }
    }
    catch (const exception& ex)
    {
        cout << ex.what() << endl;
        return;
    }
}

ArrayList::~~ArrayList()
{
}

```

main.cpp

```

#include "arrList.h"

string InputFileName();
string OutputFileName();

void Func();

int main()
{
    setlocale(LC_ALL, "ru");

    Func();

    if (_CrtDumpMemoryLeaks()) cout << endl << "Есть утечка памяти." << endl;
    else cout << endl << "Утечка памяти отсутствует." << endl;

    return 0;
}

void Func()
{
    string filename;
    ArrayList arrList;
    int value, value2;

    while (filename != "\\ex")
    {
        filename = InputFileName();

        try

```

```

        {
            arrList.ReadFile(filename);
            arrList.OutputArr();
            arrList.Task();
            filename = OutputFileName();
            arrList.SaveToFile(filename);

            cout << endl << "Введите номер массива, элемент которого хотите
получить: ";

            cin >> value;
            cout << "Введите номер элемента, который хотите получить: ";
            cin >> value2;

            arrList.CheckArr(value, value2);
            arrList.DeleteArr();
        }
        catch (const exception& ex)
        {
            if (filename == "\\exit")
            {
                cout << "Завершение работы программы." << endl;
            }
            else
            {
                cout << ex.what() << endl;
                cout << "Невозможно открыть файл с названием " << filename <<
endl;
            }
        }
    }

    return;
}

string InputFileName()
{
    string filename;

    cout << endl << "Введите название файла для чтения: ";
    cin >> filename;
    cout << endl;

    return filename;
}

string OutputFileName()
{
    string filename;

    cout << endl << "Введите название файла для записи результата: ";
    cin >> filename;

    return filename;
}

```

4 Результати тестування

```
Введите название файла для чтения: data.txt

Элементы в массиве:
3 1 3 7
2 4 2
5 6 7 8 9 10
7 11 14 9 0 -3 3 7
4 12 13 14 13
6 16 -8 10 19 20 0

Номера наибольших значений массивов:
3
1
5
2
3
5

Введите название файла для записи результата: result.txt
Данный в файл записаны.

Введите номер массива, элемент которого хотите получить: 2
Введите номер элемента, который хотите получить: 2
Элемент: 2

Введите название файла для чтения:
```

5 Опис результатів

При виконанні лабораторної роботи було набуто практичні навички роботи з виключанням. Виконали індивідуальне завдання. Результати записали у файл.