

Лабораторна робота 9. ВИКЛЮЧЕННЯ

Тема. Виключення.

Мета: навчитись розробляти програми з реалізацією виключень.

1. Завдання до роботи

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

2.1. Опис змінних

Базовий клас, що містить масив даних: [Array](#).

Клас, що має в собі масиви класів: [ArrayList](#).

2.2. Опис змінних

[int](#)* array – динамічний масив елементів (змінна класу [Array](#)).

[int](#) listSize – розмір списку масивів (змінна класу [ArrayList](#)).

[Array](#)* arrayList – масив масивів (змінна класу [ArrayList](#)).

[float](#)* task – масив результатів індивідуального завдання (змінна класу [ArrayList](#)).

2.3. Опис методів

`void ReadFile(string)` – читання даних з файлу (метод класу `ArrayList`).

`int LinesInFile(string) const` – знаходження кількості рядків у файлі (метод класу `ArrayList`).

`void OutputArr() const` – виведення даних у консоль (метод класу `ArrayList`).

`void Task()` – виконання індивідуального завдання (метод класу `ArrayList`).

`void SaveToFile(string) const` – виведення даних у файл (метод класу `ArrayList`).

`void DeleteArr()` – видалення масивів даних (метод класу `ArrayList`).

`~ArrayList()` – деструктор списку елементів (метод класу `ArrayList`).

`~Array()` – деструктор масиву (метод класу `Array`).

`Array()` – конструктор масиву (метод класу `Array`).

3. Текст програми

Array.cpp

```
#include "Array.h"
using std::ofstream;
using std::ostream;

Array::Array(): array(0){}
Array::~Array() {};
```

ArrayList.cpp

```
#include "ArrayList.h"
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);
            istream 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 << "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()
    {
        float sum = 0;
        task = new float[listSize];

        cout << "Средние значения массивов: " << endl;
        for (size_t i = 0; i < listSize; i++)
        {
            for (size_t j = 1; j <= arrayList[i].array[0]; j++)
            {
                sum += arrayList[i].array[j];
            }
            task[i] = sum / arrayList[i].array[0];
            cout << task[i] << endl;

            sum = 0;
        }
    }
    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();
        }
        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 "ArrayList.h"

string InputFileName();
string OutputFileName();
void Func();

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

    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 != "\\exit")
    {
        filename = InputFileName();

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

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

```

```

        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 << "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;
    cout << endl;

    return filename;
}

```

Array.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::istringstream;

```

```

class Array
{
public:
    int* array;

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

```

ArrayList.h

```

#pragma once
#include "Array.h"

class ArrayList
{
private:
    int listSize;           //размер динамического массива
    Array* arrayList;       //динамический массив массивов
    float* task;            //массив средних значений элементов

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

    ~ArrayList();
};

```

4. Результати роботи програми

```
Введите название файла для чтения: data.txt
Ошибка в данных файла. Переменная размера и фактический размер одного из массивов не совпадают.
Ошибка в данных файла. Переменная размера и фактический размер одного из массивов не совпадают.
Пустой массив данных.

Элементы в массиве:
5    27 67 -1 0 52

Средние значения массивов:
29

Введите название файла для записи результата: 1.txt
Данный в файл записаны.

Введите номер массива, элемент которого хотите получить: 1
Введите номер элемента, который хотите получить: 2
Элемент: 67

Введите название файла для чтения: data2.txt

Элементы в массиве:
3    1 2 3
2    4 5
5    6 7 8 9 10
1    11
4    12 13 14 15
6    16 17 18 19 20 21

Средние значения массивов:
2
4.5
8
11
13.5
18.5

Введите название файла для записи результата: 2.txt
Данный в файл записаны.

Введите номер массива, элемент которого хотите получить: 7
Введите номер элемента, который хотите получить: 5
Неверный номер массива.

Введите название файла для чтения: \exit

Завершение работы программы.

Утечка памяти отсутствует.
```

5. Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з виключеннями.

Програма протестована, витоків пам'яті немає, виконується без помилок.