

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут КНІТ  
Кафедра ПЗ

**ЗВІТ**

До лабораторної роботи № 12

**На тему:** “Виняткові ситуації в мові програмування C++ ”

**З дисципліни:** “Об’єктно-орієнтоване програмування”

**Лектор:**

доцент кафедри ПЗ  
Коротєєва Т.О.

**Виконав:**

студент групи ПЗ-16  
Коваленко Д.М.

**Прийняв:**

доцент кафедри ПЗ  
Яцишин С.І.

«\_\_\_\_\_» \_\_\_\_\_ 2022 р.  
 $\Sigma$  = \_\_\_\_\_

**Тема.** Виняткові ситуації в мові програмування C++ .

**Мета.** Ознайомитися з синтаксисом та принципами використання винятків, навчитися передбачати виняткові ситуації, які можуть виникнути в процесі роботи програмного забезпечення, а також навчитися їх перехоплювати та опрацьовувати..

## Лабораторне завдання

Реалізувати програму для роботи з одновимірним масивом дійсних чисел. Масив повинен зчитуватися з файлу. Програма повинна надавати функціонал для знаходження максимального, мінімального числа масиву, а також середнього арифметичного чисел масиву. Результати записувати в інший файл. Вхідний і вихідний файли задаються користувачем. Роботу з масивами необхідно здійснювати за допомогою розробленого класу `MyArray`. Програма повинна перехоплювати та опрацьовувати такі виняткові ситуації: а) масив не містить жодного символу, б) введення користувачем літерного символу замість числа, в) переповнення, г) не існують вхідний і вихідний файли, д) ще дві виняткові ситуації передбачити самостійно.

Всі функції повинні містити список винятків, які вони можуть генерувати.

## Код програми

Назва файлу: *main.cpp*

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

Назва файлу: *mainwindow.cpp*

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "QFileDialog"
#include "QTextStream"
#include "QMessageBox"
#include "myarray.h"
#include "iostream"
using namespace std;
MyArray * a;

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow) {
    ui->setupUi(this);
}

MainWindow::~MainWindow() {
    delete ui;
}

void MainWindow::on_inputButton_clicked() {
    QString filename = QFileDialog::getOpenFileName(this, tr("Select input file"),
"/home/dmytro/", tr("(*)"));
    try {
        a = MyArray::read(filename);
    }
```

```

    } catch (int err) {
        QMessageBox msgBox;
        msgBox.setIcon(QMessageBox::Critical);
        msgBox.setWindowTitle("Error");
        if (err == 1) msgBox.setText("File with name '" + filename + "' does not exist!");
        else if (err == 2) msgBox.setText("File is broken!");
        else if (err == 3) msgBox.setText("File is empty!");
        else if (err == 4) msgBox.setText("Overflow!");
        msgBox.exec();
        return;
    }
    ui->table->setColumnCount(0);
    for (int i = 0; i < a->getSize(); i++) {
        ui->table->insertColumn(i);
        QTableWidgetItem * item = new QTableWidgetItem();
        item->setText(QString::number(a->get(i)));
        ui->table->setItem(0, i, item);
    }
    ui->inputButton->setText(filename.split("/").back());
}

void MainWindow::on_outputButton_clicked() {
    QString filename = QFileDialog::getOpenFileName(this, tr("Select output file"), "/home/dmytro/", tr("(*)"));
    try {
        a->write(filename);
    } catch (int) {
        QMessageBox msgBox;
        msgBox.setIcon(QMessageBox::Critical);
        msgBox.setWindowTitle("Error");
        msgBox.setText("File with name '" + filename + "' does not exist!");
        msgBox.exec();
        return;
    }

    ui->outputButton->setText(filename.split("/").back());
}

void MainWindow::on_minButton_clicked() {
    ui->minLine->setText(QString::number(a->min()));
}

void MainWindow::on_maxButton_clicked() {
    ui->maxLine->setText(QString::number(a->max()));
}

void MainWindow::on_avgButton_clicked() {
    ui->avgLine->setText(QString::number(a->avg()));
}

void MainWindow::on_clearButton_clicked() {
    ui->table->setColumnCount(0);
    delete a;
}

void MainWindow::on_createButton_clicked() {
    int size = ui->table->columnCount();
    a = new MyArray(size);
    for (int i = 0; i < size; i++)
        a->set(i, ui->table->item(0, i)->text().toDouble());
}

```

```

}

void MainWindow::on_addColButton_clicked() {
    ui->table->setColumnCount(ui->table->columnCount() + 1);
}

void MainWindow::on_rmColButton_clicked() {
    ui->table->setColumnCount(ui->table->columnCount() - 1);
}

void MainWindow::on_getButton_clicked() {
    int i = ui->iLine->text().toInt();
    try {
        ui->iAnsLine->setText(QString::number(a->get(i)));
    } catch (int) {
        QMessageBox msgBox;
        msgBox.setIcon(QMessageBox::Critical);
        msgBox.setWindowTitle("Error");
        msgBox.setText("Index overflow!");
        msgBox.exec();
        return;
    }
}

void MainWindow::on_setButton_clicked() {
    int i = ui->iLine_2->text().toInt();
    double v = ui->valLine->text().toDouble();
    try {
        a->set(i, v);
        ui->table->item(0, i)->setText(QString::number(v));
    } catch (int) {
        QMessageBox msgBox;
        msgBox.setIcon(QMessageBox::Critical);
        msgBox.setWindowTitle("Error");
        msgBox.setText("Index overflow!");
        msgBox.exec();
        return;
    }
}
}

```

```

#include "myarray.h"
#include "QFileDialog"
#include "QTextStream"
#include "iostream"

using namespace std;

MyArray::MyArray(QString s) noexcept(false) {
    if (s.isEmpty()) throw 3;
    QStringList list = s.split(" ");
    this->size = list.first().toInt();
    this->array = new double[this->size];
    if (s.count(" ") > this->size) throw 4;
    int i = 0;
    bool ok;
    list.removeFirst();
    for (QString s : list) {
        this->array[i] = s.toDouble(&ok);
        if (!ok) throw 2;
        i++;
    }
}

```

```

}

double MyArray::max() {
    double max = -1000;
    for (int i = 0; i < this->size; i++)
        if (max < this->array[i]) max = array[i];
    return max;
}

double MyArray::min() {
    double min = 1000;
    for (int i = 0; i < this->size; i++)
        if (min > this->array[i]) min = array[i];
    return min;
}

double MyArray::avg() {
    double avg = 0;
    for (int i = 0; i < this->size; i++)
        avg += this->array[i];
    return avg / this->size;
}

double MyArray::get(int i) {
    if (i >= this->size) throw 5;
    return this->array[i];
}

void MyArray::set(int i, double v) {
    if (i >= this->size) throw 6;
    this->array[i] = v;
}

MyArray * MyArray::read(QString filename) noexcept(false) {
    QFile file(filename);

    if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
        throw(1);

    QTextStream in(&file);
    QString line = in.readLine();
    file.close();
    return new MyArray(line);
}

void MyArray::write(QString filename) noexcept(false) {
    QFile file(filename);
    if (!file.open(QIODevice::Append | QIODevice::Text) || filename.
isEmpty())
        throw(1);

    QString text = QString::number(this->size) + " ";
    for (int i = 0; i < this->size; i++)
        text += QString::number(this->get(i)) + " ";
    text = text.trimmed();
    text += "\nMax: " + QString::number(this->max()) + "\n";
    text += "Min: " + QString::number(this->min()) + "\n";
    text += "Avg: " + QString::number(this->avg());

    QTextStream out(&file);
    out << text << "\n";
}

```

```
        file.close();  
    }
```

```
#ifndef MYARRAY_H  
#define MYARRAY_H  
  
#include "QString"  
  
class MyArray {  
    private:  
        double * array;  
        int size;  
    public:  
        MyArray(int size):  
            array(new double[size]),  
            size(size)  
        {};  
        MyArray(QString s) noexcept(false);  
        double max();  
        double min();  
        double avg();  
  
        void write(QString filename) noexcept(false);  
        static MyArray * read(QString filename) noexcept(false);  
  
        int getSize() { return this->size; };  
  
        double get(int i);  
        void set(int i, double v);  
};  
  
#endif // MYARRAY_H
```

## Робота програми

The screenshot shows a software application window titled "MainWindow (on toolbox)". At the top, there is a list of numbers: 15, 14, 77, followed by an empty input field with "+" and "-" buttons. Below this are "Create" and "Clear" buttons. To the right, there are file input fields: "Input file:" with a button labeled "abc", and "Output file:" with a button labeled "Choose...". In the center, there is a panel with three rows: "Min" with a value of 14, "Max" with a value of 42, and "Avg" with a value of 23.6667. At the bottom, there are two rows of controls: "Get" with a button and two input fields containing 1 and 14, and "Set" with a button and two input fields containing 2 and 77.

## Висновок

Під час виконання лабораторної роботи я ознайомився з синтаксисом та принципами використання винятків, навчився передбачати виняткові ситуації, які можуть виникнути в процесі роботи програмного забезпечення, а також навчився їх перехоплювати та опрацьовувати.