Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІПІ

ЗВІТ

з виконання лабораторної роботи № 7

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 1

Виконав:

студент 1-го курсу

гр. ІП-22 ФІОТ

Андреєва Уляна Андріївна

Київ 2023

ПОСТАНОВКА ЗАДАЧІ

1. Розробити клас «Матриця» вказаної розмірності. Створити дві матриці (об'єкти даного класу) та обчислити їх добуток. При виході значення суми за заданий користувачем діапазон та при некоректній розмірності матриці згенерувати відповідні виняткові ситуації і організувати їх обробку.

main.cpp

```
#include <QApplication>
#include <QWidget>
#include <QGridLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QPlainTextEdit>
#include <QInputDialog>
#include <QString>
#include <QPropertyAnimation>
#include <exception>

#include "Functions.h"

int main(int argc, char *argv[]) {
    QApplication app(argc, argv);

    QWidget window;
    window.setStyleSheet("background: #FFF3E6; ");
    window.setGeometry(100, 100, 470, 804);

    QGridLayout layout;
```

```
    QLabel titleLabel("📙Matrix Multiplication Calculator📙");
    titleLabel.setStyleSheet("font-family: 'Nunito'; font-style: normal; font-weight: 600; font-size: 32px; line-height: 44px; display: flex; align-items: center; text-align: center; color: #000000; border: 2px solid #003570; text-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&titleLabel, 0, 0, 1, 3, Qt::AlignHCenter);  // Выравнивание заголовка по центру

    QLabel rangeLabel("Enter the range👉:");
    rangeLabel.setStyleSheet("font-size: 32px; font-weight: 500; font-style: normal; font-family: 'Nunito';"
                             "line-height: 44px; display: flex; align-items: center; text-align: center; color: #000000;");
    layout.addWidget(&rangeLabel, 1, 0, 1, 1);

    QLineEdit startEdit;
    startEdit.setStyleSheet("font-size: 14px; padding: 3px; border: 1px solid #ccc; box-sizing: border-box;"
                            "position: absolute; width: 36px; height: 23px;"
                            "background: #FFFFFF; border: 1px solid #003570; box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&startEdit, 1, 1, 1, 1);

    QLineEdit endEdit;
    endEdit.setStyleSheet("font-size: 14px; padding: 3px; border: 1px solid #ccc; box-sizing: border-box;"
                          "position: absolute; width: 36px; height: 23px;"
                          "background: #FFFFFF; border: 1px solid #003570; box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
```

```
    layout.addWidget(&endEdit, 1, 2, 1, 1);

    QLabel label1("Enter dimensions of the first matrix✏:");
    label1.setStyleSheet("font-size: 20px; font-weight: 700; font-style:
italic; color: #FF8700; text-decoration-line: underline;");
    layout.addWidget(&label1, 2, 0, 1, 3);

    QLabel rowsLabel1("Rows:");
    rowsLabel1.setStyleSheet("font-size: 25px; font-weight: 600; font-
style: normal; font-family: 'Nunito';"
                             "line-height: 44px; display: flex; align-items: center;
text-align: center; color: #000000;");
    layout.addWidget(&rowsLabel1, 3, 0);

    QLineEdit rowsEdit1;
    rowsEdit1.setStyleSheet("font-size: 14px; padding: 3px; border: 1px
solid #ccc; box-sizing: border-box;"
                            "position: absolute; width: 36px; height: 23px;"
                            "background: #FFFFFF; border: 1px solid #003570;
box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&rowsEdit1, 3, 1);

    QLabel colsLabel1("Columns:");
    colsLabel1.setStyleSheet("font-size: 25px; font-weight: 600; font-style:
normal; font-family: 'Nunito';"
                             "line-height: 44px; display: flex; align-items: center;
text-align: center; color: #000000;");
    layout.addWidget(&colsLabel1, 4, 0);
```

```
    QLineEdit colsEdit1;
    colsEdit1.setStyleSheet("font-size: 14px; padding: 3px; border: 1px
solid #ccc; box-sizing: border-box;"
                            "position: absolute; width: 36px; height: 23px;"
                            "background: #FFFFFF; border: 1px solid #003570;
box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&colsEdit1, 4, 1);

    QLabel label2("Enter dimensions of the second matrix here✏:");
    label2.setStyleSheet("font-size: 20px; font-weight: 700; font-style:
italic; color: #FF8700; text-decoration-line: underline;");
    layout.addWidget(&label2, 5, 0, 1, 3);

    QLabel rowsLabel2("Rows:");
    rowsLabel2.setStyleSheet("font-size: 25px; font-weight: 600; font-
style: normal; font-family: 'Nunito';"
                             "line-height: 44px; display: flex; align-items: center;
text-align: center; color: #000000;");
    layout.addWidget(&rowsLabel2, 6, 0);

    QLineEdit rowsEdit2;
    rowsEdit2.setStyleSheet("font-size: 14px; padding: 3px; border: 1px
solid #ccc; box-sizing: border-box;"
                            "position: absolute; width: 36px; height: 23px;"
                            "background: #FFFFFF; border: 1px solid #003570;
box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&rowsEdit2, 6, 1);

    QLabel colsLabel2("Columns:");
```

```
    colsLabel2.setStyleSheet("font-size: 25px; font-weight: 600; font-style: normal; font-family: 'Nunito';"
                             "line-height: 44px; display: flex; align-items: center; text-align: center; color: #000000;");
    layout.addWidget(&colsLabel2, 7, 0);

    QLineEdit colsEdit2;
    colsEdit2.setStyleSheet("font-size: 14px; padding: 3px; border: 1px solid #ccc; box-sizing: border-box;"
                            "position: absolute; width: 36px; height: 23px;"
                            "background: #FFFFFF; border: 1px solid #003570; box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&colsEdit2, 7, 1);

    QPushButton calculateButton("Calculate");
    calculateButton.setStyleSheet("width: 187px; height: 32px; font-family: 'Nunito'; font-size: 26px; color: #FFFFFF; background-color: #FF8700; border: 3px solid #000000; border-radius: 30px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&calculateButton, 8, 0, 1, 3);

    QPushButton exitButton("Exit");
    exitButton.setStyleSheet("width: 187px; height: 32px; font-family: 'Nunito'; font-size: 26px; color: #FFFFFF; background-color: #003570; border: 3px solid #000000; border-radius: 30px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&exitButton, 9, 0, 1, 3);

    QPushButton cleanButton("Clean All");
```

```
    cleanButton.setStyleSheet("width: 187px; height: 32px; font-family: 'Nunito'; font-size: 26px; color: #FFFFFF; background-color: #FF8700; border: 3px solid #000000; border-radius: 30px; box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.25);");
    layout.addWidget(&cleanButton, 10, 0, 1, 3);

    QPlainTextEdit resultOutput;
    resultOutput.setStyleSheet("font-size: 14px; padding: 6px; border: 1px solid #ccc;");
    layout.addWidget(&resultOutput, 11, 0, 1, 3);

    QObject::connect(&calculateButton, &QPushButton::clicked, [&]() {
        calculateButtonClicked(startEdit, endEdit, rowsEdit1, colsEdit1, rowsEdit2, colsEdit2, resultOutput);
    });

    QObject::connect(&exitButton, &QPushButton::clicked, [&]() {
        QApplication::quit();
    });

    QObject::connect(&cleanButton, &QPushButton::clicked, [&]() {
        startEdit.clear();
        endEdit.clear();
        rowsEdit1.clear();
        colsEdit1.clear();
        rowsEdit2.clear();
        colsEdit2.clear();
        resultOutput.clear();
    });
```

```
    window.setLayout(&layout);
    window.show();

    return app.exec();
}
```

Functions.cpp

```
#include "Functions.h"
#include <QInputDialog>

Matrix matrixMultiply(Matrix matrix1, Matrix matrix2, int start, int end, QPlainTextEdit&
resultOutput) {
   int rows1 = matrix1.getRows();
   int cols1 = matrix1.getColumns();
   int rows2 = matrix2.getRows();
   int cols2 = matrix2.getColumns();

   Matrix result(rows1, cols2);

   for (int i = 0; i < rows1; i++) {
      for (int j = 0; j < cols2; j++) {
         for (int k = 0; k < cols1; k++) {
            int sum = matrix1(i, k) * matrix2(k, j);

            if (sum > end || sum < start) {
               resultOutput.appendHtml("<font color='blue'></font> Calculating element at
position (" + QString::number(i) + "</font color='blue>', " + QString::number(j) + "): </font>"
+ QString::number(result(i, j)) + " + " + QString::number(sum) + " = " +
QString::number(result(i, j) + sum));
            } else {
```

```
                resultOutput.appendPlainText("Calculating element at position (" +
QString::number(i) + ", " + QString::number(j) + "): " + QString::number(result(i, j)) + " + " +
QString::number(sum) + " = " + QString::number(result(i, j) + sum));
            }

            result(i, j) += sum;
        }
    }
  }

  return result;
}

void validateRange(int start, int end) {
  if (start > end) {
    throw std::invalid_argument("Start can't be greater than end");
  }
}

void readMatrixElements(Matrix& matrix, QPlainTextEdit& resultOutput, int rows, int cols,
const QString& title) {
  resultOutput.appendPlainText(title);
  resultOutput.moveCursor(QTextCursor::End);

  for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
      QString input;
      bool validInput = false;

      while (!validInput) {
        input = QInputDialog::getText(nullptr, "Enter Element",
                                      "Element at position (" +
```

```
                                QString::number(i) + ", " +
                                QString::number(j) + "):");

            // Перевірка на валідність числа
            bool ok;
            input.toInt(&ok);

            if (!ok) {
                resultOutput.appendHtml("<font color='red'>Invalid input! Please enter a valid
number.</font>");
            } else {
                validInput = true;
            }
        }

        resultOutput.appendPlainText("Element at position (" +
                                QString::number(i) + ", " +
                                QString::number(j) + "): " +
                                input);
        matrix(i, j) = input.toInt();
    }
  }
}


void calculateButtonClicked(QLineEdit& startEdit, QLineEdit& endEdit, QLineEdit&
rowsEdit1, QLineEdit& colsEdit1, QLineEdit& rowsEdit2, QLineEdit& colsEdit2,
QPlainTextEdit& resultOutput) {
  try {
    int start = startEdit.text().toInt();
    int end = endEdit.text().toInt();
```

```
    validateRange(start, end);

    int rows1 = rowsEdit1.text().toInt();
    int cols1 = colsEdit1.text().toInt();
    int rows2 = rowsEdit2.text().toInt();
    int cols2 = colsEdit2.text().toInt();

    if (cols1 != rows2) {
        throw std::invalid_argument("Cannot multiply matrices with incompatible
dimensions");
    }

    Matrix matrix1(rows1, cols1);
    Matrix matrix2(rows2, cols2);

    resultOutput.clear();

    readMatrixElements(matrix1, resultOutput, rows1, cols1, "Enter elements of the first
matrix:");

    readMatrixElements(matrix2, resultOutput, rows2, cols2, "Enter elements of the second
matrix:");

    resultOutput.appendPlainText("\nMultiply matrices:");
    Matrix result = matrixMultiply(matrix1, matrix2, start, end, resultOutput);

    resultOutput.appendPlainText("\nMatrix product:");
    for (int i = 0; i < result.getRows(); i++) {
        QString row;
        for (int j = 0; j < result.getColumns(); j++) {
            row += QString::number(result(i, j)) + " ";
        }
```

```
            resultOutput.appendPlainText(row);
        }
    } catch (const std::exception& e) {
        resultOutput.appendHtml("<font color='red'> ✕ An error occurred: </font> <font
color='#003570'>" + QString(e.what()) + "</font>") ;
    }
}
```

Functions.h

```
#ifndef FUNCTIONS_H
#define FUNCTIONS_H

#include <QPlainTextEdit>
#include <QLineEdit>
#include "Matrix.h"

Matrix matrixMultiply(Matrix matrix1, Matrix matrix2, int start, int end,
QPlainTextEdit& resultOutput);
void validateRange(int start, int end);
void readMatrixElements(Matrix& matrix, QPlainTextEdit& resultOutput, int
rows, int cols, const QString& title);
void calculateButtonClicked(QLineEdit& startEdit, QLineEdit& endEdit,
QLineEdit& rowsEdit1, QLineEdit& colsEdit1, QLineEdit& rowsEdit2,
QLineEdit& colsEdit2, QPlainTextEdit& resultOutput);

#endif // FUNCTIONS_H
```

Matrix.cpp

```
#include "Matrix.h"
```

```
Matrix::Matrix(int rows, int columns) : rows(rows), columns(columns) {
    if (rows <= 0 || columns <= 0) {
        throw invalid_argument("Invalid matrix dimensions");
    }

    data.resize(rows, vector<int>(columns, 0));
}

int Matrix::getRows() const {
    return rows;
}

int Matrix::getColumns() const {
    return columns;
}

int& Matrix::operator()(int row, int col) {
    if (row < 0 || row >= rows || col < 0 || col >= columns) {
        throw out_of_range("Matrix indices out of range");
    }

    return data[row][col];
}
```

Matrix.h

```
#ifndef MATRIX_H
#define MATRIX_H

#include <iostream>
```

```
#include <vector>
#include <stdexcept>
using namespace std;

class Matrix {
private:
    vector<vector<int>> data;
    int rows;
    int columns;
public:
    Matrix(int rows, int columns);
    int getRows() const ;
    int getColumns() const;
    int& operator()(int row, int col) ;
};

#endif // MATRIX_H
```

Mainwindow.cpp

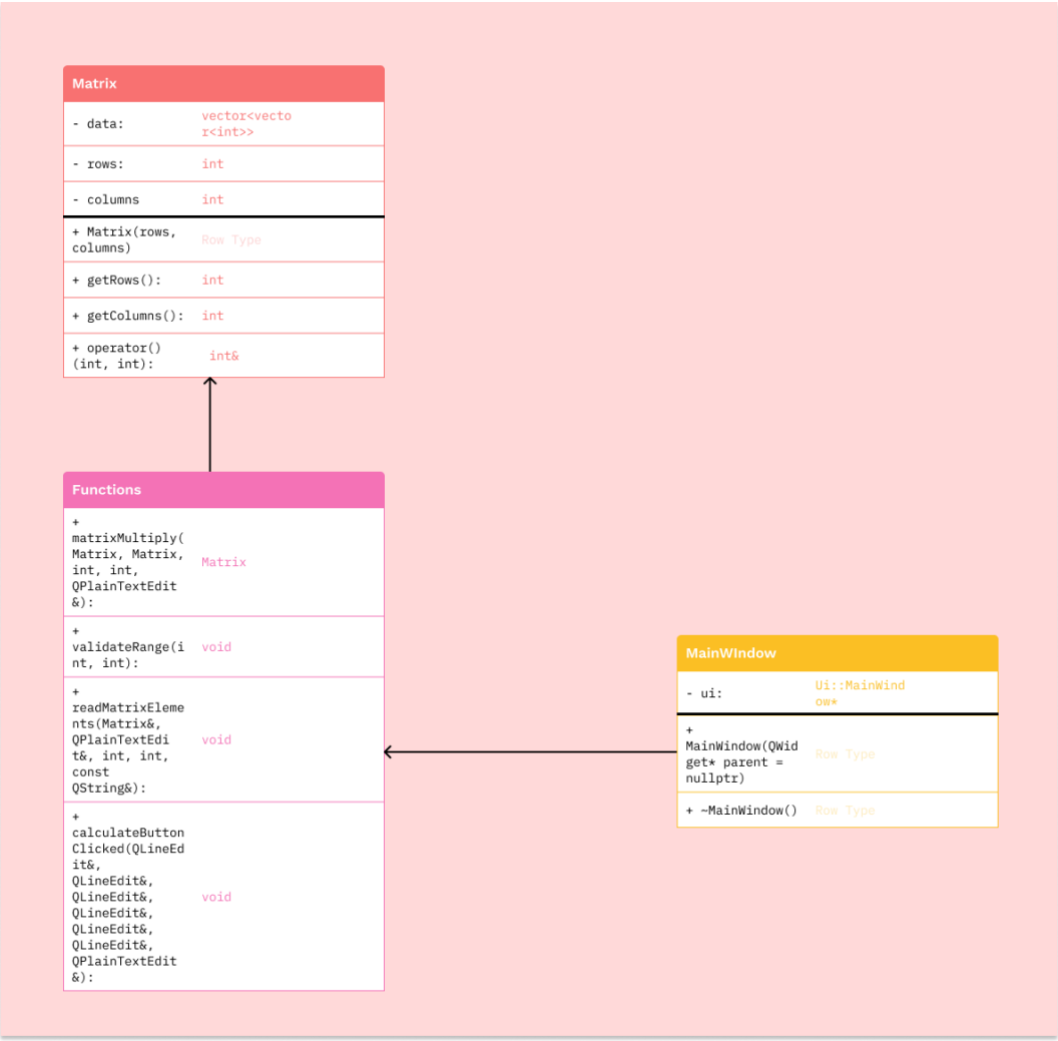
```
#include "mainwindow.h"
#include "./ui_mainwindow.h"

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

MainWindow::~MainWindow()
```

```
{
    delete ui;
}
```

Mainwindow.h

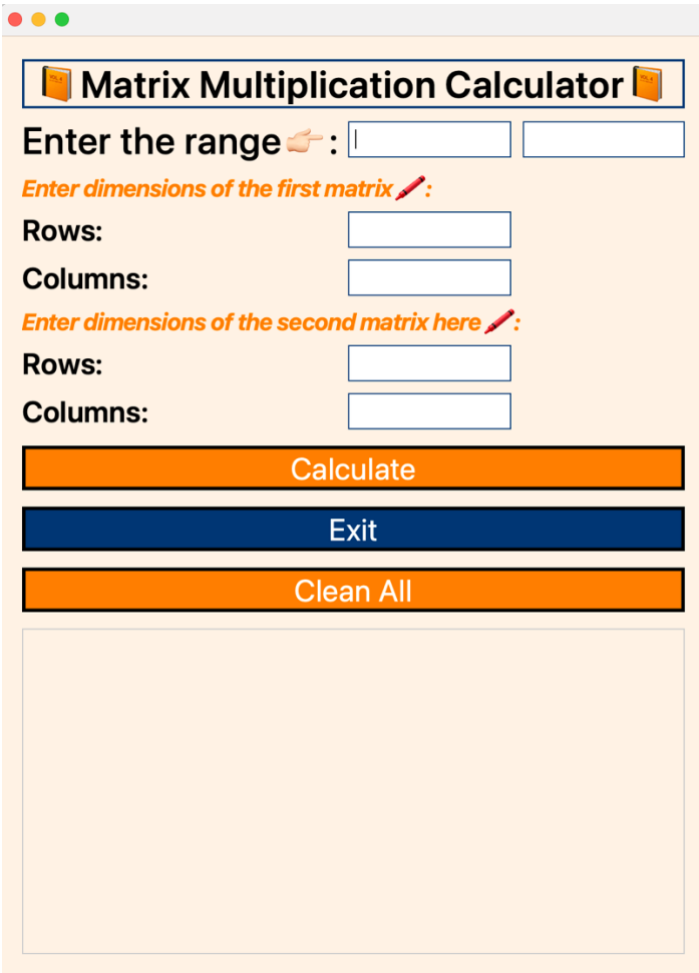
```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

Діаграма класів

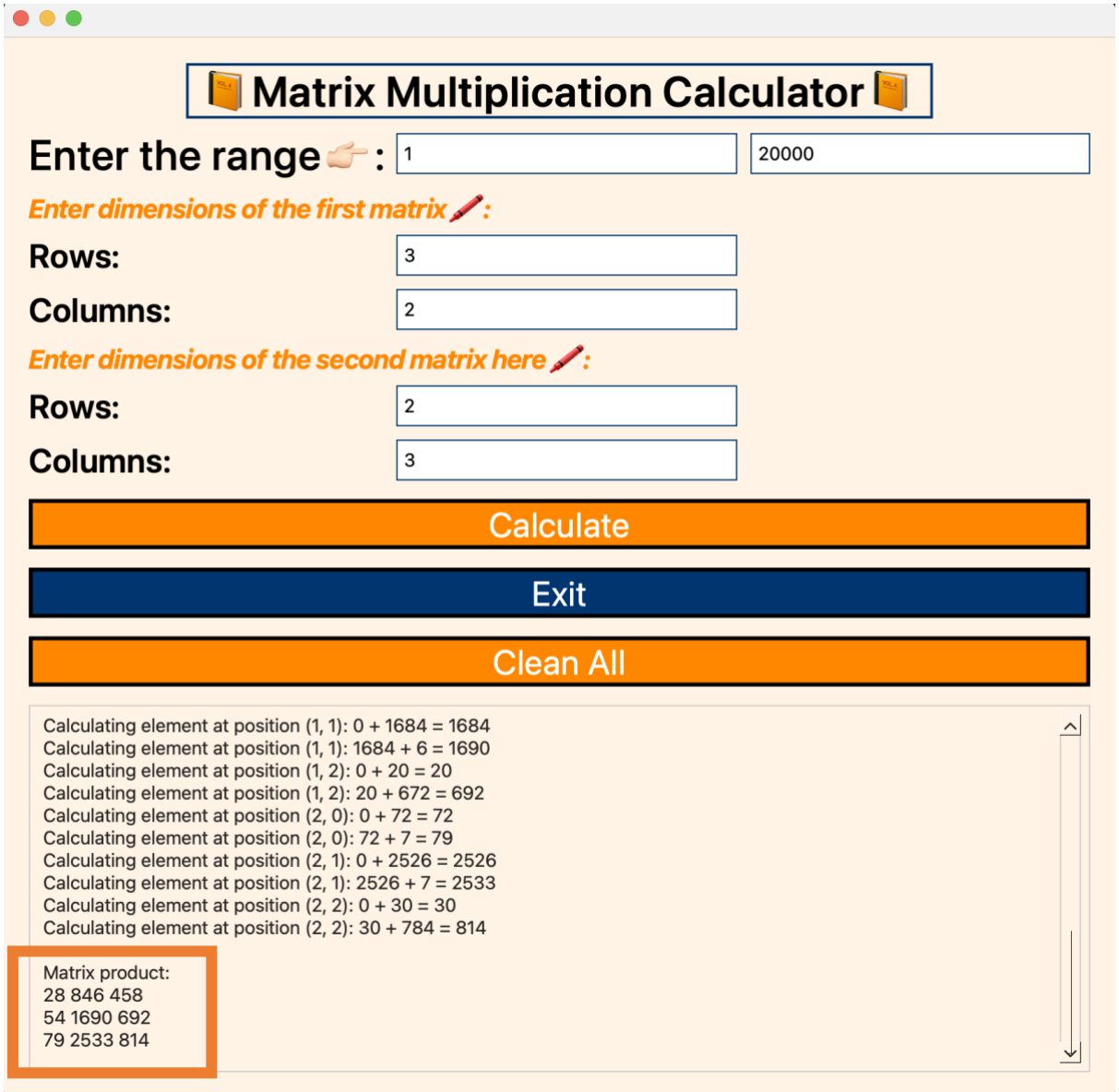


РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Вигляд програми з графічним інтерфейсом



Множення матриць програмою

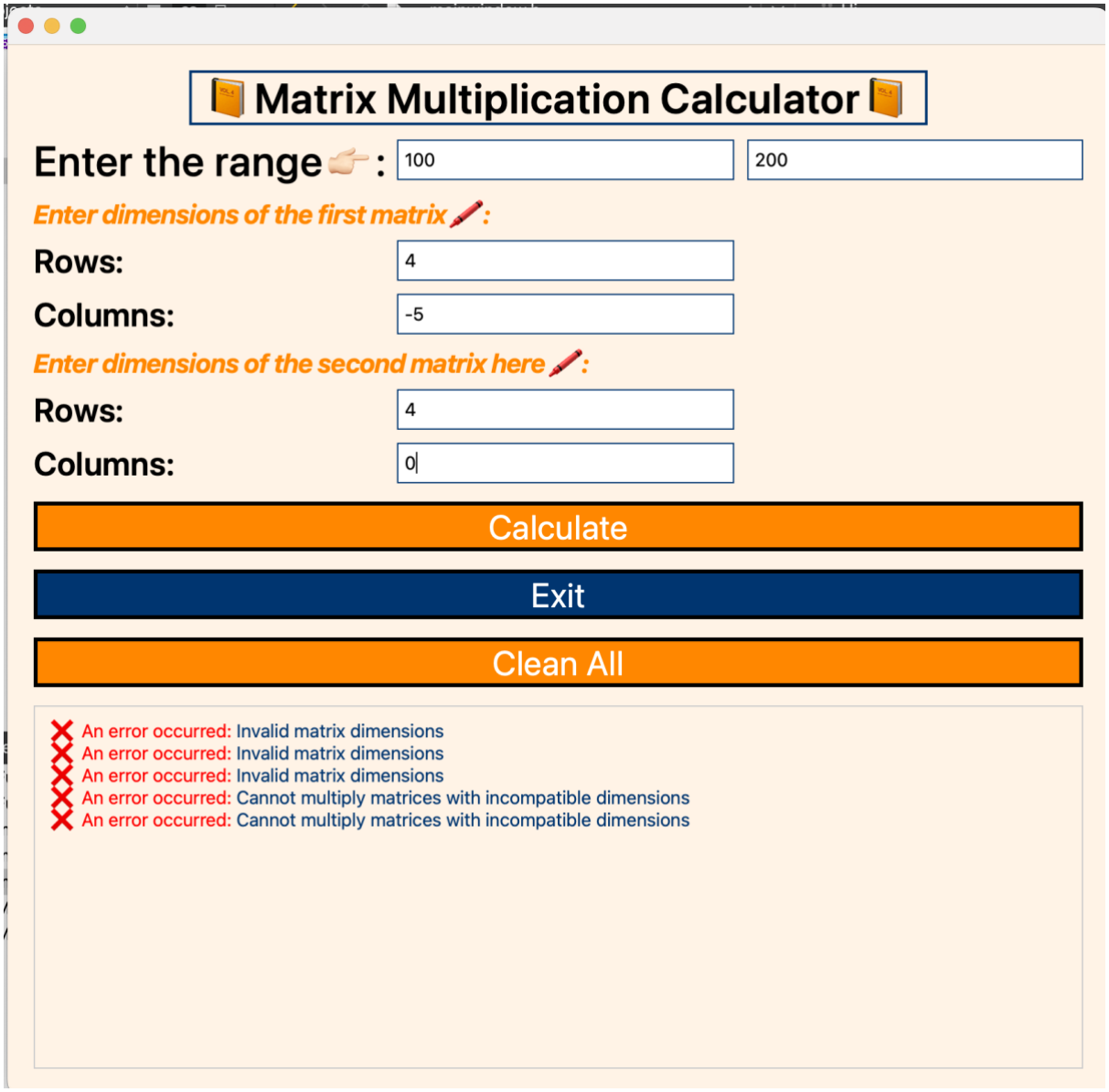


Множення матриць калькулятором

Решение:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{pmatrix} 2 & 4 \\ 4 & 6 \\ 6 & 7 \end{pmatrix} \cdot \begin{pmatrix} 12 & 421 & 5 \\ 1 & 1 & 112 \end{pmatrix} = \begin{pmatrix} 28 & 846 & 458 \\ 54 & 1690 & 692 \\ 79 & 2533 & 814 \end{pmatrix}$$

Викид exceptions



Лінк на репозиторій у GitHub:

https://github.com/Uliana200407/CppProjects-.git