**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Технічний коледж**

**НУ «Львівська політехніка»**

**Відділення**

**Інформаційних технологій**

**та комп’ютерної техніки**

**Звіт до:**

Лабораторної роботи № 4

з предмету «НП з ООП»

**Тема:**

«**Наслідування в С++. Віртуальні функції**»

**Підготував**

Студент групи – 31-ПЗ

Дячок Остап

**Львів 2021**

**Мета роботи:** Ознайомитись з поняттям та призначенням шаблонних функцій та шаблонних класів. Навчитися використовувати шаблони класів та шаблони функцій в мові С++.

**Постановка задачі:**

**Варіант 8**

Абстрактний базовий клас містить чисто віртуальні методи для отримання даних ії додавання, редагування та видалення. Два дочірні класи містять реалізацію чисто віртуальних функції для введення/виведення даних на консоль (перший клас) та у файл (другий клас).

Дана прямокутна матриця.

Визначити:

кількість рядків, які не містять жодного нульового елемента;

максимальне із чисел, що зустрічається в заданій матриці більше одного разу Перетворити матрицю таким чином, щоб всі елементи, рівні нулю, розташовувались на початку всіх інших.

**Код :**

[main.cpp](https://www.onlinegdb.com/online_c++_compiler#editor_1) :

#include <iostream>

#include <string>

#include <set>

#include <fstream>

using namespace std;

#include"rectangularMatrix.h"

#include"classChildOne.h"

#include"classChildTwo.h"

int

main ()

{

  classChildOne one;

  classChildTwo two;

  one.inputMatrix();

  one.printMatrix();

  one.rowsWithoutZeros();

  one.maxNumberWitchWasFindTwoTimes();

  one.taskX();

  two.inputMatrix();

  two.printMatrix();

}

Класс ActionsOnRectangularMatrix:

class actionsOnRectangularMatrix

{

protected:

  int row;

  int col;

  typedef set<int, greater<int>> MySet;

  string line = "----------------------------------";

  float \*\*matrix;

public:

  virtual void inputMatrix() {}

  virtual void rowsWithoutZeros()

  {

    int rowsWithZeros = 0;

    for (int i = 0; i < row; i++)

      for (int j = 0; j < col; j++)

      {

        if (matrix[i][j] == 0)

        {

          ++rowsWithZeros;

          break;

        }

      }

    cout << line << endl;

    cout << " \*Rows with out Zeros:";

    cout << (row - rowsWithZeros) << endl;

    cout << line << endl;

  }

  virtual void maxNumberWitchWasFindTwoTimes()

  {

    bool found = false;

    int rez = 0;

    MySet check;

    for (int i = 0; i < row; i++)

      for (int ii = 0; ii < col; ii++)

      {

        const int val = matrix[i][ii];

        if ((!found || val > rez) && !check.insert(val).second)

        {

          found = true;

          rez = val;

        }

      }

    if (!found)

    {

      cout << line << endl;

      cout << " \*No numbers in matrix that met more than once" << endl;

      cout << line << endl;

    }

    else

    {

      cout << line << endl;

      cout << " \*Biggest number is:" << rez << endl;

      cout << line << endl;

    }

  };

  virtual void taskX()

  {

    float \*matrixSimpleArray = new float[(row \* col)];

    int k = 0;

    for (int i = 0; i < row; i++)

      for (int j = 0; j < col; j++)

      {

        matrixSimpleArray[k] = matrix[i][j];

        k++;

      }

    k = 0;

    for (int i = ((row \* col) - 1); i >= 0; i--)

    {

      for (int j = 0; j <= i; j++)

      {

        if (matrixSimpleArray[j] == 0)

        {

          int t = matrixSimpleArray[i];

          matrixSimpleArray[i] = matrixSimpleArray[j];

          matrixSimpleArray[j] = t;

          break;

        }

      }

    }

    for (int i = 0; i < row; i++)

      for (int j = 0; j < col; j++)

      {

        matrix[i][j] = matrixSimpleArray[k];

        k++;

      }

    cout << line << endl;

    cout << " \*TaskX:" << endl;

    cout << line << endl;

    for (int i = 0; i < row; i++)

    {

      for (int j = 0; j < col; j++)

      {

        cout << matrix[i][j] << ' ';

      }

      cout << endl;

    }

    cout << endl;

  }

  virtual void printMatrix() {}

};

Клас classChildOne:

class classChildOne : public actionsOnRectangularMatrix

{

public:

    virtual void inputMatrix()

    {

        cout << " \*Creating a rectangular matrix:" << endl;

        cout << line << endl;

        cout << "Enter the number of rows:";

        cin >> row;

        cout << "Enter the number of columns:";

        cin >> col;

        cout << line << endl;

        if (row > col || col > row)

        {

            matrix = new float \*[row];

            for (int i = 0; i < row; i++)

                matrix[i] = new float[col];

            for (int i = 0; i < row; i++)

                for (int j = 0; j < col; j++)

                {

                    cout << "Enter element "

                         << "[" << i << "][" << j << "]  ";

                    cin >> matrix[i][j];

                }

        }

    }

    virtual void printMatrix()

    {

        cout << line << endl;

        cout << " \*Your matrix:" << endl;

        cout << line << endl;

        for (int i = 0; i < row; i++)

        {

            for (int j = 0; j < col; j++)

            {

                cout << matrix[i][j] << ' ';

            }

            cout << endl;

        }

        cout << endl;

    }

};

Клас classChildTwo:

class classChildTwo : public actionsOnRectangularMatrix

{

public:

    virtual void inputMatrix()

    {

        ifstream in("./input.txt");

        if (in.is\_open())

        {

            int count = 0;

            int temp;

            while (!in.eof())

            {

                in >> temp;

                count++;

            }

            in.seekg(0, ios::beg);

            in.clear();

            int count\_space = 0;

            char symbol;

            while (!in.eof())

            {

                in.get(symbol);

                if (symbol == ' ')

                    count\_space++;

                if (symbol == '\n')

                    break;

            }

            in.seekg(0, ios::beg);

            in.clear();

            row = count / (count\_space + 1);

            col = count\_space + 1;

            matrix = new float \*[row];

            for (int i = 0; i < row; i++)

                matrix[i] = new float[col];

            for (int i = 0; i < row; i++)

                for (int j = 0; j < col; j++)

                    in >> matrix[i][j];

            in.close();

        }

        else

        {

            cout << "File not found.";

        }

    }

    virtual void printMatrix()

    {

        ofstream out;

        out.open("./output.txt");

        if (out.is\_open())

        {

            out << line << endl;

            out << " \*Your matrix:" << endl;

            out << line << endl;

            for (int i = 0; i < row; i++)

            {

                for (int j = 0; j < col; j++)

                {

                    out << matrix[i][j] << ' ';

                }

                out << endl;

            }

            out << endl;

        }

        else

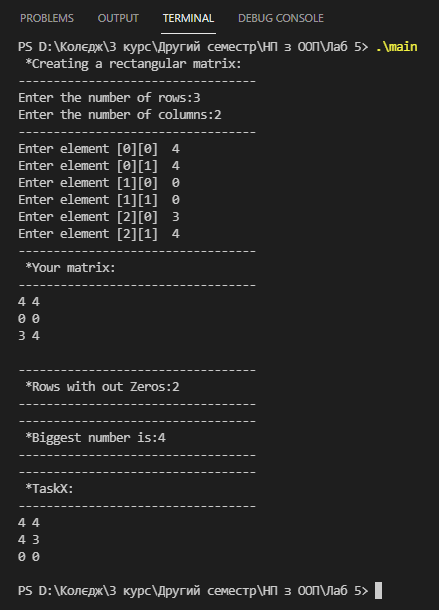
        {

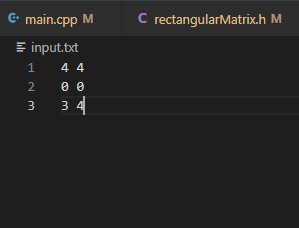
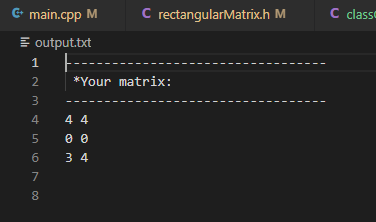
            cout << "File not found.";

        }

    }

};



**Висновок:**

Завдяки виконанню лабораторної роботи 5 ознайомився з поняттям та призначенням шаблонних функцій та шаблонних класів. Навчитися використовувати шаблони класів та шаблони функцій в мові С++.