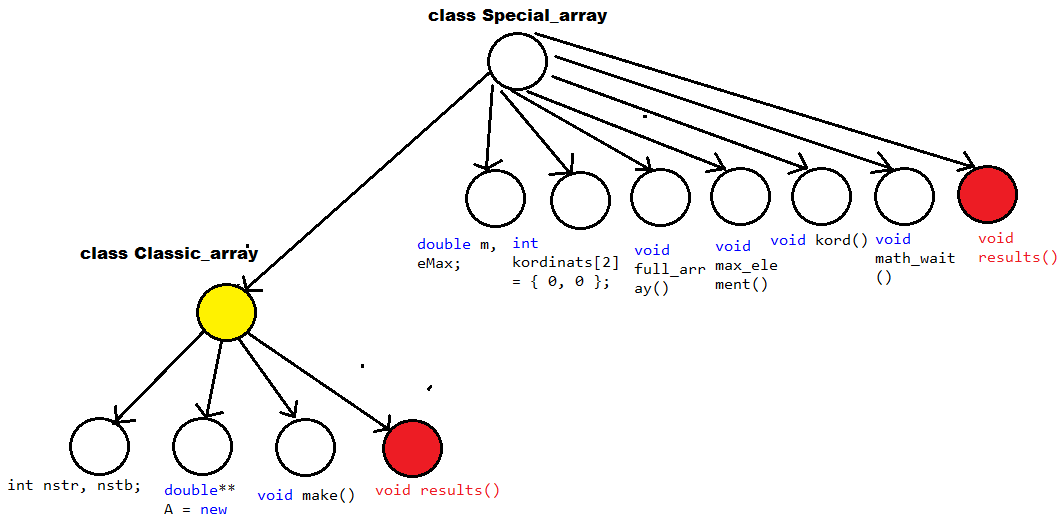
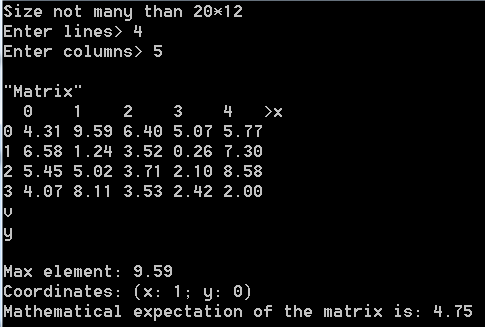
Задача:

Подсчитать математическое ожидание для элементов матрицы, принадлежащих заданному интервалу и запомнить координаты максимального из них. Размерность матрицы не более, чем 20\*12. Формат числа ±цц.ц.



Вывод:



Код:

#include <cstdlib>

#include <iostream>

#include <ctime>

using namespace std;

class Classic\_array {

public:

int nstr, nstb;

double\*\* A = new double\* [nstr];

void make() {

if (!((nstr > 1 and nstr <= 20) and (nstb > 1 and nstb <= 12))) {

cout << "The array wasnt created" << endl;

exit(0);

}

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

A[i] = new double[nstb];

}

virtual void results() {

cout << "The array was created" << endl;

}

};

class Special\_array : public Classic\_array {

public:

double m, eMax;

int kordinats[2] = { 0, 0 };

void full\_array() {

cout << "\"Matrix\"\n";

for (int x = 0; x < nstb; x++) {

cout << " " << x << " ";

for (int x = 0; x < 1; x++) {

cout << " ";

}

}

cout << " >x\n";

for (int i = 0; i < nstr; i++) {

cout << i;

for (int j = 0; j < nstb; j++) {

A[i][j] = 1. \* 10 \* rand() / RAND\_MAX;

cout << " ";

cout << fixed;

cout.precision(2);

cout << A[i][j];

}

cout << "\n";

}

cout << "v\ny\n\n";

}

void max\_element() {

eMax = -1.0;

for (int i = 0; i < nstr; i++) {

for (int j = 0; j < nstb; j++) {

if (A[i][j] > eMax) {

eMax = A[i][j];

}

}

}

}

void kord() {

for (int i = 0; i < nstr; i++) {

for (int j = 0; j < nstb; j++) {

if (eMax == A[i][j]) {

kordinats[0] = i;

kordinats[1] = j;

}

}

}

}

void math\_wait() {

m = 0.0;

for (int i = 0; i < nstr; i++) {

for (int j = 0; j < nstb; j++) {

m = m + A[i][j];

}

}

m = m / (nstr \* nstb);

}

void results() {

if (eMax == -1) cout << "There is no largest element!\n";

else {

cout << "Max element: " << eMax << "\n";

cout << "Coordinates: (x: " << kordinats[1] << "; y: " << kordinats[0] << ")\n";

}

cout << "Mathematical expectation of the matrix is: " << m << "\n";

}

};

int main() {

int a, b;

srand(time(NULL));

cout << "Size not many than 20\*12 \n";

cout << "Enter lines> "; cin >> a;

cout << "Enter columns> "; cin >> b;

cout << "\n";

Special\_array my\_array;

Classic\_array\* link\_array = &my\_array;

// тут хранится ссылка на объект класса Special\_array

// полиморфизм предоставляет доступ к методу дочернего класса через указатель

my\_array.nstr = a;

my\_array.nstb = b;

my\_array.make();

my\_array.full\_array();

my\_array.max\_element();

my\_array.kord();

my\_array.math\_wait();

link\_array->results();

// выполняется вызов метода базового класса results

// который благодаря виртуальному классу выводит результат дочернего

int strk;

cin >> strk;

return 0;

}