

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи № 3 з дисципліни
«Основи програмування 2. Модульне програмування»

„Класи та об’єкти”

Виконав

ІІІ-15 Левченко Владислав В’ячеславович
(шифр, прізвище, ім'я, по батькові)

Перевірила

Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Завдання

19. Розробити клас "квадратна матриця" вказаної розмірності. Створити масив об'єктів даного класу. Визначити матрицю з найбільшою сумою діагональних елементів (найбільшим слідом).

Код c++

class.h

```
c++ final 3 lab
1  #pragma once
2
3  #include <iostream>
4  #include <iomanip>
5  #include <cstdlib>
6
7  class Matrix {
8  public:
9      Matrix(int);
10     ~Matrix();
11     Matrix(const Matrix&);
12     void fillHandM();
13     void fillRandM();
14     void printMatrix();
15     float findDiagSum();
16 private:
17     int size;
18     double** m;
19     void allowSpace();
20 };
```

class.cpp

```

1  #include "class.h"
2
3  Matrix::Matrix(int Size) : size(Size)
4  {
5      allowSpace();
6  }
7
8  Matrix::~Matrix()
9  {
10     for (int i = 0; i < size; ++i) {
11         delete[] m[i];
12     }
13     delete[] m;
14 }
15
16 Matrix::Matrix(const Matrix& p) : size(p.size)
17 {
18     allowSpace();
19     for (int i = 0; i < size; ++i) {
20         for (int j = 0; j < size; ++j) {
21             m[i][j] = p.m[i][j];
22         }
23     }
24 }
25
26 void Matrix::allowSpace()
27 {
28     m = new double* [size];
29     for (int i = 0; i < size; ++i) {
30         m[i] = new double[size];
31     }
32 }
33

```

```

c++ final 3 lab -> Matrix
34 void Matrix::fillRandM() {
35     for (int row = 0; row < size; row++)
36     {
37         for (int col = 0; col < size; col++)
38         {
39             m[row][col] = ((float(rand()) / float(RAND_MAX)) * 100 - 50);
40         }
41     }
42 }
43
44 void Matrix::fillHandM() {
45     float n;
46     std::cout << "Enter the values of matrix: ";
47
48     for (int row = 0; row < size; row++)
49     {
50         for (int col = 0; col < size; col++)
51         {
52             std::cin >> n;
53             m[row][col] = n;
54         }
55     }
56 }
57
58 void Matrix::printMatrix() {
59     for (int row = 0; row < size; row++)
60     {
61         std::cout << "\n";
62         for (int col = 0; col < size; col++)
63         {
64             std::cout << std::setw(7) << std::fixed << std::setprecision(1) << m[row][col];
65         }
66         std::cout << "\n";
67     }
68 }

```

```

69
70 float Matrix::findDiagSum() {
71     float acc = 0;
72
73     for (int i = 0; i < size; i++)
74     {
75         acc += m[i][i];
76     }
77     return acc;
78 }
79
80
81
82

```

functions.h

```
c++ final 3 lab
1  #pragma once
2
3  #include "class.h"
4  #include <vector>
5  #include <string>
6  #include <iomanip>
7
8  int getCountMatrix();
9  int chooseM(int);
10 int getSize();
11 bool isNumber(const std::string&);
12 Matrix GenerateMatrix(int, int);
13 std::vector<Matrix> GenerateMatrixArray();
14 void showMatrix(std::vector<Matrix>);
15 void findMaxDiagSum(std::vector<Matrix>);
16 void Info();
```

functions.cpp

```

35
36 Matrix GenerateMatrix(int size, int i) {
37     int mode = chooseM(i);
38     Matrix matrix(size);
39     if (mode == 1) { matrix.fillHandM(); }
40     else { matrix.fillRandM(); }
41     return matrix;
42 }
43
44 std::vector<Matrix> GenerateMatrixArray() {
45     srand(time(NULL));
46     std::vector<Matrix> matrixes;
47
48     int amount = getCountMatrix();
49     int size = getSize();
50     for (int i = 0; i < amount; i++)
51     {
52         Matrix matrix = GenerateMatrix(size, i);
53         matrixes.push_back(matrix);
54     }
55     return matrixes;
56 }
57
58 void showMatrix(std::vector<Matrix> matrixes) {
59     std::cout << "\nInfo:\n";
60
61     int numb = 1;
62     for (int i = 0; i < matrixes.size(); i++) {
63         std::cout << "\n" << numb << " ) ";
64         numb++;
65         matrixes[i].printMatrix();
66     }
67 }
68

```

```

c++ final 3 lab (Глобальная область)
1 #include "functions.h"
2
3 int getCountMatrix() {
4     std::string num;
5     std::cout << "Enter the number of matrix (1 or more): ";
6     std::cin >> num;
7     while (!isNumber(num) || stoi(num) < 1) {
8         std::cout << "Enter correct number: ";
9         std::cin >> num;
10    }
11    return stoi(num);
12 }
13
14 int chooseM(int it) {
15     std::string n;
16     std::cout << "\nEnter the mode for values of matrix #" << it + 1 << " (1 or 2): ";
17     std::cin >> n;
18     while (!isNumber(n) || stoi(n) != 1 && stoi(n) != 2) {
19         std::cout << "Enter correct number: ";
20         std::cin >> n;
21     }
22     return stoi(n);
23 }
24
25 int getSize() {
26     std::string s;
27     std::cout << "Enter the size of matrix (1 or more): ";
28     std::cin >> s;
29     while (!isNumber(s) || stoi(s) < 1) {
30         std::cout << "Enter correct number: ";
31         std::cin >> s;
32     }
33     return stoi(s);
34 }
35

```

```
c++ final 3 lab (Глобальная область)

58 void showMatrix(std::vector<Matrix> matrixes) {
59     std::cout << "\nInfo:\n";
60
61     int numb = 1;
62     for (int i = 0; i < matrixes.size(); i++) {
63         std::cout << "\n" << numb << " " << " ";
64         numb++;
65         matrixes[i].printMatrix();
66     }
67 }

68
69 void findMaxDiagSum(std::vector<Matrix> matrixes) {
70     float max = matrixes[0].findDiagSum();
71     float acc = 0;
72     int ind = 0; std::cout << "\n";
73
74     for (int i = 0; i < matrixes.size(); i++) {
75         acc = matrixes[i].findDiagSum();
76         std::cout << "Sum of diagonal elements of " << i + 1 << " matrix = " << acc << "\n";
77         if (max < acc) { max = acc; ind = i; }
78     }
79     std::cout << "\nThe matrix with the biggest step: #" << ind + 1 << ":";
80     matrixes[ind].printMatrix();
81     std::cin.ignore(1);
82 }

83
84 bool isNumber(const std::string & word) {
85     for (char i : word) {
86         if (!isdigit(i)) return false;
87     }
88     return true;
89 }

90
91 void Info() {
92     std::cout << "Modes of entering the values\n1 - self fill\n2 - random fill\n";
93     std::cout << "_____ \n\n";
94 }
```

main.cpp

```
c++ final 3 lab

1  #include <iostream>
2  #include "class.h"
3  #include "functions.h"
4
5  int main()
6  {
7      std::vector<Matrix> matrixes;
8
9      Info();
10     matrixes = GenerateMatrixArray();
11     showMatrix(matrixes);
12     findMaxDiagSum(matrixes);
13     std::cin.ignore(1);
14 }
```

Випробування:

C:\Users\Админ\source\repos\c++ final 3 lab\Debug\c++ final 3 lab.exe

```
Modes of entering the values
1 - self fill
2 - random fill

Enter the number of matrix (1 or more): 3
Enter the size of matrix (1 or more): 2

Enter the mode for values of matrix #1 (1 or 2): 2

Enter the mode for values of matrix #2 (1 or 2): 1
Enter the values of matrix: 1.3 15 -34 -7.2

Enter the mode for values of matrix #3 (1 or 2): 2

Info:

1)
  -18.4   29.9
    29.3    6.8

2)
    1.3   15.0
  -34.0   -7.2

3)
  -29.6  -29.5
    11.7  -28.7
```

C:\Users\Админ\source\repos\c++ final 3 lab\Debug\c++ final 3 lab.exe

```
Info:

1)
  -18.4   29.9
    29.3    6.8

2)
    1.3   15.0
  -34.0   -7.2

3)
  -29.6  -29.5
    11.7  -28.7

Sum of diagonal elements of 1 matrix = -11.6
Sum of diagonal elements of 2 matrix = -5.9
Sum of diagonal elements of 3 matrix = -58.3

The matrix with the biggest step: #2:
    1.3   15.0
  -34.0   -7.2
```


Висновки:

Під час виконання третьої лабораторної роботи я навчився працювати з класами та об'єктами. Точніше я навчився створювати клас та його атрибути, створювати конструктори, методи класу і звісно самі ж об'єкти. На прикладі квадратної матриці можна побачити роботу класів і програму, яка створює масив об'єктів, задає їм значення і знаходить серед них матрицю з найбільшим кроком.

