

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 № 4 з дисципліни
«Основи програмування 2. Модульне програмування»

„Перевантаження операторів”

Виконав

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

Перевірила

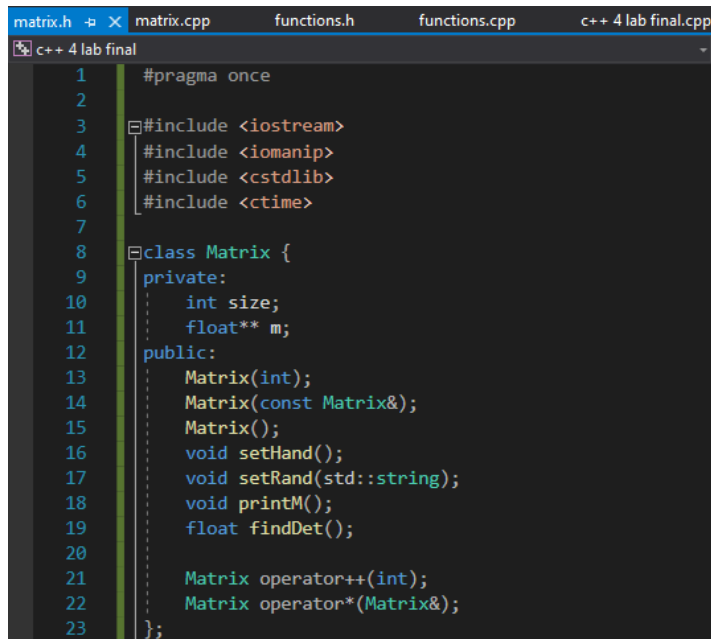
Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

19. Визначити клас "Квадратна матриця 3×3". Реалізувати для нього декілька конструкторів, геттери, метод обчислення визначника матриці. Перевантажити оператори множення "*" матриць та інкрементації її елементів "++". Створити три матриці (M1, M2, M3), використовуючи різні конструктори. Визначити матрицю M3 як добуток матриць M1 та M2. Інкрементувати елементи отриманої матриці M3. Знайти визначник зміненої матриці M3.

Варіант 19 Код c++:

matrix.h



```
1  #pragma once
2
3  #include <iostream>
4  #include <iomanip>
5  #include <cstdlib>
6  #include <ctime>
7
8  class Matrix {
9  private:
10     int size;
11     float** m;
12 public:
13     Matrix(int);
14     Matrix(const Matrix&);
15     Matrix();
16     void setHand();
17     void setRand(std::string);
18     void printM();
19     float findDet();
20
21     Matrix operator++(int);
22     Matrix operator*(Matrix&);
23 };;
```

matrix.cpp

```
matrix.h  matrix.cpp  functions.h  functions.cpp  c++ 4 lab final.
c++ 4 lab final

1      #include "matrix.h"
2
3      Matrix::Matrix(int Size) : size(Size = 3)
4      {
5          m = new float* [size];
6          for (int i = 0; i < size; i++) {
7              m[i] = new float[size];
8              for (int j = 0; j < size; j++)
9              {
10                 m[i][j] = 0;
11             }
12         }
13     }
14
15     Matrix::Matrix() : size(3)
16     {
17         m = new float* [size];
18         for (int i = 0; i < size; i++) {
19             m[i] = new float[size];
20             for (int j = 0; j < size; j++)
21             {
22                 m[i][j] = 0;
23             }
24         }
25     }
26
27     Matrix::Matrix(const Matrix& p)
28     {
29         size = 3;
30         m = new float* [size];
31         for (int i = 0; i < size; i++) {
32             m[i] = new float[size];
33             for (int j = 0; j < size; j++) {
34                 m[i][j] = p.m[i][j];
35             }
36         }
```

```
matrix.h  matrix.cpp  functions.h  functions.cpp  c++ 4 lab final.cpp
c++ 4 lab final  -> Matrix

38
39     void Matrix::setRand(std::string ans) {
40         float number;
41         for (int i = 0; i < size; i++)
42         {
43             for (int j = 0; j < size; j++)
44             {
45                 number = ((float(rand()) / float((RAND_MAX))) * 20 - 10);
46                 if(ans == "y" || ans == "Y")
47                     number = round(number * 1) / 1;
48                 else{ number = round(number * 10) / 10; }
49                 m[i][j] = number;
50             }
51         }
52     }
53
54     void Matrix::printM()
55     {
56         std::cin.ignore();
57         for (int i = 0; i < size; i++)
58         {
59             for (int j = 0; j < size; j++)
60             {
61                 std::cout << std::setw(9) << m[i][j];
62             }
63             std::cout << "\n";
64         }
65         std::cout << "\n";
66     }
```

```
matrix.h  matrix.cpp  functions.h  functions.cpp  c++ 4 lab final.cpp
c++ 4 lab final

68  float Matrix::findDet()
69  {
70      float det = 0;
71      float a = m[0][0] * m[1][1] * m[2][2];
72      float b = m[0][1] * m[1][2] * m[2][0];
73      float c = m[0][2] * m[1][0] * m[2][1];
74      float d = m[2][0] * m[1][1] * m[0][2];
75      float e = m[2][1] * m[1][2] * m[0][0];
76      float f = m[2][2] * m[1][0] * m[0][1];
77      det = a + b + c - d - e - f;
78      return det;
79  }
80
81  Matrix Matrix::operator++(int)
82  {
83      std::cin.ignore();
84      for (int i = 0; i < size; i++)
85      {
86          for (int j = 0; j < size; j++)
87          {
88              m[i][j]++;
89          }
90      }
91      return *this;
92  }
93
```

```

94  Matrix Matrix::operator*(Matrix& M2)
95  {
96      Matrix temp{};
97      for (int i = 0; i < size; i++) {
98          for (int j = 0; j < size; j++) {
99              for (int k = 0; k < size; k++) {
100                 temp.m[i][j] += (m[i][k] * M2.m[k][j]);
101                 temp.m[i][j] = round(temp.m[i][j] * 10) / 10;
102             }
103         }
104     }
105     return temp;
106 }
107
108 void Matrix::setHand() {
109     float n;
110     std::cout << "Enter the values of matrix: ";
111
112     for (int row = 0; row < size; row++)
113     {
114         for (int col = 0; col < size; col++)
115         {
116             std::cin >> n;
117             m[row][col] = n;
118         }
119     }
120 }

```

functions.h

```

matrix.h  matrix.cpp  functions.h*  functions.cpp
c++ 4 lab final
1  #pragma once
2
3  #include "matrix.h"
4  #include <string>
5  #include <time.h>
6
7  int chooseM(int);
8  bool isNumber(const std::string&);
9  Matrix GenerateMatrix(int, Matrix&);
10 void Info();
11 std::string onlyInt(int);

```

functions.cpp

```
matrix.h    matrix.cpp    functions.h*    functions.cpp  c++ 4 lab final.cpp
c++ 4 lab final    (Глобальная область)

1      #include "functions.h"
2
3      int chooseM(int it) {
4          std::string n;
5          std::cout << "\nEnter the mode for values of matrix #" << it << " (1 or 2): ";
6          std::cin >> n;
7          while (!isNumber(n) || stoi(n) != 1 && stoi(n) != 2) {
8              std::cout << "Enter correct number: ";
9              std::cin >> n;
10         }
11         return stoi(n);
12     }
13
14     Matrix GenerateMatrix(int i, Matrix& matrix) {
15         int mode = chooseM(i);
16         if (mode == 2) {
17             std::string ans = onlyInt(i);
18             matrix.setRand(ans);
19         }
20         else { matrix.setHand(); }
21         return matrix;
22     }
23
24     bool isNumber(const std::string& word) {
25         for (char i : word) {
26             if (!isdigit(i)) return false;
27         }
28         return true;
29     }
30
31     void Info() {
32         std::cout << "Modes of entering the values\n1 - self fill\n2 - random fill\n";
33         std::cout << "_____ \n\n";
34     }
35
36     std::string onlyInt(int it)
37     {
38         std::string ans;
39         std::cout << "Create only int values for matrix #" << it << " ? (y or n): ";
40         std::cin >> ans;
41         while (ans != "y" && ans != "Y" && ans != "N" && ans != "n") {
42             std::cout << "Enter correct answer: ";
43             std::cin >> ans;
44         }
45         return ans;
46     }
47
```

main.cpp

```
matrix.h    matrix.cpp    functions.h*    functions.cpp    c++ 4
c++ 4 lab final

1      #include "functions.h"
2
3      int main()
4      {
5          srand(time(NULL));
6          Info();
7
8          Matrix M1; GenerateMatrix(1, M1);
9          Matrix M2(3); GenerateMatrix(2, M2);
10         Matrix M3(M1);
11
12         std::cout << "\n1)\n "; M1.printM();
13         std::cout << "2)"; M2.printM();
14         std::cout << "3)"; M3.printM();
15
16         std::cout << "\nM3 = M1 * M2:\n";
17         M3 = M1 * M2;
18         M3.printM();
19
20         std::cout << "Increment M3:";
21         M3++;
22         M3.printM();
23
24         float det = M3.findDet();
25         std::cout << "Determinant M3 = "<<det;
26         std::cin.ignore(1);
27     }
```

Випробування:

Modes of entering the values

- 1 - self fill
 - 2 - random fill
-

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

Create only int values for matrix #1 ? (y or n): y

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

Create only int values for matrix #2 ? (y or n): y

1)

-4	-8	-4
-1	6	2
8	8	-0

2)

-4	0	-3
-7	-4	-6
-1	-6	-0

3)

-4	-8	-4
-1	6	2
8	8	-0

M3 = M1 * M2:

76	56	60
-40	-36	-33
-88	-32	-72

Increment M3:

77	57	61
-39	-35	-32
-87	-31	-71

Determinant M3 = 3820

A

$$\begin{pmatrix} -4 & -8 & -4 \\ -1 & 6 & 2 \\ 8 & 8 & 0 \end{pmatrix}$$

Розмір: 3 × 3

2

2

↔
A + B
A - B
A × B

B

$$\begin{pmatrix} -4 & 0 & -3 \\ -7 & -4 & -6 \\ -1 & -6 & 0 \end{pmatrix}$$

Розмір: 3 × 3

2

2

Вводити можна лише числа або дробі (-2.4, 5/7, ...). Більш детально читайте в [правилах вводу чисел](#).

Розв'язок:

$$C = A \cdot B = \begin{pmatrix} -4 & -8 & -4 \\ -1 & 6 & 2 \\ 8 & 8 & 0 \end{pmatrix} \cdot \begin{pmatrix} -4 & 0 & -3 \\ -7 & -4 & -6 \\ -1 & -6 & 0 \end{pmatrix} = \begin{pmatrix} 76 & 56 & 60 \\ -40 & -36 & -33 \\ -88 & -32 & -72 \end{pmatrix}$$

A

$$\begin{pmatrix} 77 & 57 & 61 \\ -39 & -35 & -32 \\ -87 & -31 & -71 \end{pmatrix}$$

Очистити

Транспонувати

Знайти визначник

Знайти ранг

Розмір: 3 × 3

Помножити на
2

Піднести до степеня
2

Обернена матриця: A⁻¹

Вводити можна лише числа або дробі (-2.4, 5/7, ...). Більш детально читайте в [правилах вводу чисел](#).

Розв'язок:

[Скористаємося формулою для обрахунку визначника матриці 3×3:](#)

$$\det A = \begin{vmatrix} 77 & 57 & 61 \\ -39 & -35 & -32 \\ -87 & -31 & -71 \end{vmatrix} = 77 \cdot (-35) \cdot (-71) + 57 \cdot (-32) \cdot (-87) + 61 \cdot (-39) \cdot (-31) - 61 \cdot (-35) \cdot (-87) - 77 \cdot (-32) \cdot$$

$$(-31) - 57 \cdot (-39) \cdot (-71) = 191345 + 158688 + 73749 - 185745 - 76384 - 157833 = 3820$$

Очистити

Висновки: під час виконання даної лабораторної роботи, я навчився механізмам створення класів та використання різних конструкторів з використанням перевантажених операторів. Я створив три різні об'єкти класу (різними конструкторами) M1, M2, M3. M3 я визначив як добуток матриць M1 та M2 (за допомогою перевантаженого оператора множення). Після цього інкрементував M3 (за допомогою перевантаженого оператора інкрементування) та знайшов визначник отриманої матриці.