

Цель:

формирование практических навыков применения систем контроля версий и сервисов хостинга проектов для организации коллективной работы.

Задачи:

1. Ознакомиться с понятием, назначением, порядком и общими правилами работы с системой контроля версий;
2. Продемонстрировать командную работу при разработке программного продукта;
3. Написать программный код для индивидуального варианта.

Вариант 4

Квадратную матрицу 17x17 заполнить случайными числами из диапазона [-10.. 10], вывести ее на экран. а) Найти наибольшие элементы в столбцах матрицы с чётными номерами. б) Проверить, совпадают ли главная и побочная диагональ поэлементно. в) Найдите минимальный элемент среди элементов матрицы, выделенных чёрным цветом (матрица квадратная).

Листинг программы:

Main.cpp:

```
1.  #include "../include/matrix.h"
2.
3.  using namespace std;
4.
5.  const int MENU_LEN {5};
6.
7.  void print_menu() {
8.      system("clear");
9.      cout << "Выберите опцию" << endl;
10.     cout << "1. Пересоздать матрицу" << endl;
11.     cout << "2. Вывести матрицу" << endl;
12.     cout << "3. Проверить, равны ли главная и побочная диагонали" << endl;
13.     cout << "4. Вывести минимальный элемент в чётных столбцах" << endl;
14.     cout << "5. Вывести минимальный элемент под главной диагональю" << endl;
15.     cout << "0. Выход" << endl << endl;
16. }
17.
18.
19. int get_value()
20. {
21.     int value{-1};
22.     while (not (cin >> value) or (value < 0) or (value > MENU_LEN))
23.     {
24.         cin.clear();
25.         cin.ignore();
26.         cout << "Вы ввели неверное значение!" << endl;
27.     }
28.     return value;
29. }
```

```

30.
31. int main() {
32.
33.     std::array<std::array<int, 17>, 17> arr{};
34.     Matrix matrix;
35.     int value{1};
36.     while (value) {
37.
38.         print_menu();
39.         value = get_value();
40.
41.         switch (value) {
42.             case 1:
43.                 matrix.generate_array();
44.                 break;
45.             case 2:
46.                 matrix.print_array();
47.                 break;
48.             case 3:
49.                 cout << (matrix.are_main_and_secondary_diagonals_equal() ? "Совпадают": "Не
совпадают") << endl;
50.                 break;
51.             case 4:
52.                 cout << matrix.get_max_elem_in_even_columns() << endl;
53.                 break;
54.             case 5:
55.                 cout << matrix.get_min_element_beneath_main_diagonal() << endl;
56.                 break;
57.             default:
58.                 break;
59.         }
60.         if (value) {
61.             system("read -rsp '$Press enter to continue...\\n'");
62.         }
63.     }
64.     return 0;
65. }
66.

```

Matrix.h:

```

1. #ifndef FUNDAMENTALS_OF_SOFTWARE_ENGINEERING_MARTIX_H
2. #define FUNDAMENTALS_OF_SOFTWARE_ENGINEERING_MARTIX_H
3.
4. #include <iostream>
5. #include <iomanip>
6. #include <chrono>
7. #include <random>
8. #include <array>
9.
10. class Matrix {
11.     const std::pair<int, int> boundaries{-10, 10};
12.     static const int LEN{17};
13.     std::array<std::array<int, LEN>, LEN> array;
14.
15. public:
16.     Matrix();
17.
18.     /// Fills a 17x17 two-dimensional array with random numbers.
19.     void generate_array();
20.
21.     /// Prints a 17x17 two-dimensional array with even spacing based on the max element contained in
each line
22.     void print_array();
23.
24.
25.     /// Returns the maximum elements contained in each line of a 2D array
26.     int get_max_elem_in_even_columns();
27.
28.     /// Checks if elements on main and secondary diagonals are equal respectively

```

```

29. /// \returns bool - result of comparison
30.     bool are_main_and_secondary_diagonals_equal();
31.
32. /// Returns the minimum element contained beneath the main diagonal of a 2D array
33. /// \returns int - the minimum element
34.     bool get_min_element_beneath_main_diagonal();
35.
36. };
37.
38. #endif //FUNDAMENTALS_OF_SOFTWARE_ENGINEERING_MARTIX_H
39.

```

Matrix.cpp

```

1. #include "../include/martix.h"
2.
3. using namespace std;
4.
5. const int MENU_LEN {5};
6.
7. void print_menu() {
8.     system("clear");
9.     cout << "Выберите опцию" << endl;
10.    cout << "1. Пересоздать матрицу" << endl;
11.    cout << "2. Вывести матрицу" << endl;
12.    cout << "3. Проверить, равны ли главная и побочная диагонали" << endl;
13.    cout << "4. Вывести минимальный элемент в чётных столбцах" << endl;
14.    cout << "5. Вывести минимальный элемент под главной диагональю" << endl;
15.    cout << "0. Выход" << endl << endl;
16. }
17.
18.
19. int get_value()
20. {
21.     int value{-1};
22.     while (not (cin >> value) or (value < 0) or (value > MENU_LEN))
23.     {
24.         cin.clear();
25.         cin.ignore();
26.         cout << "Вы ввели неверное значение!" << endl;
27.     }
28.     return value;
29. }
30.
31. int main() {
32.
33.     std::array<std::array<int, 17>, 17> arr{};
34.     Matrix matrix;
35.     int value{1};
36.     while (value) {
37.
38.         print_menu();
39.         value = get_value();
40.
41.         switch (value) {
42.             case 1:
43.                 matrix.generate_array();
44.                 break;
45.             case 2:
46.                 matrix.print_array();
47.                 break;
48.             case 3:
49.                 cout << (matrix.are_main_and_secondary_diagonals_equal() ? "Совпадают": "Не
совпадают") << endl;
50.                 break;
51.             case 4:
52.                 cout << matrix.get_max_elem_in_even_columns() << endl;
53.                 break;
54.             case 5:
55.                 cout << matrix.get_min_element_beneath_main_diagonal() << endl;
56.                 break;

```

```

57.         default:
58.             break;
59.     }
60.     if (value) {
61.         system("read -rsp '$Press enter to continue...\\n'");
62.     }
63. }
64. return 0;
65. }
66.

```

Результат выполнения:

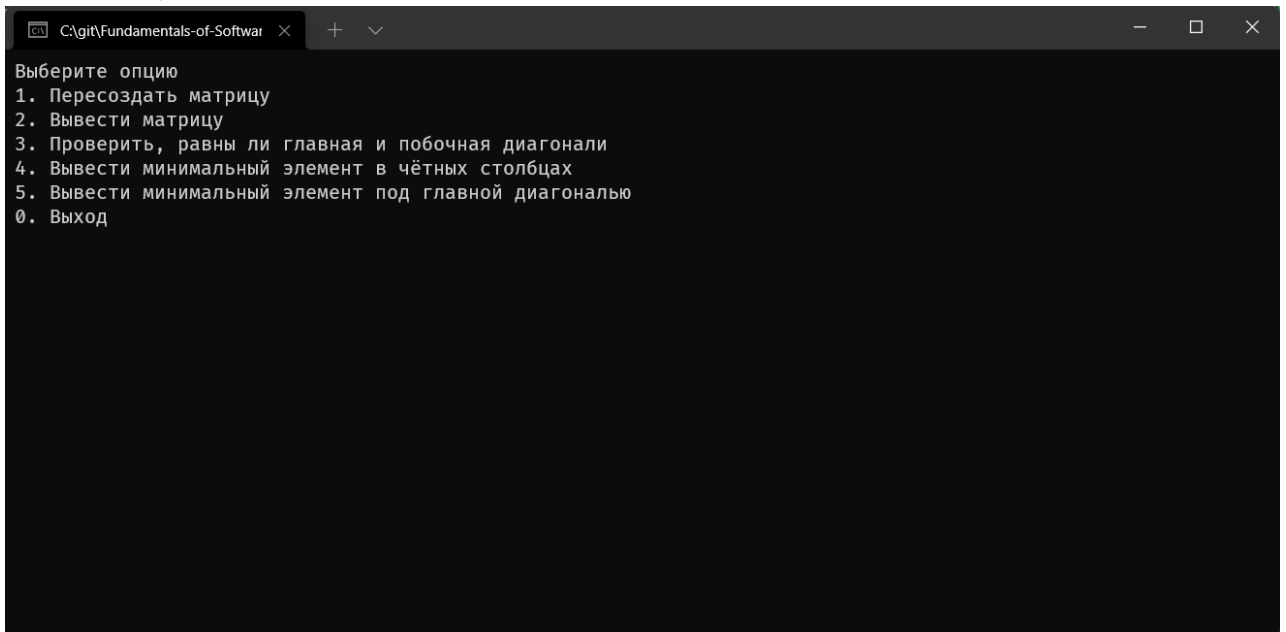


Рис. 1. Пользовательское меню программы

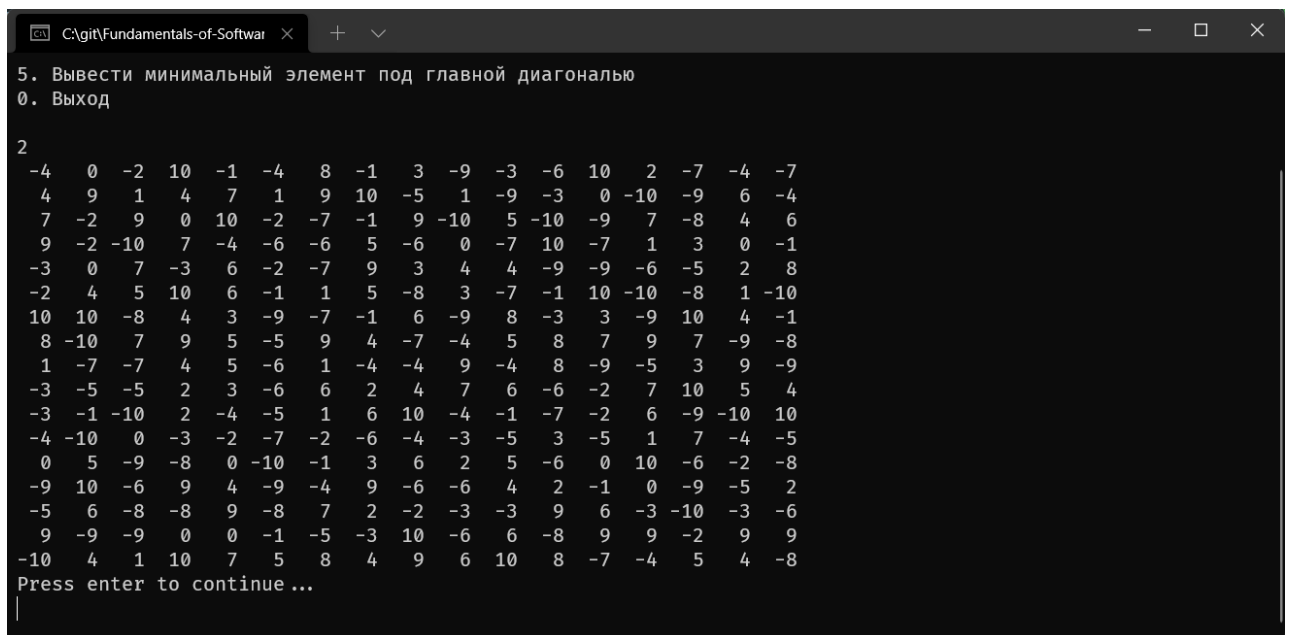


Рис. 2. Результат выполнения функции print_array

```
C:\git\Fundamentals-of-Softwar x + v - □ x
Выберите опцию
1. Пересоздать матрицу
2. Вывести матрицу
3. Проверить, равны ли главная и побочная диагонали
4. Вывести минимальный элемент в чётных столбцах
5. Вывести минимальный элемент под главной диагональю
0. Выход

3
Не совпадают
Press enter to continue...
|
```

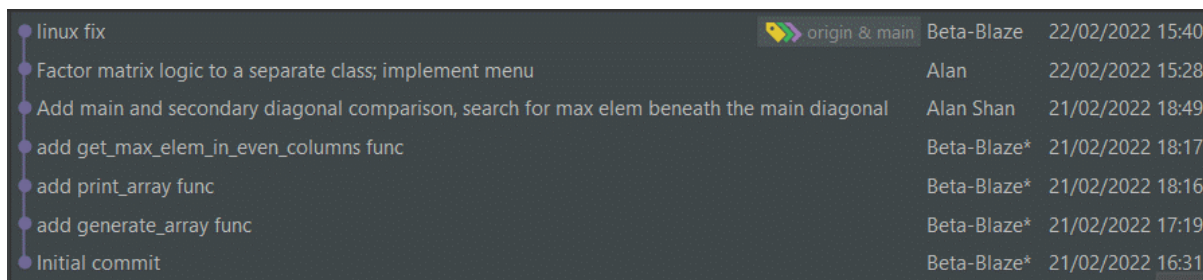
Рис. 3. Результат выполнения функции `are_main_and_secondary_diagonals_equal`

```
C:\git\Fundamentals-of-Softwar x + v - □ x
Выберите опцию
1. Пересоздать матрицу
2. Вывести матрицу
3. Проверить, равны ли главная и побочная диагонали
4. Вывести минимальный элемент в чётных столбцах
5. Вывести минимальный элемент под главной диагональю
0. Выход

4
10
Press enter to continue...
|
```

Рис. 4. Результат выполнения функции `get_max_elem_in_even_columns`

Рис. 5. Результат выполнения функции `get_min_element_beneath_main_diagonal`



Commit Message	Author	Date
linux fix	origin & main Beta-Blaze	22/02/2022 15:40
Factor matrix logic to a separate class; implement menu	Alan	22/02/2022 15:28
Add main and secondary diagonal comparison, search for max elem beneath the main diagonal	Alan Shan	21/02/2022 18:49
add get_max_elem_in_even_columns func	Beta-Blaze*	21/02/2022 18:17
add print_array func	Beta-Blaze*	21/02/2022 18:16
add generate_array func	Beta-Blaze*	21/02/2022 17:19
Initial commit	Beta-Blaze*	21/02/2022 16:31

Рис. 6. Окно истории изменений разрабатываемой программы

Вывод: в результате выполнения домашней работы была получена информация о понятии, назначении, порядке и общих правилах работы с системой контроля версий, продемонстрирована командная работа при разработке программного продукта, а также был написан программный код для индивидуального варианта.

ЛИТЕРАТУРА

Основная литература 1. Киселева, Т. В. Программная инженерия. Часть 1 [Электронный ресурс]: учебное пособие / Т. В. Киселева. — Электрон. текстовые данные —

Ставрополь : Северо-Кавказский федеральный университет, 2017. — 137 с.: <http://www.iprbookshop.ru/69425.html>

2. Программная инженерия. Часть II [Электронный ресурс]: учебное пособие / составители Т. В. Киселева. — Электрон. текстовые данные — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 100 с.: <http://www.iprbookshop.ru/83193.html>

3. Полетайкин, А. Н. Учебно-методическое пособие по выполнению лабораторных работ по дисциплине «Программная инженерия». Часть I. Реализация жизненного цикла программного обеспечения [Электронный ресурс]: учебно-методическое пособие / А. Н. Полетайкин. — Электрон. текстовые данные — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2016. — 97 с.: <http://www.iprbookshop.ru/69565.html>

Дополнительная литература

1. Мейер, Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Б. Мейер. — 3-е изд. — Электрон. текстовые данные — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 с.: <http://www.iprbookshop.ru/79706.html>