



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»
КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

«Разработка программного кода. Стандарт кодирования»

ДИСЦИПЛИНА: «Основы программной инженерии»

Выполнил: студент гр. ИУК4-21Б


(подпись)

(Суриков Н.С)
(Ф.И.О.)

Проверил:

(подпись)

(Амеличев Г. Э.)
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024

Цель: формирование практических навыков следования стандарту кодирования при разработке программного обеспечения.

Задачи:

1. Изучить стандарты кодирования
2. Решить поставленную задачу
3. Произвести рефакторинг кода
4. Составить таблицу идентификаторов

Условие задачи:

Вариант 4

Квадратную матрицу 17x17 заполнить случайными числами из диапазона [-12..12], вывести ее на экран.

а) К каждому элементу целочисленной матрицы прибавить сумму цифр элемента, стоящего в этой же строке на побочной диагонали.

б) Проверить, содержит строка, номер которой введен с клавиатуры, знаочередующиеся элементы.

в) Указать номер строки и столбца первого нулевого элемента среди элементов матрицы, выделенных чёрным цветом (матрица квадратная).



Листинг программы:

```
1 #include <iostream>
2 #include <cstdlib>
3
4 const int MATRIX_SIZE = 17;
5
6 int matrix[MATRIX_SIZE][MATRIX_SIZE];
7 int numberOfRow = 0;
8 bool tempOfNegative, flag = true;
9
10 void generateRandomMatrix()
11 {
12     for (int i = 0; i < MATRIX_SIZE; i++)
```

```
13  {
14      for (int j = 0; j < MATRIX_SIZE; j++)
15      {
16          matrix[i][j] = rand() % 25 - 12;
17      }
18  }
19 }
20
21 void printMatrix()
22 {
23     for (int i = 0; i < MATRIX_SIZE; i++)
24     {
25         for (int j = 0; j < MATRIX_SIZE; j++)
26         {
27             std::cout << matrix[i][j] << " ";
28         }
29         std::cout << std::endl;
30     }
31 }
32
33 int calculateDigitSum(int number)
34 {
35     return abs(number / 10 + number % 10);
36 }
37
38 void modifyMatrix()
39 {
40     for (int i = 0; i < MATRIX_SIZE; i++)
41     {
42         for (int j = 0; j < MATRIX_SIZE; j++)
43         {
44             matrix[i][j] += calculateDigitSum(matrix[i][MATRIX_SIZE - 1 - i]);
45         }
46     }
47 }
48
49 bool isNegative(int number)
50 {
51     return number < 0;
52 }
53
54 void checkRowForAlternatingSigns()
55 {
56     std::cin >> numberOfRow;
```

```
57     for (int j = 1; j < MATRIX_SIZE; j++)
58     {
59         if (isNegative(matrix[numberOfRow][j]) == isNegative(matrix[numberOfRow][j - 1]))
60         {
61             flag = false;
62             break;
63         }
64     }
65     std::cout << flag << std::endl;
66 }
67
68 void findFirstZeroElement()
69 {
70     flag = true;
71     for (int i = MATRIX_SIZE / 3; i < MATRIX_SIZE; i++)
72     {
73         for (int j = 0; j < MATRIX_SIZE; j++)
74         {
75             if (matrix[i][j] == 0)
76             {
77                 std::cout << i << " " << j << std::endl;
78                 flag = false;
79                 break;
80             }
81         }
82         if (!flag)
83         {
84             break;
85         }
86     }
87 }
88
89 int main()
90 {
91     generateRandomMatrix();
92     modifyMatrix();
93     checkRowForAlternatingSigns();
94     findFirstZeroElement();
95     return 0;
96 }
```

Таблица идентификаторов:

Идентификаторы – это последовательность символов, используемая для обозначения имени объекта или переменной, имени класса, структуры или объединения		
Правила именования:		
1. Американский английский 2. Upper/Lower Camel casing 3. Важно, чтобы все идентификаторы были легко читаемыми (и произносимыми вслух) и понятными, а не максимально короткими 4. Именованные константы (включая значения перечислений) должны быть записаны в верхнем регистре с нижним подчёркиванием в качестве разделителя		
Имя идентификатора	Нотация	Условный перевод
ElementMatrix	UpperCamelCase	Элемент матрицы
matrix	CamelCase	Матрица
numberOfRow	LowerCamelCase	Номер строки
В идентификаторах недопустимо:		
1.использование сокращений		
2.Использование символов подчеркивания, дефисов и других неалфавитно-цифровых символов.		
3.Применение венгерской нотации.		
4.Использование ключевых слов в качестве идентификаторов.		
В идентификаторах следует избегать:		
1.Сленг		
2.Акронимы		
3.Просторечия		
4.Элементы конкретной культуры		
Акронимы – слова, образованные из начальных букв слов или словосочетаний		
Правила именования:		
1. Акронимы, состоящие из трех или больше символов подчиняются рекомендациям для обычных слов 2. Акроним, состоящий из двух символов, в Upper Camel Case набирается двумя прописными символами, а в Lower Camel Case — двумя строчными		
Акронимы	Нотация	Дешифровка
Dry	Upper Camel Case	Don't Repeat Yourself
В акронимах недопустимо:		
1. Использование не общепринятых акронимов		
2. Сокращения акронимов		
Именование типов		
1. В идентификаторах классов и структур должен использоваться стиль Upper Camel case 2. Название класса/структуры должно быть существительным или именной группой 3. В идентификаторах интерфейсов должен использоваться стиль Upper Camel case, при этом имя должно снабжаться префиксом «I». В качестве имен интерфейсов следует выбирать существительные и именные группы, а, если интерфейс представляет возможность, — прилагательные и адъективные фразы 4. В именах абстрактных базовых классов можно использовать слова «Abstract» (как префикс) и «Base» (чаще как суффикс) 5. Идентификатор производного класса должен указывать, на что класс похож и чем отличается. Рекомендуется имена производных классов оканчивать именами базовых классов 6. В паре «интерфейс — класс», в которой класс является стандартной реализацией интерфейса, идентификаторы должны совпадать с точностью до префикса «I» у интерфейса		

7. В именах перечислений должен использоваться Upper Camel case. Идентификатор должен быть существительным или именной группой. Если значениями перечисления являются битовые флаги, то идентификатор должен быть во множественном числе, в остальных случаях — в единственном числе		
Типы	Нотация	Дешифровка
Container	Upper Camel Case	Контейнер
При объявлении типов недопустимо:		
1. Прибавление префиксов или суффиксов		
2. Использование общих слов		
Процедуры и функции – фрагмент программного кода, к которому можно обратиться из другого места программы Правила именования: <ol style="list-style-type: none"> 1. В идентификаторах методов должен использоваться стиль Upper Camel case 2. Имена должны прояснять обязанности метода и при этом быть максимально короткими 3. При именовании методов следует выработать единый согласованный лексикон и следовать ему 4. Методы-обработчики событий следует именовать, используя префикс «On». Правила создания: <ol style="list-style-type: none"> 1. При именовании параметров методов должен использоваться стиль Lower Camel case 2. Идентификаторы параметров в большинстве случаев должны представлять собой существительные или именные группы. Параметром следует давать описательные имена. 		
Процедура или функция	Нотация	Дешифровка
generateRandomMatrix	CamelCase	Создать матрицу случайными числами
printMatrix()	CamelCase	Вывести матрицу
calculateDigitSum	CamelCase	Посчитать сумму цифр
modifyMatrix	CamelCase	Модифицировать матрицу
checkRowForAlternatingSign	CamelCase	Проверка ряда на знакопеременность
При именовании недопустимо:		
1. Если из контекста понятно, над каким объектом выполняется действие, то название объекта не следует включать в идентификатор метода		
При именовании следует избегать:		
1. Использования имени типов, а следует использовать слова, проясняющие семантику		
Константы – способ адресации данных, изменение которых рассматриваемой программой не предполагается или запрещается Правила именования: <ol style="list-style-type: none"> 1. В идентификаторах констант должен использоваться стиль Upper Camel case. Константы должны иметь имена, являющиеся существительными, именными группами или прилагательными. 2. Булевы свойства рекомендуется называть в утвердительно форме. При необходимости можно также добавлять и префиксы «Is», «Can» или «Has». Правила использования: <ol style="list-style-type: none"> 1. Именованные константы (включая значения перечислений) должны быть записаны в верхнем регистре с нижним подчеркиванием в качестве разделителя. 2. Общая практика в сообществе разработчиков C++. Использование таких констант должно быть сведено к минимуму. 		
Константы	Нотация	Дешифровка
MATRIX_SIZE	CamelCase	Размер матрицы

Локальные переменные – переменные, объявленные внутри блока кода

Правила именования:

1. При именовании переменных должен использоваться стиль Lower Camel case
2. Идентификаторы переменных в большинстве случаев должны представлять собой существительные или именные группы. Переменным следует давать описательные имена.

Локальные переменные	Нотация	Дешифровка
numberOfRow	LowerCamelCase	Номер строки

Вывод: в ходе данной лабораторной работы были сформированы практические навыки следования стандарту кодирования при разработке программного обеспечения.