|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 №2**

**«Оценка качества программного продукта»**

**ДИСЦИПЛИНА: «Основы программной инженерии»**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-21Б | |  |  | ( | Суриков Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Амеличев Г. Э. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** изучение основных методов и подходов оценки качества программного

продукта.

**Задачи:**

1. изучить основы метрической теории Холстеда;
2. для написанных ранее программ произвести расчет количественных характеристик программ;
3. сравнить полученные результаты.

**Листинг программы:**

1 *#include* *<iostream>*

2 *#include* *<cstdlib>*

3

4 const int MATRIX\_SIZE = 17;

5

6 int matrix[MATRIX\_SIZE][MATRIX\_SIZE];

7

8 void generateRandomMatrix()

9 {

10 for (auto &i : matrix)

11 {

12 for (int &j : i)

13 {

14 j = std::rand() % 25 - 12;

15 }

16 }

17 }

18

19 void printMatrix()

20 {

21 for (auto &i : matrix)

22 {

23 for (int j : i)

24 {

25 std::cout << j << " ";

26 }

27 std::cout << std::endl;

28 }

29 }

30

31 int calculateDigitSum(int number)

32 {

33 number = abs(number);

34 return number / 10 + number % 10;

35 }

36

37 void modifyMatrix()

38 {

39 for (int i = 0; i < MATRIX\_SIZE; i++)

40 {

41 for (int j = 0; j < MATRIX\_SIZE; j++)

42 {

43 matrix[i][j] += calculateDigitSum(matrix[i][MATRIX\_SIZE - 1 - i]);

44 }

45 }

46 }

47

48 bool isNegative(int number)

49 {

50 return number < 0;

51 }

52

53 void checkRowForAlternatingSigns()

54 {

55 int numberOfRow{};

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 std::cout << "Строка содержит знакочередующиеся элементы" << std::endl;

62 break;

63 }

64 }

65 }

66

67 void findFirstZeroElement()

68 {

69 for (int i = MATRIX\_SIZE / 3; i < MATRIX\_SIZE; i++)

70 {

71 for (int j = 0; j < MATRIX\_SIZE; j++)

72 {

73 if (matrix[i][j] == 0)

74 {

75 std::cout << i << " " << j << std::endl;

76 return;

77 }

78 }

79 }

80 }

81

82 int main()

83 {

84 generateRandomMatrix();

85 printMatrix();

86 modifyMatrix();

87 checkRowForAlternatingSigns();

88 findFirstZeroElement();

89 printMatrix();

90 return 0;

91 }

**Таблица идентефикаторов:**

|  |  |  |
| --- | --- | --- |
| **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:   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:   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 знакочередование |
| 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 имена типов, а следует использовать слова, проясняющие семантику | | |
| **Константы** – способ адресации данных, изменение которых рассматриваемой программой не предполагается или запрещается  Правила именования:   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:   1. Именованные константы (включая значения перечислений) должны быть записаны в верхнем регистре с нижним подчёркиванием 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.