|  |  |
| --- | --- |
| **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»***

**ДОМАШНЯЯ РАБОТА**

**«Использование системы контроля версий и хостинга проектов»**

**ДИСЦИПЛИНА: «Основы программной инженерии»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-22Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличев Г.Э. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**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 9x9 заполнить случайными числами из диапазона [-21; 23], вывести ее на экран.

Найти и вывести на экран количество положительных элементов на побочной диагонали.

Заменить все отрицательные элементы ниже главной диагонали на 0.

Вывести результирующую матрицу на экран.

**Листинг:**

**CreateMatrix.h:**

#ifndef CREATE\_MATRIX\_H

#define CREATE\_MATRIX\_H

int\*\* CreateMatrix(int rows, int columns, bool mode, int minimalRandomNumber = 0, int maximalRandomNumber = 0);

#endif // !CREATE\_MATRIX\_H

**CreateMatrix.cpp:**

#include "CreateMatrix.h"

#include <ctime>;

#include <iostream>

int\*\* CreateMatrix(int rows, int columns, bool mode, int minimalRandomNumber, int maximalRandomNumber)

{

int\*\* matrix = new int\* [rows];

if (mode == false)

{

srand(time(0));

for (short i = 0; i < rows; i++)

{

matrix[i] = new int[columns];

for (short j = 0; j < columns; j++)

{

matrix[i][j] = rand() % (maximalRandomNumber - minimalRandomNumber + 1) + minimalRandomNumber;

}

}

}

else

{

for (short i = 0; i < rows; i++)

{

matrix[i] = new int[columns];

for (short j = 0; j < columns; j++)

{

std::cin >> matrix[i][j];

}

}

}

return matrix;

}

**PrintMatrix.h:**

#ifndef PRINT\_MATRIX\_H

#define PRINT\_MATRIX\_H

void PrintMatrix(int\*\* matrix, int rows, int columns, int cellWidth);

#endif // !PRINT\_MATRIX\_H

**PrintMatrix.cpp:**

#include "PrintMatrix.h"

#include <iostream>

#include <iomanip>

void PrintMatrix(int\*\* matrix, int rows, int columns, int cellWidth)

{

short numberColumnWidth{};

int tempRows = rows;

while (tempRows != 0)

{

tempRows /= 10;

numberColumnWidth++;

std::cout << " ";

}

std::cout << " ";

for (short i = 0; i < columns; i++)

{

std::cout << std::setw(cellWidth) << i << " ";

}

std::cout << "\n\n\n";

for (short i = 0; i < rows; i++)

{

std::cout << std::setw(numberColumnWidth) << i;

std::cout << " ";

for (short j = 0; j < columns; j++)

{

std::cout << std::setw(cellWidth) << matrix[i][j] << " ";

}

std::cout << "\n\n";

}

}

**FindPlusOnSideDiagonal.h:**

#ifndef FIND\_PLUS\_ON\_SIDE\_DIAGONAL\_H

#define FIND\_PLUS\_ON\_SIDE\_DIAGONAL\_H

void FindPlusOnSideDiagonal(int\*\* matrix,int rows, int columns);

#endif // !FIND\_PLUS\_ON\_SIDE\_DIAGONAL\_H

**FindPlusOnSideDiagonal.cpp:**

#include <iostream>

#include "FindPlusOnSideDiagonal.h"

void FindPlusOnSideDiagonal(int\*\* matrix, int rows, int columns)

{

setlocale(LC\_ALL, "Russian");

int countPlus{};

for (int i = 0; i < rows && i < columns; i++)

{

if (matrix[i][columns - i - 1] > 0)

{

countPlus++;

}

}

std::cout << "Количество положительных чисел на побочной диагонали: " << countPlus;

std::cout << std::endl;

}

**ReplaceMinusUnderMainDiagonal.h:**

#ifndef REPLACE\_MINUS\_UNDER\_MAIN\_DIAGONAL\_H

#define REPLACE\_MINUS\_UNDER\_MAIN\_DIAGONAL\_H

void ReplaceMinusUnderMainDiagonal(int\*\* matrix, int rows, int columns);

#endif // !REPLACE\_MINUS\_UNDER\_MAIN\_DIAGONAL\_H

**ReplaceMinusUnderMainDiagonal.cpp:**

#include <iostream>

#include "ReplaceMinusUnderMainDiagonal.h"

void ReplaceMinusUnderMainDiagonal(int\*\* matrix, int rows, int columns)

{

for (int i = 1; i < rows; i++)

{

for (int j = 0; j < i && j<columns; j++)

{

if (matrix[i][j] < 0)

{

matrix[i][j] = 0;

}

}

}

}

**Main.cpp:**

#include "CreateMatrix.h"

#include "PrintMatrix.h"

#include "FindPlusOnSideDiagonal.h"

#include "ReplaceMinusUnderMainDiagonal.h"

#include <iostream>

const int MAXIMUM\_INPUT\_LENGHT = 255;

int main()

{

setlocale(LC\_ALL, "Russian");

char command[MAXIMUM\_INPUT\_LENGHT]{};

int rows = 9;

int columns = 9;

std::cout << "Введите 1, если хотите поменять размер матрицы (по ум. 9х9): ";

std::cin.getline(command, MAXIMUM\_INPUT\_LENGHT, '\n');

if (!strcmp(command, "1"))

{

std::cout << "Введите кол-во строк: ";

std::cin >> rows;

std::cout << "Введите кол-во столбцов: ";

std::cin >> columns;

std::cin.ignore(32767, '\n');

}

int\*\* matrix{};

int cellWidth = 1;

std::cout << "Введите 1, если хотите задать значения элементов матрицы (по ум. случайное заполнение): ";

std::cin.getline(command, MAXIMUM\_INPUT\_LENGHT, '\n');

if (!strcmp(command, "1"))

{

std::cout << "Введите матрицу " << rows << "x" << columns << "\n";

matrix = CreateMatrix(rows, columns, 1);

}

else

{

int minimalRandomNumber = 0;

int maximalRandomNumber = 0;

std::cout << "Введите 1, если хотите изменить границы случайных чисел (по ум. [-21; 23]): ";

std::cin.getline(command, MAXIMUM\_INPUT\_LENGHT, '\n');

if (!strcmp(command, "1"))

{

std::cout << "Введите минимальное случайное число: ";

std::cin >> minimalRandomNumber;

std::cout << "Введите максимальное случайное число: ";

std::cin >> maximalRandomNumber;

std::cin.ignore(32767, '\n');

}

else

{

minimalRandomNumber = -21;

maximalRandomNumber = 23;

}

if (minimalRandomNumber < 0)

{

cellWidth++;

}

int tens = 10;

while (tens < abs(minimalRandomNumber) || tens < abs(maximalRandomNumber))

{

cellWidth++;

tens \*= 10;

}

matrix = CreateMatrix(rows, columns, 0, minimalRandomNumber, maximalRandomNumber);

}

std::cout << "Созданная матрица:\n\n";

PrintMatrix(matrix, rows, columns, cellWidth);

FindPlusOnSideDiagonal(matrix, rows, columns);

std::cout << '\n';

ReplaceMinusUnderMainDiagonal(matrix, rows, columns);

std::cout << "Матрица после замены отрицательных чисел, находящихся ниже главной диагонали, на 0:\n\n";

PrintMatrix(matrix, rows, columns, cellWidth);

return 0;

}

**Результат работы:**

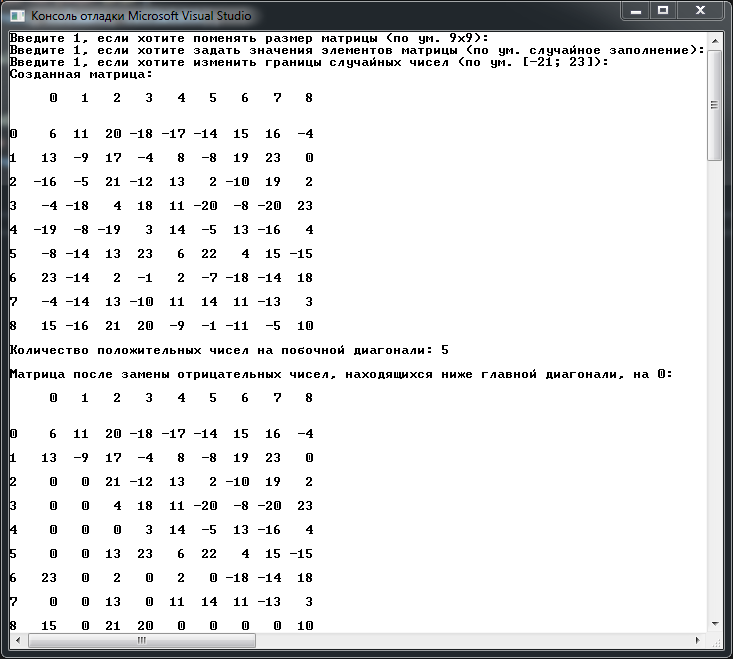
****

Рисунок 1. Результат работы

**История изменений:**

****

Рисунок 2. История изменений

**Вывод:** в ходе работы были получены практические навыки работы с веб-сервисом для хостинга проектов и их совместной разработки Bitbucket, клиентом управления версиями Git SourceTree, средой разработки для компиляции Git msysGit.

**Список литературы:**

**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 c.: <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 c.: <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 c: <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 : ИнтернетУниверситет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 c.: http://www.iprbookshop.ru/79706.html