**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ**

**ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

**Лабораторная работа №6**

**Вариант 1**

**«Двумерные массивы. Указатели. Метод сверху вниз»**

**по предмету «Алгоритмизация и программирование»**

**Выполнил: студент гр. 5130904/30002 Севостьянова А.В.**

**Руководитель: Череповский Д.К.**

**Санкт-Петербург**

**2023 г.**

Оглавление

[Задание 1 3](#_Toc149227218)

[Задание 2.1 4](#_Toc149227219)

[1. Общая постановка задачи 4](#_Toc149227220)

[2. Детальные требования и тест план 4](#_Toc149227221)

[Выбор пользователем функции(choice) должен быть корректен (choice – принимает значение 1 или 2): 4](#_Toc149227222)

[Элементы массива должны быть заданы корректно (целые числа): 4](#_Toc149227223)

[Таблица с детальными требованиями и тест планом 4](#_Toc149227224)

[3. Программа 6](#_Toc149227225)

[Main.cpp 6](#_Toc149227226)

[Functions.h 7](#_Toc149227227)

[Functions.cpp 7](#_Toc149227228)

[Приложение A 9](#_Toc149227229)

[Приложение B 11](#_Toc149227230)

[Вывод 13](#_Toc149227231)

# Задание 1

1. 0

Будет выведен элемент статического двумерного массива с индексом [1][1].

1. \*(\*(cc + p) + q) = 3
2. Test1(c)

Функция принимает статический массив размера 4 на 5 (4 строки, 5 колонок)

1. \*(\*(bb + k) + m)) = 2
2. test4(&dd[0][0], 3);

test4(&dd[0][0], 3);

# Задание 2.1

1. **Общая постановка задачи**

Написать функцию(inArray), имеющую три параметра – адрес матрицы (целые числа, встроенный массив), количество строк и количество столбцов, и выполняющую ввод всех элементов массива из потока cin, обращение к элементам массива происходит через индексы.

Написать функцию(inArrayPointer), имеющую три параметра – адрес матрицы (целые числа, встроенный массив), количество строк и количество столбцов, и выполняющую ввод всех элементов массива из потока cin, обращение к элементам массива происходит через указатели.

**Входные данные**: элементы массива

**Выходные данные**: заполненный массив

1. **Детальные требования и тест план**

### *Выбор пользователем функции(choice) должен быть корректен* (choice – *принимает значение 1 или 2):*

1. Если choice не число сообщение: «Непредвиденный символ»
2. Если choice принимает дробное сообщение: «Непредвиденный символ»
3. Если choice == 1: вызов функции inArray, заполнение массива.
4. Если choice == 2: вызов функции inArrayPointer, заполнение массива
5. Если choice != 1 и choice != 2: сообщение «Некорректный выбор»

### *Элементы массива должны быть заданы корректно (целые числа):*

1. Если элемент массива принимает нечисловое значение, сообщение: «Непредвиденный символ»
2. Если элемент массива принимает дробное значение, сообщение: «Непредвиденный символ»

### *Таблица с детальными требованиями и тест планом*

|  |  |  |  |
| --- | --- | --- | --- |
| Требования | Детальные требования | Данные | Ожидаемый результат |
| 1. ***Выбор пользователя должен быть корректен*** | | | |
| 1. choice - число | Если choice имеет нечисловое значение, то сообщение: «Непредвиденный символ», повторное введение выбора пользователя. | R | Сообщение: «Непредвиденный символ»  Повторный ввод выбора пользователя. |
| 2. choice – целое число | Если choice имеет дробное значение, то сообщение: «Непредвиденный символ», повторное введение выбора пользователя. | 5,5 | Сообщение: «Непредвиденный символ»  Повторный ввод выбора пользователя. |
| 3. choice == 1 | Если choice = 1, вызов функции inArray, заполнение массива | 1 | Вызов функции inArray.  Заполнение массива. |
| 4. choice == 2 | Если choice = 2, вызов функции inArrayPointer, заполнение массива | 2 | Вызов функции inArrayPointer.  Заполнение массива. |
| 1. ***Элементы массива должны быть заданы корректно*** | | | |
| 1. value - число | Если value имеет нечисловое значение, то сообщение: «Непредвиденный символ», повторное введение элемента массива. | R | Сообщение: «Некорректный ввод данных»  Повторное введение элемента массива.  Сообщение: «Введите элемент массива» |
| 2. value – целое число | Если value имеет дробное значение, то сообщение: «Непредвиденный символ», повторное введение элемента массива. | 5,5 | Сообщение: «Непредвиденный символ»  Повторное введение элемента массива.  Сообщение: «Введите элемент массива» |
| 3. value задана корректно | Если value задана корректно, добавление переменной в массив. После заполнения всего массива корректно заданными элементами вызов функции outArray для вывода полученного массива. | 45 | Сообщение: «Введите элемент массива:»  Ввод элементов массива.    После полного заполнения массива вывод полученной структуры на экран. |

1. **Программа**

### *main.cpp*

#include"function.h"

int main()

{

setlocale(LC\_ALL, "ru");

int arr[ROWS][COLS] = { 0 };

string value = "" ; // переменная, хранящая выбор пользователя до проверки корректности ввода

int choiсе = 0; // выбор пользователя

bool test = false;

cout << "Заполнение статичческого массива\n";

while (!test)

{

cout << " 1- обращение к элементам массива через индексы\n 2 - обращение к элементам массива через указатели\n"

<<"Выбор:\t";

cin >> value;

try

{

choiсе = checkInput(value);

switch (choiсе)

{

case 1: {inArray(arr,ROWS,COLS); test = true; break; }

case 2: {inArrayPointer(arr, ROWS, COLS); test = true; break; }

default:

{

cout << "\nНекорректный выбор\n";

test = false;

}

}

}

catch (runtime\_error &ex)

{

cout << "\n" << ex.what() << "\n";

test = false;

}

}

cout << "\n\nПолученный статического массив:\n";

outArray(arr);

return 0;

}

### *functions.h*

#pragma once

#include<iostream>

#include<regex>

#include<string>

using namespace std;

#define ROWS 2

# define COLS 3

const string ERROR = "Непредвиденный символ";

int checkInput(const string& value); // провекрка корректности вводимой переменной

void inArray(int (&arr)[ROWS][COLS],int rows,int cols); // заполнение массива (обращение к элементам ведется через индексы)

void inArrayPointer(int(&arr)[ROWS][COLS], int rows, int cols); // заполнение массива (обращение к элементам ведется через указатели)

void outArray(int (&arr)[ROWS][COLS]); // вывод полученного массива на экран

### *functions.cpp*

#include "function.h"

int checkInput(const string& value)

{

regex regular("-?\\d+");

if (regex\_match(value, regular)) return stoi(value);

else throw runtime\_error(ERROR);

}

void inArray(int (&arr)[ROWS][COLS],int rows,int cols)

{

bool test = true;

string value = "";

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

for (int j = 0; j < cols; j++)

{

do

{

cout << "Введите элемент массива\t";

cin >> value;

try

{

arr[i][j] = checkInput(value);

test = true;

}

catch (runtime\_error& ex)

{

cerr << ex.what() << "\n";

test = false;

}

} while (!test);

}

}

void inArrayPointer(int(&arr)[ROWS][COLS], int rows, int cols)

{

bool test = true;

string value = "";

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

for (int j = 0; j < cols; j++)

{

do

{

cout << "Введите элемент массива\t";

cin >> value;

try

{

\*(\*(arr+i)+j) = checkInput(value);

test = true;

}

catch (runtime\_error& ex)

{

cerr << ex.what() << "\n";

test = false;

}

} while (!test);

}

}

void outArray(int(&arr)[ROWS][COLS])

{

for (int i = 0; i < ROWS; i++)

for (int j = 0; j < COLS; j++)

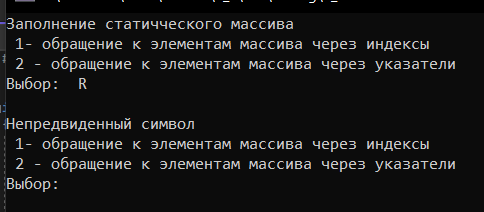
cout << "arr[" << i << "][" << j << "] =" << \*(\*(arr + i) + j) << "\n";

}

# Приложение A

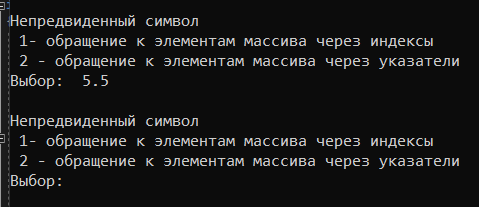
Выбор пользователя должен быть корректен

1. Выбор пользователя должен принимать числовое значение

****

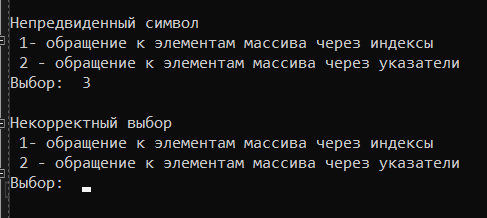
*Рис. 1 Выбор пользователя принимает нечисловое значение*

1. Выбор пользователя должен принимать целочисленное значение

****

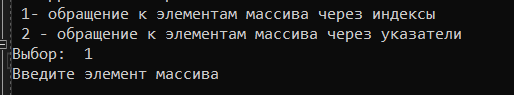
*Рис. 2 Выбор пользователя принимает дробное значение*

1. Выбор пользователя должен быть 1 или 2

****

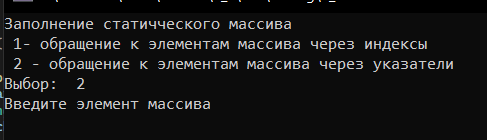
*Рис. 3 Выбор пользователя не равен 1 или 2*

1. Если выбор пользователя равен 1

****

*Рис. 4 Выбор пользователя равен 1*

1. Если выбор пользователя равен 2

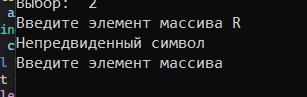
****

*Рис. 5 Выбор пользователя равен 2*

# Приложение B

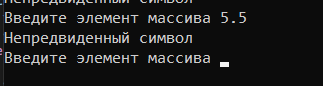
Элементы массива должны быть заданы корректно

1. Элементы массива должны принимать числовое значение

****

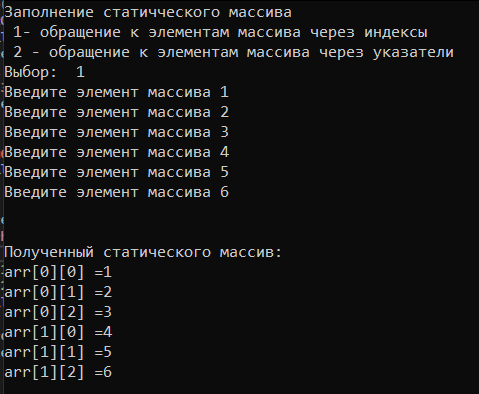
*Рис. 1 Элемент массива принимает нечисловое значение*

1. Элементы массива должны принимать целочисленное значение

****

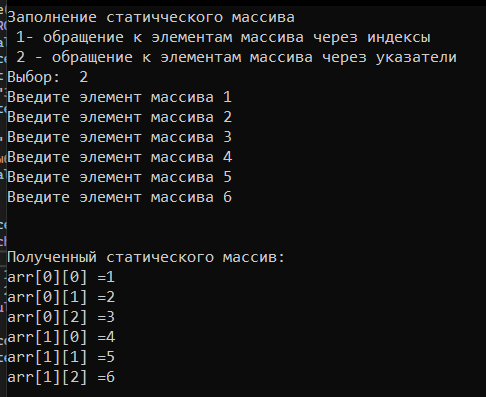
*Рис. 2 Элемент массива принимает дробное значение*

1. Элементы массива заданы корректно
2. Результат работы функции inArray



*Рис. 3 Результат работы функции inArray*

1. Результат работы функции inArrayPointer

****

*Рис. 4 Результат работы функции InArrayPointer*

# Вывод

В ходе работы мы изучили:

1. Статические двумерные массивы
2. Передача адреса статического двумерного массива в функцию
3. Обращение к элементам массива при помощи указателей
4. Конструкцию try … catch, которая позволяет эффективнее обрабатывать возникающие в ходе работы программы ошибки