ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ #2

за курсом «Алгоритмы и структуры данных»

студента группы ПА-19-2

Ильяшенко Егора Виталиевича

кафедра компьютерных технологий, ДНУ

2019/2020 уч. год

1. Постановка задачи:

Реализовать очередь на базе статического массива. Реализовать основные методы очереди. Так же сделать возможность автозаполнения очереди случайными элементами из диапазона, реализовать вывод всех элементов. Среди положительных элементов вывести сразу чётные, потом нечётные.

1. Описание решение

Разбил задачи на функции, которые вызываются с меню. Функции с задачами вызываются через Switch-case по соответствующему номеру задачи. Реализовал очередь с основными её методами. В задаче про положительные числа шёл по очереди, выводя чётные, а нечётные удаляя и добавляя. Таким образом нечётные элементы перемещались в конец очереди, где и выводились.

1. Исходный текст программы

#include <iostream>

#include <ctime>

using namespace std;

void Menu();

const int n\_max = 150;

struct Queue

{

private:

int n;

int mass[n\_max];

public:

Queue()

{

n = 0;

}

int Take\_N()

{

return n;

}

int AddElem(int elem)

{

if (n != n\_max)

{

mass[n] = elem;

n++;

return 0;

}

return -1;

}

int TakeElem()

{

if (n != 0)

{

int out = mass[0];

for (int i = 0; i < n-1; i++)

mass[i] = mass[i + 1];

n--;

return out;

}

return -1;

}

};

Queue qu = Queue();

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(0));

Menu();

system("Pause");

}

void Menu()

{

bool menu\_i = true;

while (menu\_i)

{

system("cls");

cout << "1) Додати елемент" << endl;

cout << "2) Видалити елемент" << endl;

cout << "3) Вивести елементи на екран" << endl;

cout << "4) Очистити структуру" << endl;

cout << "5) Вивести кількість елементів, індекс першого та останнього елементу" << endl;

cout << "6) Сформувати випадковий массив" << endl;

cout << "7) Парные - непарные" << endl;

cout << "8) Завершення роботи" << endl;

cout << endl;

cout << "Введите номер пункта, к которому хотите перейти - ";

int n\_; cin >> n\_; cout << endl;

switch (n\_)

{

case 1:

cout << "Введите элемент" << endl;

int k; cin >> k; cout << endl;

if (qu.AddElem(k) == -1)

cout << "Ошибка ввода" << endl;

system("Pause");

break;

case 2:

if (qu.TakeElem() == -1)

cout << "Очередь пустая" << endl;

system("Pause");

break;

case 3:

int buff;

for (int i = 0; i < qu.Take\_N(); i++)

{

buff = qu.TakeElem();

qu.AddElem(buff);

cout << "elem[" << i << "]= " << buff << endl;

}

system("Pause");

break;

case 4:

int i\_max; i\_max = qu.Take\_N();

for (int i = 0; i < i\_max; i++)

qu.TakeElem();

system("Pause");

break;

case 5:

cout << "Кол-во элементов - " << qu.Take\_N() << endl;

cout << "Индекс минимального - 0" << endl;

cout << "Индекс максимального - " << qu.Take\_N() - 1 << endl;

system("Pause");

break;

case 6:

cout << "Введите n(кол-во элементов очереди) (0, " << n\_max << ") - ";

int n\_test; cin >> n\_test; cout << endl;

cout << "Введите 2 границы случайного числа:" << endl;

cout << "Введите А (>0)- ";

int a; cin >> a; cout << endl;

cout << "Введите B (>0)- ";

int b; cin >> b; cout << endl;

if (a > b)

{

int c = a;

a = b;

b = c;

}

if (a < 0) a \*= -1;

if (b < 0) b \*= -1;

if (n\_test > 0 && n\_test <= n\_max)

{

for (int i = 0; i < qu.Take\_N(); i++)

qu.TakeElem();

for (int i = 0; i < n\_test; i++)

qu.AddElem(rand() % (b - a + 1) + a);

}

system("Pause");

break;

case 7:

int i\_maxx; i\_maxx = qu.Take\_N();

for (int i = 0; i < i\_maxx; i++)

{

int elem\_cur = qu.TakeElem();

if (elem\_cur % 2 == 0)

{

cout << elem\_cur << endl;

}

else

{

qu.AddElem(elem\_cur);

}

}

i\_maxx = qu.Take\_N();

for (int i = 0; i < i\_maxx; i++)

{

cout << qu.TakeElem() << endl;

}

system("Pause");

cout << endl;

break;

case 8:

menu\_i = 0;

break;

default:

menu\_i = 0;

break;

}

}

}

1. Описание интерфейса

При запуске программы выводится меню. Чтобы перейти к пункту меню, необходимо ввести номер пункта с клавиатуры и нажать ENTER. Дальше следовать инструкциям.

1. Описание тестовых примеров

Вводил разные значения, после чего добавлял ограничения на вводимые данные. Если данные не соответсвовали требуемым, консоль очищалась, выводилось сообщение об ошибке ввода. После этого подпрограмма перезапускается.







