Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Информатика»

Тема: Лабораторная работа 11

"Очередь"

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Варов Д.A.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

Реализация очереди

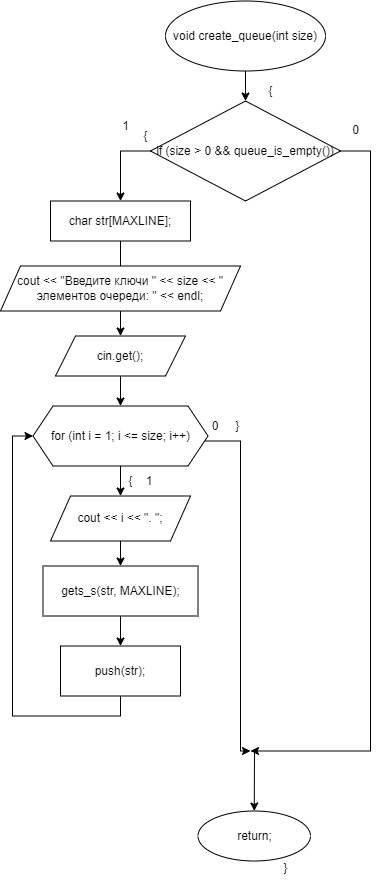


Рис.1. Создание очереди

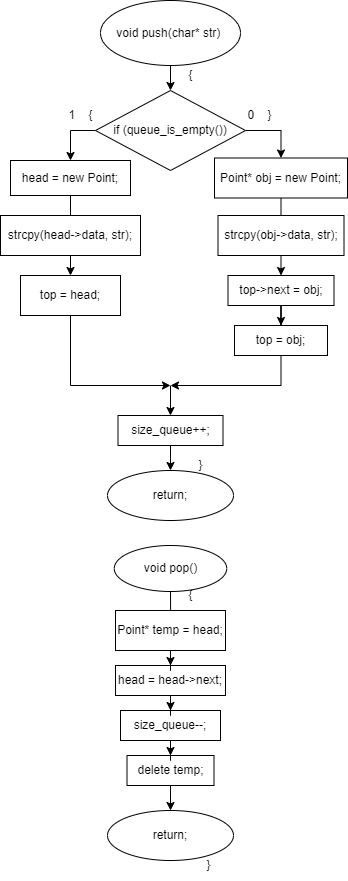


Рис.2. Добавление элементов в очереди

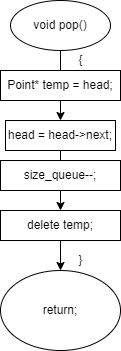


Рис.3. Удаление первого элемента

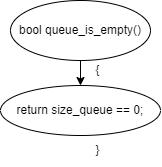


Рис. 4. Проверка на пустой стек

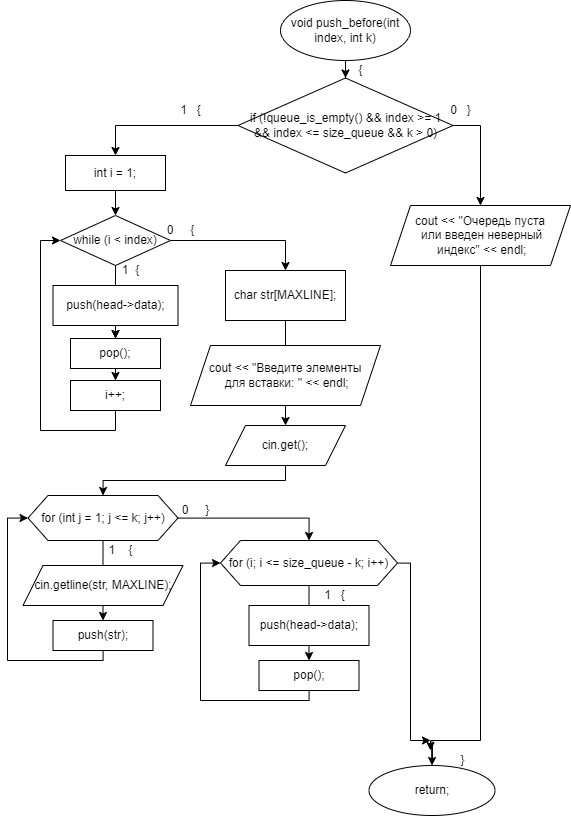


Рис.5. Вставка элемента до заданного

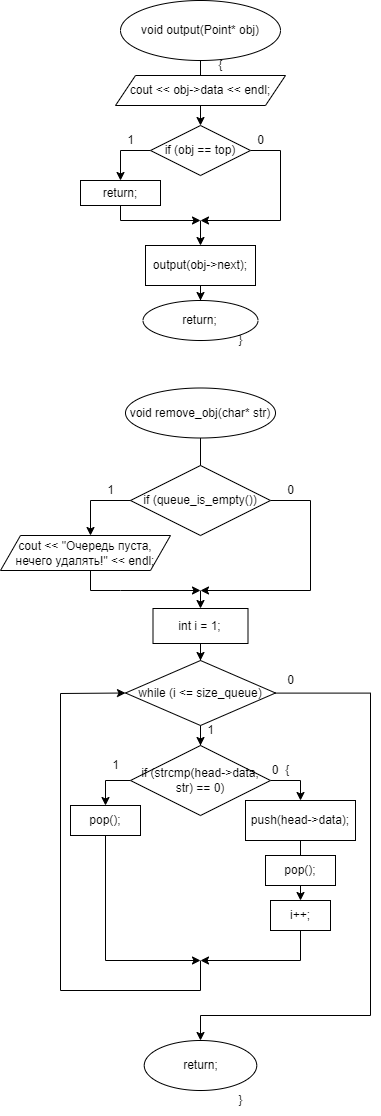


Рис.6. Вывод элементов очереди

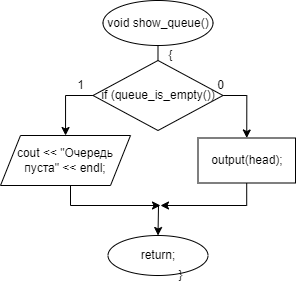


Рис.7. Вывод всей очереди

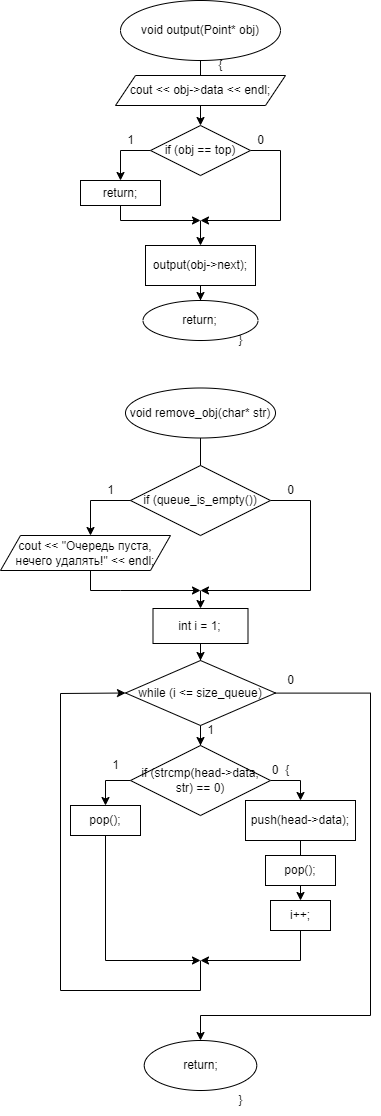


Рис.8. Удаление элементов по ключу

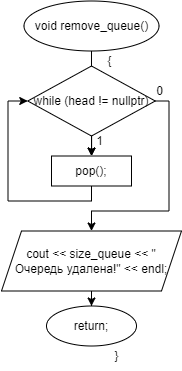


Рис.9. Удаление очереди

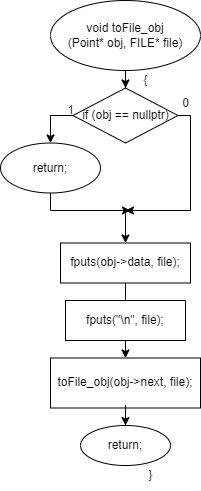


Рис.10. Запись данных в файл



Рис.11. Запись ключей в файл

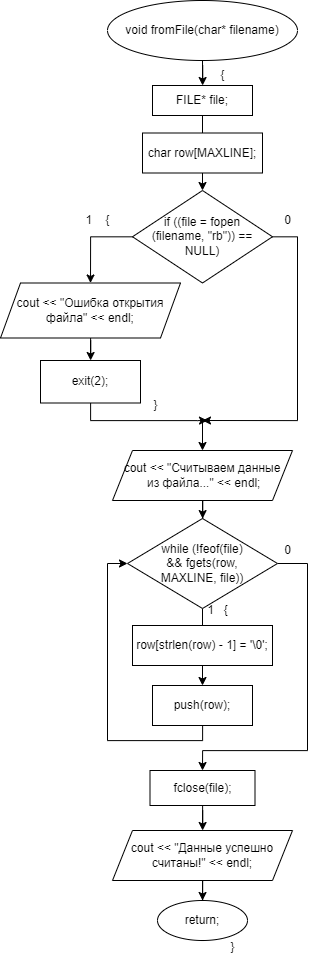


Рис.12. Выгруз ключей из файла

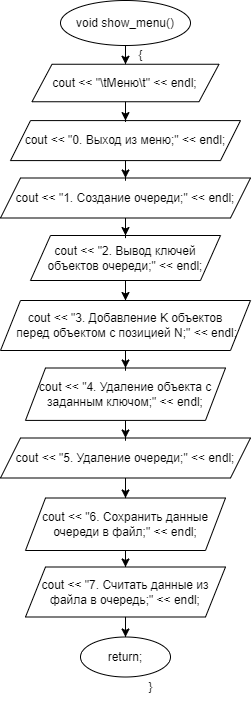


Рис.13. Вывод меню

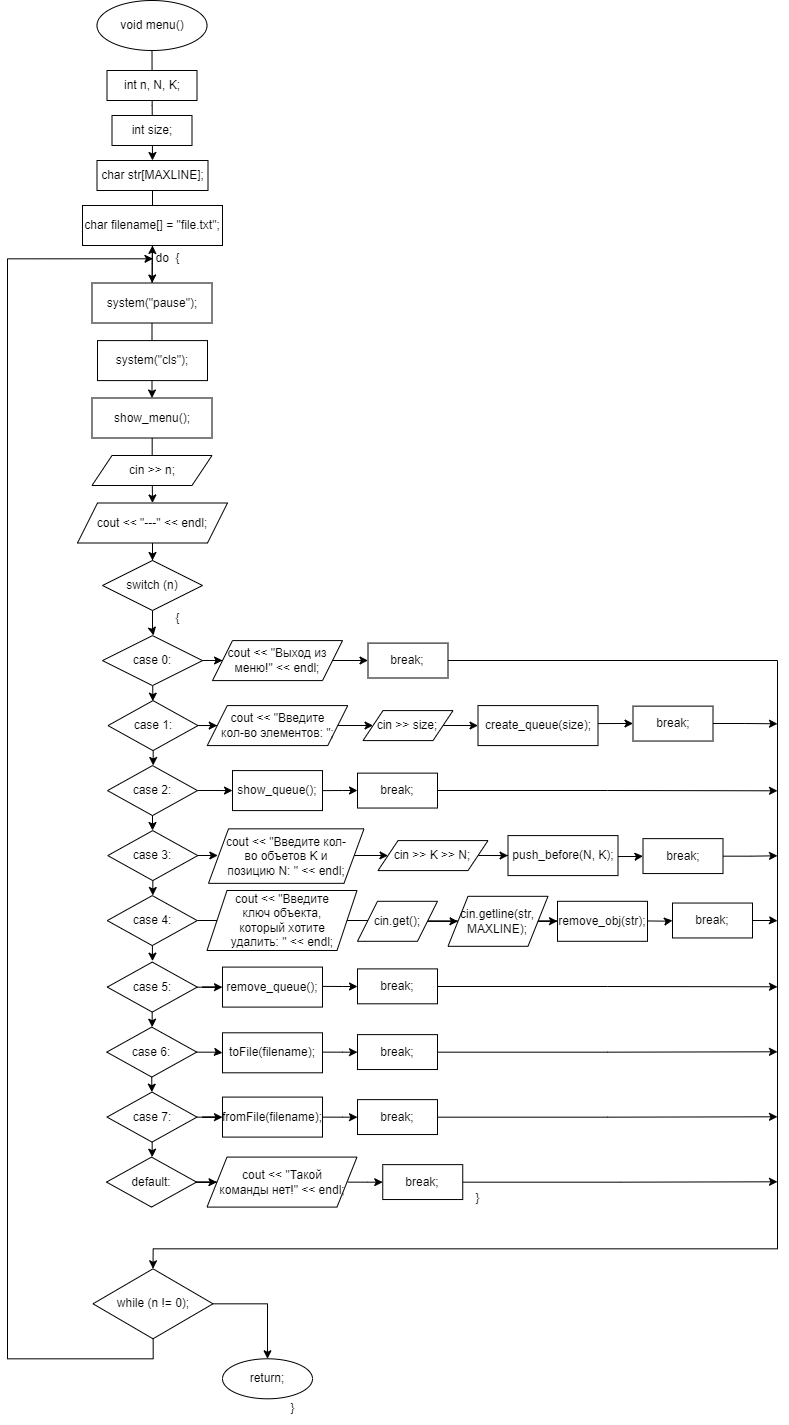


Рис.14. Реализация меню

Программный код

#define \_CRT\_SECURE\_NO\_WARNINGS //чтобы не жаловался

#include <iostream>

#include <stdio.h> //для работы с файлами

#include <windows.h> //для русского ввода-вывода штучки

using namespace std;

const int MAXLINE = 255; //макс длина строки

struct Point

{

char data[MAXLINE]; //строка, название и тд элемента очереди

Point\* next = nullptr; //следующий элемент очереди

};

Point\* head = nullptr; //указатель на первый элемент очереди

Point\* top = nullptr; //указатель на последний элемент очереди

size\_t size\_queue = 0;

void push(char\* str); //добавить новый элемент сверху

bool queue\_is\_empty(); //проверка на пустой стек

void create\_queue(int size)

{

if (size > 0 && queue\_is\_empty())

{

char str[MAXLINE];

cout << "Введите ключи " << size << " элементов очереди: " << endl;

cin.get();

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

{

cout << i << ". ";

gets\_s(str, MAXLINE);

push(str);

}

}

}

void push(char\* str) //добавить новый элемент сверху

{

if (queue\_is\_empty())

{

head = new Point;

strcpy(head->data, str);

top = head;

}

else

{

Point\* obj = new Point;

strcpy(obj->data, str);

top->next = obj;

top = obj;

}

size\_queue++;

}

void pop()

{

Point\* temp = head;

head = head->next;

size\_queue--;

delete temp;

}

bool queue\_is\_empty() //проверка на пустой стек

{

return size\_queue == 0;

}

void push\_before(int index, int k)

{

if (!queue\_is\_empty() && index >= 1 && index <= size\_queue && k > 0)

{

int i = 1;

while (i < index)

{

push(head->data);

pop();

i++;

}

char str[MAXLINE];

cout << "Введите элементы для вставки: " << endl;

cin.get();

for (int j = 1; j <= k; j++)

{

cin.getline(str, MAXLINE);

push(str);

}

for (i; i <= size\_queue - k; i++)

{

push(head->data);

pop();

}

}

else cout << "Очередь пуста или введен неверный индекс" << endl;

}

void output(Point\* obj)

{

cout << obj->data << endl;

if (obj == top) return;

output(obj->next);

}

void show\_queue() //вывод стека

{

if (queue\_is\_empty()) cout << "Очередь пуста" << endl;

else output(head);

}

void remove\_obj(char\* str)

{

if (queue\_is\_empty()) cout << "Очередь пуста, нечего удалять!" << endl;

int i = 1;

while (i <= size\_queue)

if (strcmp(head->data, str) == 0)

pop();

else

{

push(head->data);

pop();

i++;

}

}

void remove\_queue()

{

while (head != nullptr)

pop();

cout << size\_queue << " Очередь удалена!" << endl;

}

void toFile\_obj(Point\* obj, FILE\* file)

{

if (obj == nullptr) return;

fputs(obj->data, file);

fputs("\n", file);

toFile\_obj(obj->next, file);

}

void toFile(char\* filename) //загрузить в файл ключи стека

{

FILE\* file;

if ((file = fopen(filename, "wb")) == NULL) //ошибка открытия файла

{

cout << "Ошибка открытия файла" << endl;

exit(1);

}

cout << "Заносим данные в файл..." << endl;

toFile\_obj(head, file);

cout << "Данные успешно занесены!" << endl;

fclose(file);

}

void fromFile(char\* filename) //выгрузить из файла ключи списка

{

FILE\* file;

char row[MAXLINE];

if ((file = fopen(filename, "rb")) == NULL) //ошибка открытия файла

{

cout << "Ошибка открытия файла" << endl;

exit(2);

}

cout << "Считываем данные из файла..." << endl;

while (!feof(file) && fgets(row, MAXLINE, file))

{

row[strlen(row) - 1] = '\0'; //заносили с \n, нужно его убрать.

push(row);

}

fclose(file);

cout << "Данные успешно считаны!" << endl;

}

void show\_menu()

{

cout << "\tМеню\t" << endl;

cout << "0. Выход из меню;" << endl;

cout << "1. Создание очереди;" << endl;

cout << "2. Вывод ключей объектов очереди;" << endl;

cout << "3. Добавление K объектов перед объектом с позицией N;" << endl;

cout << "4. Удаление объекта с заданным ключом;" << endl;

cout << "5. Удаление очереди;" << endl;

cout << "6. Сохранить данные очереди в файл;" << endl;

cout << "7. Считать данные из файла в очередь;" << endl;

}

void menu()

{

int n, N, K;

int size;

char str[MAXLINE];

char filename[] = "file.txt";

do

{

system("pause");

system("cls");

show\_menu();

cin >> n;

cout << "---" << endl;

switch (n)

{

case 0: {

cout << "Выход из меню!" << endl;

break;

}

case 1: {

cout << "Введите кол-во элементов: ";

cin >> size;

create\_queue(size);

break; }

case 2: {

show\_queue();

break; }

case 3: {

cout << "Введите кол-во объетов K и позицию N: " << endl;

cin >> K >> N;

push\_before(N, K);

break;

}

case 4: {

cout << "Введите ключ объекта, который хотите удалить: " << endl;

cin.get();

cin.getline(str, MAXLINE);

remove\_obj(str);

break;

}

case 5: {remove\_queue(); break; }

case 6: {toFile(filename); break; }

case 7: {fromFile(filename); break; }

default: { cout << "Такой команды нет!" << endl; break; }

}

} while (n != 0);

}

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

menu();

return 0;

}

Результат работы

