Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

о лабораторной работе

Тема: «Очередь»

Выполнил

Студент группы РИС-22-2б

Прядеин И.А.

Проверил доцент кафедры

ИТАС

Полякова О. А.

Пермь 2023

**Постановка задачи**

Реализовать Очередь, используя структуру с помощью языка программирования C++.

**Исходный код программы:**

#include <iostream>

#include <string>

using namespace std;

struct Node

{

string key;

Node\* next = nullptr;

};

struct Queue

{

int size = 0;

Node\* head = nullptr;

Node\* tail = nullptr;

};

bool QueueIsEmpty(Queue\* queue)

{

return queue == nullptr || queue->size == 0;

}

void PushBack(Queue\* queue, string str)

{

if (!QueueIsEmpty(queue))

{

Node\* new\_node = new Node;

queue->size++;

new\_node->key = str;

new\_node->next = nullptr;

queue->tail->next = new\_node;

queue->tail = new\_node;

}

else

{

Node\* new\_node = new Node;

queue->size = 1;

new\_node->key = str;

queue->head = new\_node;

queue->tail = new\_node;

}

}

Queue\* CreateQueue(int size)

{

Queue\* queue = new Queue;

string str;

cout << "Enter values:\n"; getline(cin, str);

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

{

getline(cin, str);

PushBack(queue, str);

}

return queue;

}

void ShowQueue(Queue\* queue)

{

if (QueueIsEmpty(queue))

cout << "Queue is empty!" << endl;

else

{

Node\* temp = queue->head;

cout << "Begin -> ";

while (temp != nullptr)

{

cout << temp->key << " -> ";

temp = temp->next;

}

cout << "end\n";

}

}

void Pop(Queue\* queue)

{

Node\* temp = queue->head;

queue->head = temp->next;

queue->size--;

delete temp;

}

void RemoveElement(Queue\* queue, string k)

{

if (QueueIsEmpty(queue))

{

cout << "Queue is empty!" << endl;

return;

}

int i = 1;

while (i <= queue->size)

{

if (queue->head->key != k)

{

PushBack(queue, queue->head->key);

Pop(queue);

i++;

}

else

Pop(queue);

}

}

void AddElements(Queue\* queue, int count, int number)

{

int i = 1;

while (i < number)

{

PushBack(queue, queue->head->key);

Pop(queue);

i++;

}

string key;

getline(cin, key);

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

{

cout << "Enter new value: ";

getline(cin, key);

PushBack(queue, key);

}

for (i; i <= queue->size - count; i++)

{

PushBack(queue, queue->head->key);

Pop(queue);

}

}

void showMenu()

{

cout << "\tMenu\n";

cout << "Enter 0 to end the program\n";

cout << "Enter 1 to view the Queue\n";

cout << "Enter 2 to create a Queue\n";

cout << "Enter 3 to add the item to the begin of the Queue\n";

cout << "Enter 4 to add the item to the end of the Queue\n";

cout << "Enter 5 to add the items before element with given number\n";

cout << "Enter 6 to remove element with given value\n";

cout << "Enter 7 to clear Queue\n";

}

void menu()

{

int switch\_n, size\_queue, number, count;

string key;

Queue\* queue = new Queue;

do {

system("pause");

system("cls");

showMenu();

cin >> switch\_n;

switch(switch\_n)

{

case 0:

cout << "The program has ended";

break;

case 1:

ShowQueue(queue);

break;

case 2:

cout << "Enter size of Queue: ";

cin >> size\_queue;

while (size\_queue < 1)

{

cout << "Size does not exist, please enter new size: ";

cin >> size\_queue;

}

queue = CreateQueue(size\_queue);

break;

case 3:

AddElements(queue, 1, 1);

break;

case 4:

cout << "Enter new element: ";

getline(cin, key); getline(cin, key);

PushBack(queue, key);

break;

case 5:

cout << "Enter number: ";

cin >> number;

while (number < 1 || number > queue->size)

{

cout << "This number does not exist, please enter new number: ";

cin >> number;

}

cout << "Enter the number of input elements: ";

cin >> count;

while (count < 1)

{

cout << "This number of elements does not exist, please enter a new value";

cin >> count;

}

AddElements(queue, count, number);

break;

case 6:

cout << "Enter value: ";

getline(cin, key); getline(cin, key);

RemoveElement(queue, key);

break;

case 7:

while (!QueueIsEmpty(queue))

Pop(queue);

break;

}

}

while (switch\_n != 0);

}

int main()

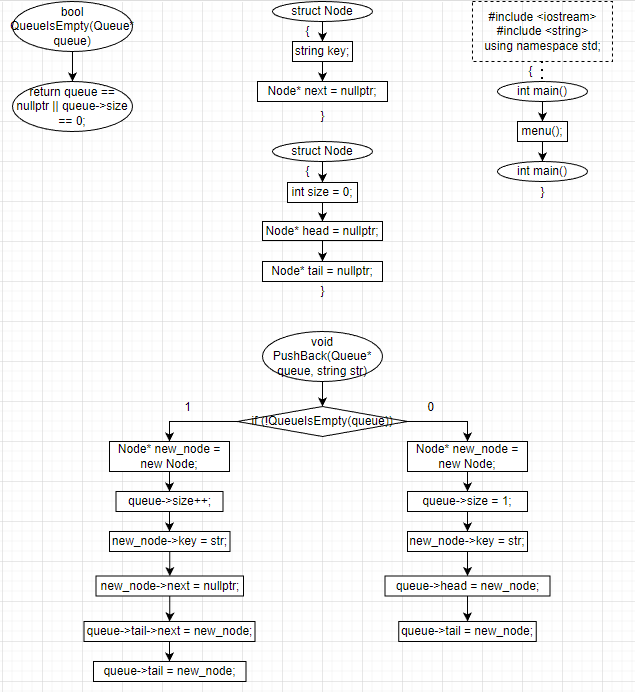
{

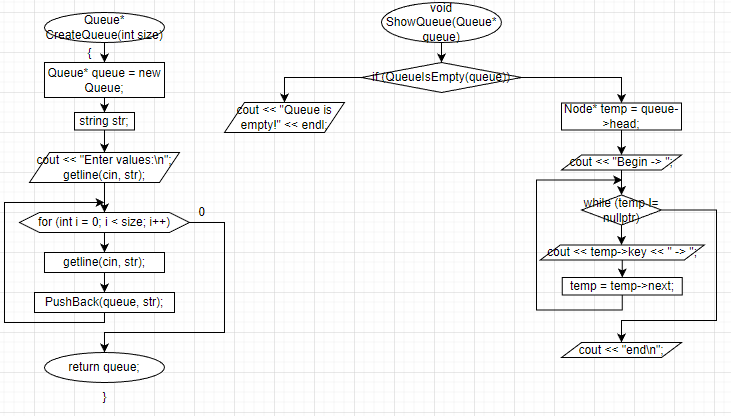
menu();

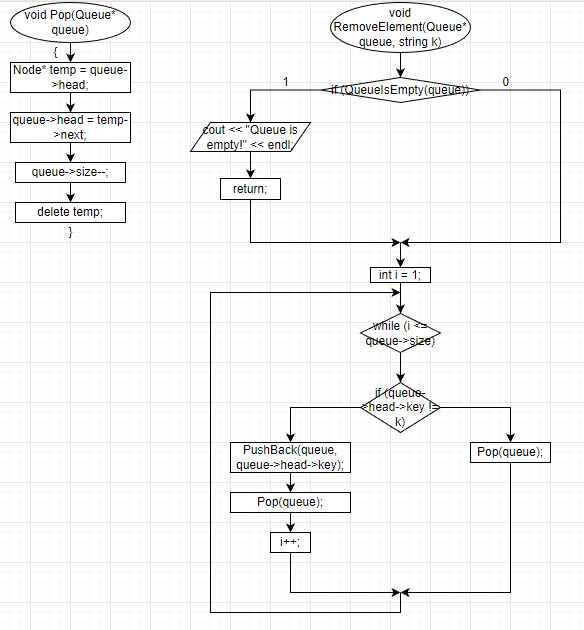
return 0;

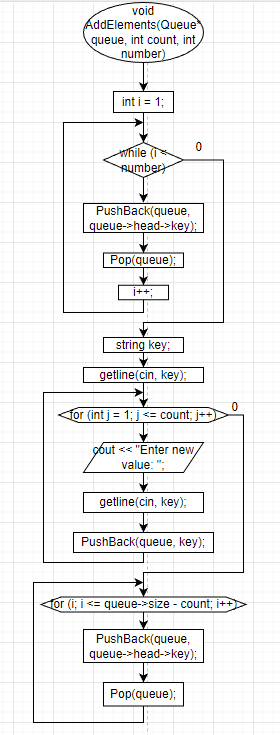
}

**Блок схема**









**Скриншоты результатов выполнения программы**

