**«Пермский национальный исследовательский  
политехнический университет»**

**О Т Ч Ё Т**

**по лабораторной работе №11.2**

Тема: Информационные и динамические структуры. Очереди

Вариант 7

Выполнил работу

студент группы ИВТ-20-2б

Копытов А.А.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

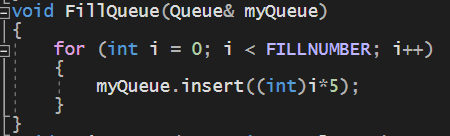
Пермь, 2021

**Постановка задачи**

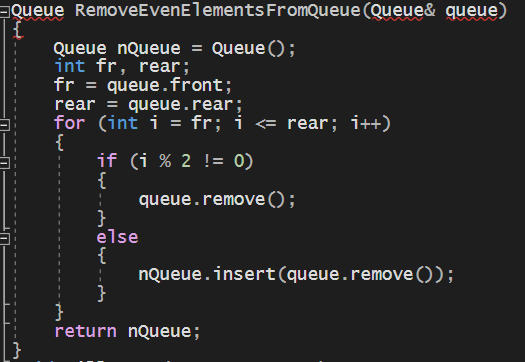
1. Сформировать информационное поле типа int.
2. Распечатать полученную очередь.
3. Удалить из очереди все элементы с четным информационным полем.
4. Распечатать получившуюся очередь.

**Анализ задачи**

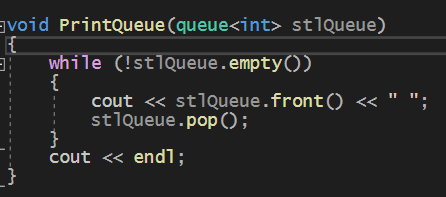
1. Для решения задачи необходимо…
   1. Создать функцию FillQueue типа void, которая заполняет очередь числами с шагом в 5 (т.е. на выходе получается очередь , состоящая из чисел 0, 5, 10, 15, 20, 25 и т.д);



* 1. Реализовать функцию RemoveEvenElementsFromQueue, которая проверяет каждый элемент очереди пошагово на четность. Если попадается четный элемент, то функция его удаляет;

****

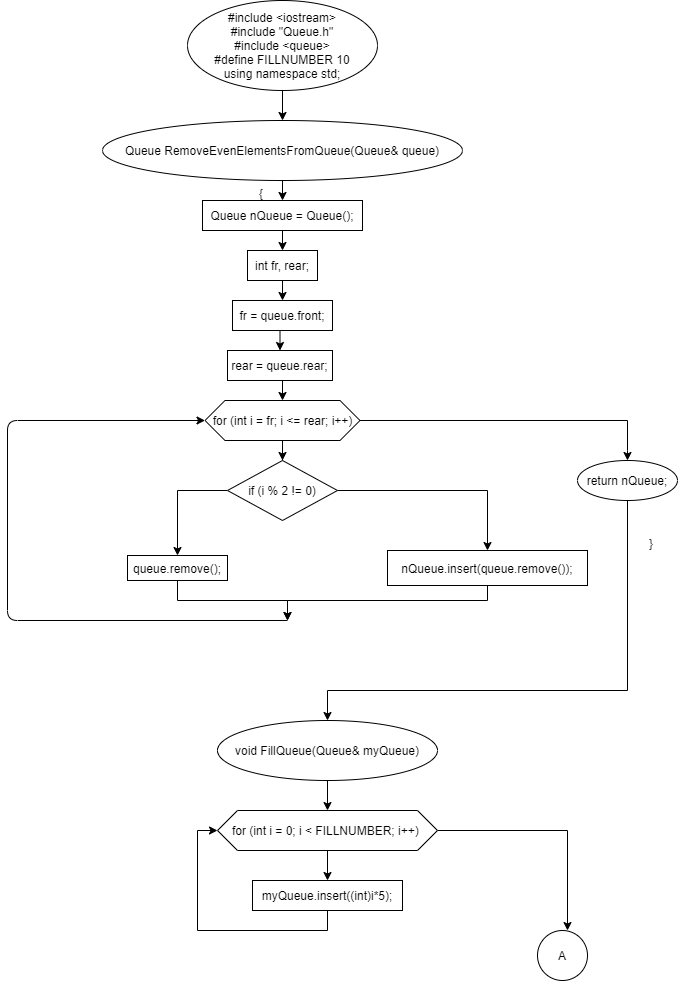
* 1. Реализовать функцию PrintQueue типа void, которая выводит в консоль очередь;

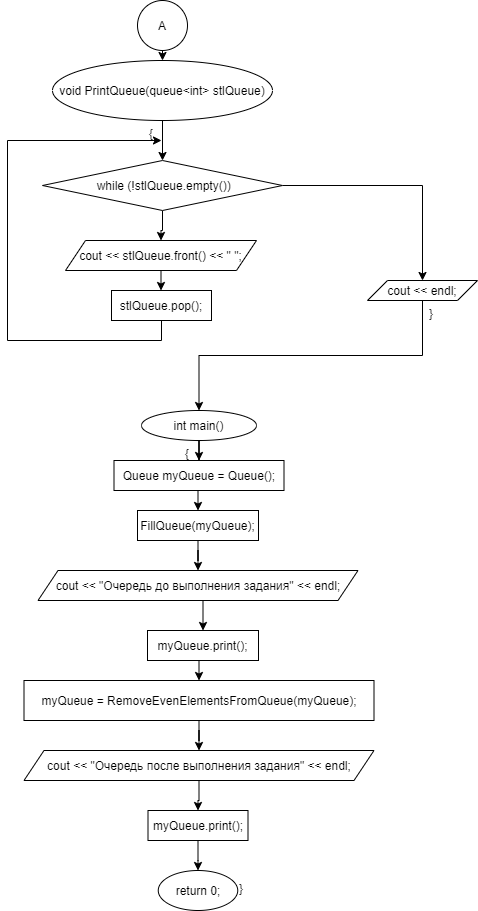


1. В программе были использованы следующие типы данных:
   1. Тип данных int для счетчика по циклу;



**Блок—схема программы**





**Решение**

#include <iostream>

#include "Queue.h"

#include <queue>

#define FILLNUMBER 10

using namespace std;

Queue RemoveEvenElementsFromQueue(Queue& queue)

{

Queue nQueue = Queue();

int fr, rear;

fr = queue.front;

rear = queue.rear;

for (int i = fr; i <= rear; i++)

{

if (i % 2 != 0)

{

queue.remove();

}

}

return nQueue;

}

void FillQueue(Queue& myQueue)

{

for (int i = 0; i < FILLNUMBER; i++) //100 элементов

{

myQueue.insert((int)i\*5);

}

}

void PrintQueue(queue<int> stlQueue)

{

//if(stlQueue.empty())

// cout << "Очередь пуста!" << endl;

while (!stlQueue.empty())

{

cout << stlQueue.front() << " ";

stlQueue.pop();

}

cout << endl;

}

int main() {

setlocale(LC\_ALL, "Russian");

Queue myQueue = Queue();

FillQueue(myQueue);

cout << "Очередь до выполнения задания" << endl;

myQueue.print();

myQueue = RemoveEvenElementsFromQueue(myQueue);

cout << "Очередь после выполнения задания" << endl;

myQueue.print();

return 0;

}

#pragma once

#include <iostream>

using namespace std;

#ifndef QUEUE\_QUEUE\_H

#define QUEUE\_QUEUE\_H

#define QMAX 100

class Queue

{

private:

int qu[QMAX];

public:

int rear, front;

Queue()

{

front = 1;

rear = 0;

return;

}

void insert(int x)

{

if (rear < QMAX - 1)

{

rear = rear + 1;

qu[rear] = x;

}

else

cout << "Очередь переполнена\n";

return;

}

bool isEmpty(Queue\* q)

{

if (rear < front) return 1;

else return 0;

}

void print()

{

/\*

int h;

if(isEmpty(this))

{

cout << "Очередь пуста!\n";

return;

}

for(h = front; h <= rear;h++)

{

cout << qu[h]<<endl;

}

return;

\*/

for (int i = front; i <= rear; i++)

{

cout << qu[i] << " ";

}

cout << endl;

}

int remove()

{

int x;

int h;

if (isEmpty(this))

{

cout << "Очередь пуста!\n";

return 0;

}

x = qu[front];

for (h = front; h < rear; h++)

{

qu[h] = qu[h + 1];

}

rear--;

return x;

}

};

#endif //QUEUE\_QUEUE\_H

**Скриншоты**

