Министерство образования и науки Российской Федерации   
Федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра «Информационные технологии и автоматизированные системы

**Отчёт   
по лабораторной работе №11.2**

Дисциплина: основы алгоритмизации и программирования

Тема: «Очереди»

Вариант №14

Выполнил работу:  
студент группы РИС-20-2б  
Вичугов Алексей Дмитриевич

Проверила:  
доцент кафедры ИТАС  
Полякова Ольга Андреевна

Пермь 2021

**Цели работы**

Получить практические навыки работы с очередями;

**Постановка задачи**

1. Сформировать очередь с помощью STL <queue>. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Сформировать очередь:   
      queue <int> q;
   2. Создать функцию, распечатывающую сформированную очередь:   
      void Observe(){

for (int i=0;i<n;i++){

cout << i+1 << ". " << q.front() << endl;

q.pop();

}

}

* 1. Удалить последний чётный элемент:  
     void Ubral(){

for (int i=0;i<n;i++) if (i!=u) q.push(a[i]);

if (u==-1) cout << "В очереди отсутствуют чётные элементы. " << endl;

else n--;

}

1. В ходе работы был использован тип данных int:  
   int n,e,u=-1;
2. Данные для решения задачи представлены в виде целочисленных переменных:  
   int n,e,u=-1;
3. Ввод данных в переменны производился с помощью оператора cin и функции push():  
   cin >> e;  
   q.push(e);
4. При решении поставленных задач использовались циклы:
   1. Для ввода данных и проверки правильности ввода цикл while:

while (n<2||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Неверный ввод! Элемент должен являться целым числом! Повторите ввод: " << endl;

cin >> n;

}

**Полный код программы**

#include <iostream>

#include <queue>

using namespace std;

queue <int> q;

int n,e,u=-1;

int\* a;

void Create(){

cout << "Количество элементов очереди: " ;

cin >> n;

while (n<2||cin.fail()){

cout << "Количество элементов должно являться числом и быть не мене 2! Повторите ввод: " ;

cin.clear();

cin.ignore(30,'\n');

cin >> n;

}

a = new int [n];

for (int i=0;i<n;i++){

cout << i+1 << "-й элемент очереди: " << endl;

cin >> e;

while (cin.fail()){

cout << "Введённое значение должно быть числом! Повторите ввод: " ;

cin.clear();

cin.ignore(30,'\n');

cin >> e;

}

q.push(e);

a[i]=e;

if (e%2==0) u=i;

}

}

void Observe(){

for (int i=0;i<n;i++){

cout << i+1 << ". " << q.front() << endl;

q.pop();

}

}

void Ubral(){

for (int i=0;i<n;i++) if (i!=u) q.push(a[i]);

if (u==-1) cout << "В очереди отсутствуют чётные элементы. " << endl;

else n--;

}

void Set(){

const int fon = system("Color F0");

system ("chcp 1251>nul");

}

int main(){

Set();

Create();

Observe();

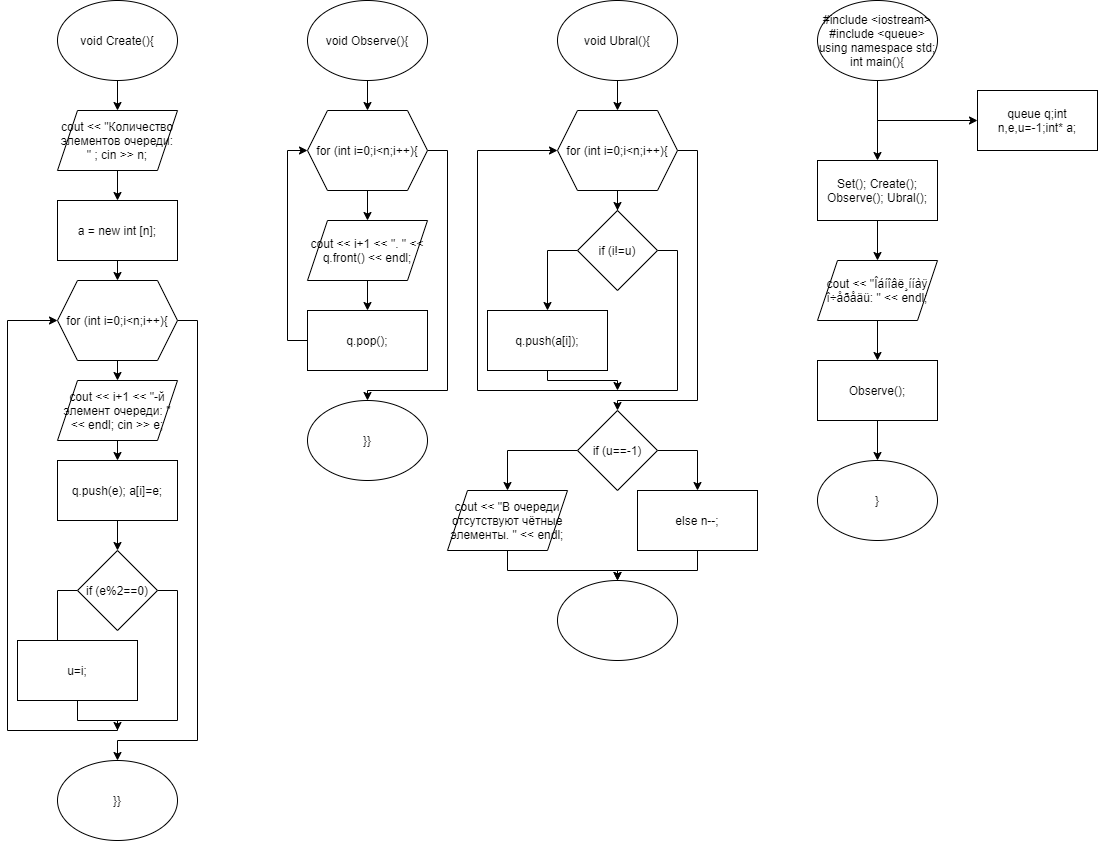
Ubral();

cout << "Обновлённая очередь: " << endl;

Observe();

}

**Блок-схема**



**Результаты тестов**

