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 №11.3**

Дисциплина: основы алгоритмизации и программирования

Тема: «Стеки»

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 со стеками;

**Постановка задачи**

1. Сформировать стек с помощью STL <stack>. 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 стек:   
      stack <char> s, b;
   2. Создать функцию, распечатывающую сформированный стек:   
      void Observe(){

int i=0;

while (!s.empty()){

cout << n-i++ << ". " << s.top() << endl;

b.push(s.top());

s.pop();

}

while (!b.empty()){

s.push(b.top());

b.pop();

}

}

* 1. Создать функцию, реализующую добавление элемента с заданным номером:  
     void Nov(){

int p;

cout << "Номер добавляемого элемента, начиная с первого пришедшего: ";

cin >> p;

while (p>n+1||p<1||cin.fail()){

cin.clear();

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

cout << "Номер добавляемого элемента должен иметь численное значение, быть положительным и не должен превышать размерность стека более чем на 1! Повторите ввод: ";

cin >> p;

}

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

b.push(s.top());

s.pop();

}

cout << "Значение нового элемента: " ;

cin >> e;

s.push(e);

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

s.push(b.top());

b.pop();

}

n++;

}

1. В ходе работы был использованы типы данных:
   1. int:  
      int n,i;
   2. char:  
      char e;
2. Данные для решения задачи представлены в виде:
   1. Целочисленных переменных:  
      int n,i;
   2. Символьных переменных:  
      char e;
3. Ввод данных в переменны производился с помощью оператора cin и функции push:  
   cin >> e;  
   s.push(e);
4. При решении поставленных задач использовались циклы:
   1. Для ввода данных и проверки правильности ввода цикл while:

while (n<1||cin.fail()){

cin.clear();

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

cout << "Размерность стека должна являться числом и иметь значение не менее 1! Повторите ввод: ";

cin >> n;

}**Полный код программы**

#include <iostream>

#include <stack>

using namespace std;

stack <char> s, b;

int n;

char e;

void Create(){

cout << "Размерность стека: " << endl;

cin >> n;

while (n<1||cin.fail()){

cin.clear();

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

cout << "Размерность стека должна являться числом и иметь значение не менее 1! Повторите ввод: ";

cin >> n;

}

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

cout << i+1 << "-й элемент: ";

cin >> e;

while (cin.fail()){

cout << "Элемент должен содержать только 1 символ! Повторите ввод! " << endl;

cin.clear();

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

cin >> e;

}

s.push(e);

}

}

void Observe(){

int i=0;

while (!s.empty()){

cout << n-i++ << ". " << s.top() << endl;

b.push(s.top());

s.pop();

}

while (!b.empty()){

s.push(b.top());

b.pop();

}

}

void Nov(){

int p;

cout << "Номер добавляемого элемента, начиная с первого пришедшего: ";

cin >> p;

while (p>n+1||p<1||cin.fail()){

cin.clear();

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

cout << "Номер добавляемого элемента должен иметь численное значение, быть положительным и не должен превышать размерность стека более чем на 1! Повторите ввод: ";

cin >> p;

}

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

b.push(s.top());

s.pop();

}

cout << "Значение нового элемента: " ;

cin >> e;

s.push(e);

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

s.push(b.top());

b.pop();

}

n++;

}

void Set(){

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

system("chcp 1251>nul");

}

int main(){

Set();

Create();

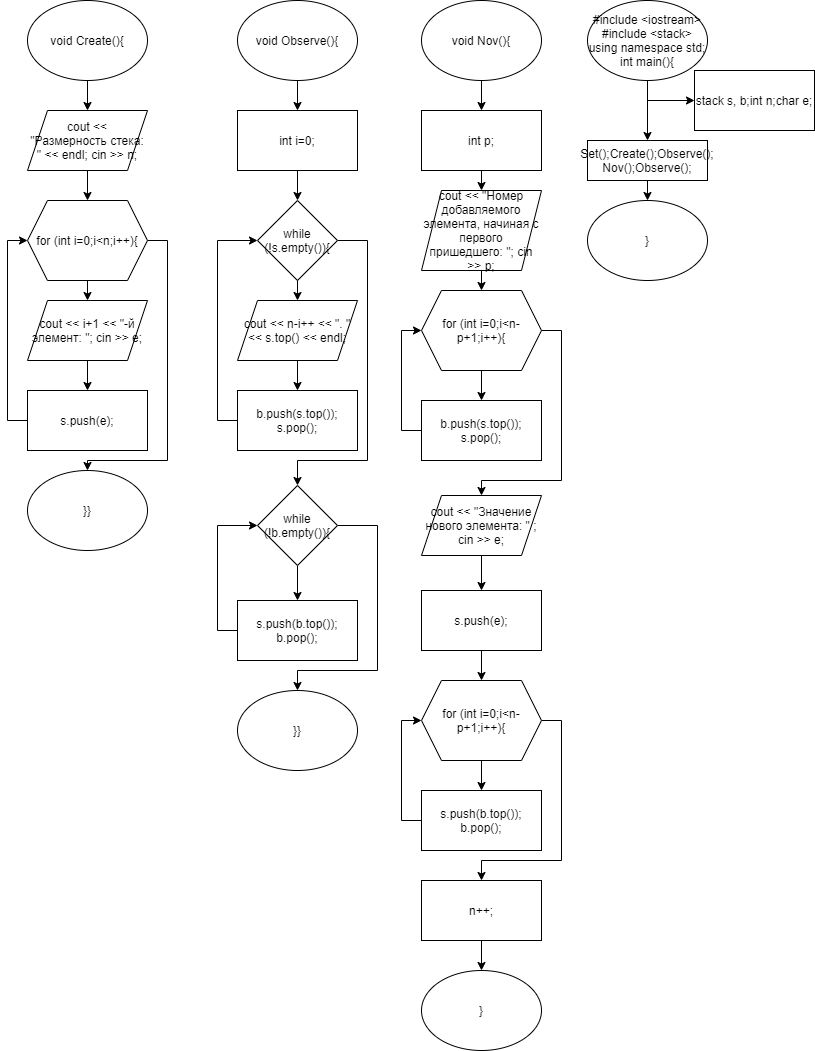
Observe();

Nov();

Observe();

}

**Блок-схема**



**Результаты тестов**

