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: Информационные и динамические структуры. Стеки

Вариант 1

Выполнил:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Получить практические навыки работы со стеками;

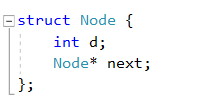
**Постановка задачи**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  варианта | Однонаправленный список | Двунаправленный список | Стек | Очередь |
| 1 | Тип информационного поля int.  Удалить из списка все элементы с четными информационными полями. | Тип информационного поля char\*.  Добавить в список элемент с заданным номером. | Тип информационного поля int.  Удалить из стека все элементы с четными информационными полями. | Тип информационного поля char\*.  Добавить в очередь элемент с заданным номером. |

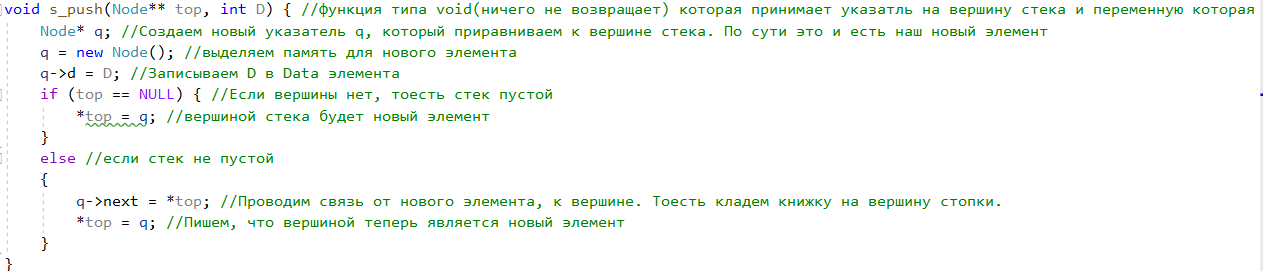
**Анализ задач**

**1.** Для решения задачи необходимо:

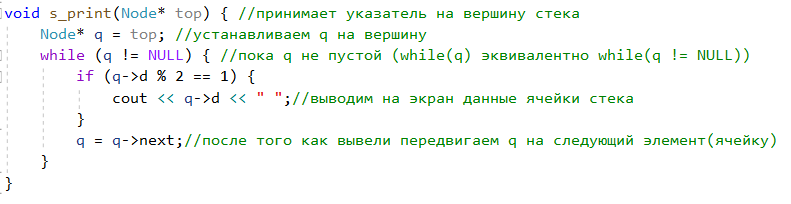
1.1.Создать структуру переменных, с указанием на следующий элемент:



**1.2.** Разработать функцию для создания стека



**1.3.**Разработать функцию print для вывода всех нечетных элементов стека:



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для функции s\_push используется входная переменная D целого и указатель на вершину стека



**2.2.** Для функции s\_print используется указатель на верхний элемент стека:



**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобной обработки элементов было решено представить элементы в виде стека

**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Ввод данных происходит благодаря функции cin:

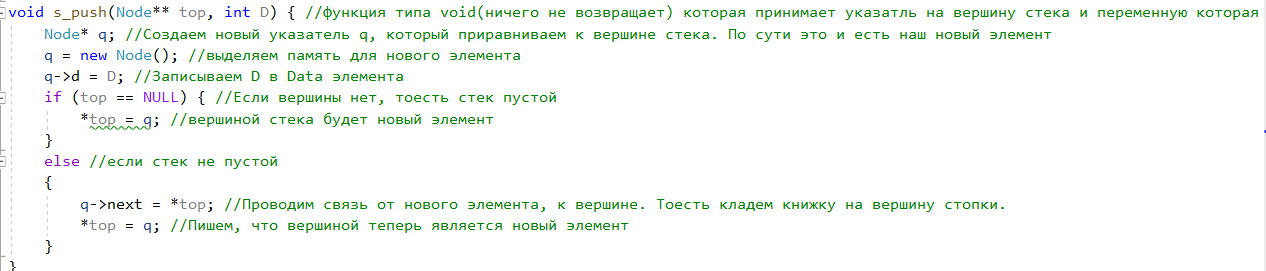


**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

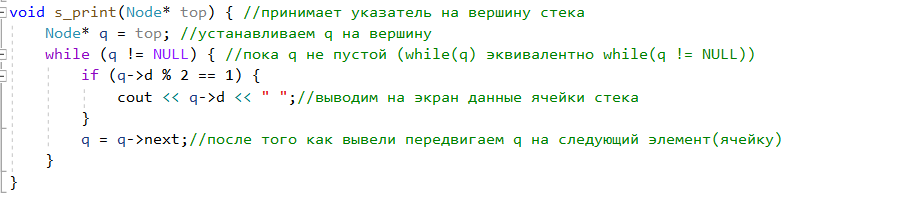


**5.**Поставленные задачи будут решены следующими действиями:

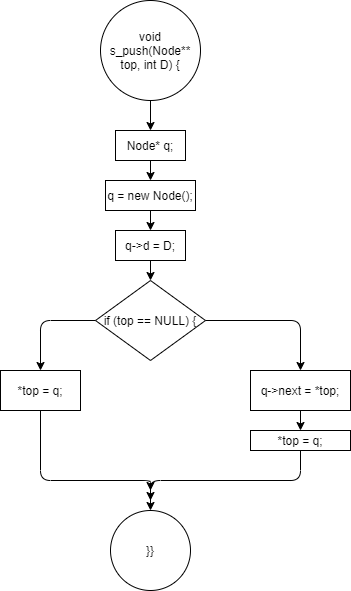
**5.1.** Создание стека происходит благодаря поочередному присваиванию указателю на вершину стека новым элементом:

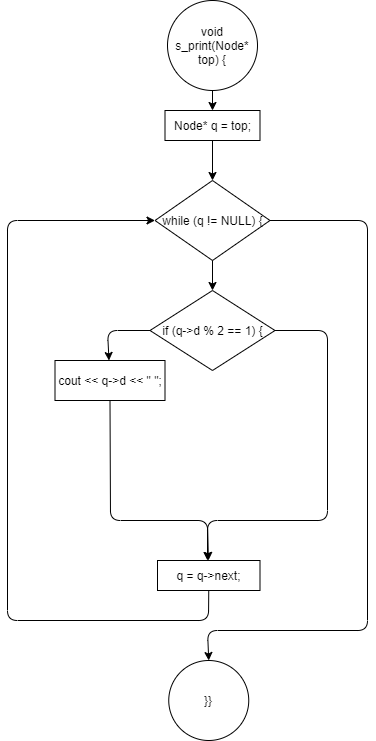


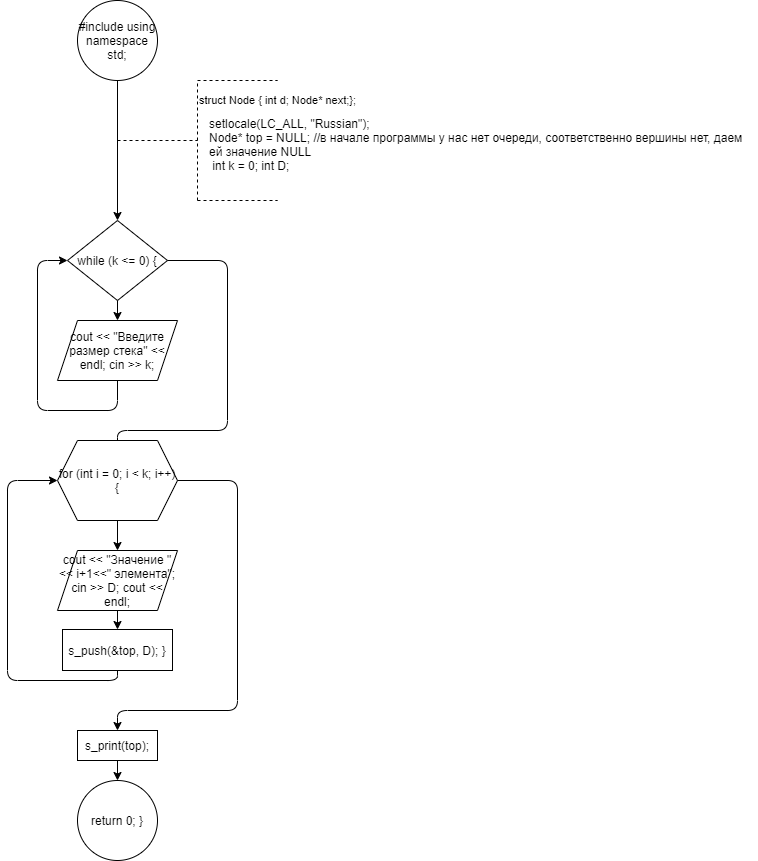
**5.2.**Вывод элементов стека реализован через вывод верхнего элемента с последовательным переходом к следующему элементу через цикл while:



**Блок-схема программы**







**Решение**

#include <iostream>

using namespace std;

struct Node {

int d;

Node\* next;

};

void s\_push(Node\*\* top, int D) { //функция типа void(ничего не возвращает) которая принимает указатль на вершину стека и переменную которая будет записываться в ячейку

Node\* q; //Создаем новый указатель q, который приравниваем к вершине стека. По сути это и есть наш новый элемент

q = new Node(); //выделяем память для нового элемента

q->d = D; //Записываем D в Data элемента

if (top == NULL) { //Если вершины нет, тоесть стек пустой

\*top = q; //вершиной стека будет новый элемент

}

else //если стек не пустой

{

q->next = \*top; //Проводим связь от нового элемента, к вершине. Тоесть кладем книжку на вершину стопки.

\*top = q; //Пишем, что вершиной теперь является новый элемент

}

}

void s\_print(Node\* top) { //принимает указатель на вершину стека

Node\* q = top; //устанавливаем q на вершину

while (q != NULL) { //пока q не пустой (while(q) эквивалентно while(q != NULL))

if (q->d % 2 == 1) {

cout << q->d << " ";//выводим на экран данные ячейки стека

}

q = q->next;//после того как вывели передвигаем q на следующий элемент(ячейку)

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

Node\* top = NULL; //в начале программы у нас нет очереди, соответственно вершины нет, даем ей значение NULL

int k = 0;

int D;

while (k <= 0) {

cout << "Введите размер стека" << endl;

cin >> k;

}

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

cout << "Значение " << i+1<<" элемента";

cin >> D;

cout << endl;

s\_push(&top, D);

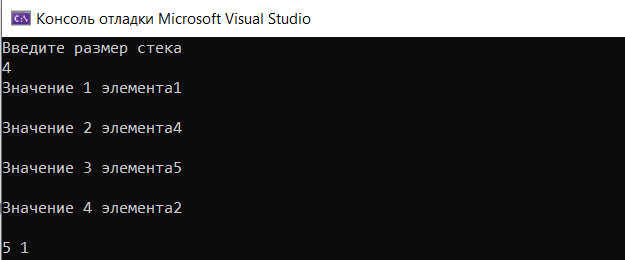
}

s\_print(top);

return 0;

}

**Скриншоты**



2))Стек(stl)

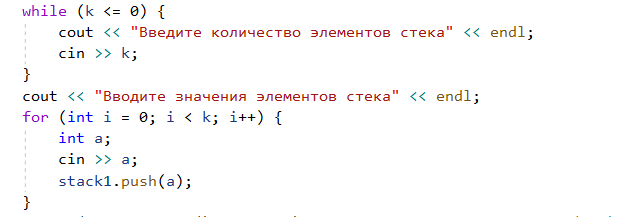
**Анализ задач**

**1.** Для решения задачи необходимо:

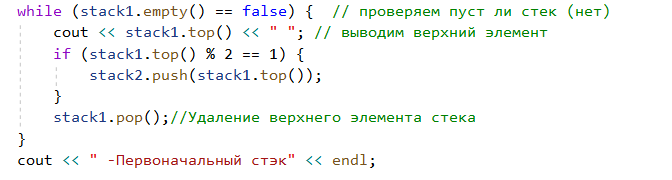
1.1.Создать структуру переменных



**1.2.** Разработать метод по заполнению стека



**1.3.**Разработать функцию для вывода всех нечетных элементов стека:



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для функции int main используется переменные целого типа данных int:



**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобной обработки элементов было решено представить элементы в виде стека

**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Ввод данных происходит благодаря функции cin:



**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

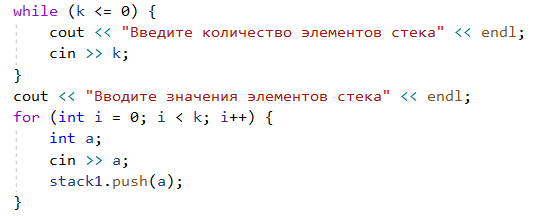


**5.**Поставленные задачи будут решены следующими действиями:

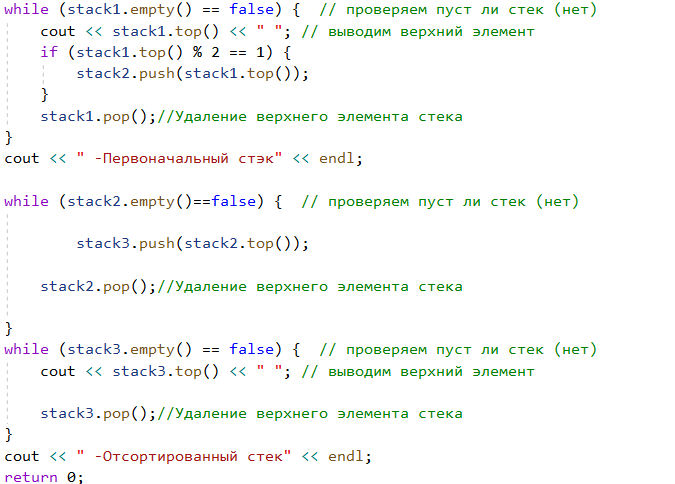
**5.1.** Создание стека происходит благодаря готовой функции по ее объявлению:



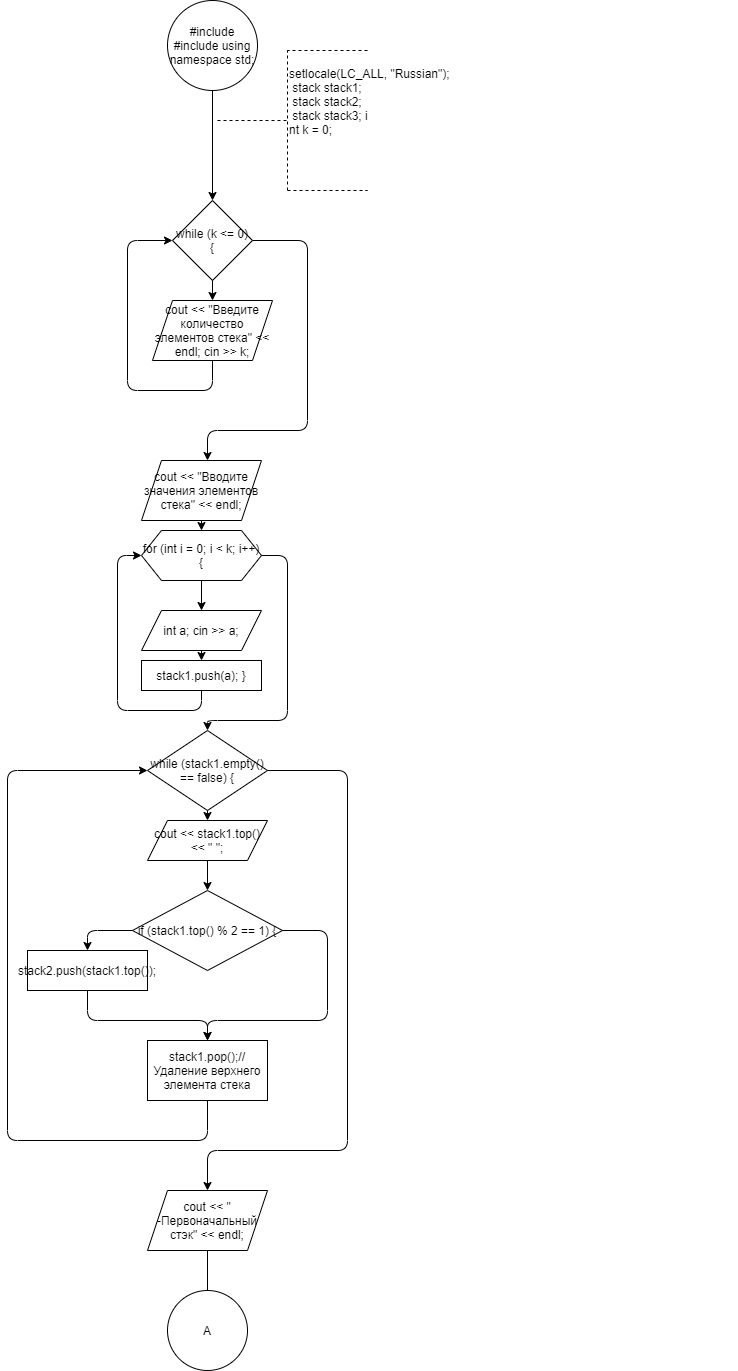
**5.2.**Добавление элементов в стек происходит благодаря циклу for принимающий значения стека до тех пор, пока не достигнет введенного пользователем значения:

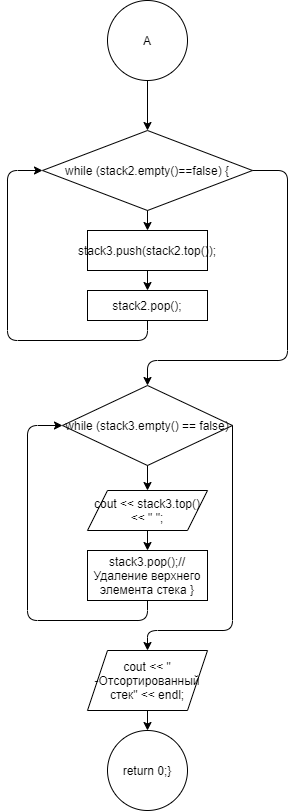


**5.3** Вывод элементов происходит благодаря готовой функции .top(),указывающий на последний элемент стека, и готовой функции .pop(), удаляющий последний элемент стека



**Блок-схема программы**





**Решение**

#include <iostream>

#include <stack>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

stack <int> stack1;

stack <int> stack2;

stack <int> stack3;

int k = 0;

while (k <= 0) {

cout << "Введите количество элементов стека" << endl;

cin >> k;

}

cout << "Вводите значения элементов стека" << endl;

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

int a;

cin >> a;

stack1.push(a);

}

while (stack1.empty() == false) { // проверяем пуст ли стек (нет)

cout << stack1.top() << " "; // выводим верхний элемент

if (stack1.top() % 2 == 1) {

stack2.push(stack1.top());

}

stack1.pop();//Удаление верхнего элемента стека

}

cout << " -Первоначальный стэк" << endl;

while (stack2.empty()==false) { // проверяем пуст ли стек (нет)

stack3.push(stack2.top());

stack2.pop();//Удаление верхнего элемента стека

}

while (stack3.empty() == false) { // проверяем пуст ли стек (нет)

cout << stack3.top() << " "; // выводим верхний элемент

stack3.pop();//Удаление верхнего элемента стека

}

cout << " -Отсортированный стек" << endl;

return 0;

}

**Скриншоты**

