

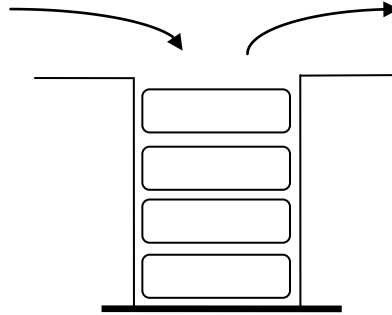
## Лабораторная работа №1

**Тема:** стеки и очереди

### **Стек**

Стек – это динамически изменяемый упорядоченный набор элементов. Данные помещаются в стек и удаляются из стека по следующему правилу: выборка элементов из стека выполняется в порядке, обратном их засылке.

Другими словами стек описывается правилом: последним пришел, первым ушел.



Основными операциями над стеками являются:

- занести элемент в стек;
- забрать элемент из стека;
- получить значение элемента, занесенного в стек последним;
- проверить, не является ли стек пустым.

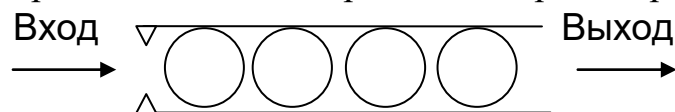
Дополнительные операции: Создание и удаление стека.

### **Очередь**

Очередь – это динамически изменяемый упорядоченный набор элементов.

Данные помещаются в очередь и удаляются из очереди в порядке поступления.

Другими словами очередь описывается правилом: первым пришел, первым ушел.



Основными операциями над очередью являются:

- занести в очередь новый элемент;
- забрать элемент из очереди;
- получить значение элемента, стоящего первым в очереди;
- проверить, не является ли очередь пустой.

Дополнительные операции: создание и освобождение очереди.

В первой лабораторной работе АТД Стек и Очередь необходимо реализовать на массиве.

### Пример решения задачи

Задача. Прочитать с клавиатуры два числа и вывести их в обратном порядке. Для решения задачи использовать стек. Для реализации стека использовать массив.

Решение

Алгоритм решения задачи

1. Ввести число с клавиатуры.
2. Положить введенное число в стек (операция Push).
3. Ввести второе число с клавиатуры.
4. Положить введенное число в стек (операция Push).
5. Взять число из стека и вывести его на экран (операция Pop).
6. Взять число из стека и вывести его на экран (операция Pop).

Программная реализация

Последовательность действий при программной реализации:

1. Задать массив необходимо типа и размера, на котором будет реализована структура данных типа стек.
2. Реализовать алгоритм решения поставленной задачи.
3. Реализовать основные функции работы со стеком (Push, Pop, IsEmpty, Top).

В результате получили следующий код:

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

#define SIZE 100
char Stack[SIZE];
int t;

void init();    // инициализация стека
void Push(int); // положить в стек
int Pop();      // забрать из стека
int Top();      // получить значение верхнего элемента стека
bool IsEmpty(); // проверить, является ли стек пустым

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int k;
    init();           // инициализация стека
    printf("Введите целое число\n");
    scanf("%d", &k);
    Push(k);          // положить введенное число в стек
```

```

printf("Введите целое число\n");
scanf("%d", &k);
Push(k);           // положить введенное число в стек
printf("Введено число %d\n", Pop()); // забрать число из стека
                                   // и вывести на печать
printf("Введено число %d\n", Pop()); // забрать число из стека
                                   // и вывести на печать
if(IsEmpty()==true) printf("Stack пустой\n"); // проверить,
                                   // остались ли числа в стеке
else printf("Stack не пустой\n");
return 0;
}

void init()        // инициализация стека
{
    t = -1;
}

void Push(int a) // положить в стек
{
    if(t==SIZE) // проверка на попытку занесения элемента в полный
стек
    {
        printf("Попытка занесения значения в полный стек");
        exit(EXIT_FAILURE);
    }
    else
    {
        t = t+1;
        Stack[t] = a;
    }
}

int Pop()          // забрать из стека
{
    if(t == -1) // проверка на попытку забрать элемент из пустого
стека
    {
        printf("Попытка чтения из пустого стека");
        exit(EXIT_FAILURE);
    }
    else
    {
        int val = 0;
        val = Stack[t];
        t = t-1;
        return(val);
    }
}

```

```

    }
}

int Top()          // получить значение верхнего элемента стека
{
    return(Stack[t]);
}

bool IsEmpty() // проверить, является ли стек пустым
{
    if(t == -1)
        return(true);
    else
        return(false);
}

```

Результаты тестирования программы:

Введите целое число

2

Введите целое число

5

Введено число 5

Введено число 2

Stack пустой

Process returned 0 (0x0) execution time : 9.888 s

Press any key to continue.

Отчет по лабораторной работе должен содержать:

- Титульный лист
- Задание на лабораторную работу
- Алгоритм решения задачи
- Листинг программы
- Пример решения задачи
- Выводы

### **Контрольные вопросы**

1. Понятие стека. Операции, выполняемые над стеком.
2. Понятие очереди. Операции, выполняемые над очередью.
3. Понятие дека. Операции, выполняемые над деком.
4. Представление стека с помощью массива. Выполнение основных операций.
5. Представление очереди с помощью массива. Выполнение основных операций.