

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

ПРЕПОДАВАТЕЛЬ:

Канд. техн. наук		Л.Н.Бариков
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №15

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: СТЕК, ДЕК, ОЧЕРЕДЬ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4143		Е.Д.Тегай
		подпись, дата	инициалы, фамилия

## **Цель работы**

Изучение способов создания и принципов использования динамических структур данных типа стек, дек, очередь; изучение стандартных средств языка C/C++ для работы с динамической памятью; совершенствование навыков модульного программирования на языке C/C++ при решении задач обработки динамических структур данных; совершенствование способов разработки многофайловых проектов.

## **Задание на программирование**

Используя технологию процедурного программирования разработать программу обработки данных, содержащихся в заранее подготовленном файле, в соответствии с индивидуальным заданием. Применить динамическую структуру указанного в задании вида: стек, очередь или дек.

## **Порядок выполнения работы**

1. Получить у преподавателя индивидуальное задание.
2. Разработать математическую модель: описать с помощью формул и рисунков вид используемой динамической структуры и процессы ее создания и использования.
3. Построить схему алгоритма решения задачи.
4. Использовать функции, реализующие полный набор операций для этой структуры: – допустимые операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала; – допустимые операции для очереди: инициализация, проверка на пустоту, добавление нового элемента в конец, извлечение элемента из начала; – допустимые операции для дека: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.
5. Создать многофайловый проект на языке C/C++.

6. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов. Обеспечить одновременный показ на экране входной и выходной информации.
7. Выходные данные должны выводиться на экран с пояснениями. Операторы вывода результатов работы должны находиться либо в функции `main()`, либо в специальной функции вывода результата, вызов которой осуществляется из функции `main()`.
8. Оформить отчет о работе в составе: постановка задачи, математическая модель, схема алгоритма решения основной (основных) задач, текст программы, контрольные примеры.
9. Текст программы в отчете не должен представлять из себя скриншот.
10. Скриншоты тестов должны легко читаться. Все их неинформативные части должны быть удалены.

### **Индивидуальное задание**

Дан текстовый файл. Используя стек, за один просмотр файла сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т. д.

### **Ход работы**

Проект состоит из нескольких файлов:

- Заголовочный файл *mods.h*, который содержит в себе определения новых типов и объявления функций:
- Исходный файл *mods.cpp*, который содержит в себе реализацию набора функций для обработки списка;
- Основной файл *stek.cpp*, который содержит в себе функцию *main()*.

Также выполняются следующие функции:

- Функция инициализации;
- Функция проверки на пустоту;
- Функция добавления элемента в стек;
- Функция извлечения элемента из стека.

## Текст программы

### *Основной файл stek.cpp*

```
#include<iostream>

#include"mods.h"

#include<fstream>

#include<string>

#include<locale.h>

using namespace std;

int main()
{
    setlocale(LC_ALL, "Rus");
    stek* head;
    t_elem ch;

    head = init_stek();           //инициализация стека
    char path[255];

    cout << "Введите, пожалуйста, путь до файла: " << endl;
    cin >> path;

    ifstream file(path);
    ofstream fin("D:\\FILE.txt", ios_base::out | ios_base::trunc);

    if (!file.is_open())
    {
        cout << "Ошибка: файл не существует!\n\n";
        return -1;
    }

    cout << "\n Полученный результат:" << endl;

    while (getline(file,ch))
    {
        head= push_stek(ch, head);
    }

    while (!pu_stek(head))
    {
        head=pop_stek( head, ch);
        fin << ch << endl;
    }
}
```

```

    }

    fin.close();
    file.close();

    ifstream fout("D:\\FILE.txt");

    while (getline(fout, ch))
    {
        cout << ch << endl;
    }
    fout.close();
    return(0);
}

```

### ***Заголовочный файл mods.h***

//Объявления типов

#pragma once

```

typedef string t_elem;           //тип информационной части элемента стека
struct stek                     //тип элемента стека
{
    t_elem data;                //поле данных
    stek* link;                 //поле адреса
};

//Объявления (прототипы) функций
stek* init_stek();              //инициализация стека
int pu_stek(stek*);             //проверка стека на пустоту
stek* push_stek(t_elem, stek*); //добавление элемента в стек
stek* pop_stek(stek*, t_elem&); //извлечение элемента из стека

```

### ***Исходный файл mods.cpp***

#include<iostream>

#include"mods.h"

#include<locale.h>

//Занесение нового элемента в стек

```

stek* push_stek(t_elem ch, stek* head)
{
    stek* nov = new(stek); //выделение памяти под элемент стека
    nov->data = ch;         //заполнение полей
    nov->link = head;
    head = nov;             //указатель на начало стека
    return(head);
}

```

```

//Извлечение элемента из стека
stek* pop_stek(stek* head, t_elem& ch)
{
    stek* tec = head->link;
    ch = head->data;      //копирование информационной части головного элемента
    delete(head);        //освобождение памяти
    return(tec);
}
//Инициализация стека
stek* init_stek()
{
    return(NULL); //в исходном состоянии стек пуст
}
//Проверка стека на пустоту
int pu_stek(stek* head)
{
    return(head == NULL); //если стек пуст - 0, иначе - не 0
}

```

## Результаты работы программы

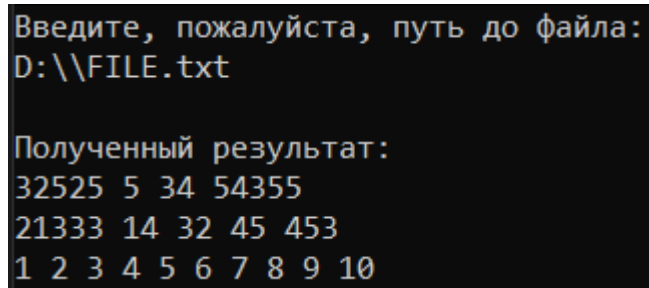
*Исходный файл имел вид:*

1 2 3 4 5 6 7 8 9 10

21333 14 32 45 453

32525 5 34 54355

Полученный результат показан на рисунке 1.



```

Введите, пожалуйста, путь до файла:
D:\\\\FILE.txt

Полученный результат:
32525 5 34 54355
21333 14 32 45 453
1 2 3 4 5 6 7 8 9 10

```

Рисунок 1 – Результат работы программы

*Исходный файл имел вид:*

55 32 41 8 9 10

11 12 13 46 5dfgdf

dfg gd d eg d 43 g

Полученный результат показан на рисунке 2.

```
Введите, пожалуйста, путь до файла:  
D:\\FILE.txt  
  
Полученный результат:  
dfg gd d eg d 43 g  
11 12 13 46 5dfgdf  
55 32 41 8 9 10
```

Рисунок 2 – Результат работы программы

## **Вывод**

В данной лабораторной работе были изучены способы создания и принципы использования динамических структур данных типа стек, дек, очередь; изучены стандартные средства языка C/C++ для работы с динамической памятью; усовершенствованы навыки модульного программирования на языке C/C++ при решении задач обработки динамических структур данных; усовершенствованы способы разработки многофайловых проектов.