МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №15

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Стек, дек, очередь»

Работу выполнил

студент гр. 4941 Н.С. Горбунов

Санкт-Петербург

2020

**Цель лабораторной 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 её 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 лабораторной работе в составе: постановка задачи, математическая модель, схема алгоритма решения основной (основных) задач, текст программы, контрольные примеры.

**Вариант 5**

Даны три стержня и*n* дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести*n* дисков со стержня*A*на стержень*C***,** сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила:

на каждом шаге со стержня на стержень переносить только один диск;

диск нельзя помещать на диск меньшего размера;

для промежуточного хранения можно использовать стержень*B***.**

Реализовать алгоритм, используя три ***стека*** вместо стержней*A***,** *B***,** *C***.** Информация о дисках хранится в исходном файле.

**Текст программы**

//Текст файла Stek.h

#pragma once

typedef int elem;

class Stek

{

private:

class mem

{

public:

elem data;

mem\* link;

mem(mem\* newLink, elem newData)

{

data = newData;

link = newLink;

}

};

mem\* head;

public:

Stek();

void NewMember(elem newData);

void showStek();

elem Take();

};

//Текст файла Stek.cpp

#include "Stek.h"

#include <iostream>

Stek::Stek()

{

this->head = NULL;

}

void Stek::NewMember(elem newData) //добавить новый элемент в стек

{

mem\* select = this->head; // переменная выбранного элемента стека

if (head != NULL) // если стек не пустой

{

head = new mem(select, newData); // вставляем в начало стека

}

else head = new mem(NULL, newData); // иначе вставляем значение в первую ячейку

}

void Stek::showStek() //вывод стека в консоль

{

std::cout << "\n";

if (head != NULL) // если стек не пустой

{

mem\* select = this->head; // переменная выбранного элемента списка

do

{

std::cout << select->data << "\n"; //вывести значение ячейки

select = select->link; // перейти на следующую

} while (select != NULL); // пока есть элемент

}

else std::cout << "NULL"; // иначе вывести в консоль NULL

}

elem Stek::Take()

{

mem\* del = head;

elem data = head->data;

if (head->link != NULL)

{

head = head->link; // смещаем 1 элемент на 0 место

}

else

{

head = NULL;

}

delete del; // удаляем бывший 0 элемент

return data;

}

//файл source.spp

#include<stdio.h>

#include<iostream>

#include<string>

#include<stdlib.h>

#include <sstream>

#include <fstream>

#include<locale.h>

#include "Stek.h"

using namespace std;

int ReadFile(char\* fname, Stek& stek) //вывод содержимого файла на печать

{

int n = 0;

stringstream stream;

ifstream input(fname);

string s;

while (getline (input, s))

{

stream << s << " ";

}

stream << ".";

int ch;

while (stream >> ch)

{

stek.NewMember(ch);

n++;

}

return n;

}

void WriteFile(char\* fname, Stek& stek, int n) //выводи стек в файл в обратном порядке

{

ofstream output(fname);

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

{

output << stek.Take() << " ";

}

}void Exchange(Stek& First, Stek& Second) //переложить из первого во второй

{

Second.NewMember(First.Take());

}

int ExchangeAll(Stek& first, Stek& second, Stek& third, int N)

{

if (N == 1) // если диск всего один

{

Exchange(first, third); // переносим диск со стрежня А на стержень С

return 0;

}

else

{

//переносим все диски, кроме самого большого со стержня А на стежень В

ExchangeAll(first, third, second, N - 1);

// Переносим самый большой диск со стержня A на стержень C

Exchange(first, third);

// Переносим все диски со стержня B на стержень C поверх самого большого диска

ExchangeAll(second, first, third, N - 1);

return 0;

}

}

int main()

{

char fname\_i[] = "F:\\input.txt";

char fname\_o[] = "F:\\output.txt";

int n;

setlocale(LC\_ALL, "Russian");

Stek A = Stek();

Stek B = Stek();

Stek C = Stek();

n = ReadFile(fname\_i, A); // ввод данных в стек из файла

//вывод изначальных стеков

cout << "\n Стержень (стек) А: \n";

A.showStek();

cout << "\n Стержень (стек) В: \n";

B.showStek();

cout << "\n Стержень (стек) С: \n";

C.showStek();

ExchangeAll(A, B, C, n);

//вывод стеков после перемещения

cout << "\n После перестановки:";

cout << "\n Стержень (стек) А: \n";

A.showStek();

cout << "\n Стержень (стек) В: \n";

B.showStek();

cout << "\n Стержень (стек) С: \n";

C.showStek();

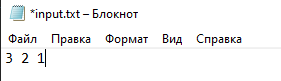
WriteFile(fname\_o, C, n); // вывод стека в файл

}

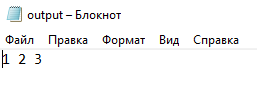
**Скриншоты работы программы**

Пример 1

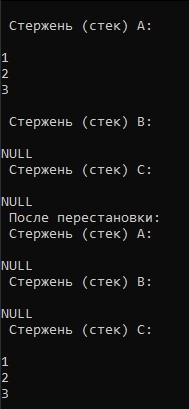
Входной файл:

****

Выходной файл:

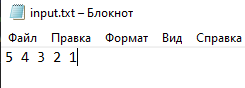


Консоль:

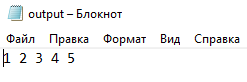


Пример 2

Входной файл:



Выходной файл:



Консоль:

