МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

**«Структура хранения мультистека»**

**Выполнил:** студент группы 381706-1

Кольтюшкина Янина Вадимовна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc1069359)

[2. Постановка задачи 4](#_Toc1069360)

[3. Руководство пользователя 5](#_Toc1069361)

[4. Руководство программиста 7](#_Toc1069362)

[4.1. Описание структуры программы 7](#_Toc1069363)

[4.2. Описание структур данных 7](#_Toc1069364)

[4.3. Описание алгоритмов 8](#_Toc1069365)

[5. Заключение 10](#_Toc1069366)

# Введение

**Целью** данной лабораторной работы является практическое освоение динамтической структуры данных – мультистека на массиве.

Разберемся с несколькими базовыми понятиями.

**Мультистек** – это структура данных, представляющая собой упорядоченный набор из N стеков фиксированного размера. Хранятся они последовательно друг за другом. Каждый из стеков реализован по одному и тому же принципу (LIFO: last in – first out или последним пришел – первым ушел). Более подробно об устройстве одного стека уже говорилось ранее. Нам важно помнить, что доступ мы будем иметь только к элементу, который добавили последним.

Стеки хранятся в памяти единым блоком друг за другом. Размер такого блока m.

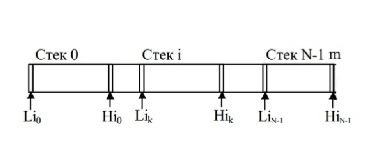


Рисунок 1 Мультистек в памяти компьютера

Здесь, – индекс начала j-го стека, – индекс последнего элемент j-го стека.

Некоторые свойства такой структуры памяти:

1. Условие неподвижности первого стека: ;
2. Условие пустоты: ;
3. Условие неперекрытия: ;
4. Условие переполнения: ;

# Постановка задачи

Важной задачей данной лабораторной работы является эффективная реализация структуры данных мультистека на массиве.

С программной точки зрения это будет значить следующее:

1. Разработка и реализация вспомогательного класса стека TNewStack, а так же класса мультистека TMStack
2. Пример программы, демонстрирующий работу класса TMStack.
3. Написание набора автоматических тестов с использованием Google C++ Testing Framework и проверка работоспособности методов классов.
4. Класс для обработки исключений – MyException.

# Руководство пользователя

При запуске программы пользователю предлагается ввести общее число элементов в будущем мультистеке. После чего количество стеков в нем.

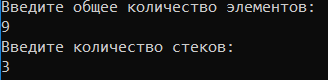


Рисунок 2 Задание параметров мультистека

Далее, мультистек заполняется значениями от 1 до введенного пользователем значения. Каждое действие программа выводит на экран для того, чтобы в дальнейшем можно было проверить правильность заполнения стеков. Закончив с этим, на экран выводится весь мультистек с заданными пользователем параметрами.

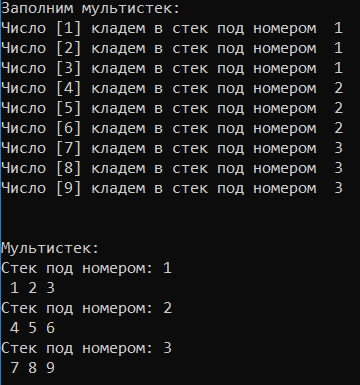


Рисунок 3 Заполнение мультистека значениями и вывод его на экран

Затем, пользователь может выбрать стек, из которого он бы хотел взять один элемент. Значение этого элемента выводится на экран вместе с измененным мультистеком.

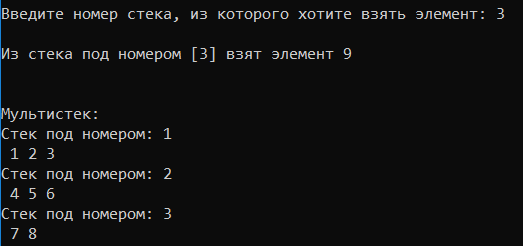


Рисунок 4 Извлечение элемента из мультистека

После всего этого, пользователю предлагается ввести новый элемент в произвольный стек. Для этого его просят ввести номер стека, куда бы он хотел поместить свой элемент и его значение. Если пользователь выберет стек, который уже был заполнен, то память в мультистеке перепакуется.

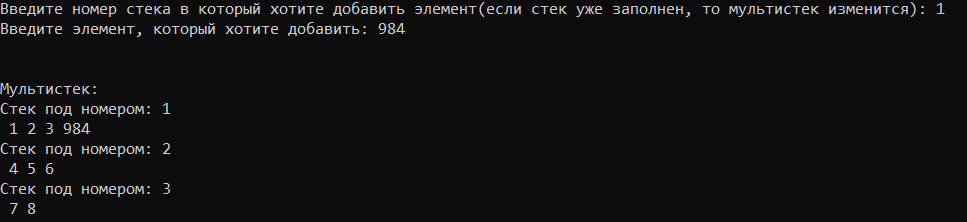


Рисунок 5 Добавление элемента в мультистек

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль MultiStack. Содержит пример использования мультистека. Реализация в файле MultiStack*\_main.cpp.*
* Модуль MultiStackLib - статическая библиотека. Содержит файл NewStack.h, в котором описан интерфейс и реализация шаблонного вспомогательного класса *TNewStack.* А так же файл MultiStack.h, в котором описан интерфейс и реализация шаблонного класса *TMStack*.
* Модуль MultiStackTest. Содержит 28 тестов, описанных в файле MultiStack*Test.cpp.*иразработанными с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, позволяющая создавать собственные исключения.

## Описание структур данных

*Класс TNewStack наследуется от класса TStack:*

Конструкторы:

TNewStack(int \_leng, T\* \_elem);

TNewStack(TNewStack<T>& newst);

Методы:

int GetFreeCell(); - возвращает количество свободных ячеек в стеке

T Get(); - возвращает элемент из стека (с удалением)

int GetLeng(); - возвращает длину стека

int GetTop(); - возвращает позицию вершины стека

void Push(T \_elem); - положить элемент в стек

void SetMas(int \_leng, T\* \_elem); - преобразовать массив в стек

void PrintNewStack(); - вывод стека на экран

*Класс TMStack наследуется от класса TNewStack:*

Поля:

int leng; - длина мультистека

T\* elem; - элементы мультистека (массив)

int countSt; - количество стеков в мультистеке

TNewStack<T>\*\* stackMas; - массив указателей на начало каждого стека в мультистеке

Защищенные методы:

int CountFree(); - количество свободных элементов в мультистеке

void Repack(int \_st); - перепаковка стека с увеличением свободной памяти в выбранном стеке

Конструкторы:

TMStack(int \_countSt = 1, int \_leng = 10);

TMStack(TMStack &mst);

Методы:

int GetLeng(); - возвращает длину мультистека

T Get(int \_st); - возвращает элемент из выбранного стека

void Set(int \_st, T \_elem); - положить в выбранный стек введенный элемент

bool IsFull(int \_st); - проверка на полноту выбранного стека

bool IsEmpty(int \_st) – проверить на пустоту выбранного стек

void PrintMStack(); - вывод мультистека на экран

## Описание алгоритмов

**Перепаковка мультистека.**

Необходимость в перепаковке возникает, когда нужно положить новый элемент в стек, который и без того уже полон, в то время как в остальных есть хотя бы одна свободная ячейка.

Опишем алгоритм перепаковки мультистека относительно k-го стека:

* + - 1. Количество свободных ячеек, которые можно добавить в каждый стек add\_ev находим путем деления количества свободных ячеек во всем мультистеке free на количество стеков n:

|  |  |
| --- | --- |
|  | (1) |

* + - 1. Увеличиваем старые размеры size[i] стеков на add\_ev, получаем новые размеры стеков new\_size[i]:

|  |  |
| --- | --- |
|  | (2) |

* + - 1. Если количество свободных ячеек во всем мультистеке free не кратно количеству стеков n, то оставшиеся свободные ячейки добавляем в k-й стек:

|  |  |
| --- | --- |
|  | (3) |

* + - 1. Определяем новый индекс начала каждого стека new\_ind[i], с учетом их размера new\_size[i]:

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |

* + - 1. Перемещение элементов стеков на новые позиции:
  1. Если индекс нового начала i-го стека new\_ind[i] не больше, чем индекс старого начала i-го стека ind[i], то копируем элементы по порядку, в котором они хранятся в старом стеке.
  2. Иначе идем по новым позициям стеков до тех пор, пока не выполняется 5.1. Затем копируем элементы, в котором они хранятся в старом стеке, но в обратном порядке. Иначе произойдет затирание памяти.

# Заключение

В этой лабораторной работе я смогла справиться со всеми поставленными задачами. Разобравшись в том, как функционирует мультистек на массиве, я реализовала класс TMStack. Для этого была реализована библиотека MultiStackLib, в которой полностью описан данный шаблонный класс. Также там описан и реализован и вспомогательный шаблонный класс TNewStack. Написанный мной набор автоматических тестов, проверяет работоспособность всех реализованных методов и делает это весьма успешно. Пример использования данных методов класса очередь для пользователя полностью функционирует и готов к использованию.

1. **Литература**
2. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [http://www.itmm.unn.ru/files/2018/11/Primer-1.5.-Struktury-hraneniya-neskolkih-stekov-v-obshhej-pamyati.pdf], 2015.
3. Википедия: свободная электронная энциклопедия: на русском языке: <https://ru.wikipedia.org/wiki/Стек>
4. Университет ИТМО – Викиконспекты

http://neerc.ifmo.ru/wiki/index.php?title=Стек