

**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
«Реализация мультитека»**

Выполнил:
студент группы 361706-1
Резанцев Сергей Алексеевич
_____ Подпись

Научный руководитель:
ассистент каф. МОСТ ИИТММ
_____ Лебедев И. Г.

Нижний Новгород
2018 г.

Содержание	
Введение	3
Постановка задачи	4
Руководство пользователя	5
Руководство программиста	6
Описание структуры программы	6
Описание структур данных	6
Описание алгоритмов	8
Заключение	9
Литература	10

1. Введение

Целью данной лабораторной работы является реализовать мультистек , и для создания этой программы понадобится создать классы с шаблонами и различными функциями.

Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (last in — first out, «последним пришёл — первым вышел»).

Мультистек - абстрактный тип данных, содержащий несколько списков элементов.

Область применения:

Программный вид стека используется для обхода структур данных, например, дерево или граф. При использовании рекурсивных функций также будет применяться стек, но аппаратный его вид. Кроме этих назначений, стек используется для организации стековой машины, реализующей вычисления в обратной инверсной записи.

Для отслеживания точек возврата из подпрограмм используется стек вызовов.

Арифметические сопроцессоры, программируемые микрокалькуляторы и язык Forth используют стековую модель вычислений.

Идея стека используется в стековой машине среди стековых языков программирования.

Применение стека упрощает и ускоряет работу программы, так как идет обращение к нескольким данным по одному адресу.

2. Постановка задачи

Реализовать класс `TMStack` и `TNewStack` и `TStack` для работы со стеками. Поля классов должны быть закрыты.

В классах обязательно должны присутствовать методы:

- 1) добавление элемента
- 2) удаление элемента
- 3) проверки на полноту и пустоту

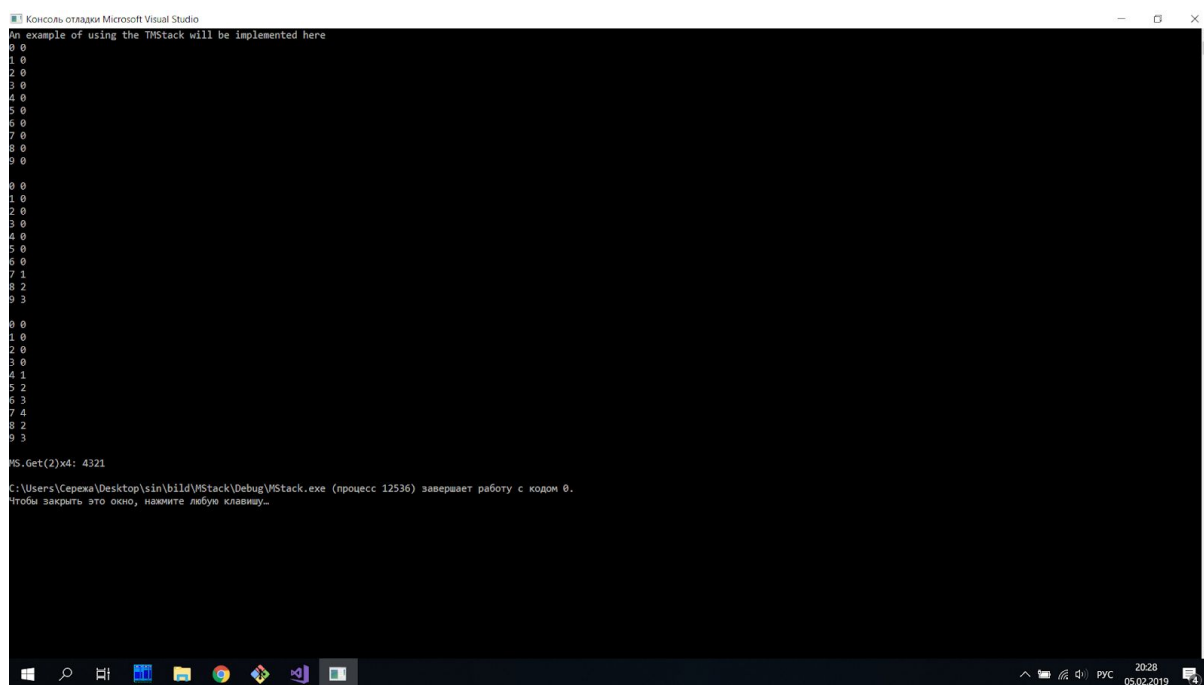
Должны быть реализованы конструкторы: копирования и инициализатор.

Предоставить пример использования и обеспечить работоспособность тестов, покрывающих все методы классов `TMStack` и `TNewStack`.

3. Руководство пользователя

Чтобы начать работу с программой запустите приложение MStack.

На экране появится следующее:



```
Консоль отладки Microsoft Visual Studio
An example of using the TMStack will be implemented here
0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 1
8 2
9 3
0 0
1 0
2 0
3 0
4 1
5 2
6 3
7 4
8 2
9 3
MS.Get(2)x4: 4321
C:\Users\Сергей\Desktop\sin\build\MStack\Debug\MStack.exe (процесс 12536) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
```

Результат программы, выведенный на консоль.

Затем программа завершится.

4. Руководство программиста

4.1. Описание структуры программы

Для реализации алгоритмов будут использованы классы TStack и TMSStack и TNewStack .

Лабораторная работа состоит из следующих модулей:

StackLib

Библиотека, содержащая заголовочный файл TStack.h, в котором содержится класс TStack и реализация его методов, и файл TStack.cpp

TMSStackLib

Библиотека, содержащая заголовочный файл TMSStack.h, в котором содержится класс TMSStack , и реализация его методов, и файл TMSStack .cpp

И содержит заголовочный файл TNewStack.h, в котором содержится класс TNewStack - наследник TStack , и реализация его методов, и файл TNewStack.cpp

TMSStack

Пример использования программы.

test

В файле test_mstack.cpp прописаны тесты, покрывающие каждый метод класса TMSStack и TNewStack .

4.2. Описание структур данных

В программе описаны классы

Шаблонный класс TMSStack:

В нем 4 поля:

int size; - размер мультистека

T* mas; - указатель на область памяти

int n; - кол-во стеков в мультистеке

TNewStack<T>** ns; - массив указателей на начало каждого из стеков

И реализованы следующие функции:

TMStack(int _size, int _n); конструктор по умолчанию

TMStack(TMStack<T> &MS); конструктор копирования

void Put(int _n, T A); положить эл-т в n-ый стек

T Get(int _n); взять эл-т из n-ого стека

bool IsFull(int _n); проверка на полноту

bool IsEmpty(int _n); проверка на пустоту

void Print() вывод мультистека

TNewStack, наследник класса TStack:

В нем реализованы следующие функции:

TNewStack(int _size = 0, T* mas = 0) конструктор по умолчанию

TNewStack(TNewStack <T> &A) : TStack(A) конструктор копирования

int CountFree() кол-во свободных ячеек в стеке

int GetSize() получить размер стека

int GetTop() получить позицию вершины стека

void SetMas(int _size, T* mas) преобразование массива в стек

4.3. Описание алгоритмов

В данном разделе не будут рассматриваться тривиальные методы.

Перепакровка мультистека:

Потребность в перепакровке мультистека возникает, когда нужно добавить элемент в n -й стек, а он уже переполнен, в то время как в других стеках еще осталась хотя бы одна свободные ячейка.

Сперва находим количество свободных ячеек, которые можно добавить в каждый стек. Оно находится путем деления количества свободных ячеек во всем мультистеке на количество стеков. Затем увеличиваем размер каждого стека на это число(если количество свободных ячеек не кратно кол-ву стеков, тогда остаток добавляется в n -ый стек). Переопределяем индексы начала новых стеков и перемещаем элементы на новые позиции в обратном порядке(иначе происходит затирание памяти).

5. Заключение

В данном курсовом проекте при разработке программы был реализован и рассмотрен мультитекстек с помощью класса стека. Классы, наследование оказались очень полезными и удобными и упростили работу с программой.

Предоставлено описание примера работы со списком на массивах в разделе «Руководство пользователя».

Также разработаны и доведены до успешного выполнения тесты, проверяющие корректность методов класса TNewStack и TMSStack.

6. Литература

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Т. А. Павловская С/С++ Программирование на языке высокого уровня. - СПб.:Питер, 2011. - 461 с.
- 3.Крапенко С. Н. и др. Методы объектно-ориентированного программирования. <http://e-learning.unn.ru/course/view.php?id=251>.
- 4.Страуструп. Б. Курс «Язык программирование С++ для профессионалов» <http://www.intuit.ru/studies/courses/98/98/info>
- 5.Гергель В.П. Методические материалы по курсу “Методы программирования 2”: [<http://www.itmm.unn.ru/files/2018/10/Primer-1.1.-Struktury-hraneniya-mnozhestva.pdf>], 2015.