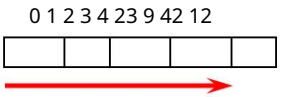
Выпадающий стек Шаблонный контейнер

Этот проект включает в себя реализацию простой полудинамической структуры данных: стек выпадающего списка на основе массива.

Выпадающий стек — это просто вариант стека, который модифицирует операцию push() таким образом, что нижний элемент в стеке выпадает (теряется), если стек заполнен. Приложения выпадающих стеков включают списки истории и списки отмены в приложениях. Пользователи могут захотеть хранить всевозможные вещи в виде задач в выпадающем стеке. Для этого нам потребуется, чтобы реализация использовала шаблоны C++.

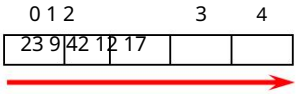
Как и следовало ожидать, выпадающий стек часто реализуется как , чтобы была фиксированная максимальная емкость . Это будет иметь место в этом назначении, хотя емкость будет указана как параметр конструктора. Базовая физическая структура может быть либо динамически размещаемым массивом, либо связанным списком узлов. В этом задании вы должны использовать массив, позволяя вершине «плавать» при добавлении и удалении элементов. Никогда не будете сдвигать содержимое массива для выполнения Push() или Pop().

Выпадающий стек может демонстрировать явление, похожее на «расползание очереди», когда он удерживается в массиве. Предположим, у нас есть выпадающий стек емкостью пять элементов, и мы поместили в стек четыре элемента:

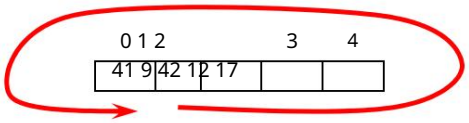


Здесь Top будет равен 4 (индекс ячейки, которая будет использоваться при следующей операции push), а нижний элемент будет иметь индекс 0.

Теперь предположим, что мы архивируем еще один, заполняя стек:



А теперь предположим, что мы толкаем другой. Куда это идет? В выпадающем стеке на основе массива новый элемент будет помещаться поверх нижнего элемента, используя массив по кругу:



Здесь новый элемент был 41, и теперь верхний элемент равен 1, а нижний элемент имеет индекс 1.

Объявления шаблонов

Это задание будет оцениваться тестовой программой. Чтобы приспособиться к этому, мы должны довольно строго ограничить вашу реализацию. Короче говоря, ваши шаблоны должны точно соответствовать декларациям, приведенным далее в этой спецификации. Детали реализации зависят от вас, но если вы измените или опустите члены данных или функции, ваш код почти наверняка не скомпилируется с помощью тестовой системы

Вы также должны поместить всю реализацию шаблона в отдельный файл, используя имя, указанное в заголовочном файле шаблона, который следует ниже.

Обратите внимание, что шаблон DropOutStackT объявит класс друзей, как показано ниже:

```
#ifndef DROPOUTSTACKT_H #define
DROPOUTSTACKT_H #include
<iostream> #include <iomanip>
#include <new> using namespace std;



шаблон <имятипа T> класс DropOutStackT {

друг класса Жавер;

частный:
    T *Стек;                        // указатель на массив стека //
    беззнаковый int Cap;            емкость (размерность) массива стека // индекс для
    беззнаковое целое Верх;         следующего нажатия// количество сохраненных элементов
    беззнаковый целочисленный размер;

публичный:
    DropOutStackT (беззнаковая целая емкость = 0);
    DropOutStackT(const DropOutStackT<T>& Source);
    DropOutStackT<T>& operator=(const DropOutStackT<T>& RHS);

    bool Push(const T&Elem); bool Pop(T&Elem);      // вставляем элемент сверху //
                                                    удаляем самый верхний элемент и возвращаем //
    T* const Peek() const; Пусто                    предоставляем доступ к самому верхнему элементу //
    Очистить();                                     сбрасываем стек в пустое состояние

    логическое значение пусто ()                    // проверка на пустой стек // проверка
    константа; //bool isFull()const;                на «полный» стек // отчето емкости
    беззнаковая целая емкость () константа;         стека

    void Display(ostream& Out) const; // отображаем содержимое стека

    ~DropOutStackT();                               // уничтожить стек
};

// код реализации шаблона импорта: #include
"DropOutStackTImplementation.cpp"

#endif
```

НЕ реализуйте метод isFull*()

Требования к дизайну и реализации

В дополнение к тем, которые указаны на странице «Стиль кодирования» на веб-сайте курса, существуют некоторые явные требования:

- Вы должны реализовать шаблон C++ (очевидно), соответствующий данному интерфейсу.
- Код из C++ STL или любой другой неутвержденной библиотеки НЕ может использоваться в вашей реализации. Нарушение этого ограничения приведет к нулевой оценке!
- Вы должны правильно справляться с проблемами копирования. Мы обязательно проверим это. Сбои в реализации почти наверняка приведут к сбоям программы и нулевым оценкам.
  - Вы можете предположить, что любой тип данных, хранящийся в вашем шаблоне, будет корректно обрабатывать собственные проблемы с копированием (как и должно быть).
  - Вы можете предположить, что любой тип данных, хранящийся в вашем шаблоне, будет предоставлять оператор <<() и оператор ==().
- Вы должны правильно выделять и освобождать память по мере необходимости.
- Вы должны предоставить клиенту обратную связь в случае сбоя операции. Данные прототипы функций-членов указывают, где это необходимо. Ни при каких обстоятельствах никакая из функций шаблона, кроме Display(), не должна записывать вывод.

Функция Display() должна записывать содержимое выпадающего стека в следующем формате:

```
Вместимость: 40:
    453: 35

    2:25 1:15
```

Сообщается емкость стека, за которой следуют сохраненные элементы, перечисленные сверху вниз, с меткой, указывающей индекс массива, в котором хранится каждый элемент. Если стек пуст, напишите:

```
Вместимость: 4
    Стек пуст
```

Проблема: когда выполняется операция Pop(), индекс Top должен быть перемещен назад к предыдущей ячейке. Если Top равен нулю, он должен быть сброшен до самого большого допустимого индекса. Очевидно, что это можно сделать, рассматривая эту ситуацию как частный случай и используя оператор выбора. Задача заключается в следующем: реализовать Pop() так, чтобы корректировка Top не требовала какой-либо специальной логики. Подсказка: какова аддитивное значение, обратное 1, если выполняется циклическая арифметика над целыми числами {0, 1, 2, ... , n-1}? Эту задачу НЕЛЬЗЯ обсуждать на веб-форумах CS.

Процедура тестирования DropOutStack

Ниже приводится схема рекомендуемой процедуры тестирования кода шаблона. По сути, это стратегия тестирования, которая будет использоваться тестовой программой Curator. Выдолжнысоответствующим образом разработать собственную стратегию тестирования. Учащиеся, желающие составить код тестового комплекта и поделиться им с другими учащимися курса, могут сделать это, отправив код тестового комплекта своему инструктору для предварительного одобрения перед публикацией на форуме веб-класса CS.

I. Тестирование логики глубокого копирования

- создать стек и поместить в него некоторые элементы•
передать его функции по значению • проверить содержимое
локального стека на соответствие точной копии• проверить содержимое
исходного стека на соответствие точной копии Примечание: если здесь есть
КАКИЕ-ЛИБО несоответствия, тест считается прервано.

II. Протестируйте push/pop/peek/clear

- создайте стек и поместите в него несколько элементов *вытащите и проверьте несколько элементов сверху вниз •
заполните стек, затем нажмите еще один, чтобы он обернул *проверьте верхний элемент *вытолкните и проверить несколько элементов •
вытолкнуть его, пока он не станет пустым, и проверить последний элемент • очистить его • вставить элементы, пока он снова не завернется
- вытолкнуть, пока он не развернется• проверить содержимое

III. Протестируйте

большой стек • создайте
большой стек • вставляйте элементы, пока он не
заполнится• проверьте некоторые элементы

- щелкайте, пока не завершится•
проверьте некоторые элементы•
очистите его

Оценка:

Вы должны задокументировать свою реализацию в соответствии со страницей стилей кода на веб-сайте курса. Возможно, ваша программа будет оцениваться на предмет документации и дизайна, а также на правильность результатов. Если это так, ваша заявка, набравшая наивысший балл, может быть оценена одним из ассистентов, который оценит вычет (в идеале нулевой) по сравнению с вашей оценкой от куратора.

Обратите внимание: если вы сделаете две или более работ с одинаковым количеством баллов, будет оценена самая ранняя из них. Из этой политики не будет абсолютно никаких исключений!

Мораль: кодируйте и документируйте, чтобы соответствовать требованиям с самого начала, вместо того, чтобы планировать переделывать документацию в готовую программу.

Обратите внимание, что оценка вашего проекта может существенно зависеть от качества вашего кода и документации.

Что сдавать и как: отправьте

файл C++ DropOutStackTImplementation.cpp, содержащий реализацию вашего шаблона, в кураторскую систему. Отправьте только файл реализации шаблона. Больше ничего не отправляйте. Инструкции по подаче Куратору приведены в Руководстве для студентов на сайте Куратора: http://www.cs.vt.edu/curator/. Обязательно внимательно следуйте этим инструкциям. URL-адрес подачи также размещен на веб-сайте куратора. Вам будет разрешено представить свое решение до шести раз.

Приносить присягу:

Каждая заявка на участие в программе должна соответствовать требованиям Кодекса чести для этого курса. В частности, вы должны включить следующее заявление о залоге в комментарий заголовка файла:

> В мою честь:
>
> - Я не обсуждал код языка C++ в своей программе ни с кем, кроме моего преподавателя или ассистентов преподавателя, назначенных на этот курс.
>
> - Я не использовал код языка C++, полученный от другого учащегося или любого другого неавторизованного источника, модифицированный или немодифицированный.
>
> - Если какой-либо код языка C++ или документация, используемые в моей программе был получен из другого источника, такого как учебник или заметки о курсе, это было четко отмечено с соответствующей ссылкой в комментариях к моей программе.
>
> - Я не разработал эту программу таким образом, чтобы победить или помешать нормальной работе Системы Куратора.
>
> - Я не оказывал и не принимал несанкционированную помощь в завершение этого задания.
>
> // // // // // <Имя учащегося>/<ИД учащегося>

Неспособность включить это обещание в заявку является нарушением Кодекса чести VT.