***Шаблонный класс стек.***

1. Постановка задачи.

Реализация шаблонного класса, реализующего стек и поддерживающего операции помещения объекта в стек, извлечения объекта из стека и получение размерности стека.

В случае попытки вызова операции извлечения объекта из стека при условии, что стек пуст, должно генерироваться исключение класса EStackEmpty (наследник класса EStackException). Данный класс должен содержать публичный метод char\* what(), возвращающий диагностическое сообщение.

1. Предлагаемое решение.

Стек описывается как шаблонный класс на основе односвязного списка. Узел данного списка описывается структурой, состоящей из значения и указателя на предыдущее звено стека. Структуру определяем в приватном доступе вместе с указателем на последний элемент стека и целочисленной переменной, позже используемой для получения размера стека.

Реализуем метод Push, кладущий элемент в стек. Данный метод с типом void, то есть не возвращающий ничего после исполнения; на вход принимает значение по ссылке. Сперва создается новое звено списка, затем проверяется создался ли объект. Если после создания указатель на звено остается пустым, то выбрасываться исключение. Далее новое звено делается последним, и увеличивается размер стека на единицу.

Следом реализуем метод Pop, позволяющий извлекать последний объект из стека. Возвращает константное значение, для невозможности это значение, зачем-то, изменить. Сперва проверяется пустота стека, т.е. не указывает ли последний элемент стека на ничто, если это так, то выбрасывается исключение. Далее последнее значение сохраняется в константную переменную, а конец стека переносится на предпоследний элемент, счётчик элементов в стеке уменьшается. Наконец, возвращается сохранённое значение.

Реализуем функцию Clear, очищающую стек. Суть заключается в том, что функция проходит по всем элементам стека, каждый из них помещается во временную переменную, которая после сдвига последнего элемента на предыдущий, удаляется.

1. Коды программы.

К данной реализации относятся файлы:

EStackEmpty.h

EStackException.h

Stack.h

1. Инструкция пользователя.

Так как стек шаблонный, у пользователя есть выбор значения элементов стека. Предположим, что пользователь выбрали простое - целочисленное значение.

Объявление стека будет выглядеть так:

Stack<int> Stack;

Далее, пользователь захочет стек чем-нибудь заполнить:

Stack.Push(2);

Stack.Push(3);

Stack.Push(4);

Stack.Push(5);

Пользователь может убедиться, что ему это удалось, проверив размер стека:

std::cout << Stack.Size() << std::endl;

Также, он может извлечь последний элемент стека:

int lastElement = Stack.Pop();

std::cout << lastElement << std::endl;

1. Тестирование.

Протестировать данную программу можно с помощью попытки извлечения значения с ещё не заполненного стека:

Объявим стек и вызовем метод Pop:

Stack<int> Stack;

Stack.Pop();

Программа выбросит исключение о пустом стеке, что и предполагалось при формулировке задания.

***Класс PersonKeeper.***

1. Постановка задачи.
2. Предлагаемое решение.
3. Коды программы.
4. Инструкция пользователя.
5. Тестирование.