

Задание №5.

НИУ ВШЭ, Департамент программной инженерии,
курс «Построение и анализ алгоритмов»

Дворянский Л.В., 2015/16 г.

Темы задания: шаблоны, пара (pair), стек (stack), очередь (queue)

Необходимо:

- реализовать шаблон NiceStack<class T>;
- понять устройство очереди из 2х стеков
- реализовать шаблон NiceQueue<class T>;

Описание:

Обычно, для поиска минимального элемента в стеке, требуется пройти все значения стека (извлечь все элементы), что требует $O(N)$ времени, где N число элементов в стеке. Однако в ситуации, когда стек используется в высокопроизводительной системе, требуется узнавать минимальный элемент мгновенно.

Шаблон NiceStack<class T> это стек, который позволяет за время $O(1)$ узнать значение минимального элемента в стеке. Для этого каждый элемент стека, кроме значения, содержит еще и минимум среди всех элементов стека глубже его. Тогда вершина стека хранит минимум среди всех элементов стека. Конечно, при этом стек занимает в 2 раза больше памяти, т.к. каждый элемент хранит свое значение и минимум элементов ниже его в стеке. Но такова плата за скорость проверки минимума стека.

Рассмотрим пример заполнения такого стека элементами:

7 15 13 5 6 5 3 2 15

push 7	push 15	push 13	push 5	push 6	push 5	push 3	push 2	push 15
<7,7>	<15,7>	<13,7>	<5,5>	<6,5>	<5,5>	<3,3>	<2,2>	<15,2>
	<7,7>	<15,7>	<13,7>	<5,5>	<6,5>	<5,5>	<3,3>	<2,2>
		<13,7>	<13,7>	<13,7>	<13,7>	<5,5>	<3,3>	<3,3>
		<15,7>	<15,7>	<15,7>	<15,7>	<5,5>	<5,5>	<5,5>
		<7,7>	<7,7>	<7,7>	<7,7>	<6,5>	<6,5>	<6,5>
			<5,5>	<5,5>	<5,5>	<5,5>	<5,5>	<5,5>
			<13,7>	<13,7>	<13,7>	<13,7>	<13,7>	<13,7>
			<15,7>	<15,7>	<15,7>	<15,7>	<15,7>	<15,7>
			<7,7>	<7,7>	<7,7>	<7,7>	<7,7>	<7,7>

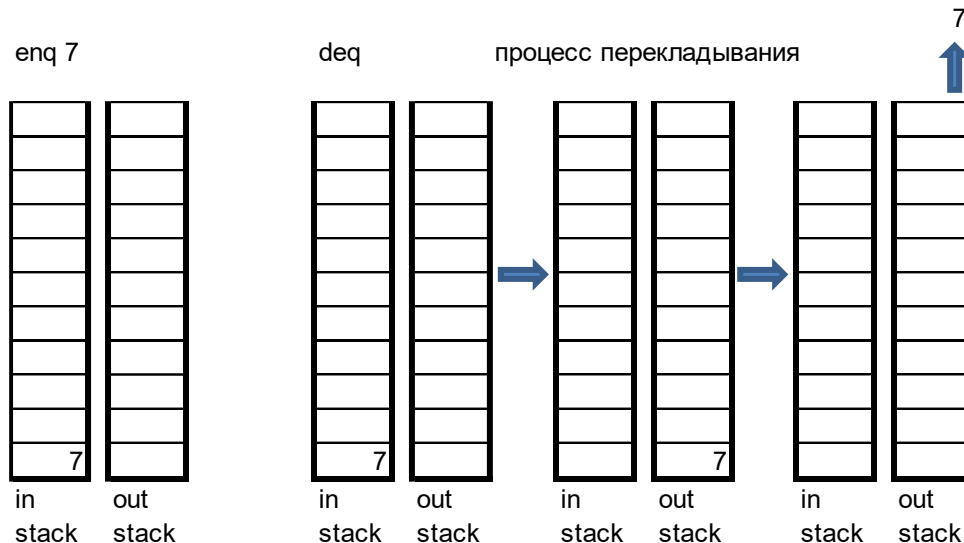
Заголовок шаблона `NiceStack<class T>` содержится в файле `NiceStack.h`. Этот заголовок менять нельзя! Методы шаблона `NiceStack` необходимо реализовать в файле `NiceStack.hpp`.

Очередь это такая структура данных, которая извлекает элементы в том же порядке, в котором они были размещены в очереди. Т.е. самый старый элемент извлекается первым. Чтобы положить значение в очередь используется операция **enqueue** (сокращенно `enq`). Чтобы извлечь – **dequeue** (сокращенно `deq`).

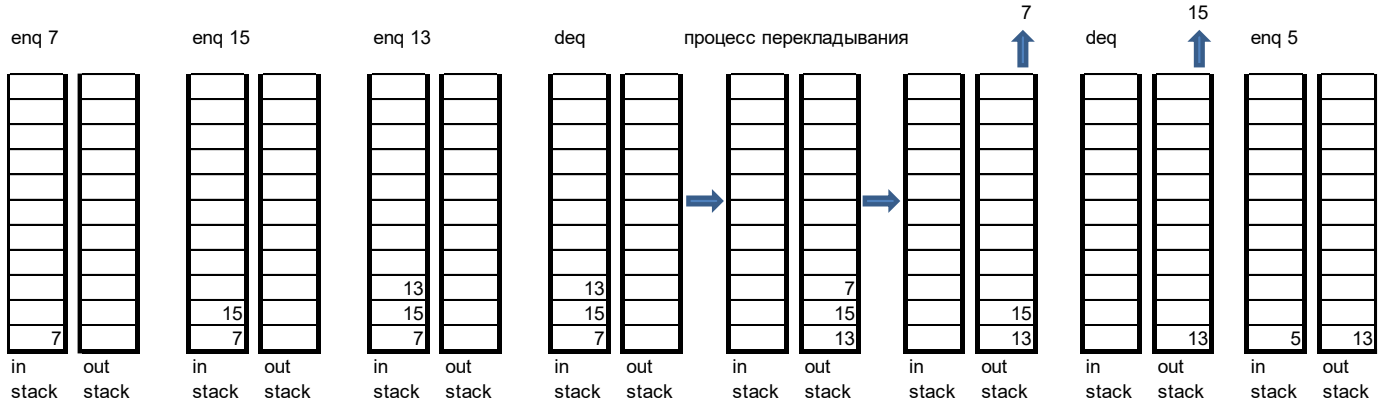
Необычным применением стеков является реализации очереди. При этом очередь состоит из входного и выходного стека. Новые элементы кладутся во входной стек. Элементы очереди извлекаются из выходного стека. Если при извлечении (**deq**) элемента из очереди, выходной стек пуст, а входной стек содержит элементы, то, сначала, элементы перекадываются из входного стека в выходной, а затем извлекается элемент из выходного стека. Если оба стека пусты, то при вызове (**deq**) возникает ошибка.

Примеры работы очереди из 2х стеков:

1) Положить 7 и извлечь: `enq 7, deq`



2) enq 7, enq 15, enq 13, deq, deq, enq 5



Если такую очередь реализовать не из простых стеков, а из 2-х NiceStack, то мы получим NiceQueue у которой можно быстро (за $O(1)$) узнать минимальный элемент. (Подумайте как).

Заголовок шаблона NiceQueue<class T> содержится в файле NiceQueue.h. Этот заголовок менять нельзя! Методы NiceQueue необходимо реализовать в файле NiceQueue.hpp.