Лекция 1.4. Стеки

№ слайда	Текст
1	Здравствуйте. На этой лекции мы познакомимся с таким простым, но полезным типом данных, как стек, и рассмотрим разные способы реализации этого типа данных.
2	Что вообще такое тип данных? Это понятие включает в себя две компоненты: - множество значений, которые могут принимать объекты данного типа; - и множество операций, применимых к этим объектам. Любой язык программирования включает набор встроенных типов данных с соответствующими операциями (целые и вещественные числа, символы, строки, логические значения и т.п.). Например, для числовых типов определены операции арифметики и сравнения. Современные языки позволяют программисту конструировать сложные типы данных, адекватные решаемой задаче. Одни и те же пользовательские типы данных часто можно реализовать на основе различных структур данных и алгоритмов, и выбор наиболее эффективной реализации зависит от требований конкретной задачи. Рассмотрим и сравним различные реализации несложного типа данных – стека.
3	Стек – это совокупность однотипных элементов, для которой определены две основные операции: - добавление элемента в стек; - извлечение из стека элемента, который был добавлен последним. Тип элементов может быть различным в зависимости от конкретной задачи: целые или вещественные числа, строки, указатели, записи и т.п. Размер стека (количество элементов) теоретически неограничен, хотя возможны ограничения конкретной реализации.
4	Пусть элементы стека являются значениями некоторого типа данных Element. Например, тип Element определен как тип целых чисел int. Пусть также максимальный размер стека задается константой N. Тогда стек можно определить, как структуру, состоящую из массива элементов, а также числа top – текущего количества элементов в стеке. Начальная часть массива с индексами от 0 до top-1 – это часть, занятая элементами, помещенными в стек. Оставшаяся часть массива свободна, не содержит данных.

№ слайда	Текст
5	На следующем слайде приведены фрагменты программного кода для основных операций со стеком: - создание пустого стека целых чисел (при этом top = 0); - занесение в стек двух чисел с увеличением счетчика top; - извлечение из стека того числа, которое было добавлено последним; - и проверка, что стек не пуст, то есть top > 0.
6	Как альтернативу, рассмотрим возможность реализации стека как списка. В данном случае вполне удобен простейший вариант линейного списка. Определение стека по сути сводится к определению элемента стека, который состоит из указателя на следующий (добавленный раньше) элемент и собственно значения данного элемента. Создание стека – это просто описание указателя на голову списка, первоначально пустого. Непустой стек представляет собой список, в голове которого последний добавленный элемент.
7	Как выполняются операции с таким стеком? Занесение каждого числа в стек требует выполнения нескольких действий: создания нового элемента, присваивание ему значения, а затем включение этого элемента в список в качестве головного элемента, как было рассмотрено ранее. Эти действия повторяются для каждого добавляемого элемента. Извлечение числа из стека – это просто удаление прочитанного элемента из головы списка. Наконец, проверка на непустоту выполняется как проверка ненулевого значения указателя на голову стека.
8	Сравним две рассмотренные реализации стека с точки зрения их эффективности. Обе реализации позволяют очень быстро выполнить основные операции. Преимущества массива перед списком: - меньше расход памяти; - операции выполняются проще и быстрее (но надо проверять на переполнение массива!). Преимущество списка перед массивом обычное для списка: - нет ограничения по памяти (кроме полного объема памяти программы, который может достигать гигабайтов).

№ слайда	Текст
9	Для чего нужны стеки? Надо признать, что в условиях реальных задач довольно редко исходные данные задачи требуют представления в виде стека. Чаще стеки используются как инструмент при решении задач в некоторых алгоритмах. Надо прежде всего понимать, что использование в программировании рекурсивных функций всегда опирается на использование стека. Просто этот стек ведется системой, а не описывается в явном виде. В стек помещаются адрес возврата, значения параметров функции и описанных в ней локальных переменных, Любую рекурсивную функцию можно переписать без использования рекурсии. Такое преобразование, как правило, требует явного использования стека. При этом исходный текст функции обычно становится более длинным и менее понятным, но работает нерекурсивная функция несколько быстрее. Многим знакома задача-головоломка «Ханойские башни», придуманная еще в XIX веке французским математиком Эдуардом Люка. Речь там идет о перекладывании колец с одного столба на другой по определенным правилам. Поскольку снять со столба всегда можно только верхнее, положенное последним кольцо, эти столбы можно описать как стеки.
10	Итак, что мы узнали о типах данных вообще, о стеках и очередях в частности? Что тип данных определяется множеством возможных значений объектов этого типа и набором операций над ними. Что один и тот же тип может быть реализован на основе разных структур данных и алгоритмов, эти реализации могут различаться по эффективности. Что такой тип данных, как стек, может быть эффективно реализован на основе массивов или списков, причем оба варианта имеют свои преимущества и недостатки. Что выбор реализации определяется потребностями конкретной задачи. На этом лекция закончена. Благодарю за внимание.