



Программирование на Python

Презентация занятия

Алгоритмы и структуры данных. Часть 2.

11 занятие



инжинириум[®]

МГТУ им. Н.Э. Баумана

2019

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ. ОРГАНИЗАЦИОННАЯ ИНФОРМАЦИЯ

- Тема занятия
- Цели и задачи занятия
- Результаты занятия
- Материалы для преподавателя
- Материалы для ученика
- Тайминг проведения занятия

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

- Стэк
- Очередь
- Знакомство с графами и деревьями

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

- Реализация стэка
- Особенности работы со стэком
- Реализация очереди
- Базовые методы очередей



ВВЕДЕНИЕ. ОРГАНИЗАЦИОННАЯ ИНФОРМАЦИЯ

Тема: Алгоритмы и структуры данных. Часть 2.

Цели и задачи:

- Рассказать о новых структурах данных.
- Пояснить особенности работу со стэком.
- Определить различия в области применимости стэка и очереди.
- Рассказать о графах.
- Описать терминологию в области теории графов.
- Рассказать про деревья.

По результатам занятия слушатель будет знать:

- Какие существуют дополнительные структуры данных.
- Какие приложения существуют у графов.



Тема: Алгоритмы и структуры данных. Часть 2.

По результатам занятия слушатель будет уметь:

- Реализовывать простейший стэк и очередь.
- Понимать отличие между графом и деревом.
- Оптимизировать решения при помощи алгоритмов стэка.

Тайминг занятия

Таб.1

№	Этапы	время	Сумма
1	Структуры данных 2.0	20 мин.	20
2	Пишем первый стэк	25 мин.	25 мин.
3	<i>Перерыв</i>	5 мин.	5 мин.
4	Графы	40 мин.	40 мин.



Тема: Алгоритмы и структуры данных. Часть 2.

1. Какие еще есть структуры данных?

1.1 Стэк

Стэк- структура данных, представляющая собой список элементов, организованный по принципу LIFO(last in - frist out, "последний вошел - первый ушел"). Часто принцип работы стэка сравнивают со стопкой фишек в стакане: чтобы взять вторую фишку сверху, нужно снять верхнюю.

1.2 Очередь

Очередь - структура данных, похожая на стэк, только организована по принципу LILLO(last in - last out, "последний вошел - последний ушел"). Сначала из очереди удаляются старые элементы, затем новые. Принцип работы очереди сравнивают с классической очередью в магазине: последний человек в очереди будет обслужен самым последним.



Тема: Алгоритмы и структуры данных. Часть 2.

1. Какие еще есть структуры данных?

1.1 Стэк

Стэк- структура данных, представляющая собой список элементов, организованный по принципу LIFO(last in - frist out, "последний вошел - первый ушел"). Часто принцип работы стэка сравнивают со стопкой фишек в стакане: чтобы взять вторую фишку сверху, нужно снять верхнюю.

1.2 Очередь

Очередь - структура данных, похожая на стэк, только организована по принципу LIFO(last in - last out, "последний вошел - последний ушел"). Сначала из очереди удаляются старые элементы, затем новые. Принцип работы очереди сравнивают с классической очередью в магазине: последний человек в очереди будет обслужен самым последним.



Тема: Алгоритмы и структуры данных. Часть 2.

1.3 Как с ними работать?

Для работы со стэком и очередью есть множество алгоритмов, которые решают разнообразные задачи. Не нужно придумывать новые сложные алгоритмы так как 99.99% задач программиста покрываются уже существующими решениями.



Тема: Алгоритмы и структуры данных. Часть 2.

Стек

Наиболее часто встречающаяся аналогия для объяснения стека — стопка тарелок. Вне зависимости от того, сколько тарелок в стопке, мы всегда можем снять верхнюю. Чистые тарелки точно так же кладутся на верх стопки, и мы всегда будем первой брать ту тарелку, которая была положена последней.

Если мы положим, например, красную тарелку, затем синюю, а затем зеленую, то сначала надо будет снять зеленую, потом синюю, и, наконец, красную. Главное, что надо запомнить — тарелки всегда ставятся и на верх стопки. Когда кто-то берет тарелку, он также снимает ее сверху. Получается, что тарелки разбираются в порядке, обратном тому, в котором ставились.



Тема: Алгоритмы и структуры данных. Часть 2.

Реализация стека в Python

Преподавателю рекомендуется ознакомиться с реализацией стэка на Python и объяснить ключевые моменты.

Пример реализации:

```
stack = []  
def push(value):  
    stack = stack + [value]  
def pop():  
    return stack[len(stack)-1:len(stack)]
```

2.2 Какие алгоритмы нужны для работы со стеком

Поиск элемента в стэке, наибольший / наименьший элемент, подсчет количества элементов в стэке с определенным условием и т.д.

2.3 Как их применять и для чего

Классическая задача - правильная скобочная последовательность.



Тема: Алгоритмы и структуры данных. Часть 2.

Реализация стека в Python

Преподавателю рекомендуется ознакомиться с реализацией стэка на Python и объяснить ключевые моменты.

Пример реализации:

```
stack = []  
def push(value):  
    stack = stack + [value]  
def pop():  
    return stack[len(stack)-1:len(stack)]
```

2.2 Какие алгоритмы нужны для работы со стеком

Поиск элемента в стэке, наибольший / наименьший элемент, подсчет количества элементов в стэке с определенным условием и т.д.

2.3 Как их применять и для чего

Классическая задача - правильная скобочная последовательность.



Тема: Алгоритмы и структуры данных. Часть 2.

3. Графы

Преподавателю рекомендуется рассказать про задачу о Кёнингсбергских мостах.

3.1 Что такое граф?

Граф - абстрактный математический объект, представляющий набор вершин и ребер. Очень часто используется в программировании и понимается как структура данных состоящая из p -узлов (вершин) и q -ребер. Преподавателю необходимо пояснить, что мы понимаем под узлом и ребром графа.

3.2 Как его использовать?

Теория графов находит широкое применение в химии, в экономике и логистике, в схемотехнике и информатике. Везде, где необходимо детальное исследование внутреннего состояния какой-либо структуры - используются графы.



Тема: Алгоритмы и структуры данных. Часть 2.

3.4 Что можно делать с их помощью

Задачи поиска кратчайшего пути, задачи сложного структурного поиска, задачи поиска структуры в наборе данных, оптимизация топологии и т.

д.

3.5 Дерево - это граф, только

Дерево - это связный ациклический граф. Преподавателю необходимо пояснить значения терминов "связность" и "ацикличность".

Связность - наличие путей между любыми парами вершин.

