УДК 004.43/93

АНАЛИЗ ВРЕМЕНИ ИСПОЛНЕНИЯ JAVA COLLECTION FRAMEWORK

Ковальчук Е.Д
Професор, зав.кафедры КС Ситников В.С
Одесский национальный политехнический університет, УКРАИНА

АННОТАЦИЯ. Материал по исследованию структур данных, в котором проводится исследование времени исполнения каждого отельного класса интерфейса за худшее и лучшее время.

Коллекции являются одним из столпов Java Core. Они используются почти в каждом приложении, поэтому необходимо уметь использовать Java Collections Framework эффективно. Под коллекцией в **java** подразумевается структура данных, позволяющая оперировать множеством объектов одного типа. Примерами коллекций могут служить динамические массивы, стек, связной список и т.п. Коллекции содержат только один элемент в каждой ячейке. К этой категории относятся список (**List**), где в определенной последовательности хранится группа элементов, множество (**Set**), в которое можно добавлять только по одному элементу определённого типа, и очередь (**Queue**).

Основу библиотеки составляют открытые интерфейсы, которые можно использовать для создания собственных коллекций. Каждый интерфейс объявляет набор методов, которые пользователь обязан реализовать в своей программе.

Сложность алгоритмов влияет на время его исполнения, а сама сложность еще обуславливается тем, насколько хорошо выполняется та или иная характеристика для любого количества элементов. Имеются средние оценки выполнения [1], рис. 1.

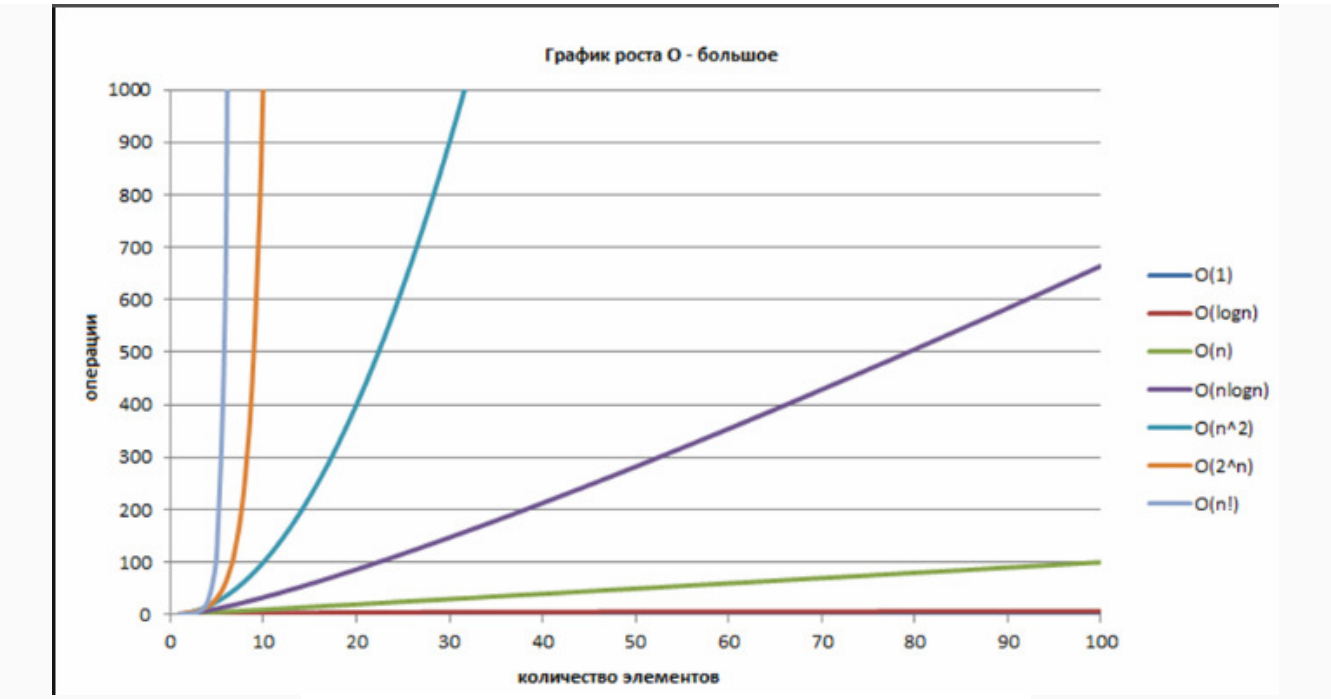

Рисунок 1 – Временная сложность каждого отдельного времени

Имея данные характеристики, можно оценить временную сложность коллекций. Сняв экспериментальным путем временную сложность коллекций, для всех трех интерфейсов и их реализаций одновременно представив их результат в худшем и лучшем времени, в табл. 1, 2.

	Лучшее время			
	Вставка	Удаление	Поиск	Поиск по индексу
Vector	O(n)	O(n)	O(n)	O(1)
ArrayList	O(n)	O(n)	O(n)	O(1)
LinkedList	O(1)	O(1)	O(n)	O(n)
HashTable	O(1)	O(1)	O(1)	н
HashMap	O(1)	O(1)	O(1)	н
LinkedHashMap	O(1)	O(1)	O(1)	н
TreeMap	O(log(n))	O(log(n))	O(log(n))	н
HashSet	O(1)	O(1)	O(1)	н
LinkedHashSet	O(1)	O(1)	O(1)	н
TreeSet	O(log(n))	O(log(n))	O(log(n))	н

Таблица 1 – Временная сложность коллекций при лучшем времени

	Худшее время			
	Вставка	Удаление	Поиск	Поиск по индексу
Vector	O(n)	O(n)	O(n)	O(1)
ArrayList	O(n)	O(n)	O(n)	O(1)
LinkedList	O(1)	O(1)	O(n)	O(n)
HashTable	O(n)	O(n)	O(n)	н
HashMap	O(n)	O(n)	O(n)	н
LinkedHashMap	O(n)	O(n)	O(n)	н
TreeMap	O(log(n))	O(log(n))	O(log(n))	н
HashSet	O(n)	O(n)	O(n)	н
LinkedHashSet	O(n)	O(n)	O(n)	н
TreeSet	O(log(n))	O(log(n))	O(log(n))	н

Таблица 2 – Временная сложность коллекций при худшем времени

Проанализировав лучшее и худшее время можно было определить качественные характеристики всех интерфейсов и классов. В интерфейсе **List** в классе **ArrayList** лучшая временная характеристика у поиска по индексу, так как в любом случае его сложность - O(1). У другого класса **LinkedList** – лучшее и худшее время у вставки и удаления(правда из конца и начала списка, в виду своей структуры), так же сложность константная. У интерфейса **Set** в лучшем случае все характеристики кроме поиска по индексу(он попросту отсутствует) – имеет константное время, но в худшем случае – линейное для тех же характеристик. Идентичную временную сложность имеют все реализации интерфейса **Map**. Отдельно стоит сказать о таких классах как **TreeMap** и **TreeSet** которые в виду своей структуры имеют постоянное время в любом случае – логарифмическое(O(log(n))).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Роберт Лафоре, “Структуры данных и алгоритмы Java”, издательство «Питер», 2013