



Увод

в обектно ориентираното програмиране



Структури от данни Стекове и опашки

Съдържание

- Преговор
- Стекове
- Опашки
- Речници и хеш-таблицы

Структури от данни

Структури от данни

В зависимост от задачата, която трябва да решим с програмиране, се налага да организираме данните, с които работим, по различен начин (например подредба на някакви елементи или връзки между тях.)

Структурите от данни са множество от данни, организирани по определен начин.

Линейни структури от данни

Линейни структури от данни

Има различни структури от данни. Линейните структури от данни са най-често срещаните.

Представяват описание (абстракция) на поредица (списък) от обекти от реалния свят.

Структури от данни в Java

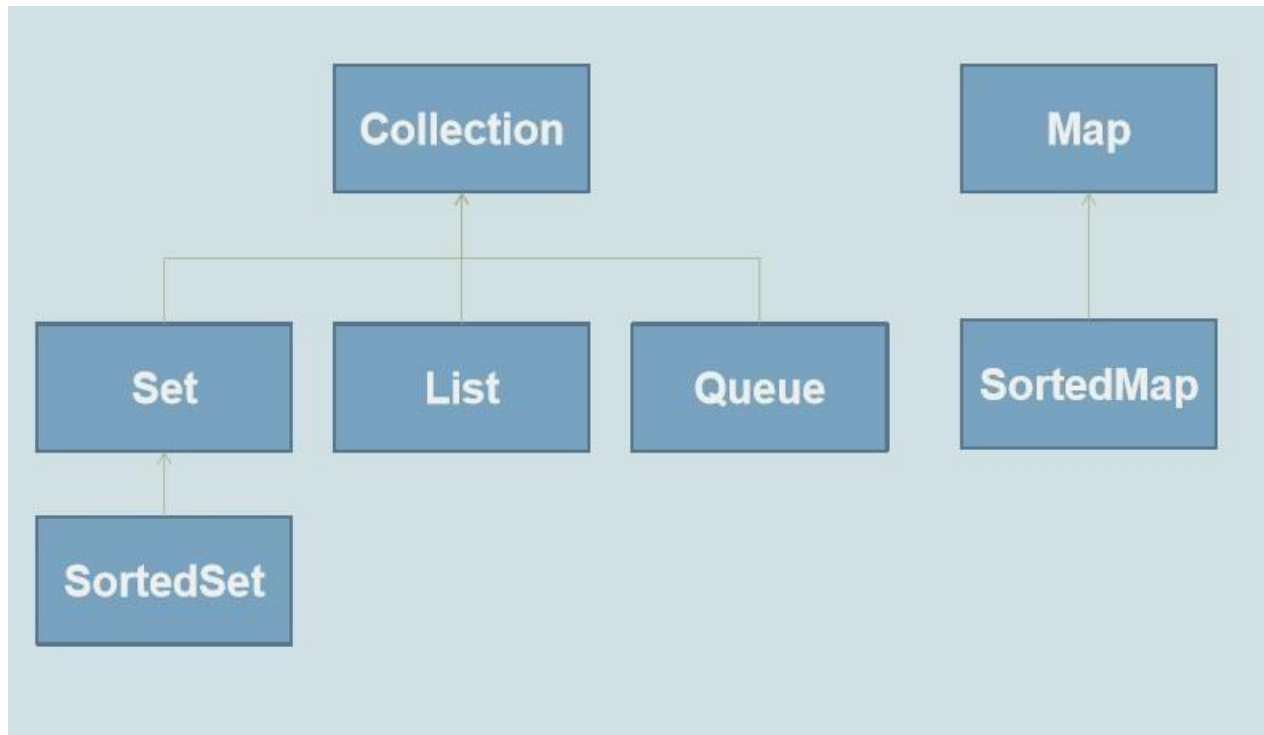
Структури от данни в Java

Колекциите са структурите от данни в стандартната библиотека на Java.

Collections Framework в Java включва:

- интерфейси
- конкретни реализации (класове) на тези интерфейси
- алгоритми

Интерфейси



Стандартни имплементации

	Hash table	Resizable Array	Tree	Linked List	Hash table + linked list
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Queue				LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

Стек (Stack)

Структура от данни, при която можем да добавяме и махаме елементи само в единия край. В Java структурата от данни стек е реализирана в класа `java.util.Stack`.

LIFO (Last In First Out)



Стек (Stack)

push(T) – позволява ни добавянето на нов елемент на върха на стека

pop() – връща ни най-горния елемент като го премахва от стека

peek() – връща най горния елемент без да го премахва

size() – връща броя на елементите в стека

clear() – премахва всички елементи

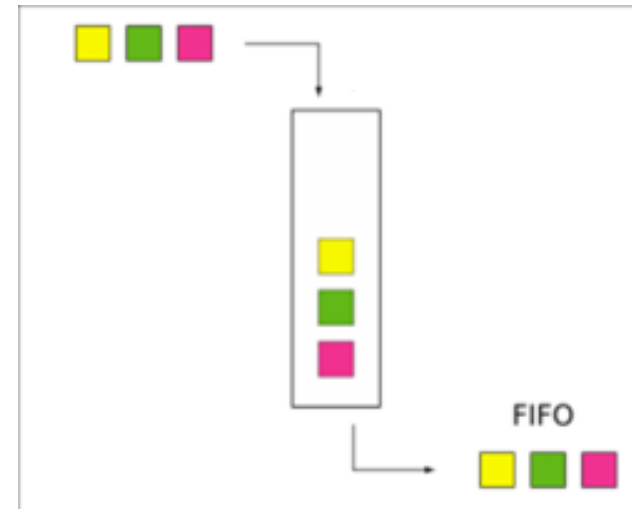
contains(T) – проверява дали елемента се съдържа в стека

toArray() – връща масив, съдържащ елементите от стека

Опашка (Queue)

Структура от данни, в която можем да добавяме елементи в само единия край и да премахваме елементи само в другия край.

FIFO (First In First Out)



Опашка (Queue)

Интерфейсът Queue дефинира основните действия за структурата опашка:

- **offer(T)** – добавя елемент накрая на опашката
- **poll()** – взима елемента от началото на опашката и го премахва
- **peek()** – връща елементът от началото на опашката без да го премахва
- **clear()** – премахва всички елементи от опашката
- **contains(T)** – проверява дали елемента се съдържа в опашката

Може да се използва клас LinkedList за работа с опашка.

Речници и хеш-таблицы

При речниците заедно с данните, които държим, пазим и ключ, по който ги намираме. Елементите на речниците са двойки (ключ, стойност), като ключът се използва при търсене.

Използвайте реализация на речник чрез хеш-таблицы, когато се нуждаете от максимално бързо намиране на стойностите по ключ.

HashMap<K,V>

- **put(K, V)** добавя нова стойност за даден ключ или презаписва вече съществуващата за този ключ
- **putAll(Map<K, V>)** добавя всички наредени двойки от друг речник в текущия. Извикването на този метод е еквивалентно на извикването на put(K, V) за всеки един елемент на речника, който е подаден като параметър.
- **get(Object)** връща стойността за дадения ключ или null, ако няма елемент с такъв ключ.
- **clear()** премахва всички елементи от речника.
- **boolean containsKey(K)** проверява дали в речника присъства наредена двойка с посочения ключ.
- **boolean containsValue(V)** проверява дали в речника присъстват една или повече наредени двойки с посочената стойност. Тази операция работи бавно, тъй като проверява всеки елемент на хеш-таблицата.

HashMap<K,V>

- **boolean isEmpty()** Проверява дали HashMap-а е празен
- **int size()** връща броя на наредените двойки в речника.
- **remove(K)** изтрива от речника елемента с този ключ.
- **boolean values().remove("key value")** Премахва елемент от HashMap<K,V> по стойност

Задачи

Задача

Даден е числов израз като стринг. Да се направи проверка дали скобите в този списък са сложени правилно.

Напр.: „(1+2) * ((5+6) + (7/5)))“

Задача

Да се напише програма, която записва съобщения в списък и ги извежда в реда, в който са записани в него. Използвайте структурата от данни Queue.

Задача

Направете речник с помощта на структурата от данни HashMap.

Домашно

Задача

Прочетете това:

- <http://www.introprogramming.info/intro-java-book/read-online/glava16-lineini-strukturi-ot-danni/>
- <http://www.introprogramming.info/intro-java-book/read-online/glava18-rechnichesh-tablici-i-mnojestva/>