Java

Collections, Annotations, HTTP

Сергей Рыбалкин

Collections, Annotations, HTTP

Отметьтесь на портале.

План лекции

Collection

Map

Annotations

HTTP

План лекции

Collection

Map

Annotations

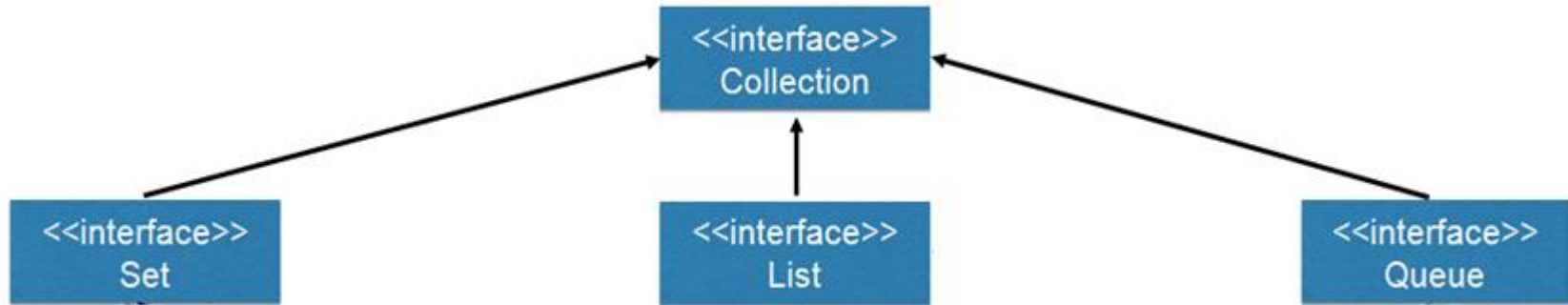HTTP

Iterator

interface Iterator<E>

- boolean hasNext();
- E next();
- default void remove() {
    throw new UnsupportedOperationException("remove");
  }

Iterable

interface Iterable<T>

- Iterator<T> iterator();
- E next();
- default void forEach(Consumer<? super T> action) {
    Objects.requireNonNull(action);
    for (T t: this) {
        action.accept(t);
    }
  }

Collection

interface Collection<E> extends
Iterable<E>

- boolean contains(Object o)
- boolean add(E e);
- boolean remove(Object o);
- Iterator<E> iterator()
- int size();

Collection

List

interface List<E> extends Collection<E>

- E get(int index);
- E set(int index, E element);
- void add(int index, E element);
- E remove(int index);
- List<E> subList(int fromIndex, int toIndex);

Ordered collection (sequence)

Random access

interface RandomAccess

Marker interface.
Indicate that List support fast (generally constant time) random access.
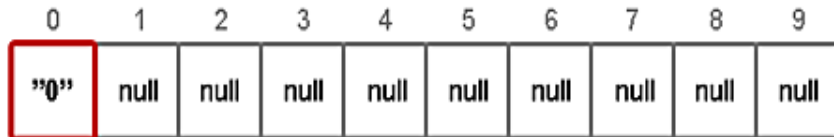
For generics algorithms.

List. Implementations

- ArratList
- CopyOnWriteArrayList
- LinkedList
- @Deprecated Vector
- …

ArrayList

class ArrayList<E> implements List<E>

Also:
extends   AbstractList<E>
Implements RandomAccess, Cloneable, Serializable

0	1	2	3	4	5	6	7	8	9
"0"	null	null	null	null	null	null	null	null	null

ArrayList

	0	1	2	3	4	5	6	7	8	9
	"0"	null	null	null	null	null	null	null	null	null

ArrayList. Complexity

TEXHOATOM

contains	add	get	set	remove
O(n)	O(1) *	O(1)	O(1)	O(n)

CopyOnWriteArrayList

class CopyOnWriteArrayList<E>
                     implements List<E>

Also:
Implements RandomAccess, Cloneable, Serializable

A thread-safe variant of ArrayList

CopyOnWriteArrayList. Complexity

TEXHOATOM

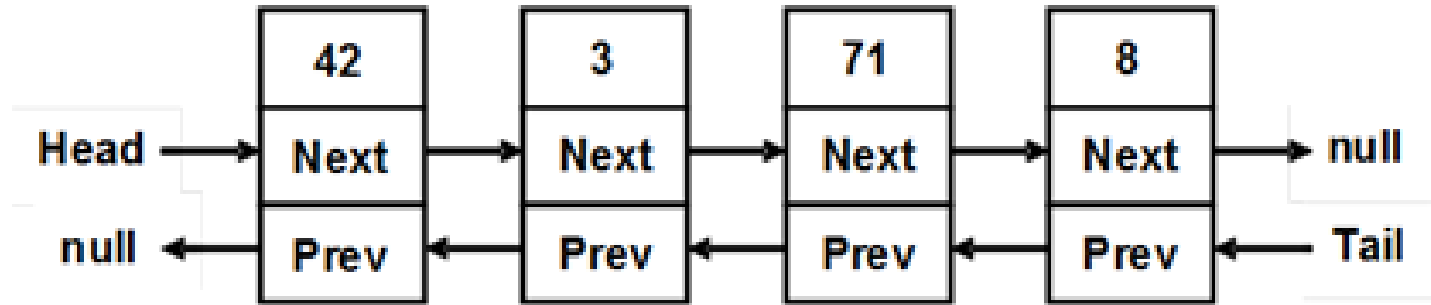| contains	add	get	set	remove
O(n)	O(n)	O(1)	O(n)	O(n)

LinkedList

class LinkedList<E> implements List<E>

Also:

Implements Deque<E>, Cloneable, Serializable

Doubly-linked list implementation

LinkedList

LinkedList. Complexity

contains	add	get	set	remove
O(n)	O(1)	O(n)	O(n)	O(n)

Set

interface Set<E> implements Collection<E>

A collection that contains no duplicate elements

Duplicate elements

How to check?

Duplicate elements. Formal

For each A != B in Set<E>
A.equals(B) == false

What about null?

Duplicate elements. Formal

What about null?

At most one null element

Set. Restrictions

Care about mutable objects!

The behavior of set is not specified if changes in object affects comparison.

Set. Implementation

- HashSet
- LinkedHashSet
- EnumSet
- TreeSet
- CopyOnWriteArraySet
- …

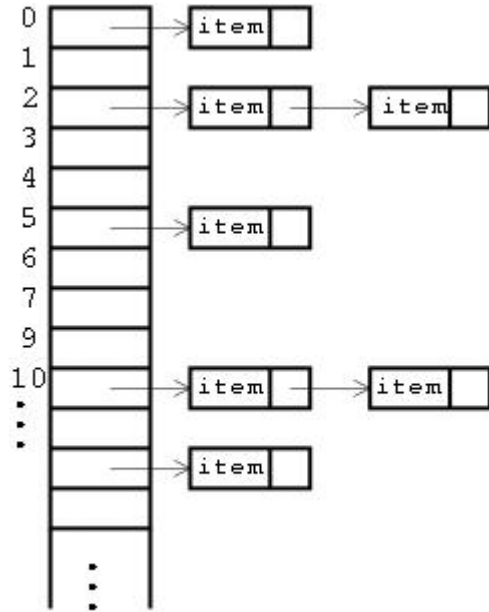## class HashSet<E> implements Set<E>

Also:
extends   AbstractSet<E>
Implements Cloneable, Serializable

hashCode metters
Object::hashCode(): int

HashSet

General contract

For objects a and b:

- a.equals(b) => a.hashCode() == b.hashCode()
- if a.hashCode() == b.hashCode()
    a may be not equal b
- a.hashcode() is the same during object lifetime

HashSet. Complexity

TEXHOATOM X

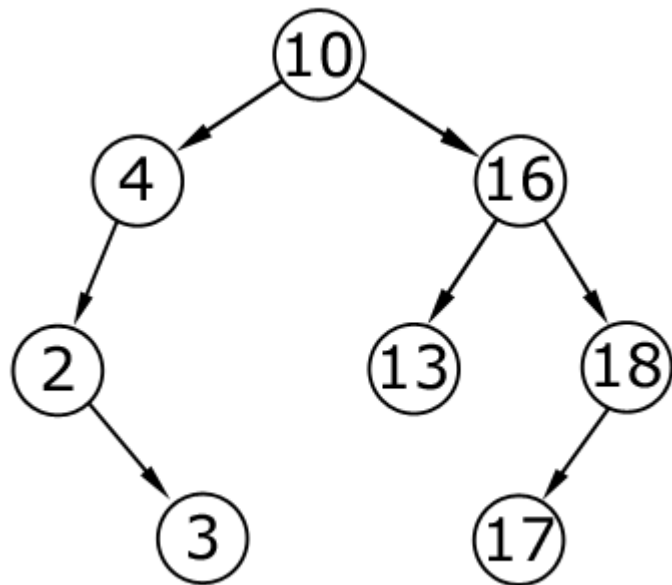| contains	add	remove
O(1)	O(1)	O(1)

class HashSet<E> implements Set<E>

Also:
extends   AbstractSet<E>
Implements SortedSet<E>, NavigableSet<E>, Cloneable, Serializable

ordering metters

TreeSet

Comparator

Functional interface Comparator<T>

static Comparator<Integer> *intComparator* =
                     (o1, o2) -> o1 - o2;

Comparable

Functional interface Comparable<T>

```
@Override
public int compareTo(T o) {
        return this.field – o.field;
}
```

compareTo & equals

Any type of contract?

compareTo & equals

a.equals(b) == true => a.compareTo(b) == 0

What about null?

TreeSet. Complexity

contains	add	remove
O(log(n))	O(log(n))	O(log(n))

План лекции

Collection

Map

Annotations

HTTP

Map

interface Map<K, V>

- An object that maps keys to values
- Cannot contain duplicate keys
- each key map to at most one value

- boolean containsKey(Object key);
- V get(Object key);
- V put(K key, V value);
- V remove(Object key);

Why, Map?

Ну реально, почему не Collection.

Why, Map?

From official FAQ:

>> This was by design.
>> We feel that mappings are not collections and collections are not mappings.
>> If a Map is a Collection, what are the elements?

Map. Implementations

- HashMap
- TreeMap
- LinkedHashMap
- EnumMap
- …

HashMap

HashSet is cutted HashMap

HashMap. Complexity

containsKey	get	put	remove
O(1)	O(1)	O(1)	O(1)

TreeMap

TreeSet is cutted TreeMap

TreeMap. Complexity

containsKey	get	put	remove
O(log(n))	O(log(n))	O(log(n))	O(log(n))

Reference

Java Collections – Performance (Time Complexity)

http://infotechgems.blogspot.ru/2011/11/java-collections-performance-time.html

План лекции

Collection

Map

Annotations

HTTP

Annotations

A form of syntactic metadata

May be available in run-time and compile-time

Annotations

@Override
@Deprecated
@FunctionalInterface
@SuppressWarnings

…
@NotNull
@Nullable

Annotations. Example

```
import java.lang.annotation.*;

@Target(ElementType.METHOD)
@Retention(RetentionPolicy.SOURCE)
public @interface Override {
}
```

Reflection API

About:
https://docs.oracle.com/javase/tutorial/reflect/

In brief:
you can look for annotations and find annotated "things"
and then …

План лекции

Collection

Map

Annotations

HTTP

HTTP

Spec:
https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html

In brief:

GET HTTP/1.1
Host: google.com

HTTP. Method

- Get
- Post
- Put
- Delete
- …

HTTP. Post

POST /auth/login
HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: localhost:8080

login=user&password=password

HTTP. curl

```
curl -X POST
    -H "Content-Type: application/x-www-form-urlencoded"
    -H "Host: localhost:8080"
    -d 'login=admin&password=admin'
http://localhost:8080/auth/login
```

```
curl
   -H 'Authorization: Bearer 2133e36c-8f31-455f-840e-1e034d4975fd'
http://localhost:8080/auth/dummy
```

```
OkHttpClient client = new OkHttpClient();
MediaType mediaType = MediaType.parse("application/x-www-form-
                              urlencoded");
RequestBody body = RequestBody.create(mediaType,
                              "login=admin&password=admin");
Request request = new Request.Builder()
        .url("http://localhost:8080/auth/login")
        .post(body)
        .addHeader("content-type", "application/x-www-form-urlencoded")
        .addHeader("host", "localhost:8080")
        .build();

Response response = client.newCall(request).execute();
```

Collections, Annotations, HTTP

Оставьте отзыв.

Спасибо
за внимание!

Сергей Рыбалкин

s.rybalkin@corp.mail.ru

ТЕХНОАТОМ

MVC