JavaOne™

Field guide to Java collections

Mike Duigou (@mjduigou)
Java Core Libraries

MAKE THE
FUTURE
JAVA

ORACLE

Required Reading

- Should have used most at some point
  - List, Vector, ArrayList, LinkedList, Arrays.asList
  - Set, HashSet, TreeSet
  - Queue, PriorityQueue
  - Map, Hashtable, HashMap, TreeMap
- Bonus Points
  - Deque, ConcurrentHashMap, CopyOnWriteArray, etc.

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Program Agenda

- Collection interfaces
  - Implementations tour
  - Unexpected usages
- Collections outside the JDK

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Collections Categories

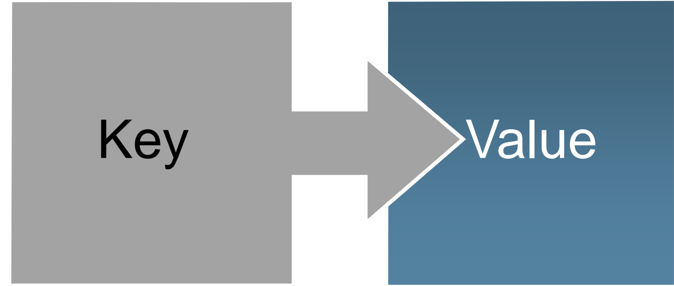
Establishing the categories

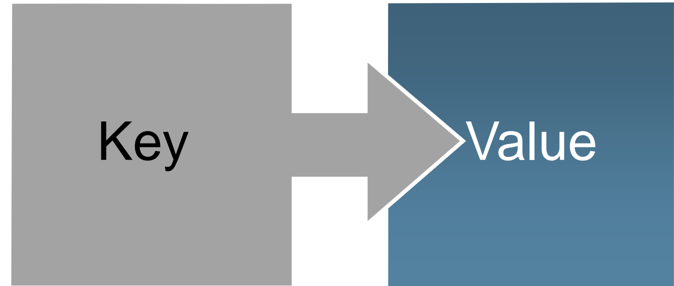
- Many collection classes
- Each implements one or more collection interfaces
- We will be classifying according to interfaces

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Two Fundamental “Shapes”



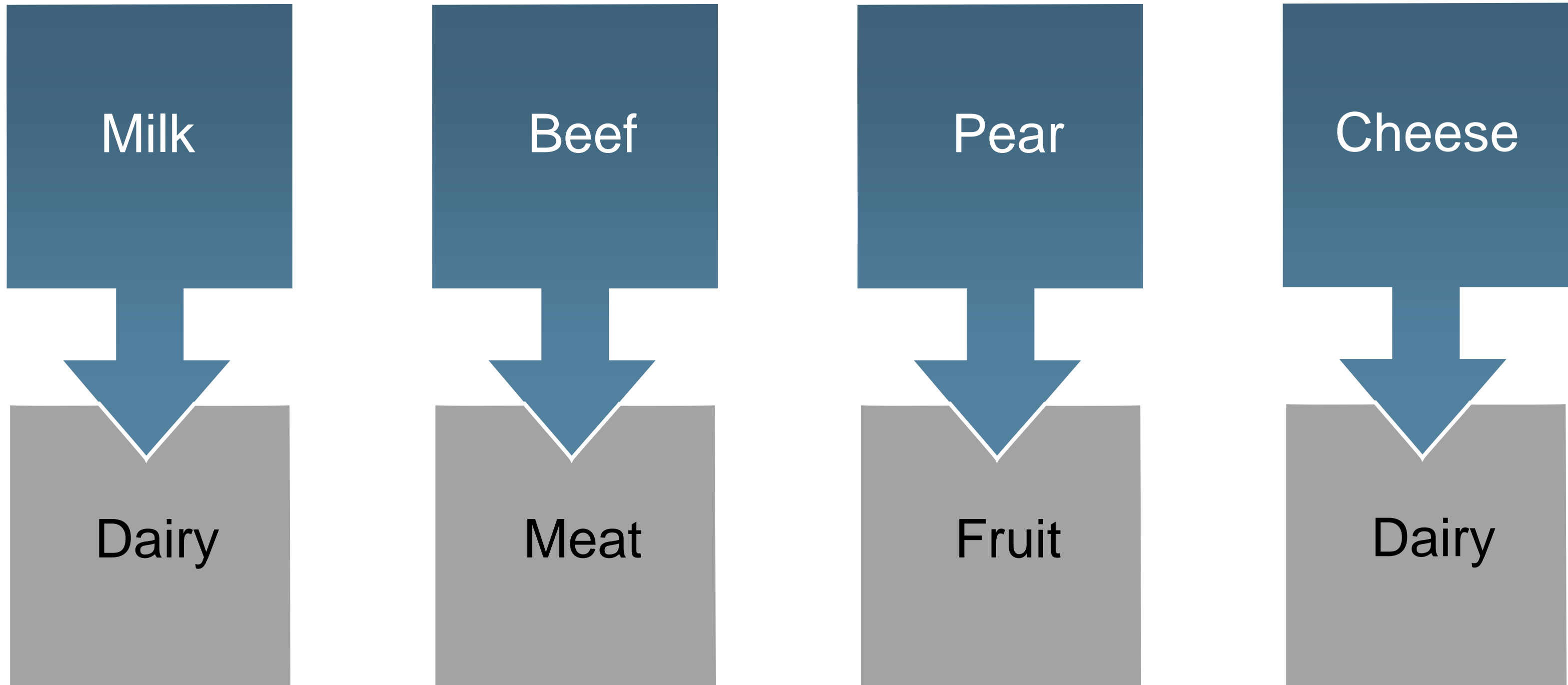


Collection **Map**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Map



Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Map

Not that kind of map

- Aggregation of key-value mappings
- Keys are unique
- Values can be anything
- Simple set of operations
  - get, contains<key|value>, remove, {put}<single|bulk>
  - iterate<keys|values|mappings>
  - equivalence, size
- The reason why `Object.hashCode()` exists
- `hashCode()` or `compareTo()` **must never** change while in Map

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

HashMap

Making a hash of it

- Array of buckets with chaining for collisions
- Bucket array is power of two sized
- Bucket index derived from hash code
  - Performance depends upon quality of **`hashCode()`** implementation
  - **O**(1) with good hash code, **O**(n) with worst hash code
- Expansion (aka rehashing) occurs when fullness threshold exceeded
- Iteration order for keys, values, elements is unspecified
  - and may become unpredictable (7u6 alternative hashing & Java 8)

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

LinkedHashMap

A link to the past

- Subclass of HashMap
- Tracks insertion order of mappings
- Iteration order is insertion order (predictable)
- Some are surprised it’s not a SortedMap
- Often used for LRU caches or where key order matters

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

WeakHashMap

Actually quite powerful

- Map which holds keys with weak references
- When key reference is cleared value is removed
- Commonly used for look up tables and caches
- Cache pattern: soft reference value holding strong reference to key

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

IdentityHashMap

Some objects are more equal than others

- Map which uses identity equality (==) rather than `equals()`
- Originally used for structure preserving object graph traversal (Serialization)
- Suggested use: associating meta-data with specific instances
- It is sometimes possible to use for higher performance map
  - How often this is worthwhile is debatable

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

EnumMap

Optimization is key

- Map specifically for enums (Java 5)
- Keys restricted to a single enum
- Very space efficient and high performance
  - Two implementations internally
- **O**(1) for put, get, remove, contains
- Elements are ordered according to enum order
- Not a SortedMap (frequently requested)

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Hashtable

It's not faster. Really. Stop saying that.

- Retrofitted into Collections framework
- All public methods are synchronized
- Does not allow `null` keys or values
- Array of buckets with chaining for collisions.
  - Bucket index derived from hash code. HashMap has better distribution
- Bucket array expansion by doubling when fullness threshold exceeded
- Iteration order for keys, values, elements is unspecified
  - and may become unpredictable

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

ConcurrentHashMap

Firing on all cylinders

- Fully concurrent
- Multiple arrays of buckets with chaining for collisions
- Individual bucket arrays grow by doubling
- Allergic to null (not supported)
  - Bonus: no need for separate contains() to determine if key present
- Iteration order for keys, values, elements is unspecified
- Iterators are consistent
  - But map may change *while* you are iterating
- Implements ConcurrentMap

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

ConcurrentMap

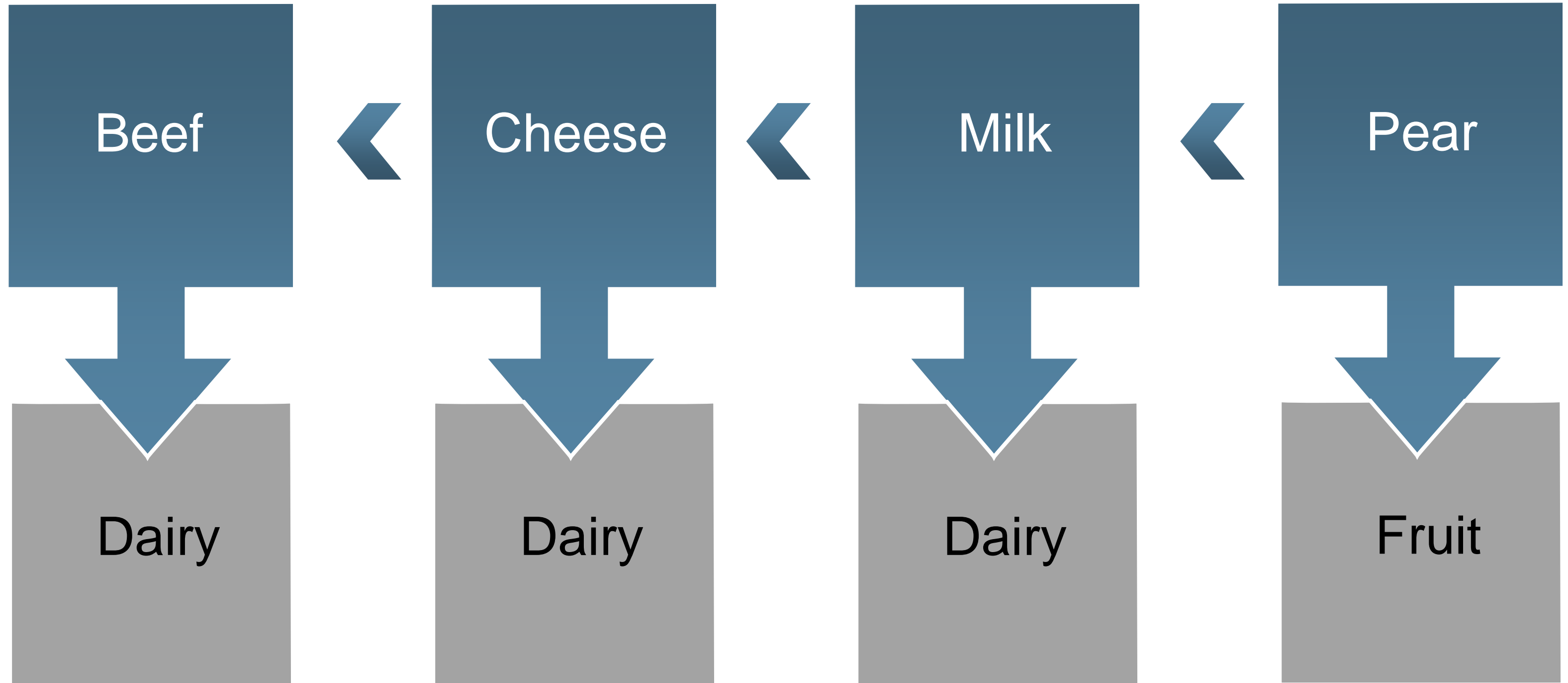
Atomics-r-us

- Adds several methods to Map interface
- Operations that *would* require synchronized block for safety
  - **`putIfAbsent`**
  - **`remove(key,value)`**
  - **`replace(key,value)`**
  - **`replace(key,value,newvalue)`**

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

SortedMap/NavigableMap



 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

SortedMap/NavigableMap

Which came first?

- Map with sorted keys
- Iteration order is key sort order
- Map partitioning features for creating sub-map views
- Any SortedMap can be used to make a SortedSet

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

SortedMap vs NavigableMap

What's the deal with that?

- `SortedMap/SortedSet` came first in Java 1.2
- `NavigableMap/SortedSet` added in Java 6
- `NavigableMap extends SortedMap`
- `NavigableSet extends SortedSet`
- `TreeMap/TreeSet` were upgraded to `NavigableMap/Set`
- Prefer Navigable to Sorted—more features
  - `lower()`, `floor()`, `higher()`, `ceiling()`
  - inclusive `headMap()` and `tailMap()`

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

TreeMap

Branchin out

- Red black balanced tree
- Basic operations (put, get, contains) are **O**(log2 n)
- Sorting via **`Comparator`** or **`Comparable`**
- **`null`** keys not permitted for “natural ordering” Comparable
- Comparators can support **`null`**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

ConcurrentSkipListMap

Massive backup at the maze

- Fully concurrent map implementation based on skip lists
- A much better choice for heavy concurrent use than synchronizedMap
  - Higher overhead though
- Implements ConcurrentNavigableMap
- Does not support null keys or values

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Collection

A place for your stuff

Fruit

Bread

Milk

Meat

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Collection

Not one to brag

- Aggregation of values
- Simple set of operations
  - {add, remove, contains}<single|bulk>, equivalence, iterate, size
- No implementations of `Collection` in JDK
  - Lots of sub-interface implementations
- Extra operations on `Collections`

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Collection-what it's not

Warranty void in Tennessee

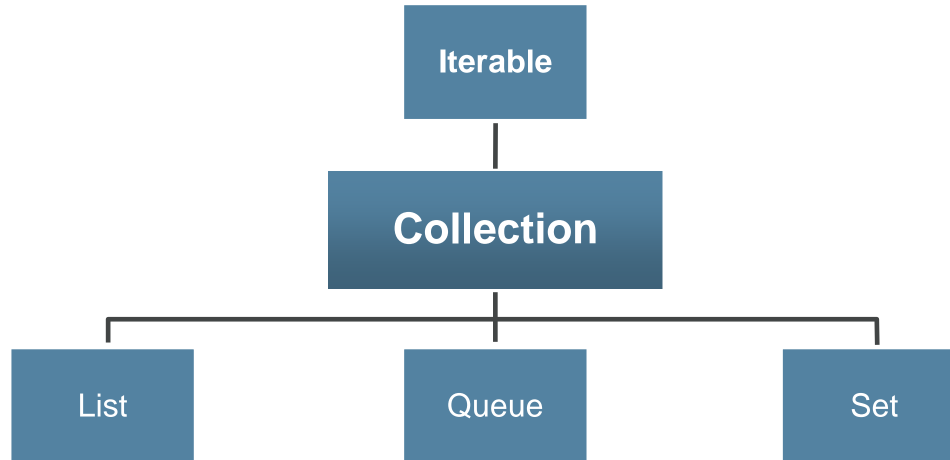
- Sub-interfaces add behaviour
  - Definition of `Collection.equals()` almost never used
- Elements might be ordered
- Duplicates may be allowed
- Mutability may be allowed
- Concurrency is (mostly) not part of definition

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Collection

Ignoring the differences for fun and profit

- Inheritance isn't just for specialization
- Commonality is expressed as well



 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Collection

Unspecific Specifics

- Variables and Fields

```
List<String> foo = new ArrayList<>();
```

- Method Parameters

```
public void calculate(ArrayList<String> bar);
```

- Return types

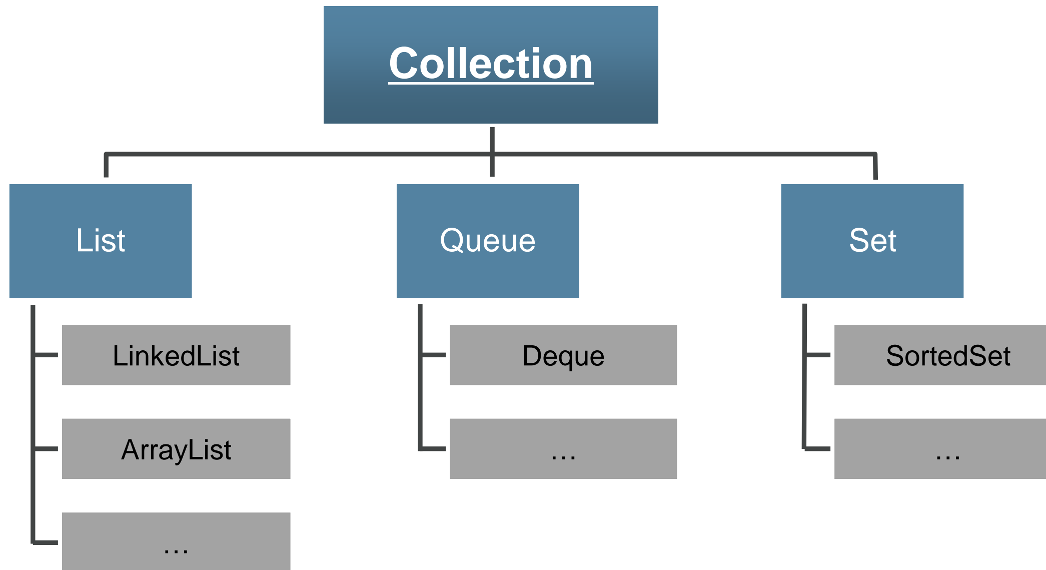
```
public List<String> sort(List<String> bar);
public <A extends List<String>> A sort(A bar)
ArrayList<String> sorted = sort(list);
```

- If you ain't going to need it, don't declare it!
  - You may want to keep reference with original type for some methods

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Collection

Finally, an inheritance hierarchy



 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Collection : Set

My Grocery ~~List~~ Set

Fruit

Bread

Milk

Meat

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Set

Are you in or out?

- Collection of unique elements—no duplicates
- Elements are not ordered
- Iteration order is not defined
- Iteration order may not be consistent
- SortedSet/NavigableSet also available
- Make any **`Map`** a **`Set`** with **`Collections.newSetFromMap()`**

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

EnumSet

The “members only” Set

- Set specifically for enums (Java 5)
- Instance restricted to a single enum
- Very space efficient and high performance
  - Two implementations internally
- **O**(1) for add, remove, contains
- Elements are ordered according to enum order
- Not a SortedSet (frequently requested)

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

HashSet

Hashing it out (or is it in?)

- Set based upon HashMap
  - Can be a little wasteful (high overhead) for small objects
- **O**(1) for add, remove, contains
- Elements are unordered
- Iteration order may be unpredictable

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

LinkedHashSet

Follow the breadcrumbs

- Set based upon LinkedHashMap
  - Can be a little wasteful (high overhead) for small objects
- **O**(1) for add, remove, contains
- Iteration order is insertion order
- Often used for LRU caches
- Some are surprised it's not a SortedSet

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Collection : SortedSet

For the highly organized

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Collection : SortedSet

More than just sorted

- Collection of unique ordered elements
- Iteration order is sort order
- Subset partitioning is very useful

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

TreeSet

Every leaf is unique

- Based upon TreeMap
  - Can be a little wasteful (high overhead) for small objects
- Elements stored in a balanced binary tree
- **O**(log n) for add, remove, contains
- Elements ordered via **`Comparator`** or **`Comparable`** elements
- **`null`** allowed if supported by **`Comparator`**
- Concurrent via **`Collections.synchronizedSortedSet()`**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

ConcurrentSkipListSet

Easier to use than type

- Based upon ConcurrentSkipListMap
- Elements stored in a skip list (aka super linked list)
- **O**(log n) average for add, remove, contains
- Elements ordered via **`Comparator`** or **`Comparable`** elements
- Iteration order is sort order
- Fully concurrent
  - Surprising: Iterating concurrently with add and remove
  - **`size()`** is **O**(n) and also subject to concurrent modification

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Collection : List

First, make a List

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

List

More than meets the eye

- Commonly used like a more flexible array
  - There is a huge difference between `List<Integer>` and `int[]`
- Elements are ordered, duplicates are allowed
- Random access is offered (but beware)
  - `RandomAccess` marker interface on truly random access Lists

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

ArrayList

Everybody knows your name

- List backed by an array
- Random Access
- Grows automatically, shrinking by explicit request
  - **`trimToSize()`** is a method of **`ArrayList`** not of **`List`**
- Grows by 1.5x as needed by creating a new larger array (copies data)
- Single threaded only (use **`Collections.synchronizedList`**)
  - Which means you will lose **`trimToSize()`** and **`ensureCapacity()`**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Arrays.asList()

aka “The Fake ArrayList”

- Created from an array
- **`java.util.Arrays.ArrayList`** is it’s name
- List backed by an array
- Random Access
- Size is fixed
- Unable to grow or shrink
- May still need **`Collections.synchronizedList`**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

CopyOnWriteArray

Where did all these clones come from?

- Fully concurrent array-based List implementation
- Random Access
- Modification copies the backing array
- Iterators use a snapshot of List
- Great when you need concurrent modification but only rarely
- Heavy updates will cause too much copying

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Vector

Faded Hero

- List backed by an array
- Random Access
- Grows automatically, shrinking by explicit request
  - `trimToSize()` is a method of `Vector` not of `List`
- Grows by 2x or increment by creating a new larger array (copies data)
- All public methods are synchronized
  - No need to call `Collections.synchronizedList`
  - No way to turn of synchronization. (But HotSpot is magic)

 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

LinkedList

Indexed access is the weakest link

- Implemented as a doubly linked list
- Cheaper add/remove within list
- Not random access
- Object overhead per element
- Single threaded only
- Also supports `Queue` and `Deque` interfaces

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Collection : Queue

Things are poppin'

Add

Remove

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

Queue/Deque

Get in line

- Ordered and allows duplicates like List
- Special semantics for adding and removing
  - And possibly “full” and “empty”
- Deque provides additional add/remove options
- Not particularly useful as immutable structure

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

ArrayDeque

So you don’t like LinkedList

- Deque implementation similar to ArrayList
- Grows as needed but never shrinks
  - LinkedList may be a better choice for highly elastic queues
- Performance better for LIFO (stack) for FIFO (queue)

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

PriorityQueue

Getting a better position

- Sorts according to **`Comparator`** or **`Comparable`**
- Removes "least" element next
- Sorting happens only at addition
  - Need to change priority? **Must** remove and re-add
- Not concurrent by default
  - Use **`PriorityBlockingQueue`** rather than **`synchronized`**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

ConcurrentLinkedQueue

Getting a better position

- Unbounded concurrent queue
- Main advantage is latency for push/pop
- **Not a BlockingQueue**

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Blocking Queue/Deque

We are experiencing heavy call volume

- Block until space available
- Block until element available
- Only make sense when used concurrently

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

ArrayBlockingQueue

- Fixed size
- Fully concurrent
- Offers optional “fairness”
  - For very unbalanced producer/consumer more fairness might be needed

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

LinkedBlockingQueue/Deque

- Optionally unbounded useful for periodic long queues
- Fully concurrent
- Somewhat higher throughput than blocking
- High throughput can increase GC pressure
- No “fairness” behaviour needed
  - Adding and removing only contend at empty <-> non empty state.

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

DelayQueue

- Concurrent unbounded Queue
- Elements can only be retrieved after they have expired
- Next element returned is the “most expired”
- Can consider DelayQueue as a time oriented PriorityQueue

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

SynchronousQueue

- Concurrent queue with no capacity
- Pushing blocks until it is retrieved by another thread
- Retrieving blocks until an element is available from another thread
- Optional fairness policy to make waiting more FIFO
- Useful when you must know that handoff has occurred

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
JavaOne ORACLE

Lambda and Collections

Same collections, new features

- Java 8 will add lambdas to the language
- Major libraries upgrade focused on use of lambda with collections
- Lambda for bulk data operations including parallel
- JDK Collections will be extended to be data sources for bulk data APIs

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Other collections

GS Collections -- github.com/goldmansachs/gs-collections

- Improves readability and reduces duplication of iteration code (enforces DRY/OAOO)
- Implements several, high-level iteration patterns (select, reject, collect, inject into, etc.) on "humane" container interfaces which are extensions of the JDK interfaces. Provides a consistent mechanism for iterating over Collections, Arrays, Maps, and Strings
- Provides replacements for ArrayList, HashSet, and HashMap optimized for performance and memory usage. Adds new containers including Bag, Interval, Multimap, and immutable versions of all types
- Encapsulates a lot of the structural complexity of parallel iteration and lazy evaluation. Performs more "behind-the-scene" optimizations in utility classes
- Has been under active development since 2005 and is a mature library

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Other collections

Guava -- code.google.com/p/guava-libraries

Guava is the open-sourced version of Google's core Java libraries: the core utilities that Googlers use every day in their code. The Guava utilities have been carefully designed, tested, optimized and used in production at Google. You don't need to write them, test them, or optimize them: you can just use them.

- Additional collections, collections utilities
- Lots of non-collections utilities
- Some very small bits are planned to be added to Java 8

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Q & A

@mjduigou

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

JavaOne ORACLE

JavaOne™