

Collection is Iterable (method "iterator()" returns an  
Iterator)

Generic

Interfaces

List

Set

SortedSet

Queue

Deque

Classes

ArrayList

LinkedList

HashSet

TreeSet

PriorityQueue

Stack (library says they prefer Deque but it  
breaks contract)

TreeSet

Collections

Comparable

Comparator

Natural order

binarySearch(List)  
copy(List)  
replaceAll(List, T oldval, T newVal)  
reverseOrder()  
shuffle(List)  
sort(List)

Java api

You can switch on strings

Every non-primitive is a pointer

Can't treat arrays as pointers

Arrays

Allocated using new

Create n-dimensional at "new"

or creation time

Run time vs compile time

Class loading order

Annotations

@Override

\u code for special characters

Unicode

"Good Methods"

Empty constructor

Initialization constructor

toString

equals

hashCode

Getters/Setters

can be generated automatically

Style conventions

Collection is Iterable ( single method iterator()) interface

Interfaces

Deque

List

Queue

Set

SortedSet

SubClasses

ArrayDeque

ArrayList

HashSet

LinkedList

Stack (prefer Deque but this breaks assumed  
contracts)

TreeSet

Vector

## Collections

- Comparable

- Comparator

- Operation example

  - binarySearch

  - replaceAll

  - reverseOrder

  - sort

## Copy Objects

- Copying of primitives is just a real copy

- Objects

  - copy pointers

  - copy constructor

    - shallow copy

    - deep copy

## Classes defined in

- Same file as another class

- Inner classes

- Anonymous inner classes

## Move .class files to other directory

## Printing

- Printf

- String concatenation

## Overriding methods

Everything is virtual

Final methods

Autoboxing

String

== vs equals

s.charAt, s.trim, s.substring(int, [int]),  
startsWith, indexOf, Integer.toString

NOT array

Special characters : \n, \t, etc

Copy of string = String copy = new String(s)

NOT string = s

Exceptions

Throwable, Error, Exceptions,

RuntimeExceptions

Just objects, treated differently

Message (getMessage)

printStackTrace

Customize by extension

Try/Catch/Finally

Resource defining try

Order of catches

Can catch anything

Interfaces

Special for loops



Class defined in same file as class name

Make a file's class public

String[] args

System.out.println()

primitive, Primitive pairing (different names)

    parsePrimitive

true is true, false is false

System.err vs System.out

String concatenation

    NOT fast

StringBuilder – insert, delete, append, etc.

Mix ints and chars to an extent, cannot mix

    boolean s and ints

array.length

user defined types

    constructors

    toString

Automatically imported classes

    Java.lang.\*

        Math

        Object

        Reflection

String

Java.lang.System

Atomic Classes

String

Literals

Immutable

Concatenation +

" vs ""

StringBuilder

I/O: Scanner, PrintWriter, Readers vs Streams, File

Don't forget to close it