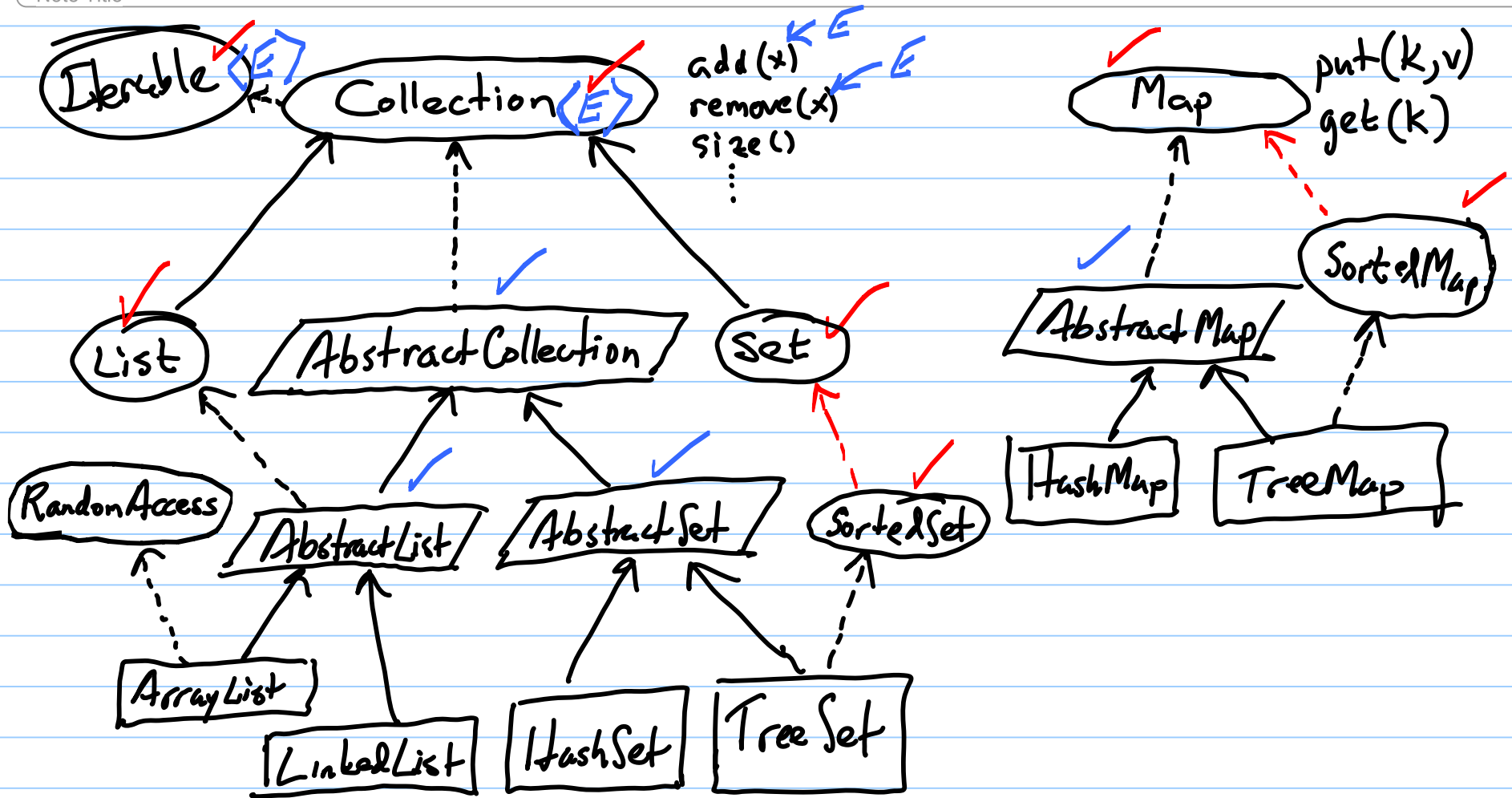


Java Generics, Polymorphic Types, Orthogonality

Note Title



Seen: Set<String>

Map<Fruit, Color>

Better than: Set<Object>

Map<Object, Object>

What about sorting within collections?

natural order

define an ordering application specific

⇒ objects in the collection implement

Comparable

x.compareTo(y)

⇒ application provides a Comparator

int compare(x, y)

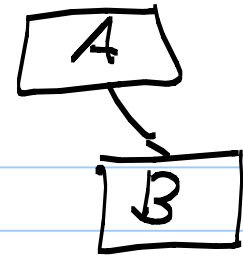
Collection $\langle E \rangle$

OR

objects implement
Comparable $\langle E \rangle$

Comparator $\langle E \rangle$

Comparator $\langle ? \text{ super } E \rangle$



What about add w.r.t. substitution & types?

add (Collection $\langle E \rangle$ x)

instead: add (Collection $\langle ? \text{ extends } E \rangle$ x)

java.util.Collections

Collections.sort(List<E> l) ← natural order

Collections.sort(List<E> l, Comparator<? super E> c)