

1 الفرق بين List و Set

إجابة الإنرفيو:

List يحفظ على الترتيب ويسمح بالتكرار ،
Set لا تسمح بالتكرار ويعتمد على equals و hashCode.

التوبي الأساي:

Data Structure Behavior + Equality Contract

الشرح:

- List:
 - Ordered
 - Duplicates allowed
 - Access by index
- Set:
 - No duplicates
 - مفيش index
 - بيستخدم:
 - equals()
 - hashCode()

⚠ Tricky:

غلط hashCode و equals →
يسمح بتكرار بدون ما تحس HashSet.

2 ArrayList vs LinkedList

إجابة الإنرفيو:

ArrayList أسرع في access،
LinkedList أسرع في add/remove.

التوبي الأساي:

Internal Data Structure



- **ArrayList:**

- Dynamic Array
- $\text{get}(i) \rightarrow O(1)$
- add وسط $\rightarrow O(n)$

- **LinkedList:**

- Doubly Linked List
- $\text{get}(i) \rightarrow O(n)$
- $\text{add/remove} \rightarrow O(1)$



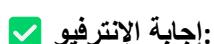
LinkedList أكثر بتسهيلاتك memory (nodes + pointers).

ArrayList vs LinkedList 2			
	LinkedList	ArrayList	البنية
Doubly Linked List	بطيء	سرع	Access
O(n)	سرع	بطيء	وسط Add/Remove
أكبر	أقل	أقل	Memory

ArrayList → كثير Access ✘
LinkedList → كثير Insert/Delete ✘

↓

3 HashMap internal working



إجابة الإنترفيو:

HashMap بتنستخدم hashing لتحديد bucket، collisions بتعالج بـ LinkedList أو Tree.

 التوبك الأساسي:

Hashing + Buckets + Collision Resolution

 الشرح:

1. hashCode() → hash
2. hash → index
3. لو bucket فاضي → insert
4. لو collision:
 - Java < 8 → LinkedList
 - Java ≥ 8 → Tree (Red-Black)

 Tricky:

Worst case:

- قبل Java 8 → O(n)
 - بعد Java 8 → O(log n)
-

 equals() & hashCode()

 إجابة الإنترفيو:

مع بعض الاتنين لازم يتعمل لهم.

 التوبك الأساسي:

Object Equality Contract

 الشرح:

- equals():
 - مقارنة منطقية
- hashCode():
 - توزيع العناصر في Hash-based collections

Rules:

- equals true → hashCode same
 - hashCode same ≠ equals true
-

5 ConcurrentHashMap

 إجابة الإنترفيو:

Thread-safe Map بدون locking. كامل.

 التوبك الأساسي:

Concurrency + Fine-Grained Locking

 الشرح:

- No synchronized on whole map
- Uses:
 - CAS
 - Segment-level locking (قدیم)
 - Node-level locking (حديث)

Tricky:

-  لا يسمح بـ null key أو value
 - iteration fail-safe
-

6 Fail-Fast vs Fail-Safe

 إجابة الإنترفيو:

Fail-fast يرمي exception،

Fail-safe يشتغل على نسخة.

 التوبك الأساسي:

Iterator Behavior



الشرح:

- Fail-Fast:
 - ArrayList
 - HashMap
 - ConcurrentModificationException
- Fail-Safe:
 - CopyOnWriteArrayList
 - ConcurrentHashMap



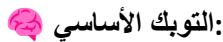
Fail-safe أبطأ ومستهلك memory.

7 TreeSet vs HashSet



إجابة الإنترفيو:

TreeSet أسرع من HashSet.



التوبك الأساسي:

Sorting + Comparable / Comparator



الشرح:

- HashSet:
 - hash-based
 - O(1)
- TreeSet:
 - Red-Black Tree
 - O(log n)
 - sorted



Tricky:

TreeSet محتاج:

- Comparable أو
 - Comparator
-

8 Map vs Collection

✓ إجابة الإنترفيو:

Collection، عناصر،

Map علاقات key-value.

💡 التوبك الأساسي:

Data Modeling Concept

🔍 الشرح:

- Collection:
 - add(E)
- Map:
 - put(K,V)

Map بتشتغل على:

- keys
 - values
 - entries
-

9 Why Map is not Collection?

✓ إجابة الإنترفيو (مهمة جداً):

لا تمثل عناصر منفردة Map لأن.

💡 التوبك الأساسي:

Design Philosophy of Java Collections

🔍 الشرح:

- Collection = element-centric
- Map = relationship-centric

عasan كده:

Map does NOT extend Collection

10 When to use LinkedHashMap

 إجابة الإنترفيو:

بترتيب Map لما تحتاج.

 التوبك الأساسي:

Ordering + LRU Cache

 الشرح:

- insertion order
- access order

 Tricky:

ممكن تستخدمه في:

- Cache
 - Session management
-

1 الفرق بين hashCode() و equals()

 إجابة إنترفيو مختصرة:

equals() يقارن منطقياً،

hashCode() يحدد مكان التخزين في الـ hash-based collections.

 التوبك:

Object Equality Contract

الشرح التفصيلي:

equals()

- بيرد على سؤال:

ده مساوي للثاني منطقياً؟ object هل الـ

مثال:

```
String a = new String("Omar");
```

```
String b = new String("Omar");
```

```
a.equals(b); // true
```

```
a == b; // false
```

hashCode()

- بيرد على سؤال:

أنهـي؟ ده فيـي bucket أحـط الـ

بـيرجـع رقم (int).

بيـستخدمـهمـ؟ إـزـايـ HashMap

1. يـحدـدـ الـ hashCode() يـنـادـيـ bucket

2. لو فيه عـاـصـرـ:

- عـلـشـانـ يـتـأـكـدـ equals() يـنـادـيـ

القـوـاعـدـ المـهـمـةـ (Contract):

- equals() رـجـعـتـ true يـبـقـىـ مـتـسـاوـيـ hashCode() لـازـمـ →
- مـتـسـاوـيـ hashCode() لوـ شـرـطـ equals() true مـشـ شـرـطـ →

 **Tricky:**

override equals → بس يسمح بتكرار HashSet ✗

 **2** الفرق بين Comparable و Comparator

 إجابة إنترفيو مختصرة:

Comparable ترتيب داخلي،

Comparator ترتيب خارجي.

 التوبك:

Sorting Strategy

 الشرح:

Comparable

- يحدد الترتيب الطبيعي
- جوه الكلاس نفسه

```
class User implements Comparable<User> {  
    public int compareTo(User u) {  
        return this.age - u.age;  
    }  
}
```

 ترتيب واحد بس.

Comparator

- ترتيب خارجي
- أكثر من ترتيب

```
Comparator<User> byName =  
(u1, u2) -> u1.name.compareTo(u2.name);
```

 مرن أكثر.

Tricky:

TreeSet ملزم:

- Comparable أو
 - Comparator
-

الفرق بين Iterable و Iterator

 إجابة إنترفيو مختصرة:

Iterable يتسمح بالـ for-each،
Iterator يبني على العناصر.

التوبك:

Iteration Mechanism

الشرح:

Iterable

- Interface فيها method:

```
Iterator<T> iterator();
```

علشان:

```
for (T t : collection)
```

Iterator

- مسؤول عن traversal

```
Iterator<Integer> it = list.iterator();
while (it.hasNext()) {
    System.out.println(it.next());
}
```

⚠ Tricky:

- remove() في Iterator
 - remove() في collection أثناء loop → Exception
-

💡 الفرق بين Collection و Collections

✓ إجابة إنترفيو مختصرة:

Collection Interface،
Collections Utility Class.

التوبك:

API Design

🔍 الشرح:

Collection

- Interface
- parent لـ List, Set, Queue

```
Collection<String> c = new ArrayList<>();
```

Collections

- Class
- فيها static methods

```
Collections.sort(list);  
Collections.synchronizedList(list);
```

 **Tricky:**

- Collection = data
 - Collections = helper methods
-