

Java Interview Diagrams + Cheat Sheets 2026

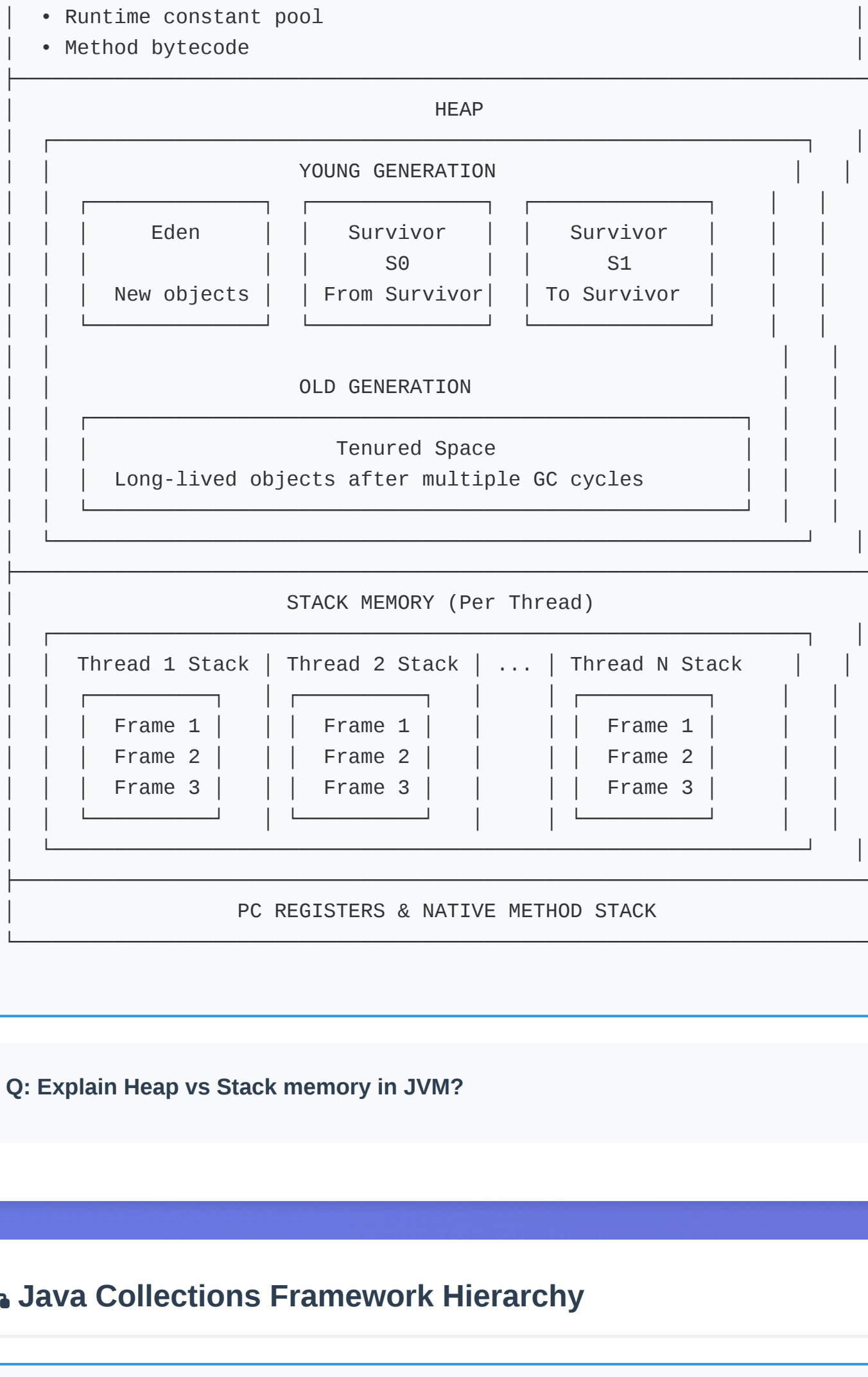
Essential Visual Guides & Quick References for Java Interviews

Java 21 LTS

Interview Focused

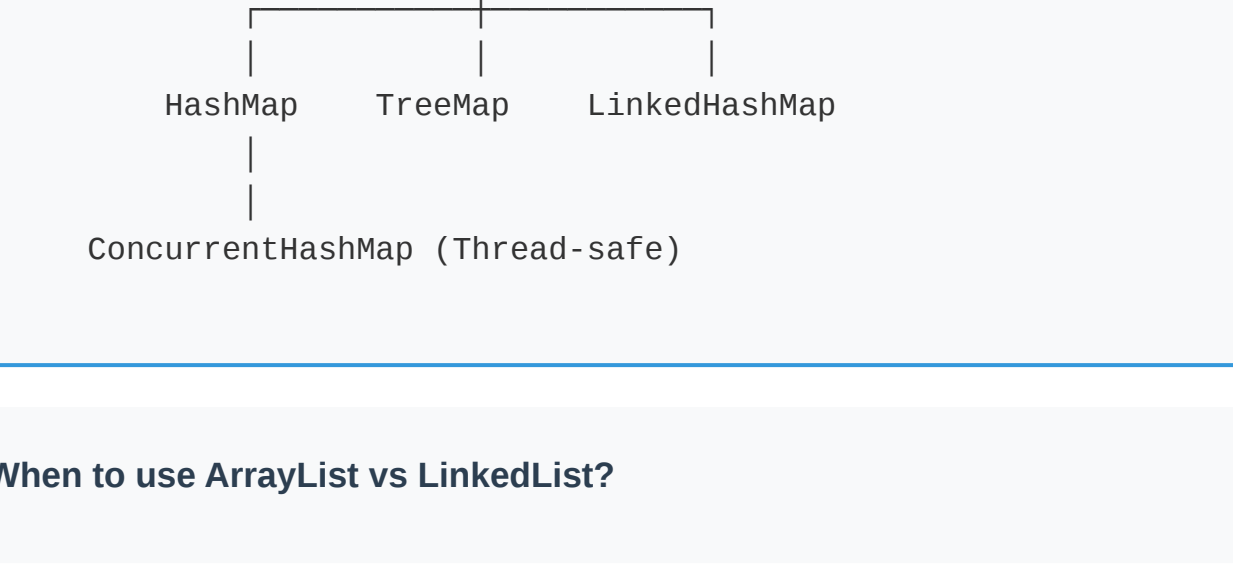
Virtual Threads

JVM Memory Architecture (Interview MUST KNOW)



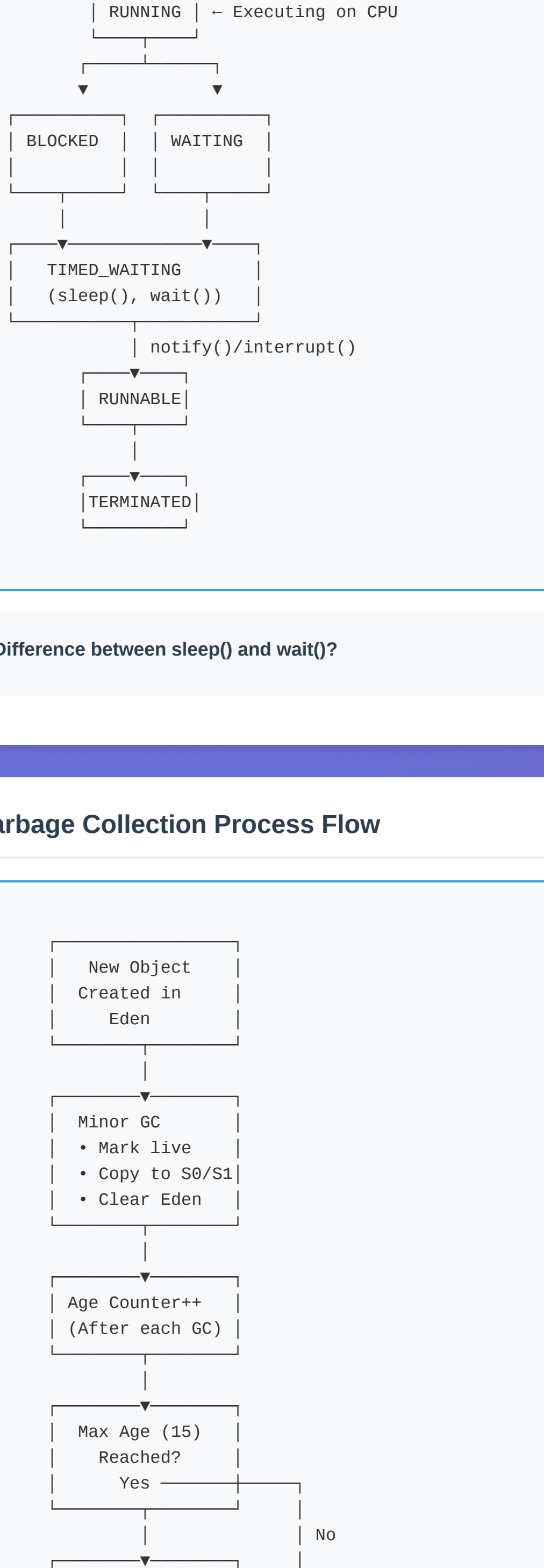
Q: Explain Heap vs Stack memory in JVM?

Java Collections Framework Hierarchy



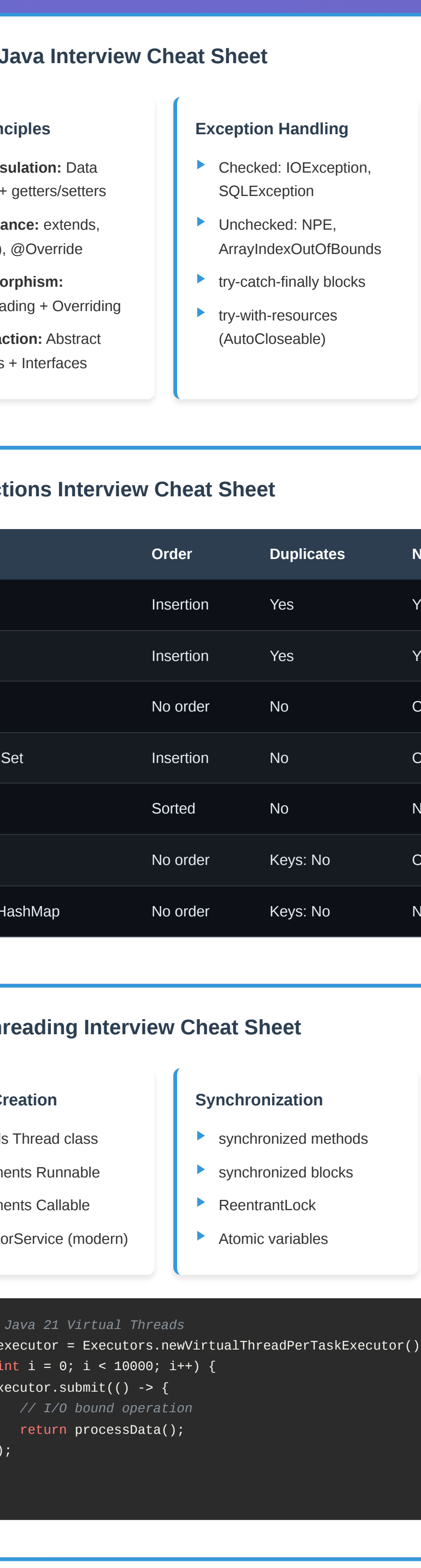
Q: When to use ArrayList vs LinkedList?

Thread Lifecycle States



Q: Difference between sleep() and wait()?

Garbage Collection Process Flow



Q: Explain G1GC vs CMS garbage collectors?

Core Java Interview Cheat Sheet

OOP Principles

- ▶ **Encapsulation:** Data hiding + getters/setters
- ▶ **Inheritance:** extends, super(), @Override
- ▶ **Polymorphism:** Overloading + Overriding
- ▶ **Abstraction:** Abstract classes + Interfaces

Exception Handling

- ▶ **Checked:** IOException, SQLException
- ▶ **Unchecked:** NPE, ArrayIndexOutOfBoundsException
- ▶ **try-catch-finally blocks**
- ▶ **try-with-resources (AutoCloseable)**

String vs StringBuilder

- ▶ **String:** Immutable, thread-safe
- ▶ **StringBuilder:** Mutable, not thread-safe
- ▶ **StringBuffer:** Mutable, thread-safe
- ▶ Use StringBuilder for loops

equals() vs ==

- ▶ **==:** Reference comparison
- ▶ **equals():** Content comparison
- ▶ Always override hashCode() with equals()
- ▶ Contract: `a.equals(b) → a.hashCode() == b.hashCode()`

Collections Interview Cheat Sheet

Collection	Order	Duplicates	Null Values	Thread-Safe	Best For
ArrayList	Insertion	Yes	Yes	No	Frequent read
LinkedList	Insertion	Yes	Yes	No	Frequent add/remove
HashSet	No order	No	One null	No	Unique elements
LinkedHashSet	Insertion	No	One null	No	Unique + Order
TreeSet	Sorted	No	No	No	Sorted unique
HashMap	No order	Keys: No	One null key	No	General purpose
ConcurrentHashMap	No order	Keys: No	No	Yes	Multi-threaded

Multithreading Interview Cheat Sheet

Thread Creation

- ▶ Extends Thread class
- ▶ Implements Runnable
- ▶ Implements Callable
- ▶ ExecutorService (modern)

Synchronization

- ▶ synchronized methods
- ▶ synchronized blocks
- ▶ ReentrantLock
- ▶ Atomic variables

Thread Pools

- ▶ FixedThreadPool
- ▶ CachedThreadPool
- ▶ ScheduledThreadPool
- ▶ Virtual Threads (Java 21)

Concurrent Collections

- ▶ ConcurrentHashMap
- ▶ CopyOnWriteArrayList
- ▶ BlockingQueue
- ▶ ConcurrentLinkedQueue

```
// Modern Java 21 Virtual Threads
try (var executor = Executors.newVirtualThreadPerTaskExecutor()) {
    for (int i = 0; i < 10000; i++) {
        executor.submit(() -> {
            // I/O bound operation
            return processData();
        });
    }
}
```

Spring Framework Interview Cheat Sheet

Core Concepts

- ▶ IoC (Inversion of Control)
- ▶ DI (Dependency Injection)
- ▶ Bean Lifecycle
- ▶ AOP (Aspect Oriented)

Annotations

- ▶ @Component, @Service
- ▶ @Repository, @Controller
- ▶ @Autowired, @Qualifier
- ▶ @Value, @Configuration

Spring Boot

- ▶ Auto-configuration
- ▶ Starter dependencies
- ▶ ClassCastException
- ▶ Actuator endpoints

Spring Data JPA

- ▶ Repository interfaces
- ▶ @Entity, @Table
- ▶ @Id, @GeneratedValue
- ▶ @OneToMany, @ManyToOne

Modern Java Patterns (Java 17+)

Old Pattern (Pre-Java 14)

```
public class Person {
    private final String name;
    private final int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Getters, equals, hashCode, toString...
}
```

New Pattern (Java 14+ Records)

```
public record Person(String name, int age) {
    // Auto-generated: constructor, getters,
    // equals, hashCode, toString
}
```

Pattern Matching (Java 21)

```
if (obj instanceof String s && s.length() > 5) {
    System.out.println(s.toUpperCase());
}
```

Sealed Classes (Java 17)

```
public sealed interface Shape
    permits Circle, Rectangle {
    double area();
}

public record Circle(double radius) implements Shape {
    public double area() {
        return Math.PI * radius * radius;
    }
}
```

Top 10 Java Interview Questions & Answers

1. What is the difference between JDK, JRE, and JVM?

2. Explain String immutability and why it's important?

3. What are the different ways to create threads in Java?

4. Difference between ArrayList and Vector?

5. What is the diamond problem in Java?

6. Explain Spring Bean lifecycle?

7. What is the volatile keyword in Java?

8. Difference between @Autowired and @Resource?

9. What are Java 8 features most asked in interviews?

10. How does HashMap work internally?

System Design: Microservices Architecture



Quick Interview Reference

Time Complexities

- ▶ ArrayList get(): O(1)
- ▶ ArrayList add(): O(1) amortized
- ▶ LinkedList get(): O(n)
- ▶ LinkedList add(): O(1)
- ▶ HashMap get/put: O(1)
- ▶ TreeMap get/put: O(log n)

Memory Sizes

- ▶ boolean: 1 bit
- ▶ byte: 8 bits
- ▶ char: 16 bits
- ▶ int: 32 bits
- ▶ long: 64 bits
- ▶ Object header: 12-16 bytes

Common Exceptions

- ▶ NullPointerException
- ▶ ArrayIndexOutOfBoundsException
- ▶ ClassCastException
- ▶ IllegalArgumentException
- ▶ IllegalStateException
- ▶ ConcurrentModification

Design Patterns

- ▶ Singleton
- ▶ Factory
- ▶ Builder
- ▶ Observer
- ▶ Strategy
- ▶ Decorator

Java Interview Success Formula 2026

Core Concepts (30%) + Problem Solving (40%) + System Design (20%) + Soft Skills (10%)

Practice with: LeetCode, HackerRank, Pramp, Grokking the Coding Interview

Updated: January 2026 | Java 21 LTS Focus

Remember: Understanding > Memorization | Diagrams > Text | Practice > Theory