Period vs Duration in Java (Full Concept Guide)

1️⃣ Introduction

In Java 8, a new Date & Time API was introduced under the package:

👉 java.time

This modern API replaced the old, confusing date classes and introduced clear separation between:

- Date-based calculations
- Time-based calculations

Two important classes for calculating differences are:

Period
Duration

Many students get confused between them.
This document gives complete clarity.

2️⃣ What is Period?

Period represents:

A date-based amount of time in Years, Months, and Days.

It works only with:

- LocalDate
- Year
- YearMonth

📌 Full Class Name

java.time.Period

📌 When to Use Period

Use Period when your logic is based on:

- Age calculation
- Experience in years
- Subscription duration in months
- Calendar-based differences

- Adding months or years to a date

📌 Example – Age Calculation

```
import java.time.LocalDate;
import java.time.Period;

public class AgeExample {
   public static void main(String[] args) {

      LocalDate birth = LocalDate.of(2000, 5, 10);
      LocalDate today = LocalDate.now();

      Period age = Period.between(birth, today);

      System.out.println("Years: " + age.getYears());
      System.out.println("Months: " + age.getMonths());
      System.out.println("Days: " + age.getDays());
   }
}
```

📌 Important Characteristics of Period

✓ Measures: Years, Months, Days
✓ Calendar-based
✓ Considers leap years
✓ Considers month length differences
✓ Immutable
✓ Thread-safe

3️⃣ What is Duration?

Duration represents:

A time-based amount of time in seconds and nanoseconds.

It works with:

- LocalTime
- LocalDateTime
- Instant
- ZonedDateTime

📌 Full Class Name

java.time.Duration

📌 When to Use Duration

Use Duration when your logic involves:

- Execution time measurement
- Stopwatch
- API response time
- Countdown timer
- Time differences in hours/minutes
- Logging timestamps

📌 Example – Execution Time

```
import java.time.Instant;
import java.time.Duration;

public class ExecutionTimeExample {
   public static void main(String[] args) {

      Instant start = Instant.now();

      for (int i = 0; i < 1_000_000; i++) {
         Math.sqrt(i);
      }

      Instant end = Instant.now();

      Duration timeTaken = Duration.between(start, end);

      System.out.println("Milliseconds: " + timeTaken.toMillis());
   }
}
```

📌 Important Characteristics of Duration

✓ Measures: Hours, Minutes, Seconds
✓ Based on exact seconds
✓ Does NOT understand months

✓ Immutable
✓ Thread-safe

4️⃣ Core Concept Difference

Feature	Period	Duration
Based On	Calendar	Clock
Measures	Years, Months, Days	Hours, Minutes, Seconds
Use Case	Age calculation	Execution time
Handles Months?	Yes	No
Handles Hours?	No	Yes

5️⃣ Why Duration Cannot Handle Months

Months are not fixed:

- February → 28 or 29 days
- April → 30 days
- January → 31 days

Because month length varies, Duration cannot measure months.

Duration measures only:

Total seconds
Total nanoseconds

That is exact time measurement.

6️⃣ Calendar-Based vs Clock-Based Logic

🗓 Period = Calendar Logic

Example:

Jan 31 → Feb 28
It understands this as one month difference.

It adjusts according to calendar rules.

⏱ Duration = Exact Time Logic

If difference is:

48 hours

It calculates exactly:

172800 seconds

No calendar logic involved.

7️⃣ Real-World Usage Scenarios

Use Period When:

✓ Calculating age
✓ Finding work experience
✓ Subscription expiry (in months)
✓ Adding 2 years to a date
✓ Academic year calculation

Use Duration When:

✓ Measuring API response time
✓ Timer application
✓ Countdown logic
✓ Server performance monitoring
✓ Execution time calculation

8️⃣ Common Mistake

Wrong usage:

Duration.between(LocalDate1, LocalDate2);

This throws exception because:

Duration works with time-based classes.

Correct:

Period.between(date1, date2);

9️⃣ Internal Storage Concept

Period Stores:

int years
int months
int days

Duration Stores:

long seconds
int nanos

That's why Duration is more precise for time measurement.

🔟 Quick Memory Trick

Period → Date logic
Duration → Time logic

OR

Period → Calendar
Duration → Clock

1️⃣1️⃣ Interview-Level Summary

- Both are part of java.time (Java 8 API)
- Both are immutable
- Both are thread-safe
- Period is date-based
- Duration is time-based
- Period handles months
- Duration handles seconds

🎯 Final Conclusion

Java 8 clearly separated:

Date difference → Period
Time difference → Duration

This separation solved confusion from older Date APIs and made logic more precise and safe.