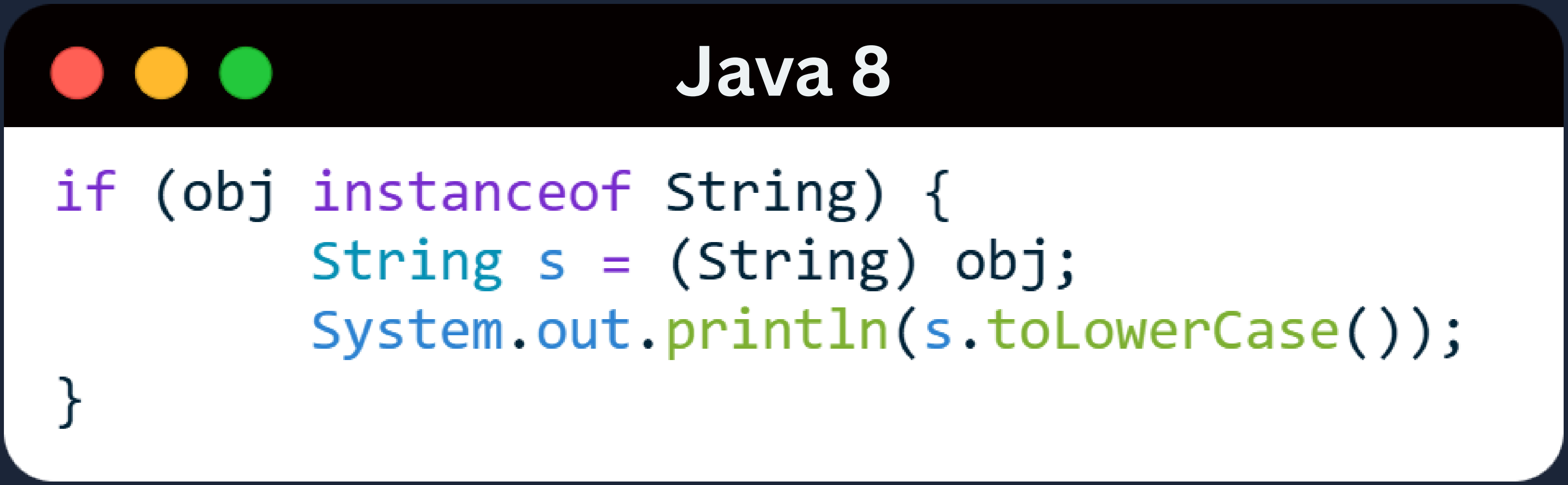
What I learned migrating from

Java 8 → Java 21

- Pattern Matching (instanceof)
- Text Blocks (Multiline Strings)
- Records (Data Classes)
- Switch Expressions
- Virtual Threads
- Sealed Classes

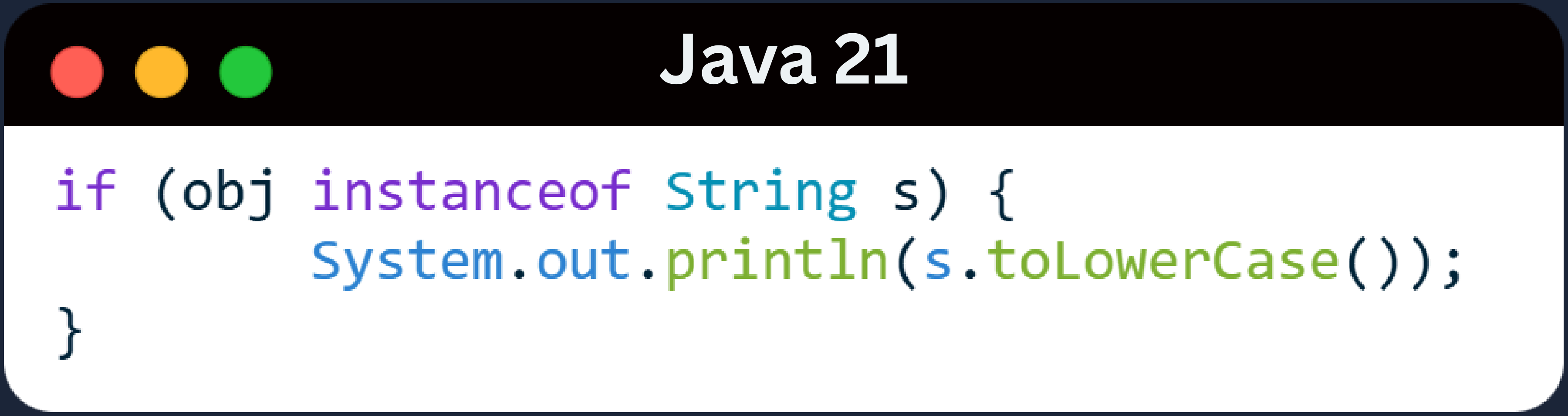
@rakesh-yadav-dasari

Pattern Matching (instanceof)



Java 8

```java
if (obj instanceof String) {
        String s = (String) obj;
        System.out.println(s.toLowerCase());
}
```



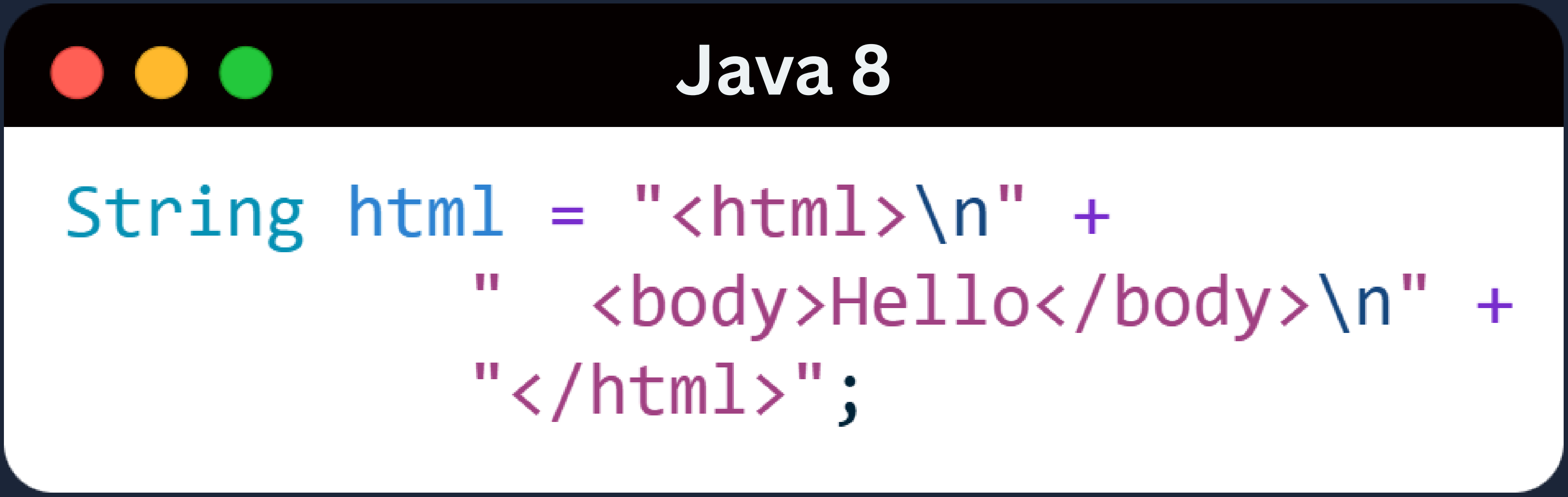
Java 21

```java
if (obj instanceof String s) {
        System.out.println(s.toLowerCase());
}
```

☑ Cleaner type checks and less boilerplate.

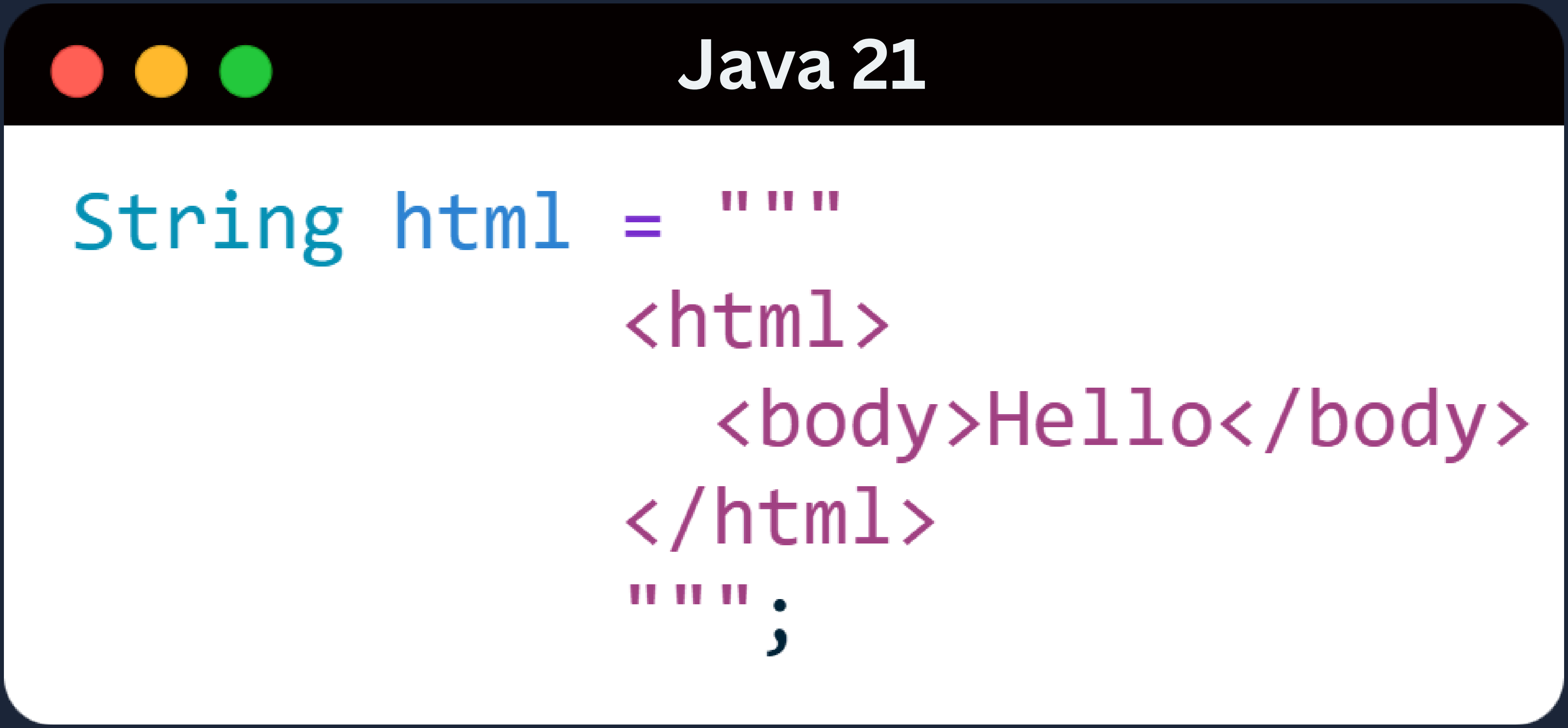
@rakesh-yadav-dasari

Text Blocks (Multiline Strings)



Java 8

```
String html = "<html>\n" +
          "  <body>Hello</body>\n" +
          "</html>";
```



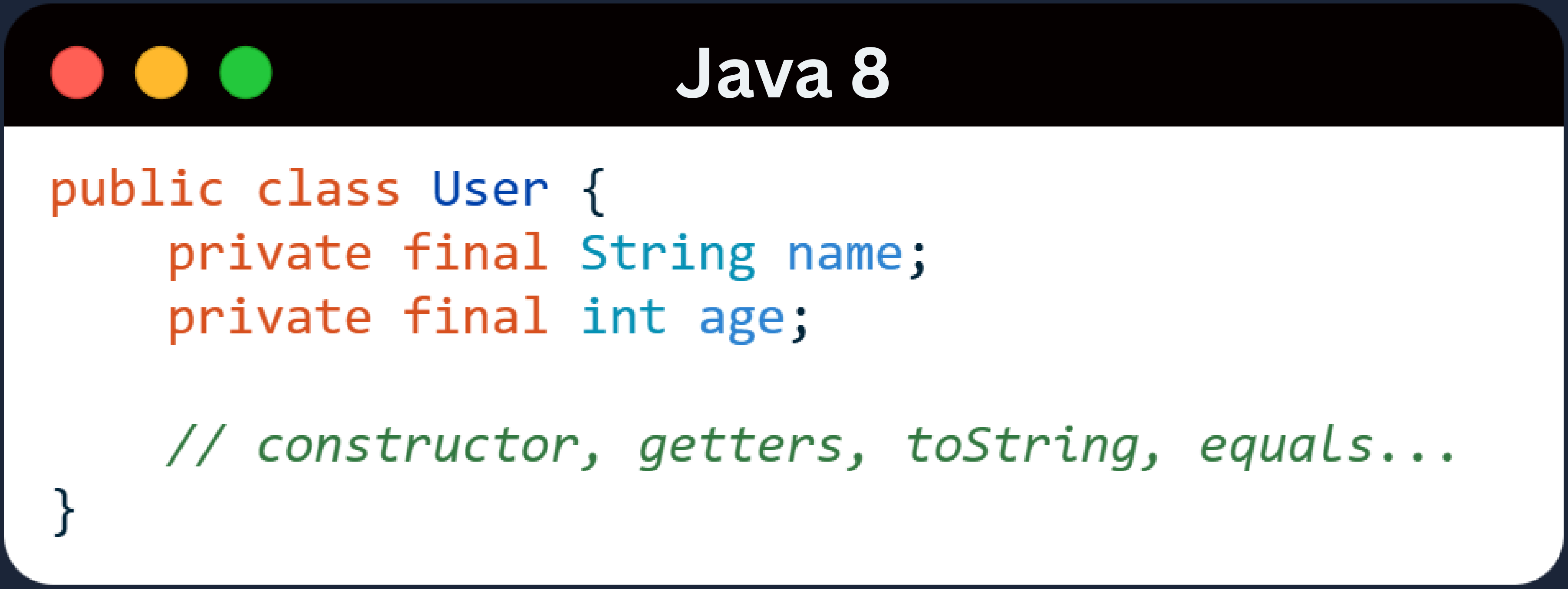
Java 21

```
String html = """
            <html>
              <body>Hello</body>
            </html>
            """;
```

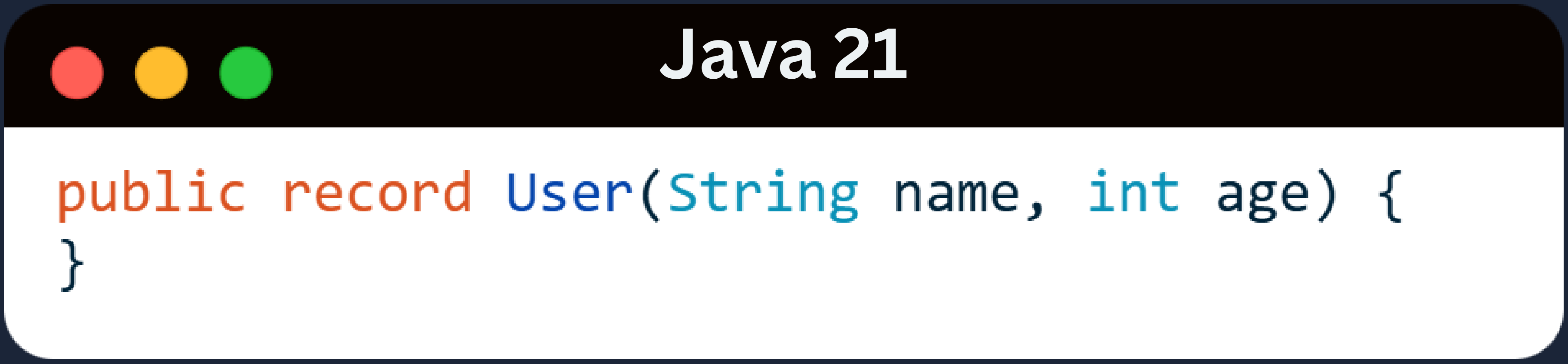
☑ Easier for SQL, HTML, JSON, etc.

@rakesh-yadav-dasari

Records (Data Classes)



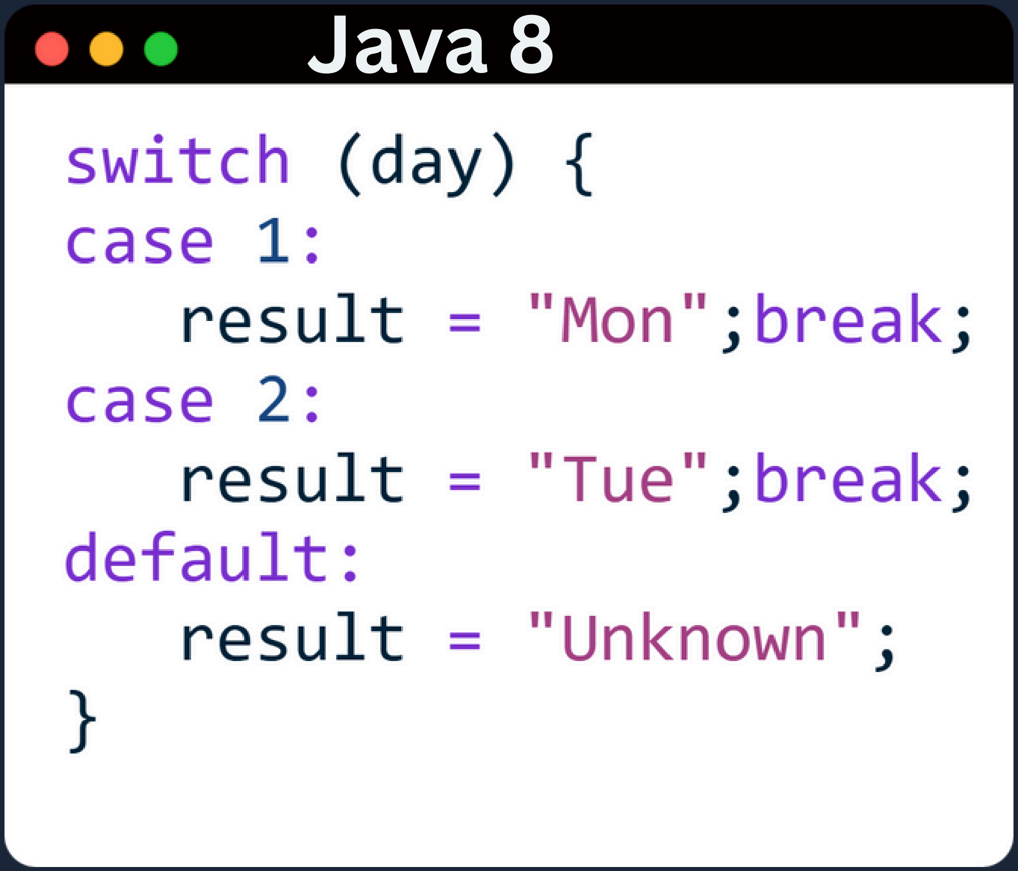
```java
public class User {
    private final String name;
    private final int age;

    // constructor, getters, toString, equals...
}
```



```java
public record User(String name, int age) {
}
```

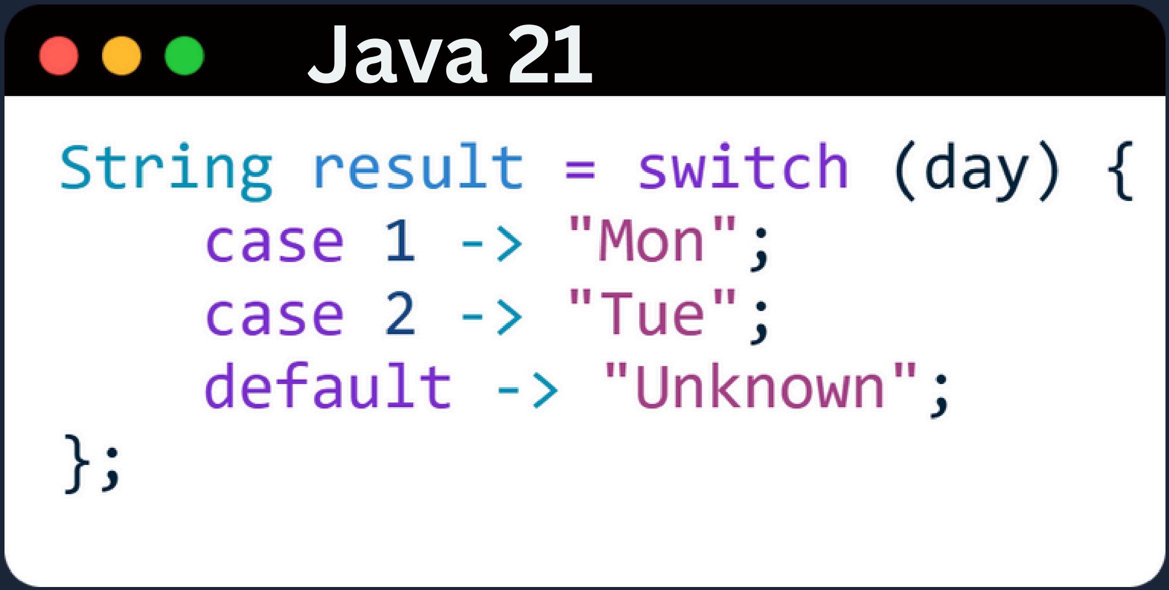
☑ Less boilerplate for data models.

Switch Expressions



Java 8

```
switch (day) {
case 1:
    result = "Mon";break;
case 2:
    result = "Tue";break;
default:
    result = "Unknown";
}
```



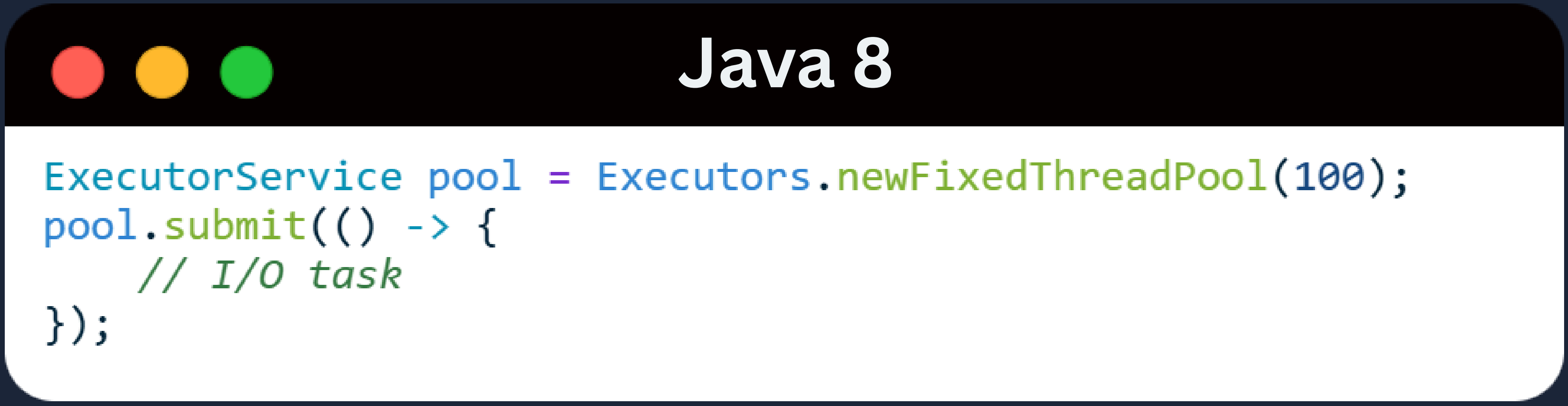
Java 21

```
String result = switch (day) {
    case 1 -> "Mon";
    case 2 -> "Tue";
    default -> "Unknown";
};
```

☑ More concise, less error-prone.

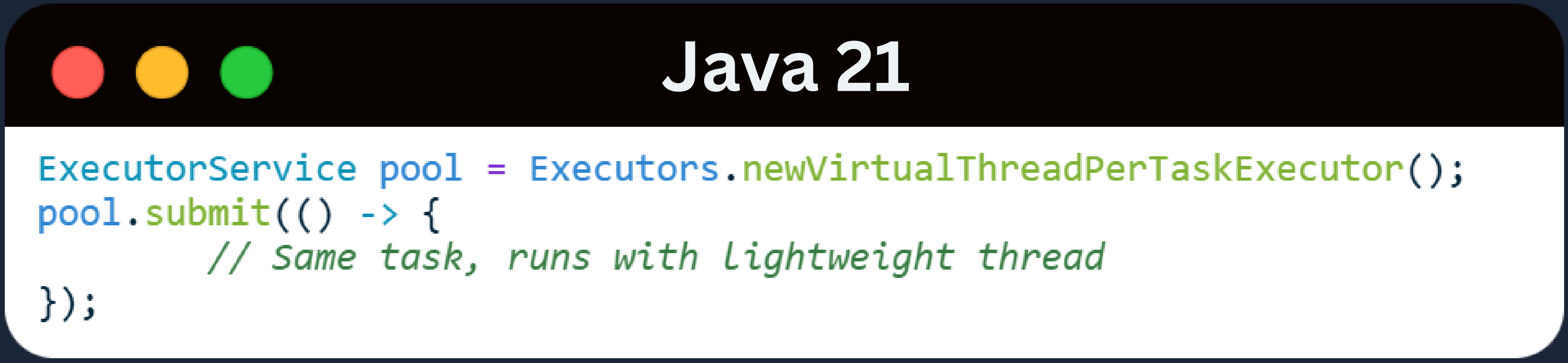
@rakesh-yadav-dasari

Virtual Threads



Java 8

```java
ExecutorService pool = Executors.newFixedThreadPool(100);
pool.submit(() -> {
    // I/O task
});
```

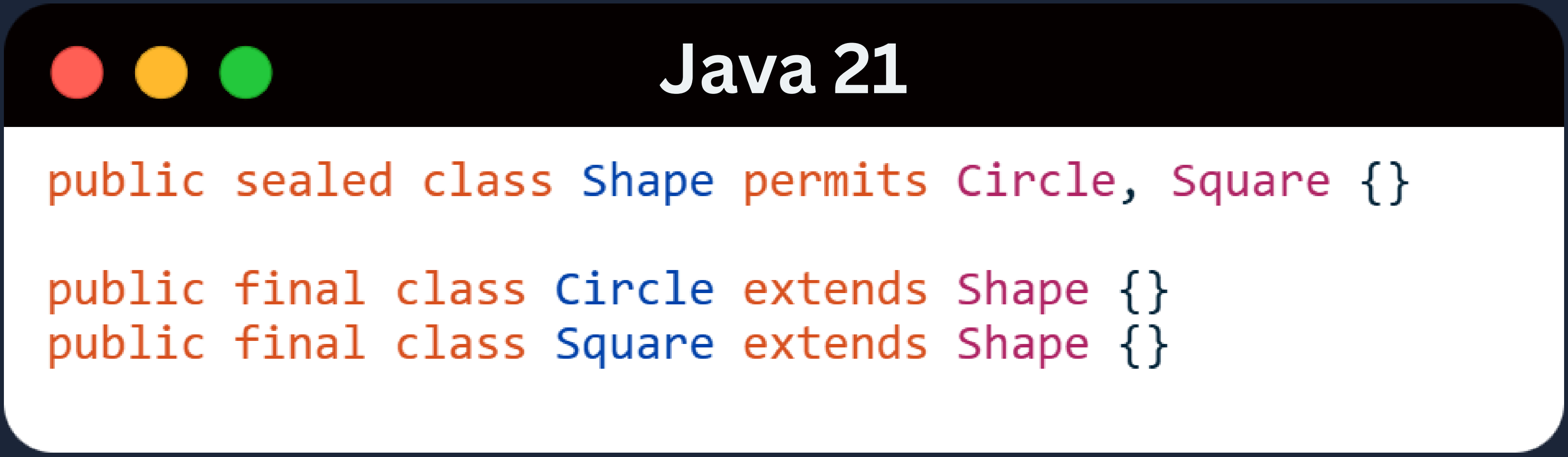


Java 21

```java
ExecutorService pool = Executors.newVirtualThreadPerTaskExecutor();
pool.submit(() -> {
        // Same task, runs with lightweight thread
});
```

- ☑ Run 1000s of concurrent tasks efficiently
- ☑ No context-switching overhead like OS threads
- ☑ Ideal for high-throughput servers & reactive apps

Sealed Classes



```java
public sealed class Shape permits Circle, Square {}

public final class Circle extends Shape {}
public final class Square extends Shape {}
```

☑ **Safer APIs with controlled inheritance.**

Thank you!

Found this helpful?

Repost to help others

Follow me for more Java & Full-Stack tips

Drop a “Comment” if you want a deep-dive PDF!

@rakesh-yadav-dasari