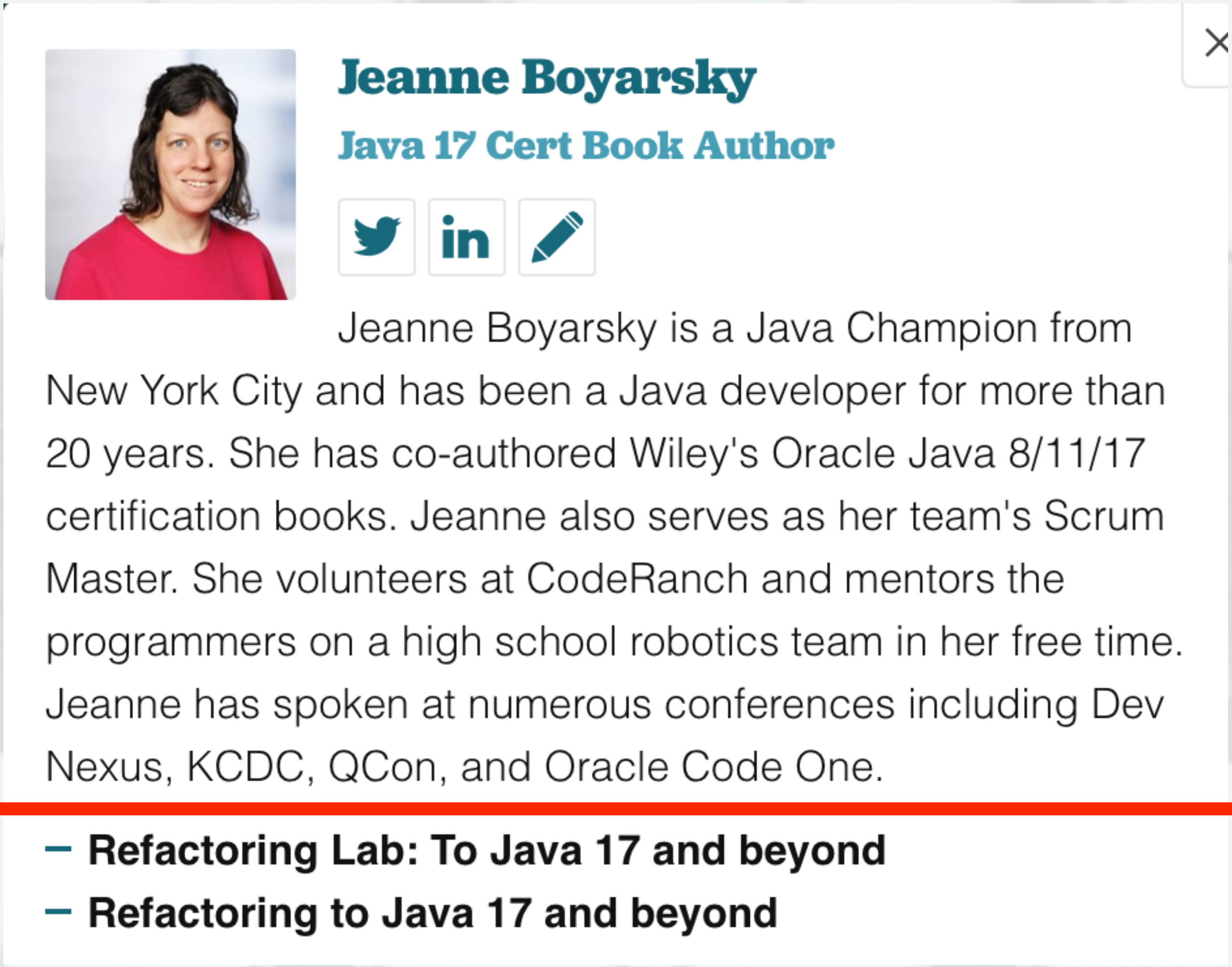
# Refactoring to Java 17 and beyond

Jeanne Boyarsky
August 8, 2022
KCDC

speakerdeck.com/boyarsky



**Jeanne Boyarsky**

**Java 17 Cert Book Author**

Jeanne Boyarsky is a Java Champion from New York City and has been a Java developer for more than 20 years. She has co-authored Wiley's Oracle Java 8/11/17 certification books. Jeanne also serves as her team's Scrum Master. She volunteers at CodeRanch and mentors the programmers on a high school robotics team in her free time. Jeanne has spoken at numerous conferences including Dev Nexus, KCDC, QCon, and Oracle Code One.

- **Refactoring Lab: To Java 17 and beyond**
- **Refactoring to Java 17 and beyond**

@jeanneboyarsky

# Pause for a Commercial

**tanium Sponsors**

LaunchDarkly → Headspring Part of Accenture Progress okta

**Speaker Dinner Sponsor**

ARCHITECT

**Platinum Sponsors**

circleci ionic amazon FusionAuth Auth. Built for Devs, by Devs. outsystems

solace. NAIC NIPR JFrog JOHN DEERE ORION snyk VIAGI TECHNOLOG

DEEPGRAM MuleSoft Red Hat twilio sonatype epiq Microsof

Gradle Enterprise Scout APM WellSky Veterans United Home Loans azul bitris

CONCENTRIX CATALYST

J.B. HUNT ChaiOne DataStax American Century Investments REPLY improving It's what we do.

**Gold Sponsors**

ORACLE Cerner New Relic. BUILDERTREND ascend LEARNING touchnet A Global Payments Company Xpirit EVENT STORE fnbo First National Bank of Omaha VMLY&R

Superblocks C.H. ROBINSON ALARM.COM SHAMROCK TRADING CORPORATION Abbott QUEST ANALYTICS REQUEST METRICS. SMART WAREHOUSING rxss

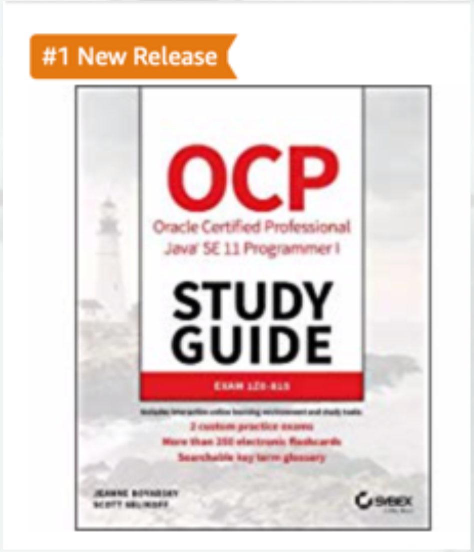
@jeanneboyarsky

# Another Commercial

Amazon Best Sellers

Our most popular products based on sales. Updated hourly.

Best Sellers in Oracle Certification



Java certs: 8/11/17

Book giveaway at end!

@jeanneboyarsky

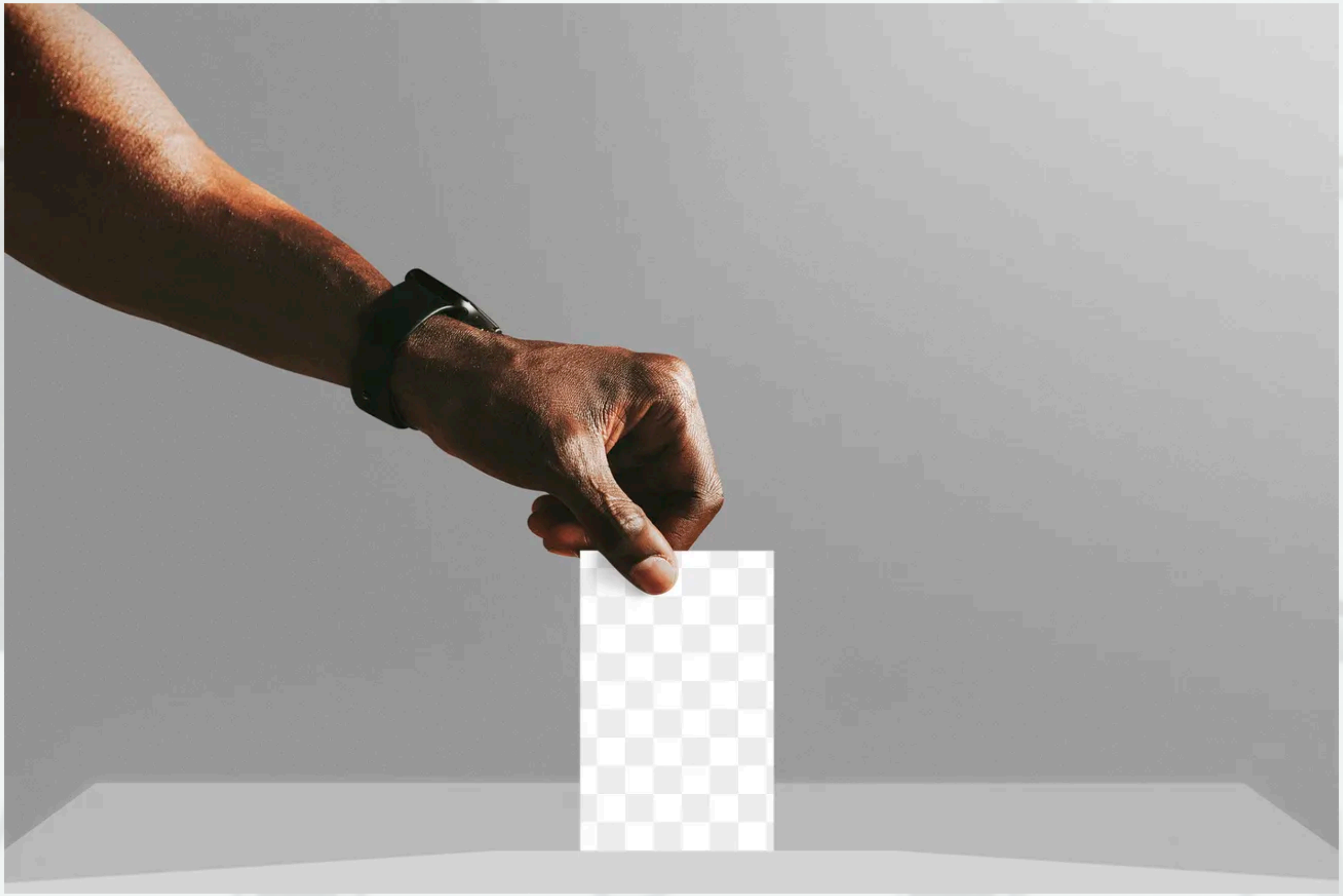
## At end of session

https://speakerdeck.com/boyarsky

## Disclaimer

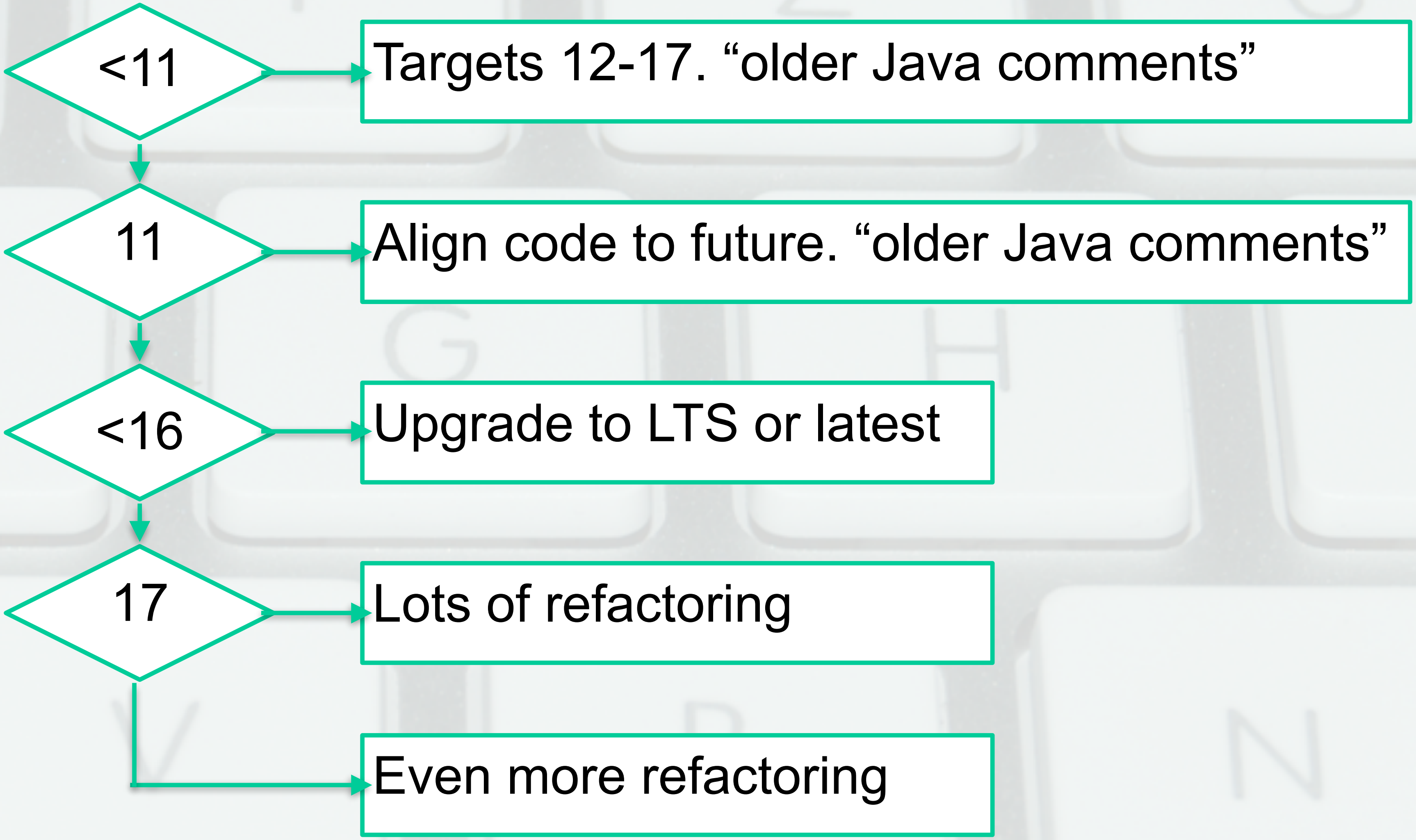
- A bit of the material is from my books.

# Version of Java?



@jeanneboyarsky

# Version of Java?


<11
Targets 12-17. “older Java comments”
11
Align code to future. “older Java comments”
<16
Upgrade to LTS or latest
17
Lots of refactoring
Even more refactoring

## For each Topic

- Example
- About the feature
- Opportunities
- IDE Support
- What to do if on older Java

- What will be explored in more detail in the lab version Wednesday….

## Refactoring

- We are writing legacy code now!
- Refactor for future compatibility

## Text blocks and Strings

## Example: REST API Params

```
public String getJson(String search) {
    String json = "{" +
            "   \"query\": \"%s\"" +
            "   \"start\": \"1\"," +
            "   \"end\": \"10\"" +
            "}";
    return String.format(json, search);
}
```

This is hard to read

# Take Two

```
public String getJson(String search) {
    Path path = Path.of(
       “src/main/resources/query.json");
    String json = null;
    try {
        json = Files.readString(path);
    } catch (IOException e) {
        throw new UncheckedIOException(e);
    }
    return String.format(json, search);
}
```

Now the String is far away

## Text Block

15

```
public String getJson(String search) {
    String json = """
            {
            "query": "%s"
            "start": "1"
            "end": "10"
            }""";
    return String.format(json, search);
}
```

It's a string literal!

Adds line breaks, but still works

@jeanneboyarsky

# Text Block Syntax

15

start block

```
String textBlock = """
                   kcdc,Kansas City,"session,workshop"
                   meetup,Various,lecture
                   """;
```

incidental whitespace

end block

@jeanneboyarsky

# Essential Whitespace

15

```
String textBlock = """
    <session>
        <speaker>
            Jeanne Boyarsky
        </speaker>
    </session>
    """;
```

incidental whitespace

essential whitespace

@jeanneboyarsky

## Ending lines

15

new escape character keeps trailing whitespace

```
String textBlock = """
        <session>
            <speaker>
                Jeanne Boyarsky    \s
            </speaker>
            <title>
                Becoming one of the first Java 17 \
                certified programmers \
                (and learning new features)
            </title>
        </session>
        """;
```

tab

continue on next line without a line break

# New lines

15

```
String textBlock = """
        <session>\n
            <speaker>
                Jeanne\nBoyarsky
            </speaker>
        </session>""";
```

Two new lines
(explicit and implicit)

One new line (explicit)

no line break at end

@jeanneboyarsky

# Escaping Three Quotes

15

```
String textBlock = """
        better \"""
        but can do \"\"\"
        """;
```

@jeanneboyarsky

## Opportunities

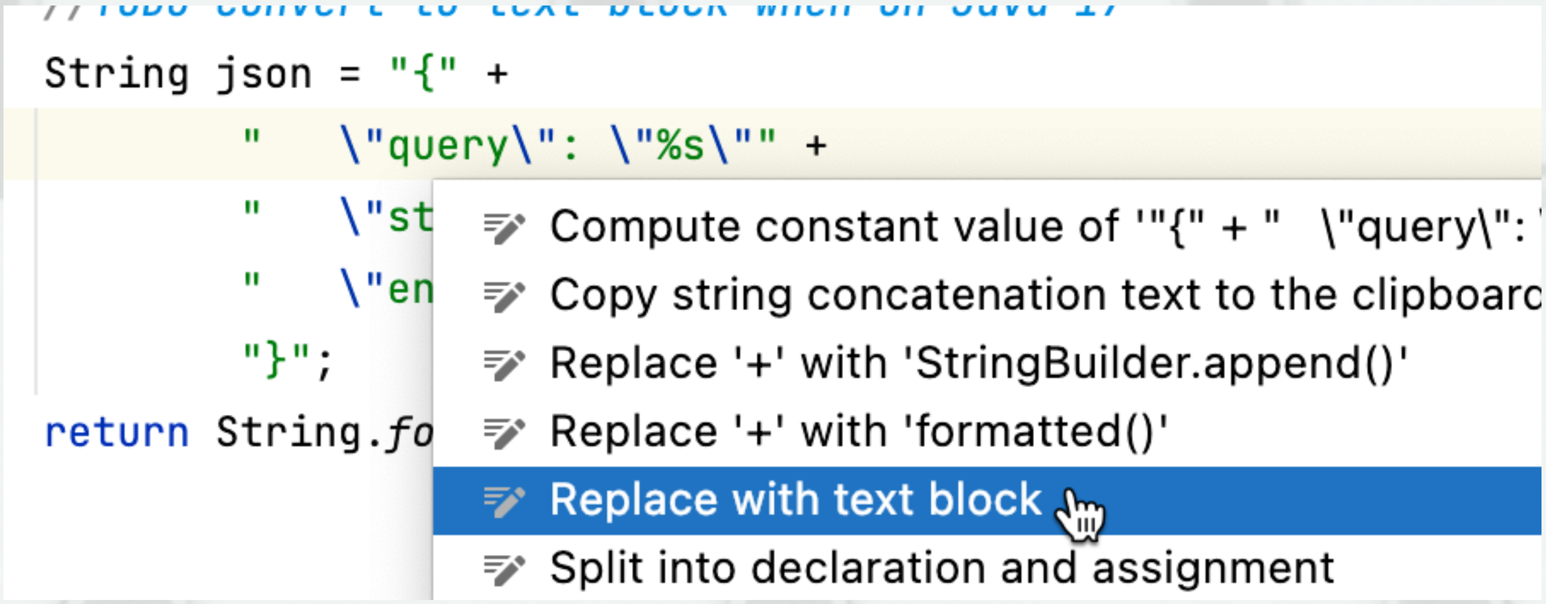
15

- Externalized data
- Expected values in JUnit

- Formats - CSV, GraphQL, JSON, SQL, Text, XML, YAML, etc

- Others?

# IDE Support



```
// TODO convert to text block when on Java 17
String json = "{" +
        "   \"query\": \"%s\"" +
        "   \"st
        "   \"en
        "}";
return String.fo
```

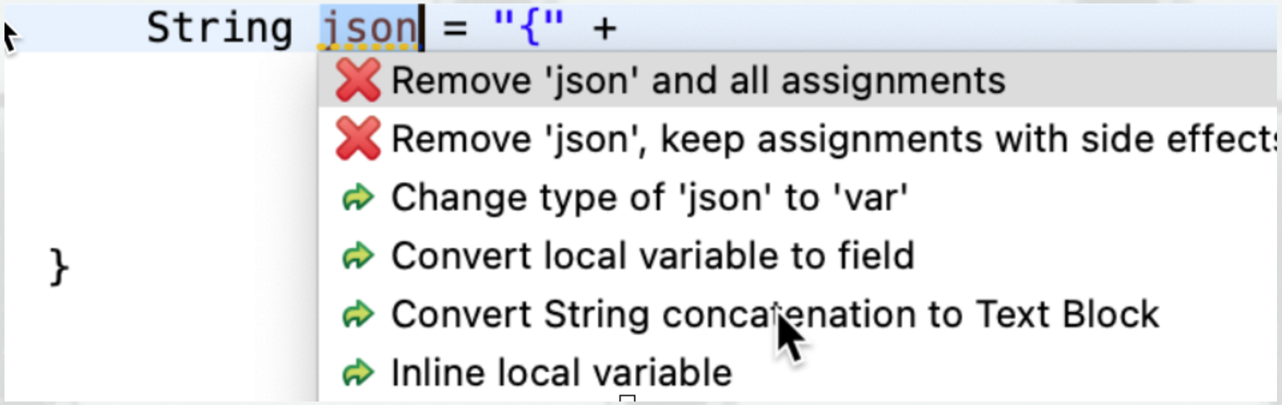
Compute constant value of '"{" + "   \"query\": '
Copy string concatenation text to the clipboarc
Replace '+' with 'StringBuilder.append()'
Replace '+' with 'formatted()'
Replace with text block
Split into declaration and assignment

```
String json = """
           {    "query": "%s"    "start": "1",    "end": "10"}""";
```

Literal refactoring - no \n

@jeanneboyarsky

# IDE Support



```
String json = "{" +

}
```

- Remove 'json' and all assignments
- Remove 'json', keep assignments with side effects
- Change type of 'json' to 'var'
- Convert local variable to field
- Convert String concatenation to Text Block
- Inline local variable

```
String json = """
        {\
            "query": "%s"\
            "start": "1",\
            "end": "10"\
        }""";
```

Preserve lines but still no \n

# On older Java?

```
public String getJson(String search) {
    //TODO convert to text block when on Java 17
    String json = "{" +
            "    \"query\": \"%s\"" +
            "    \"start\": \"1\"," +
            "    \"end\": \"10\"" +
            "}";
    return String.format(json, search);
}
```

Hard to read but positions for future

@jeanneboyarsky

## Wed Lab Version

- Practice identifying valid/invalid text blocks
- Related String APIs
- Hands on practice

## Instanceof

@jeanneboyarsky

## Casting

```
if (num instanceof Integer) {
    Integer numAsInt = (Integer) num;
    System.out.println(numAsInt);
}
if (num instanceof Double) {
    Double numAsDouble = (Double) num;
    System.out.println(numAsDouble.intValue());
}
```

## Casting

16

```
if (num instanceof Integer numAsInt) {
    System.out.println(numAsInt);
}
if (num instanceof Double numAsDouble) {
    System.out.println(numAsDouble.intValue());
}
```

Pattern variable

@jeanneboyarsky

## Flow Scope

16

```
if (num instanceof Double d1
    && d1.intValue() % 2 == 0) {

    System.out.println(d1.intValue());
}

if (num instanceof Double d2
    || d2.intValue() % 2 == 0) {

    System.out.println(d2.intValue());
}
```

Compiles

Does not compile because d2 might not be double

@jeanneboyarsky

# Does this compile?

16

```
if (num instanceof Double n)
    System.out.println(n.intValue());

if (num instanceof Integer n)
    System.out.println(n);
```

Yes. Only in scope for if statement

# Does this compile?

```
if (num instanceof Double n)
    System.out.println(n.intValue());

System.out.println(n.intValue());
```

No. If statement is over

## Does this compile?

16

```
if (!(num instanceof Double n)) {
     return;
}
System.out.println(n.intValue());
```

Yes. Returns early so rest is like an else

@jeanneboyarsky

# Does this compile?

16

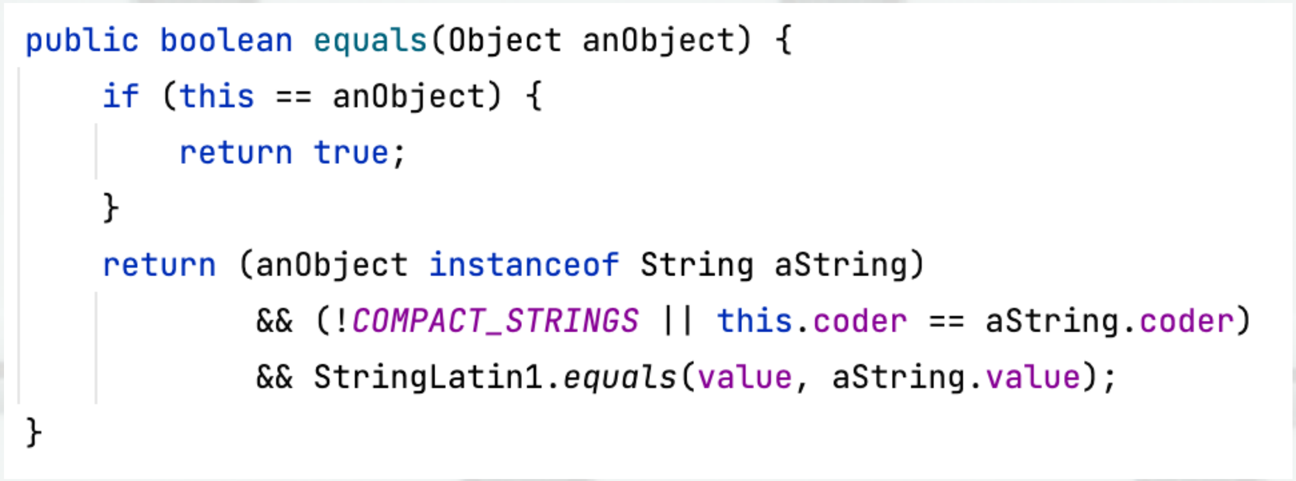
```
if (!(num instanceof Double n)) {
     return;
}
System.out.println(n.intValue());

if (num instanceof Double n)
     System.out.println(n.intValue());
```

No. n is still in scope

## Opportunities

16

- Library code
- Equals methods

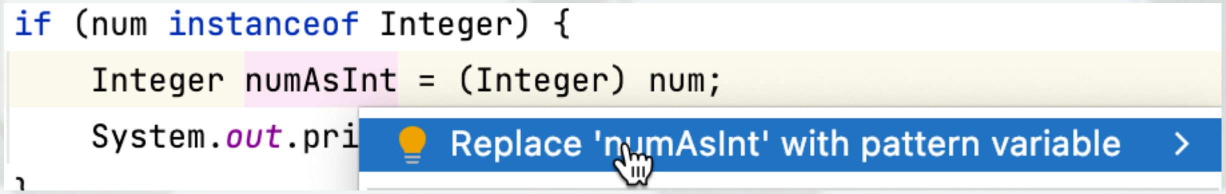


```
public boolean equals(Object anObject) {
    if (this == anObject) {
        return true;
    }
    return (anObject instanceof String aString)
            && (!COMPACT_STRINGS || this.coder == aString.coder)
            && StringLatin1.equals(value, aString.value);
}
```

- Others?

@jeanneboyarsky

# IDE Support



```
if (num instanceof Integer) {
    Integer numAsInt = (Integer) num;
    System.out.pri
```

```
if (num instanceof Integer numAsInt) {
    System.out.println(numAsInt);
}
```

## On older Java?

```
//TODO convert to pattern var when on Java 17

if (num instanceof Double) {
    Double numAsDouble = (Double) num;
    System.out.println(numAsDouble.intValue());
}
```

Positions for future

## Wed Lab Version

- Explore edge cases
- Sealed classes
- Hands on practice

## Switch expressions

## Originally

```
public String getLocation(String store) {
    String result = "";
    switch (store) {
        case "Hallmark":
            result = "KC";
            break;
        case "Crayola":
            result = "PA";
            break;
        default:
            result = "anywhere";
    }
    return result;
}
```

You remembered the breaks, right?

## Switch Expressions

14

```
public String getLocation(String store) {
    return switch (store) {
        case "Hallmark" -> "KC";
        case "Crayola" -> "PA";
        default -> "anywhere";
    };
}
```

Arrow labels

No break keyword

## Missing value

14

```
enum Position { TOP, BOTTOM };


Position pos = Position.TOP;

int stmt = switch(pos) {
    case TOP: yield 1;
};

int expr = switch(pos) {
    case BOTTOM -> 0;
};
```

Does not compile because assigning value

(poly expression)

# Pattern matching for switch

19
preview

```
public int toInt(Object obj) {
    return switch (obj) {
        case Integer i -> i;
        case Double d -> d.intValue();
        case String s -> Integer.parseInt(s);

        default -> throw new
            IllegalArgumentException("unknown type");
    };
}
```

Reminder: Syntax can change

# But wait, there's more

19
preview

```
static void printOddOrEven(Object obj) {
    switch (obj) {

        case Integer i when i % 2 == 1 ->
           System.out.println("odd");

        case Integer i when i % 2 == 0 ->
           System.out.println("even");

        default -> System.out.println("not an int");
    };
}
```

Reminder: Feature can still change

@jeanneboyarsky

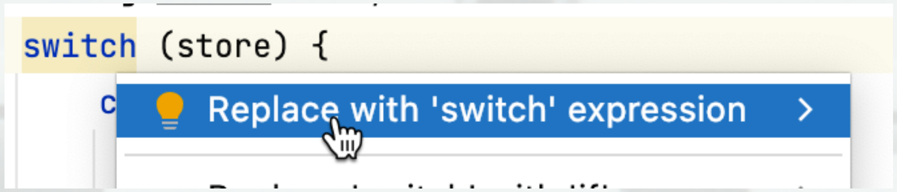
## Opportunities

17

- Many if/else chains!
- Switch statements with many breaks
- Sets the stage for advanced matching

- Others?

# IDE Support



```
switch (store) {
```

```
String result = switch (store) {
    case "Hallmark" -> "KC";
    case "Crayola" -> "PA";
    default -> "anywhere";
};
return result;
```

@jeanneboyarsky

## On older Java?

```
public String getLocation(String store) {
    //TODO convert to switch expression on Java 17
    String result = "";
    switch (store) {
        case "Hallmark":
            result = "KC";
            break;
        case "Crayola":
            result = "PA";
            break;
        default:
            result = "anywhere";
    }
    return result;
}
```

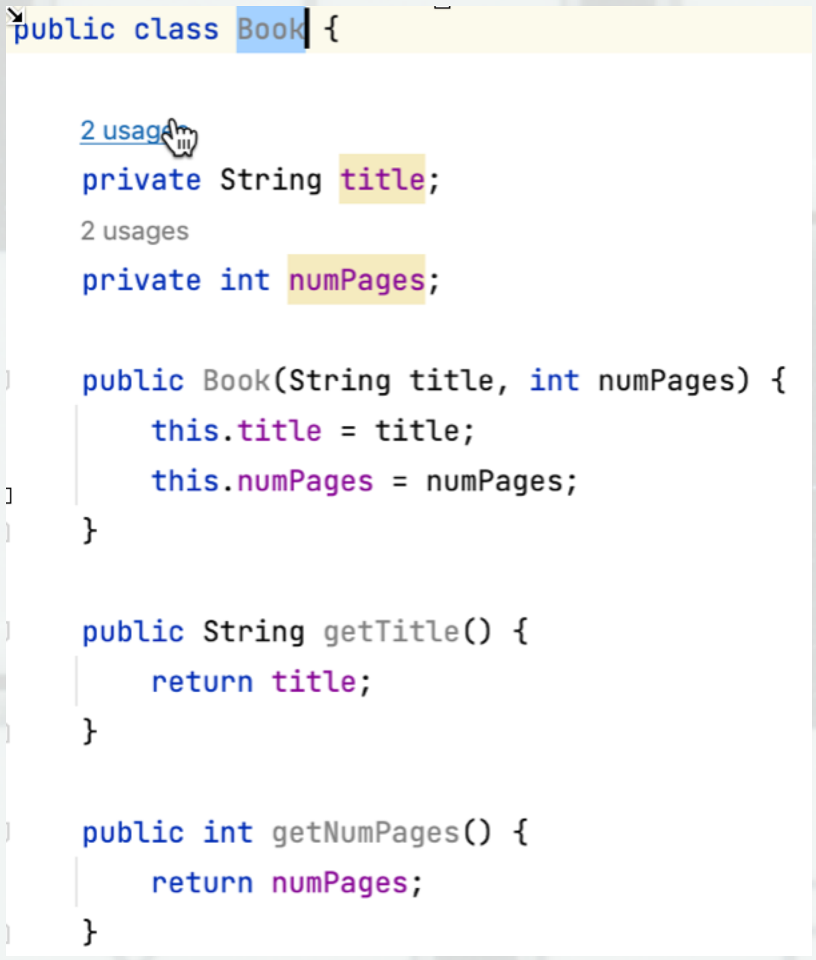
## Wed Lab Version

- Blocks and yield
- Switch with records
- More edge cases
- Hands on practice

## Records

@jeanneboyarsky

# Originally



```
public class Book {

    2 usages
    private String title;
    2 usages
    private int numPages;

    public Book(String title, int numPages) {
        this.title = title;
        this.numPages = numPages;
    }

    public String getTitle() {
        return title;
    }

    public int getNumPages() {
        return numPages;
    }
```

Ran out of room on screen!

@jeanneboyarsky

# Record

16

```
public record Book (String title, int numPages) {
}
```

New type

Automatically get
* final record
* private final instance variables
* public accessors
* constructor taking both fields
* equals
* hashCode

@jeanneboyarsky

## Using the Record

16

```
Book book = new Book("Breaking and entering", 289);

System.out.println(book.title());
System.out.println(book.toString());
```

← No “get”

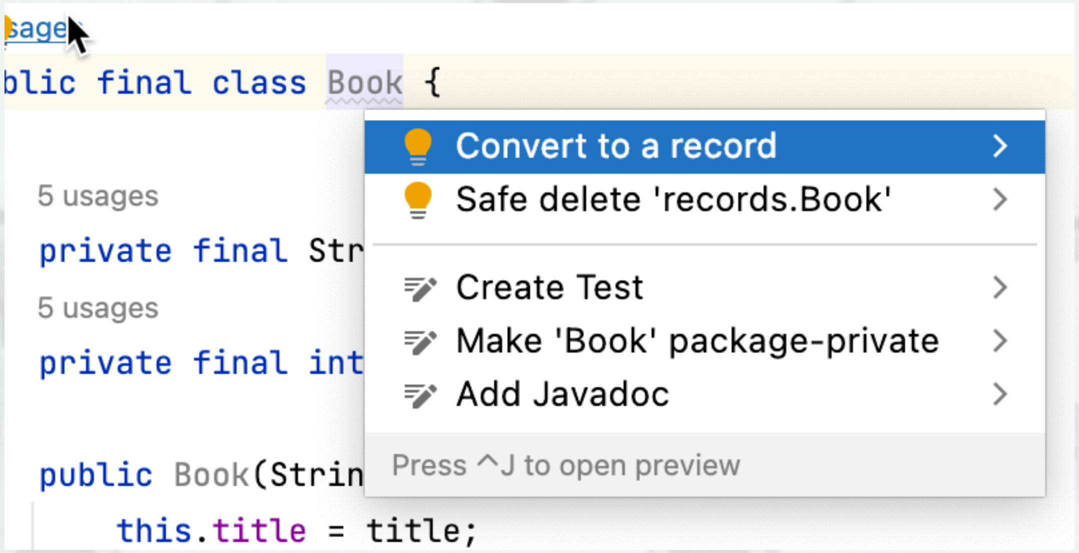
Outputs:
```
Breaking and entering
Book[title=Breaking and entering, numPages=289]
```

## Opportunities

17

- Immutable POJOs
- Don't have to write equals/ hashCode
- Vs reflection - EqualsBuilder
- Make code coverage tool happy
- Others?

# IDE Support



```
sages
blic final class Book {

  5 usages
  private final Str
  5 usages
  private final int

  public Book(Strin
      this.title = title;
```

```
public record Book(String title, int numPages) {
}
```

Had to make instance variables final. Also didn't remove my equals() even though generated by IntelliJ

@jeanneboyarsky

## On older Java?

```java
//TODO convert to record when on Java 17
public final class Book {
    private String title;
    private int numPages;

    public Book(String title, int numPages) {
        this.title = title;
        this.numPages = numPages;
    }
    public String title() {
        return title;
    }
    public int numPages() {
        return numPages;
    }
    // hash code, equals
```

Be sure to use al fields for equals/ hashCode

@jeanneboyarsky

## Wed Lab Version

- Compact constructors
- Custom methods
- More edge cases
- Hands on practice

## APIs

@jeanneboyarsky

## toList()

16

```java
public List<String> listLonger(
    Stream<String> stream) {

     return stream.collect(Collectors.toList());
}

public List<String> listShorter(
    Stream<String> stream) {

     return stream.toList();
}
```

# Teeing Collector

12

```
record Separations(String spaceSeparated,
   String commaSeparated) {}

var list = List.of("x", "y", "z");
Separations result = list.stream()
        .collect(Collectors.teeing(
              Collectors.joining(" "),
              Collectors.joining(","),
              (s, c) -> new Separations(s, c)));

System.out.println(result);
```

@jeanneboyarsky

## Formatting a String

12

```
String firstName = "Jeanne";
String lastName = "Boyarsky";
String str = String.format(
   "Hi %s %s!", firstName, lastName);
System.out.println(str);

System.out.println("Hi %s %s!".formatted(
    firstName, lastName));
```

Outputs:
Hi Jeanne Boyarsky!
Hi Jeanne Boyarsky!

@jeanneboyarsky

# Common Conversions

| Conversion | What it does |
|---|---|
| %s | Formattable as String |
| %d | Decimal integer (no dot) |
| %c | Char |
| %f | Float (decimal) |
| %n | New line |

Many more out of scope. Examples:

- %e - scientific notation
- %t - time
- %S - converts to all uppercase

# Conversion Examples

12

| Code | Output |
|---|---|
| `"%d%%".formatted(1.2)` | exception |
| `"%d%%".formatted(1)` | 1% |
| `"%s%%".formatted(1)` | 1% |
| `"%s%%".formatted(1.2)` | 1.2% |
| `“%f%%".formatted(1.2)` | 1.200000f |

# Formatting a Number

| Char | What it does |
|---|---|
| - | Left justified |
| + | Always include +/- |
| space | Leading space if positive |

| Char | What it does |
|---|---|
| 0 | Zero padded |
| , | Group numbers |
| ( | Negative # in parens |

## Flag Examples

| Code | Output |
|---|---|
| `"%,d".formatted(1234)` | 1,234 |
| `"%+d".formatted(1234)` | 1234 |
| `“% d".formatted(1234)` | 1234 |
| `“%,(d”.formatted(-1234)` | (1,234) |
| `“%,f”.formatted( 1.23456789)` | 1.234568 |

# Compact Number

12

```
NumberFormat defaultFormat =
NumberFormat.getCompactNumberInstance();
NumberFormat shortFormat = NumberFormat
    .getCompactNumberInstance(
         Locale.US, NumberFormat.Style.SHORT);
NumberFormat longFormat = NumberFormat
    .getCompactNumberInstance(
         Locale.US, NumberFormat.Style.LONG);

System.out.println(defaultFormat.format(1_000_000));
System.out.println(shortFormat.format(1_000_000));
System.out.println(longFormat.format(1_000_000));
```

```
1M
1M
1 million
```

@jeanneboyarsky

## New Files.mismatch()

12

```
Path kcdc = Path.of("files/kcdc.txt");
Path kc = Path.of("files/kc.txt");

System.out.println(Files.mismatch(kcdc, kc));
System.out.println(Files.mismatch(kcdc, kcdc));
```

11 (index of first character different)
-1 (same file contents regardless of whether exists)

@jeanneboyarsky

## Wed Lab Version

- Hands on practice

## Book Giveaway

@jeanneboyarsky