Zero Tolerance for Red code

Migrating to newer Java versions with IntelliJ IDEA

Mala Gupta (@eMalaGupta)
Java Champion & Developer Advocate

Red Code during migrations? Nah! I use IntelliJ IDEA

Release cadence – Java and IntelliJ IDEA

Support of newer Java features in IntelliJ IDEA

Supporting Preview language features.. is tricky :-)

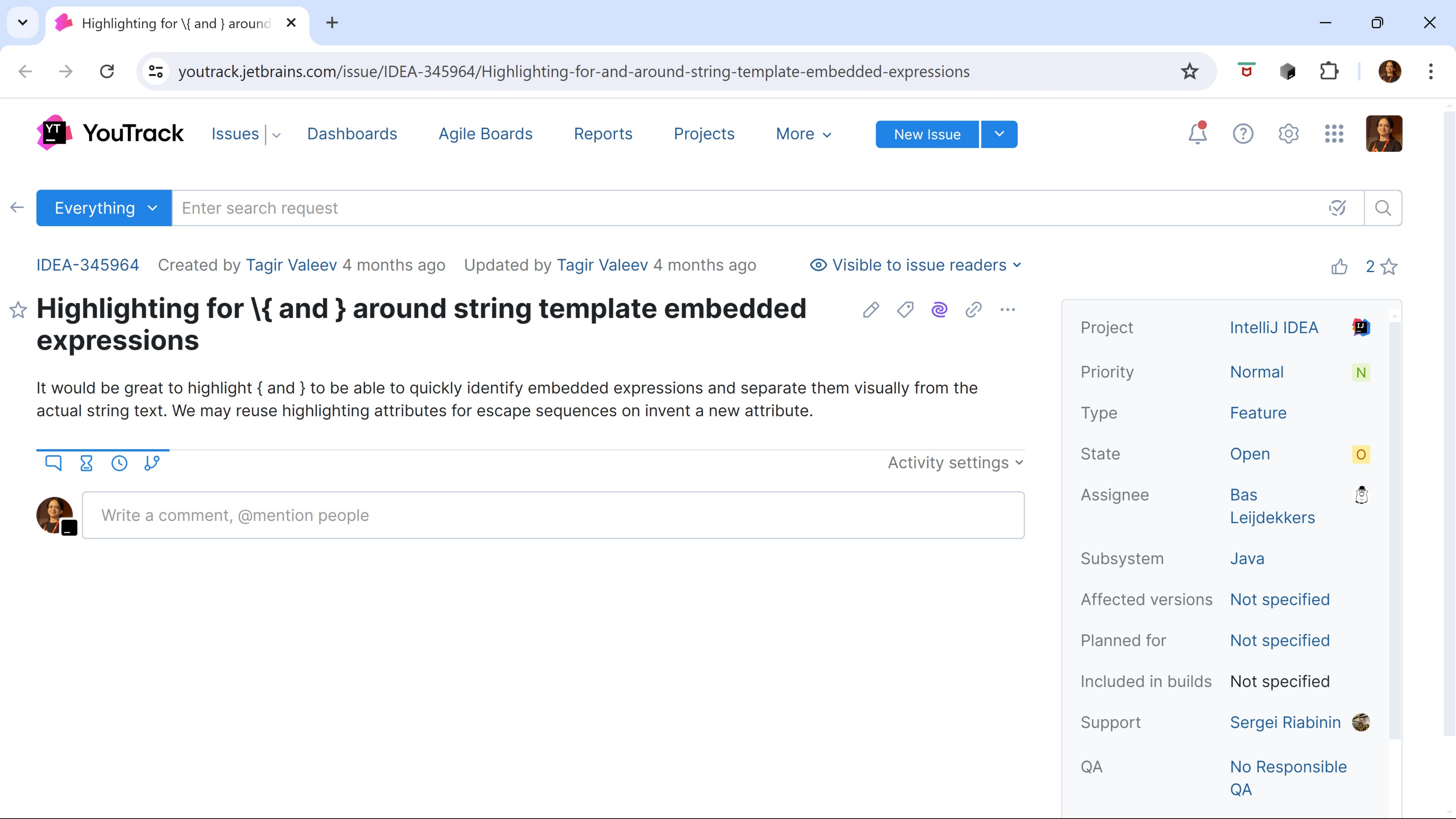
Moving Java forward together.
Oracle and JetBrains.

Workflow

Two-way communication

Mailing lists

Custom Builds/ EAP Builds



Highlighting for \{ and } around

youtrack.jetbrains.com/issue/IDEA-345964/Highlighting-for-and-around-string-template-embedded-expressions

YouTrack   Issues   Dashboards   Agile Boards   Reports   Projects   More   New Issue

Everything   Enter search request

IDEA-345964   Created by Tagir Valeev 4 months ago   Updated by Tagir Valeev 4 months ago   Visible to issue readers   2

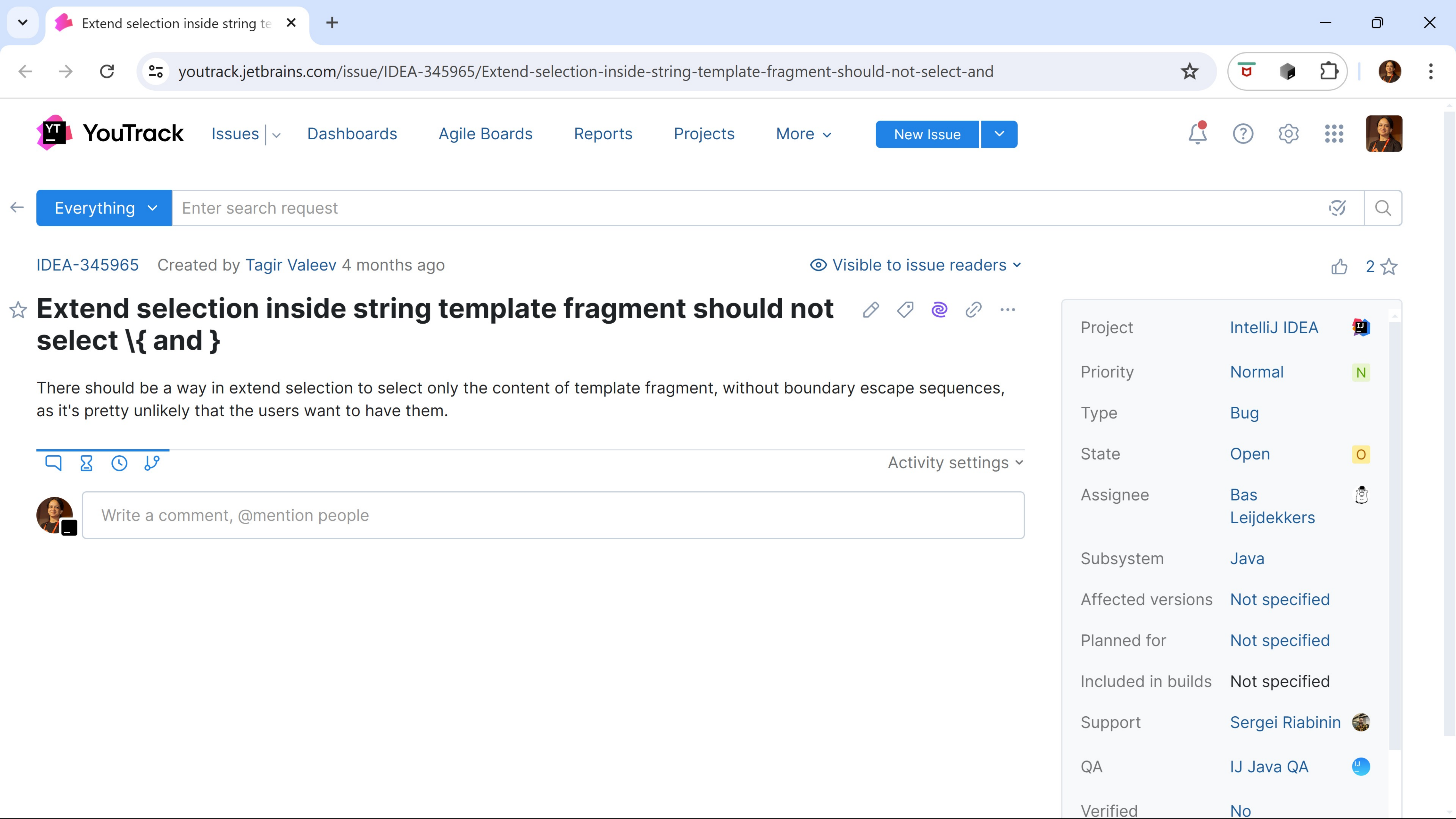
Highlighting for \{ and } around string template embedded expressions

It would be great to highlight { and } to be able to quickly identify embedded expressions and separate them visually from the actual string text. We may reuse highlighting attributes for escape sequences on invent a new attribute.

Activity settings

Write a comment, @mention people

Project	IntelliJ IDEA
Priority	Normal
Type	Feature
State	Open
Assignee	Bas Leijdekkers
Subsystem	Java
Affected versions	Not specified
Planned for	Not specified
Included in builds	Not specified
Support	Sergei Riabinin
QA	No Responsible QA



Extend selection inside string te

youtrack.jetbrains.com/issue/IDEA-345965/Extend-selection-inside-string-template-fragment-should-not-select-and

YouTrack Issues Dashboards Agile Boards Reports Projects More New Issue

Everything Enter search request

IDEA-345965 Created by Tagir Valeev 4 months ago

Visible to issue readers

2

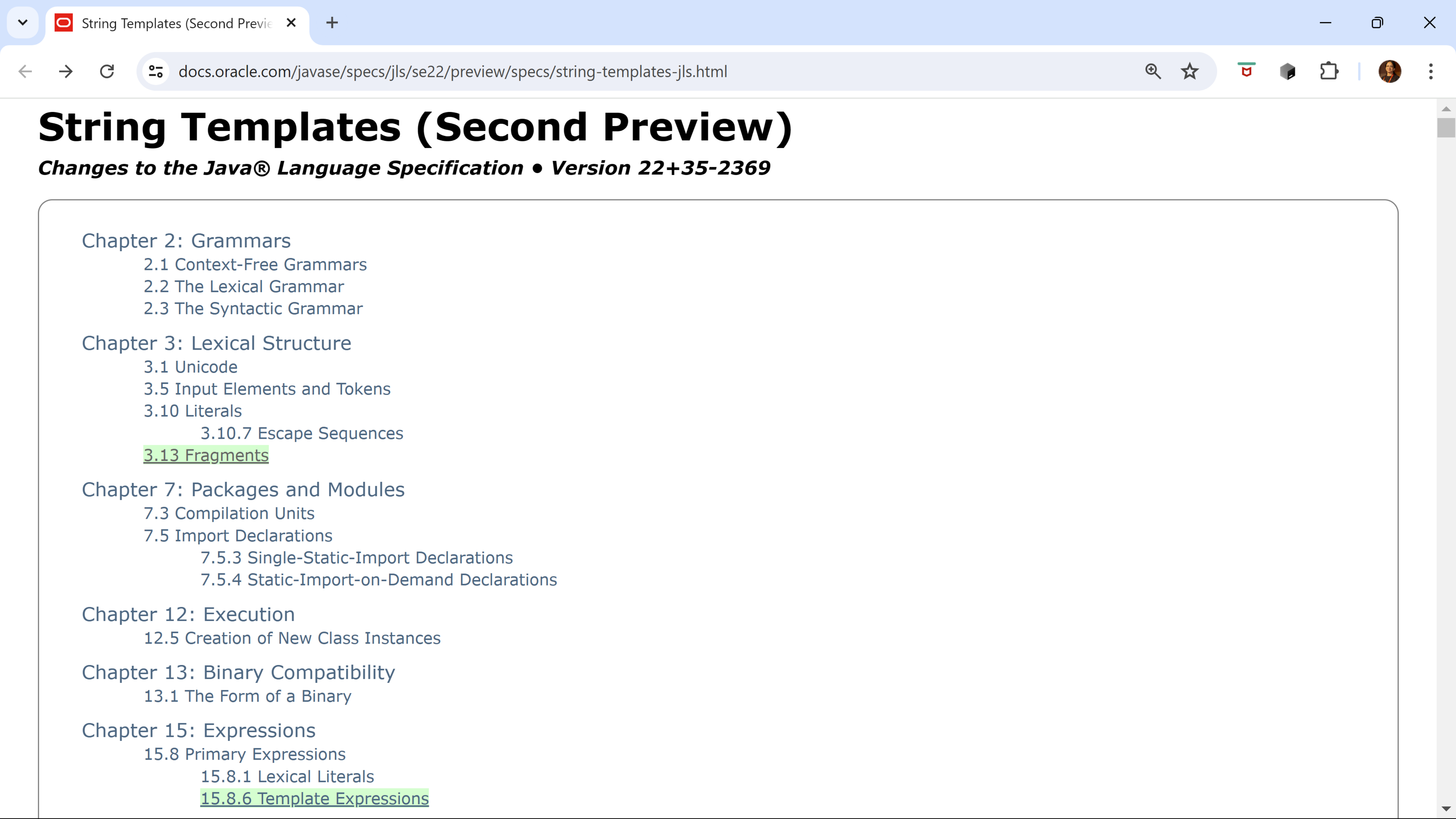
Extend selection inside string template fragment should not select \{ and }

There should be a way in extend selection to select only the content of template fragment, without boundary escape sequences, as it's pretty unlikely that the users want to have them.

Activity settings

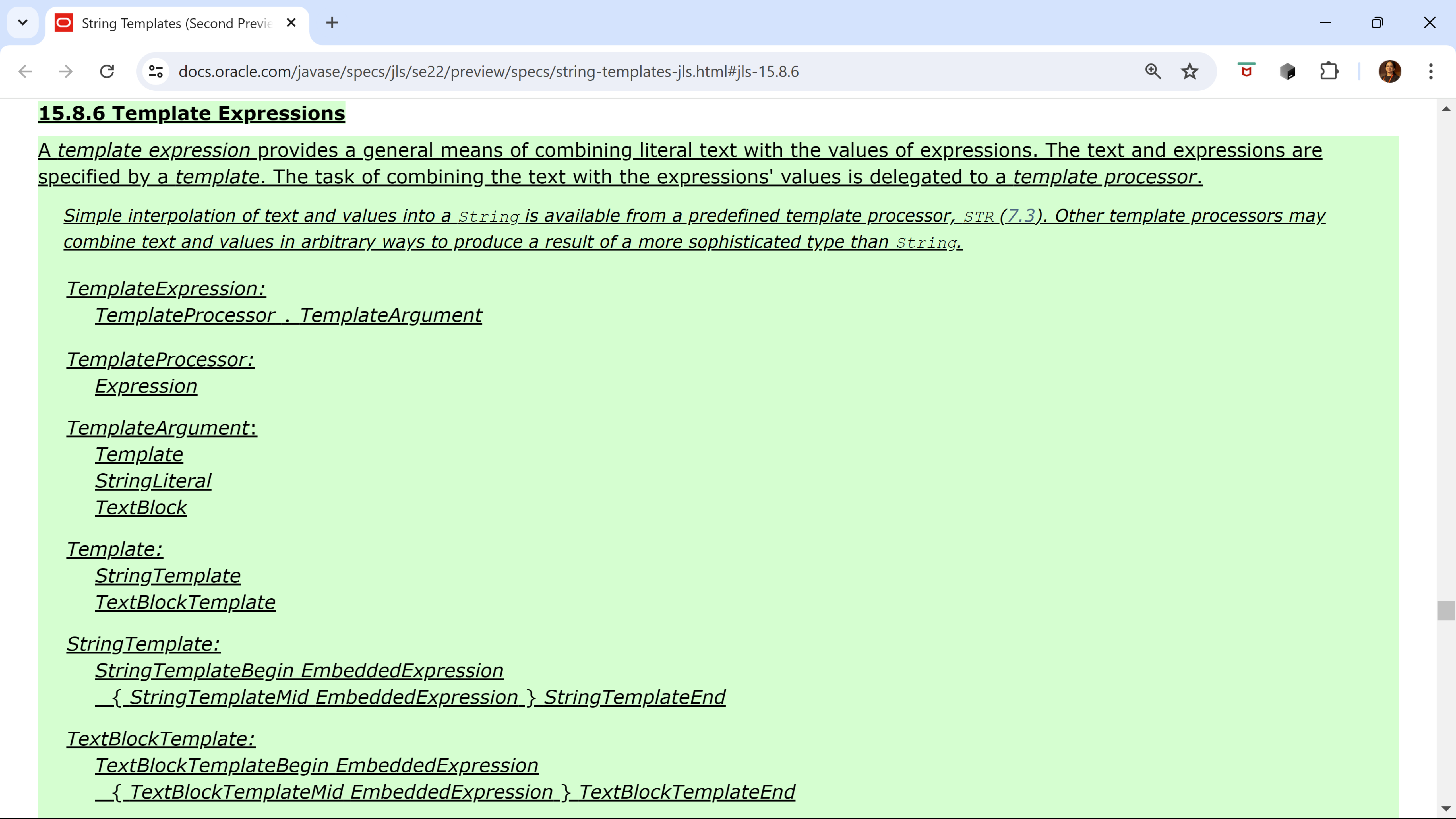
Write a comment, @mention people

Project	IntelliJ IDEA
Priority	Normal
Type	Bug
State	Open
Assignee	Bas Leijdekkers
Subsystem	Java
Affected versions	Not specified
Planned for	Not specified
Included in builds	Not specified
Support	Sergei Riabinin
QA	IJ Java QA
Verified	No


String Templates (Second Previe
docs.oracle.com/javase/specs/jls/se22/preview/specs/string-templates-jls.html

String Templates (Second Preview)

Changes to the Java® Language Specification • Version 22+35-2369


String Templates (Second Previe
docs.oracle.com/javase/specs/jls/se22/preview/specs/string-templates-jls.html#jls-15.8.6

15.8.6 Template Expressions

A *template expression* provides a general means of combining literal text with the values of expressions. The text and expressions are specified by a *template*. The task of combining the text with the expressions' values is delegated to a *template processor*.

Simple interpolation of text and values into a `String` is available from a predefined template processor, `STR` (7.3). Other template processors may combine text and values in arbitrary ways to produce a result of a more sophisticated type than `String`.

TemplateExpression:
  TemplateProcessor . TemplateArgument

TemplateProcessor:
  Expression

TemplateArgument:
  Template
  StringLiteral
  TextBlock

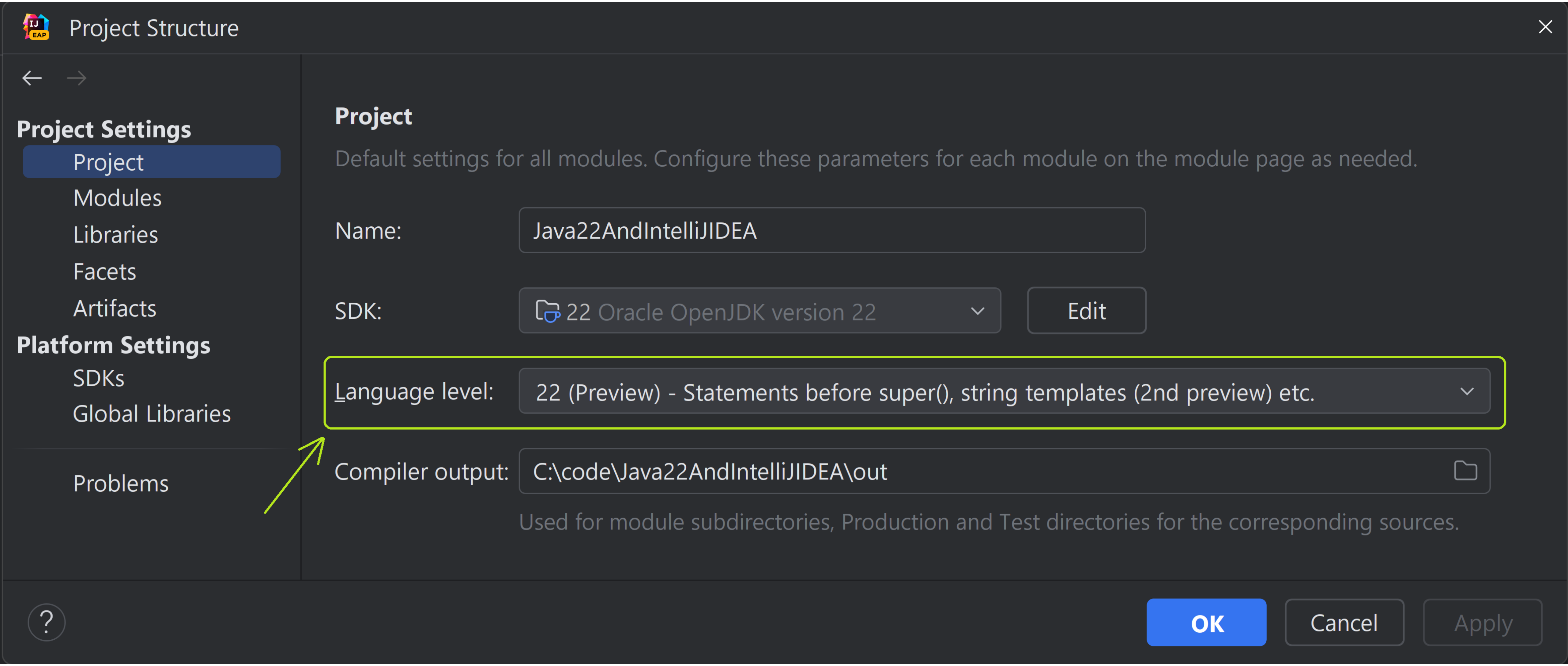
Template:
  StringTemplate
  TextBlockTemplate

StringTemplate:
  StringTemplateBegin EmbeddedExpression
  { StringTemplateMid EmbeddedExpression } StringTemplateEnd

TextBlockTemplate:
  TextBlockTemplateBegin EmbeddedExpression
  { TextBlockTemplateMid EmbeddedExpression } TextBlockTemplateEnd

Zero Tolerance to red code

#1 EAP Support


Project Structure

Project Settings
- Project
- Modules
- Libraries
- Facets
- Artifacts

Platform Settings
- SDKs
- Global Libraries

Problems

Project

Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name: Java22AndIntelliJIDEA

SDK: 22 Oracle OpenJDK version 22

Language level: 22 (Preview) - Statements before super(), string templates (2nd preview) etc.

Compiler output: C:\code\Java22AndIntelliJIDEA\out

Used for module subdirectories, Production and Test directories for the corresponding sources.

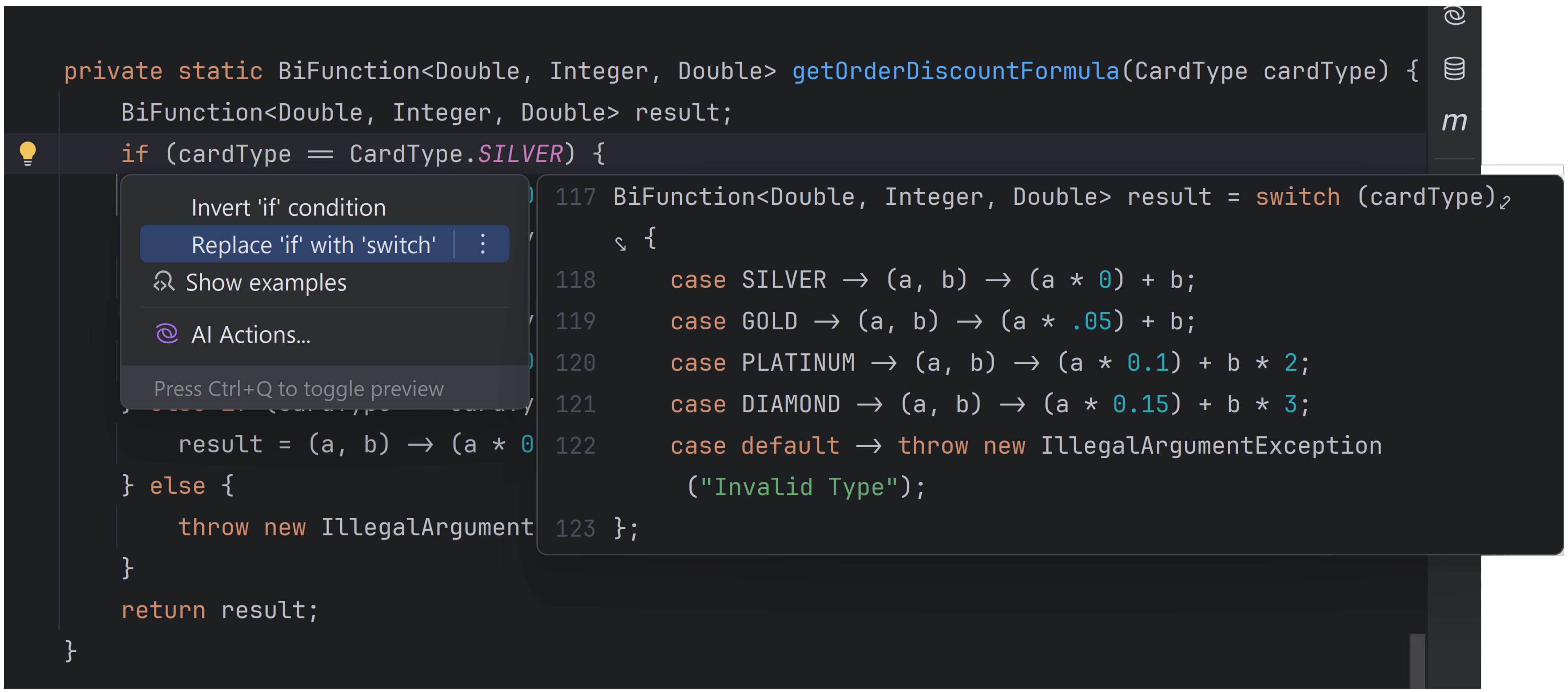
#2 False appearance of red code

#3 Incompatible API Changes

Inspections

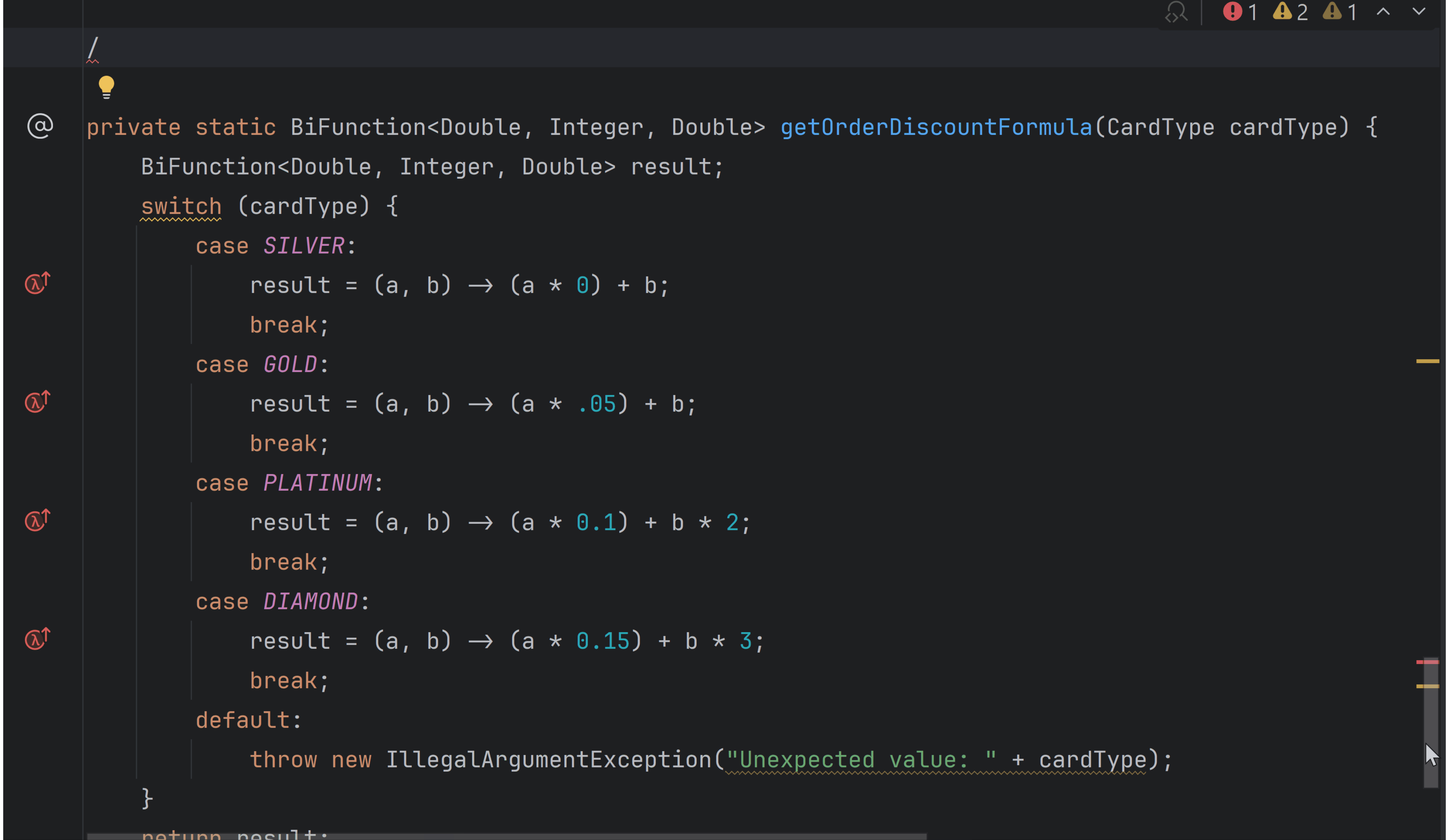
Intentions (Context Actions)

Examples –
Switch expressions



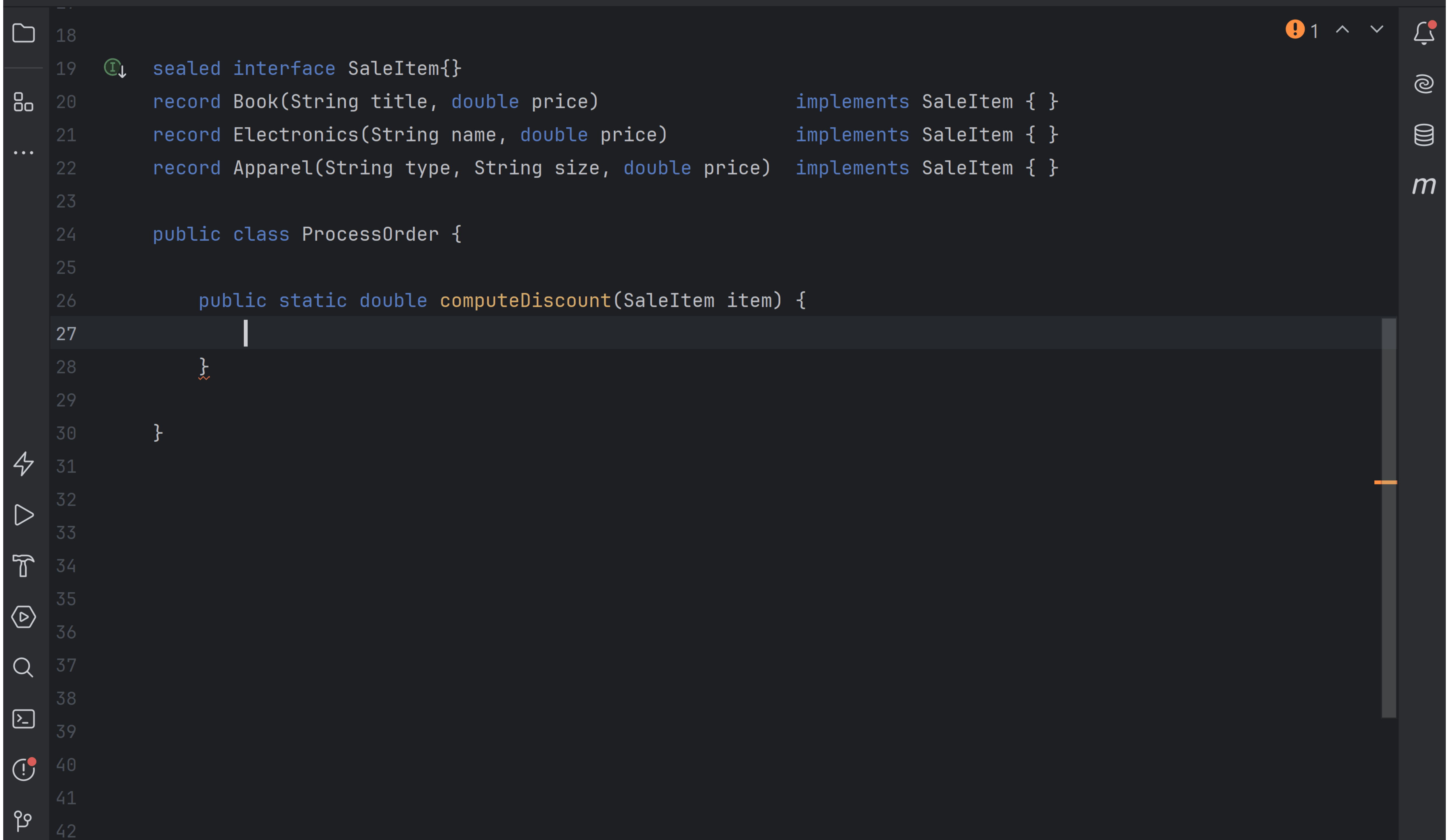
```
private static BiFunction<Double, Integer, Double> getOrderDiscountFormula(CardType cardType) {
    BiFunction<Double, Integer, Double> result;
    if (cardType == CardType.SILVER) {

        result = (a, b) -> (a * 0
    } else {
        throw new IllegalArgument
    }
    return result;
}
```

```
BiFunction<Double, Integer, Double> result = switch (cardType)
 {
    case SILVER -> (a, b) -> (a * 0) + b;
    case GOLD -> (a, b) -> (a * .05) + b;
    case PLATINUM -> (a, b) -> (a * 0.1) + b * 2;
    case DIAMOND -> (a, b) -> (a * 0.15) + b * 3;
    case default -> throw new IllegalArgumentException
     ("Invalid Type");
};
```

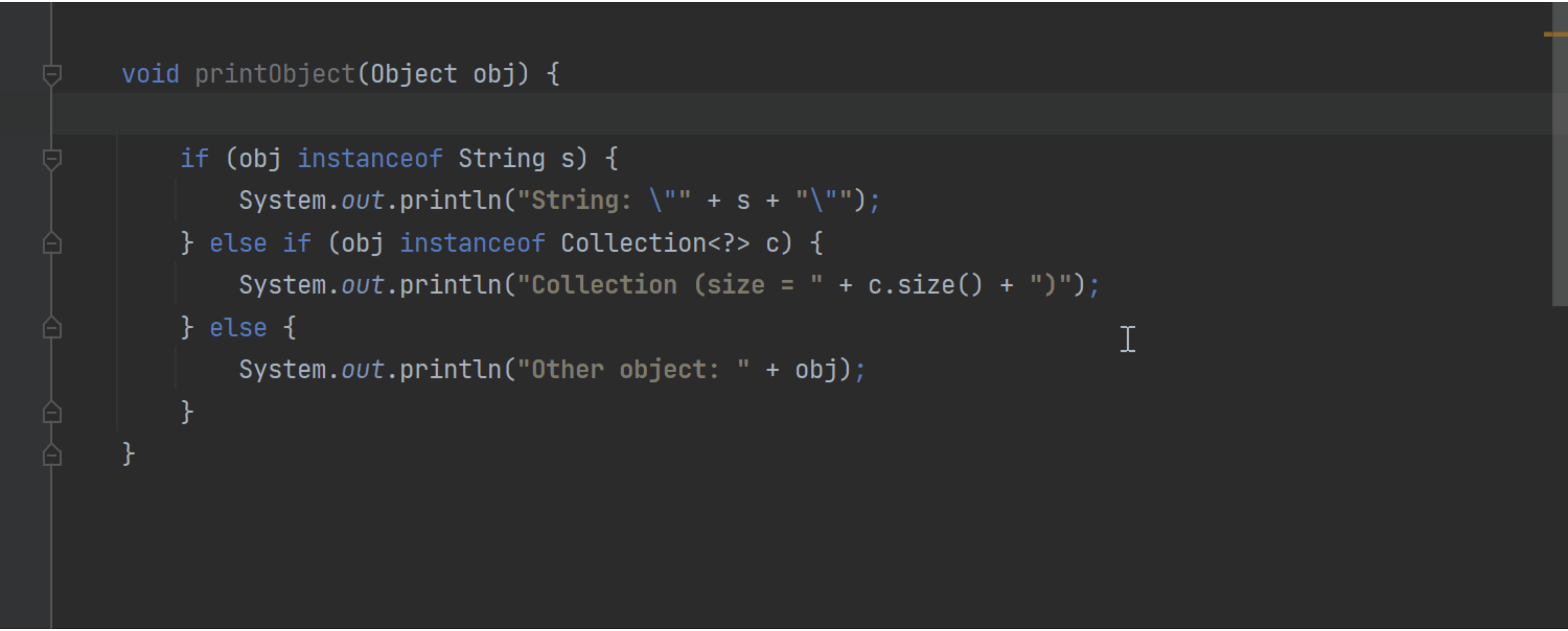


```java
/

private static BiFunction<Double, Integer, Double> getOrderDiscountFormula(CardType cardType) {
    BiFunction<Double, Integer, Double> result;
    switch (cardType) {
        case SILVER:
            result = (a, b) -> (a * 0) + b;
            break;
        case GOLD:
            result = (a, b) -> (a * .05) + b;
            break;
        case PLATINUM:
            result = (a, b) -> (a * 0.1) + b * 2;
            break;
        case DIAMOND:
            result = (a, b) -> (a * 0.15) + b * 3;
            break;
        default:
            throw new IllegalArgumentException("Unexpected value: " + cardType);
    }
    return result;
```

Sealed classes, Pattern Matching and Switch Expressions

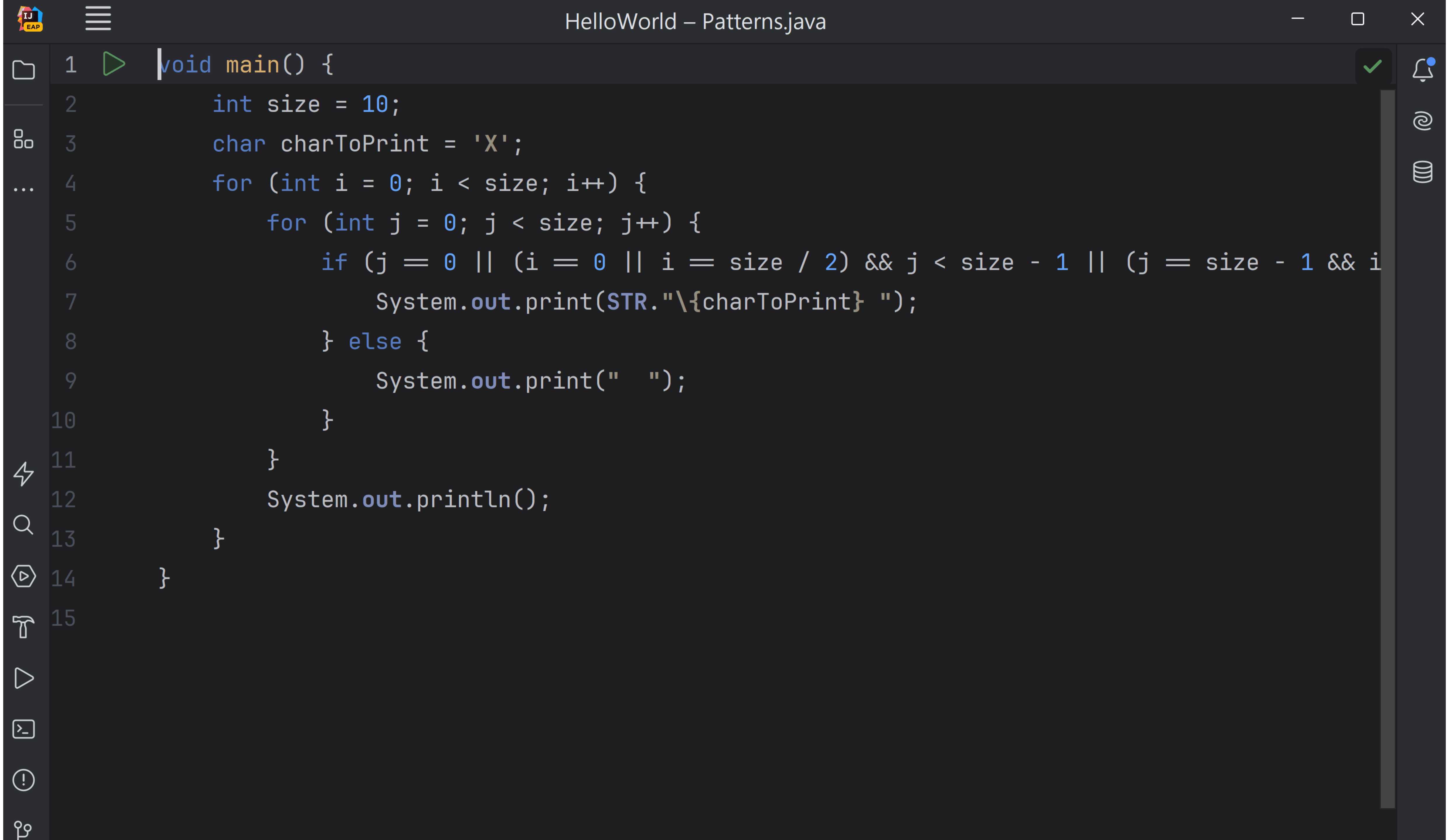


```java
sealed interface SaleItem{}
record Book(String title, double price)                  implements SaleItem { }
record Electronics(String name, double price)            implements SaleItem { }
record Apparel(String type, String size, double price)  implements SaleItem { }

public class ProcessOrder {

    public static double computeDiscount(SaleItem item) {
        
    }

}
```

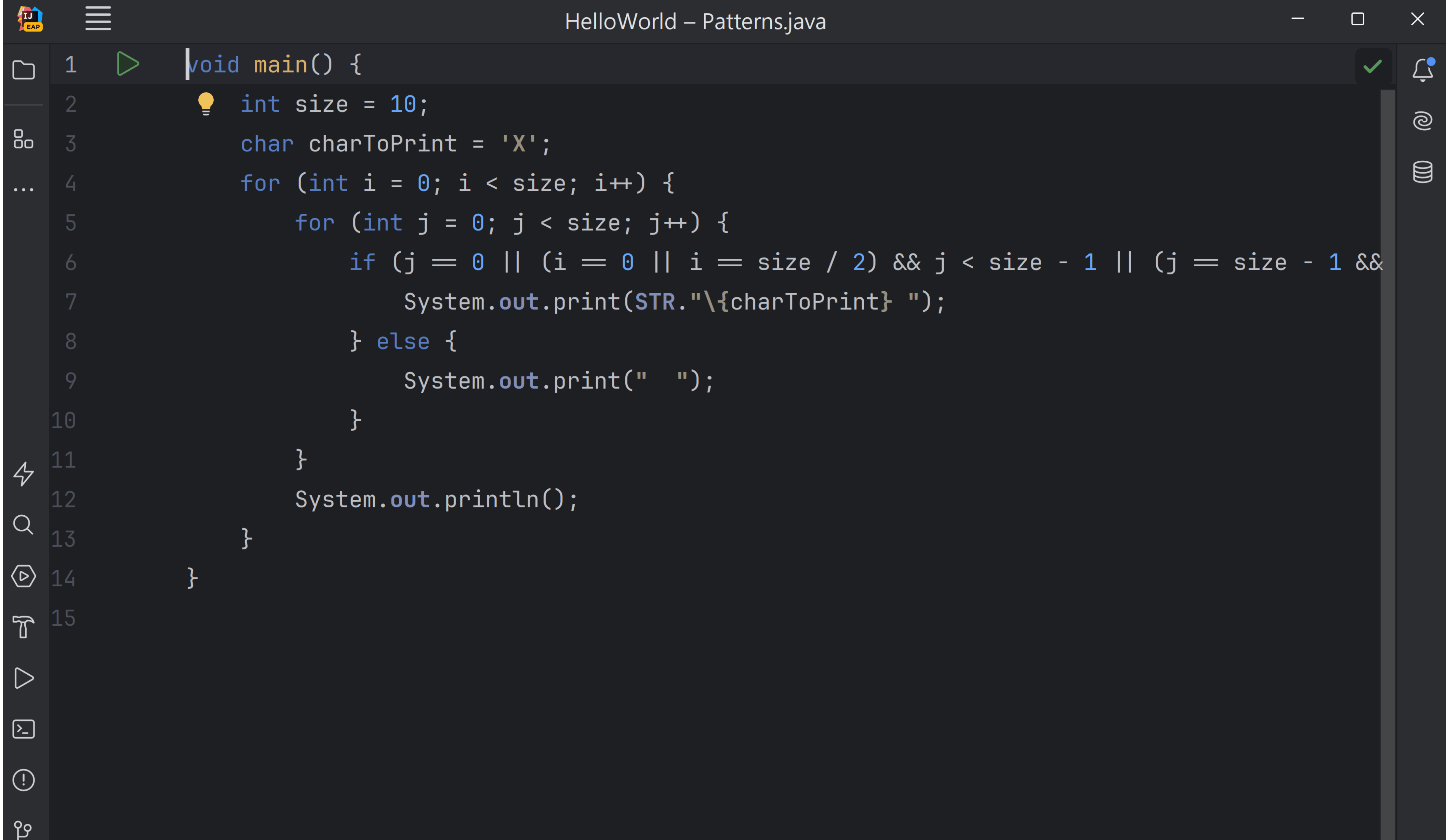


```java
void printObject(Object obj) {

    if (obj instanceof String s) {
        System.out.println("String: \"" + s + "\"");
    } else if (obj instanceof Collection<?> c) {
        System.out.println("Collection (size = " + c.size() + ")");
    } else {
        System.out.println("Other object: " + obj);
    }
}
```

Implicit classes

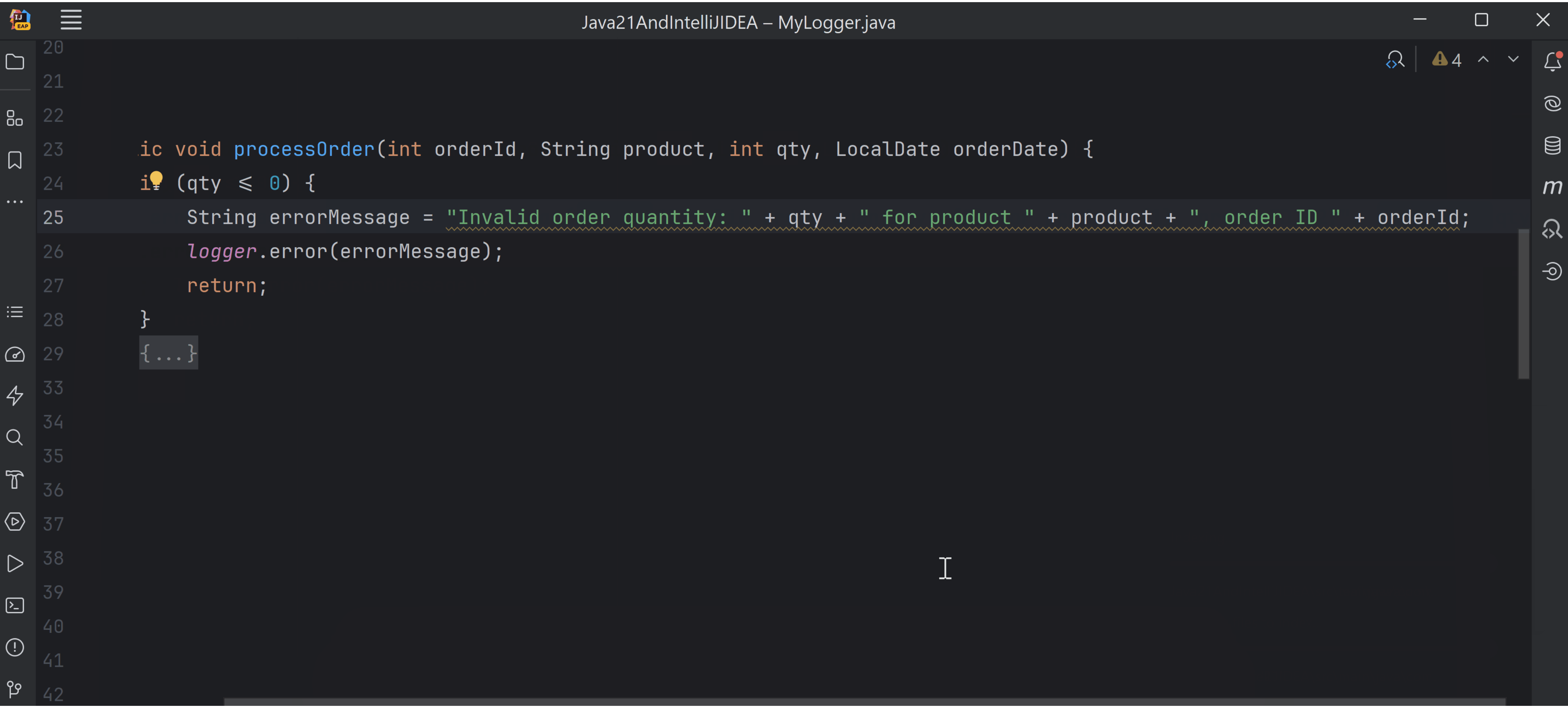

HelloWorld – Patterns.java

```java
void main() {
    int size = 10;
    char charToPrint = 'X';
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (j == 0 || (i == 0 || i == size / 2) && j < size - 1 || (j == size - 1 && i
                System.out.print(STR."\{charToPrint} ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
}

```


HelloWorld – Patterns.java

```java
void main() {
    int size = 10;
    char charToPrint = 'X';
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (j == 0 || (i == 0 || i == size / 2) && j < size - 1 || (j == size - 1 &&
                System.out.print(STR."\{charToPrint} ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
}
```

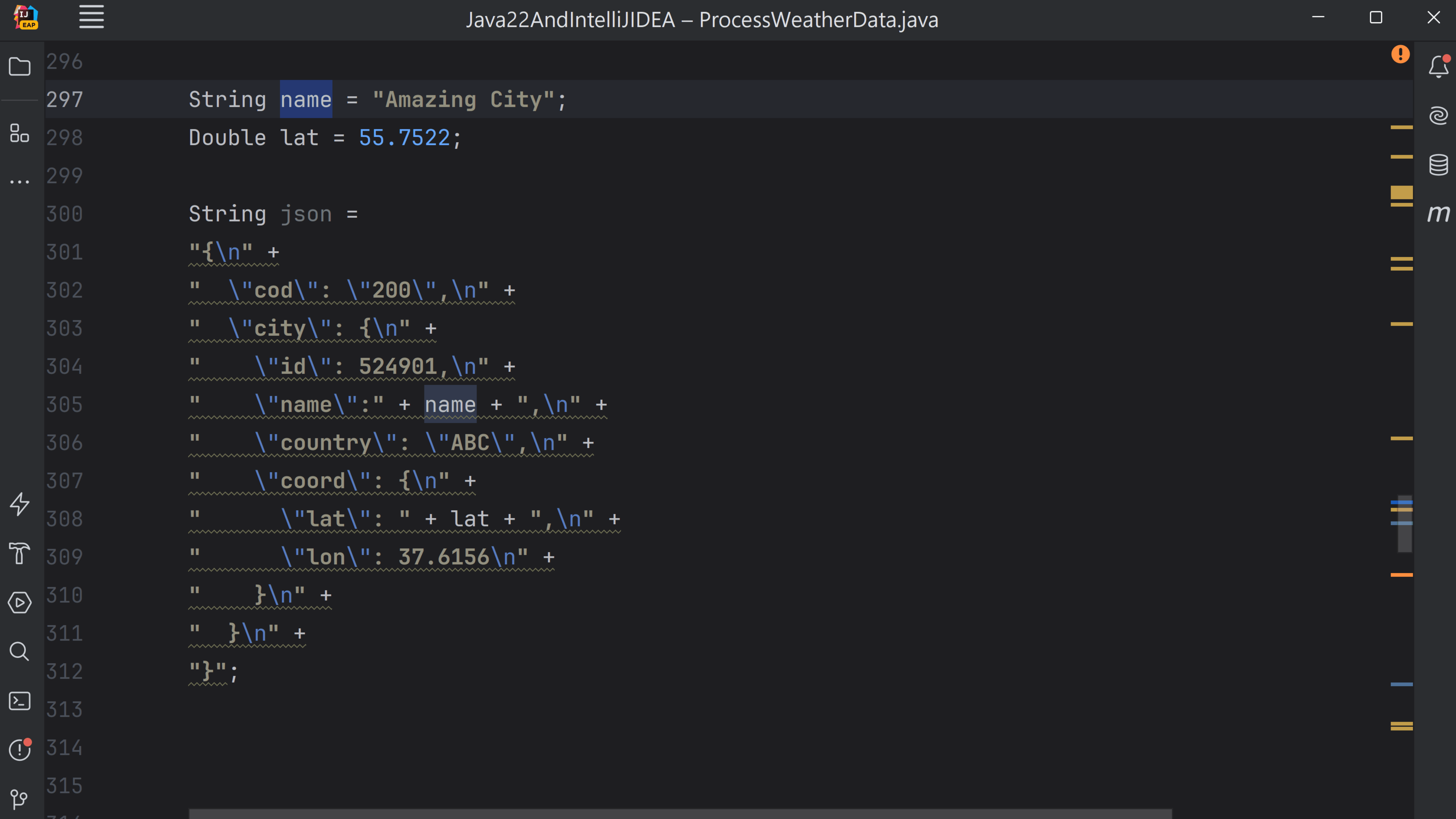
String Templates



Java21AndIntelliJIDEA – MyLogger.java

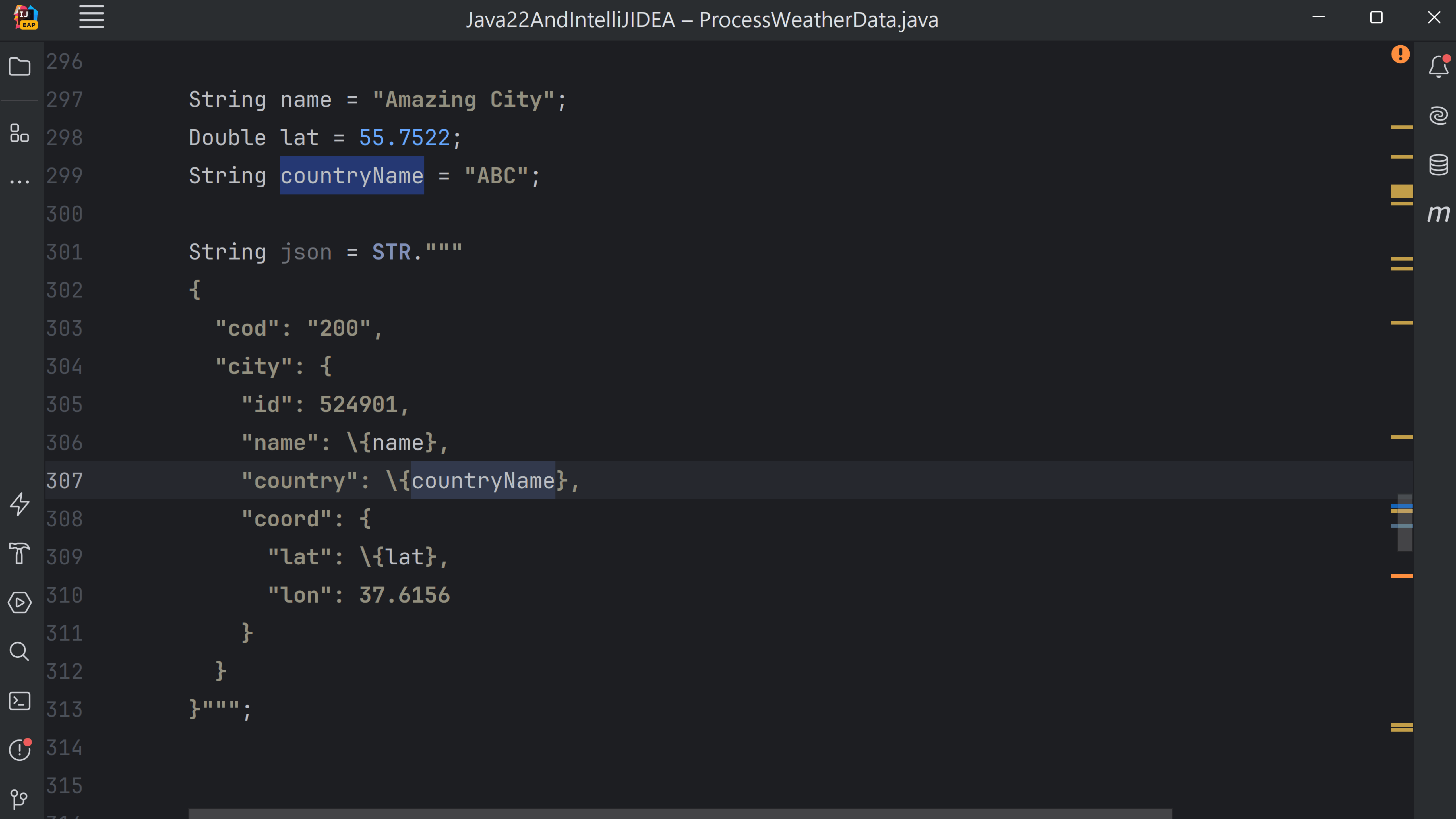
```
ic void processOrder(int orderId, String product, int qty, LocalDate orderDate) {
i  (qty ≤ 0) {
       String errorMessage = "Invalid order quantity: " + qty + " for product " + product + ", order ID " + orderId;
       logger.error(errorMessage);
       return;
   }
   {...}
```

String Templates and Textblocks


Java22AndIntelliJIDEA – ProcessWeatherData.java

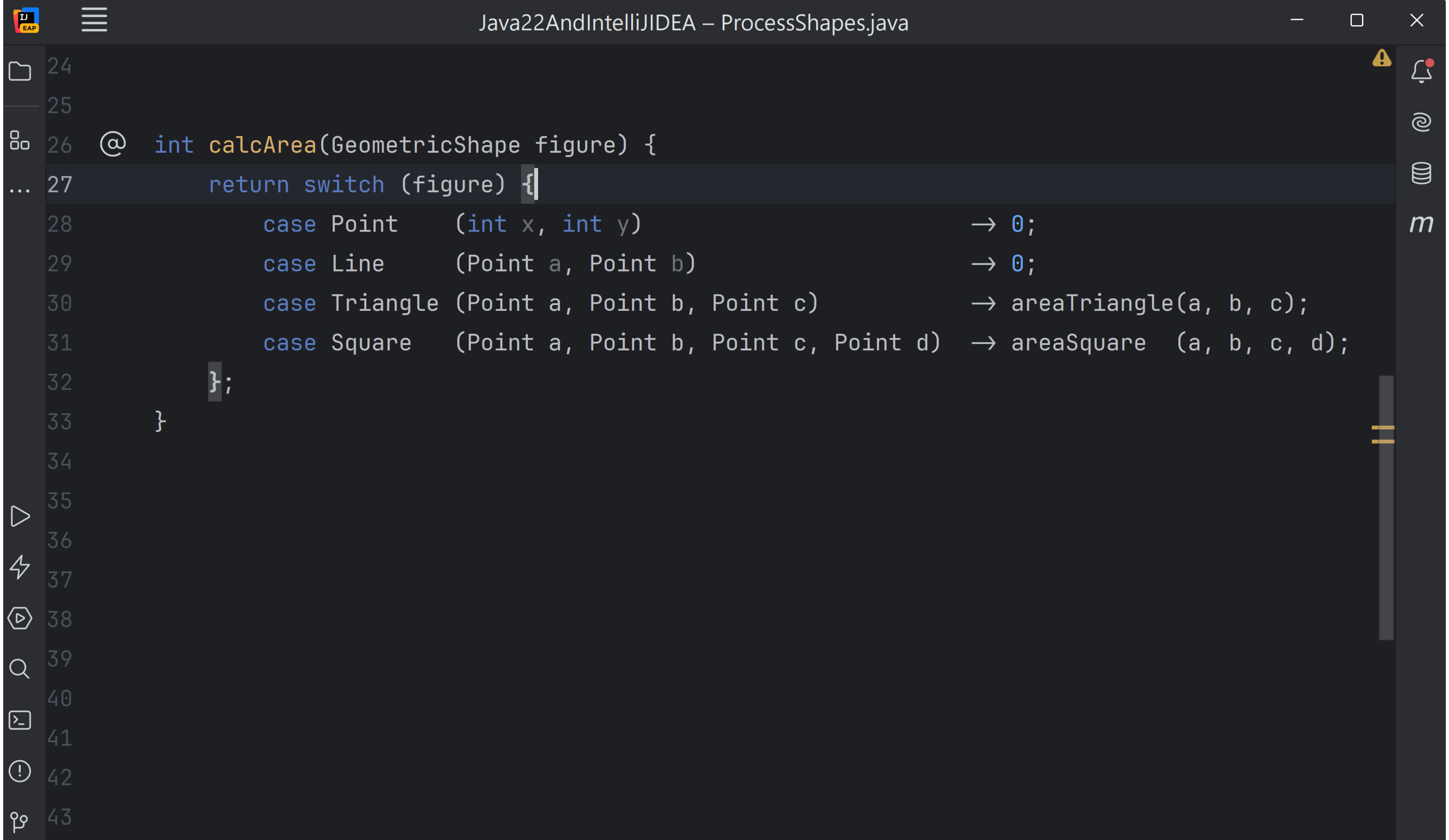
```java

String name = "Amazing City";
Double lat = 55.7522;

String json =
"{\n" +
"  \"cod\": \"200\",\n" +
"  \"city\": {\n" +
"    \"id\": 524901,\n" +
"    \"name\":" + name + ",\n" +
"    \"country\": \"ABC\",\n" +
"    \"coord\": {\n" +
"      \"lat\": " + lat + ",\n" +
"      \"lon\": 37.6156\n" +
"    }\n" +
"  }\n" +
"}";



```

Language Injection and String Templates


Java22AndIntelliJIDEA – ProcessWeatherData.java

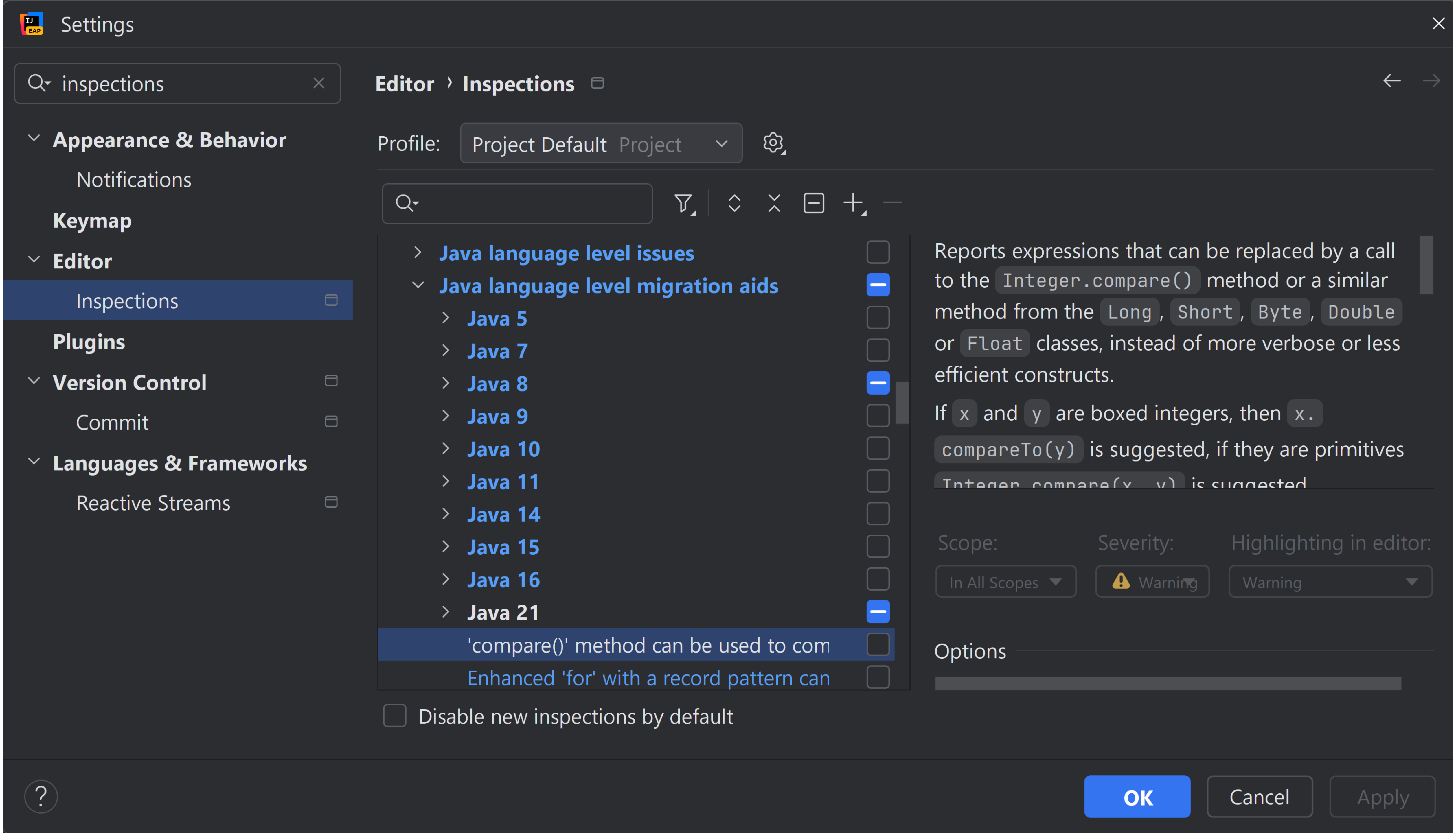
```
        String name = "Amazing City";
        Double lat = 55.7522;
        String countryName = "ABC";

        String json = STR."""
        {
          "cod": "200",
          "city": {
            "id": 524901,
            "name": \{name},
            "country": \{countryName},
            "coord": {
              "lat": \{lat},
              "lon": 37.6156
            }
          }
        }""";
```

Unused pattern variables


Java22AndIntelliJIDEA – ProcessShapes.java

```java
int calcArea(GeometricShape figure) {
    return switch (figure) {
        case Point    (int x, int y)                        → 0;
        case Line     (Point a, Point b)                    → 0;
        case Triangle (Point a, Point b, Point c)           → areaTriangle(a, b, c);
        case Square   (Point a, Point b, Point c, Point d)  → areaSquare  (a, b, c, d);
    };
}
```

Migration inspections



Settings

inspections

Appearance & Behavior
Notifications
Keymap
Editor
Inspections
Plugins
Version Control
Commit
Languages & Frameworks
Reactive Streams

Editor › Inspections

Profile: Project Default Project

Java language level issues
Java language level migration aids
Java 5
Java 7
Java 8
Java 9
Java 10
Java 11
Java 14
Java 15
Java 16
Java 21
'compare()' method can be used to com
Enhanced 'for' with a record pattern can

Reports expressions that can be replaced by a call to the `Integer.compare()` method or a similar method from the `Long`, `Short`, `Byte`, `Double` or `Float` classes, instead of more verbose or less efficient constructs.

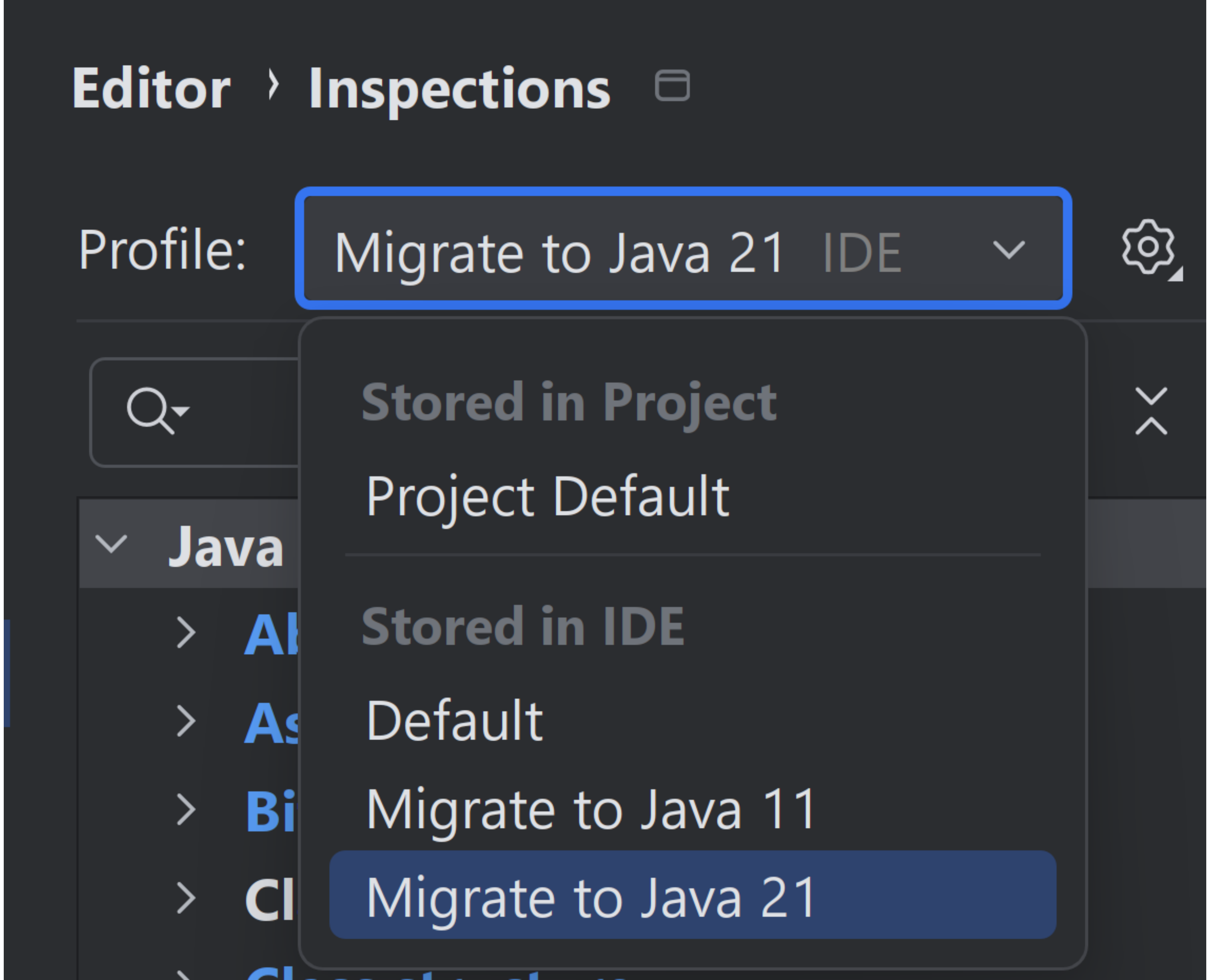
If `x` and `y` are boxed integers, then `x.compareTo(y)` is suggested, if they are primitives `Integer.compare(x, y)` is suggested

Scope: In All Scopes
Severity: Warning
Highlighting in editor: Warning

Options

Disable new inspections by default

OK Cancel Apply



Editor › Inspections

Profile: Migrate to Java 21 IDE

Stored in Project

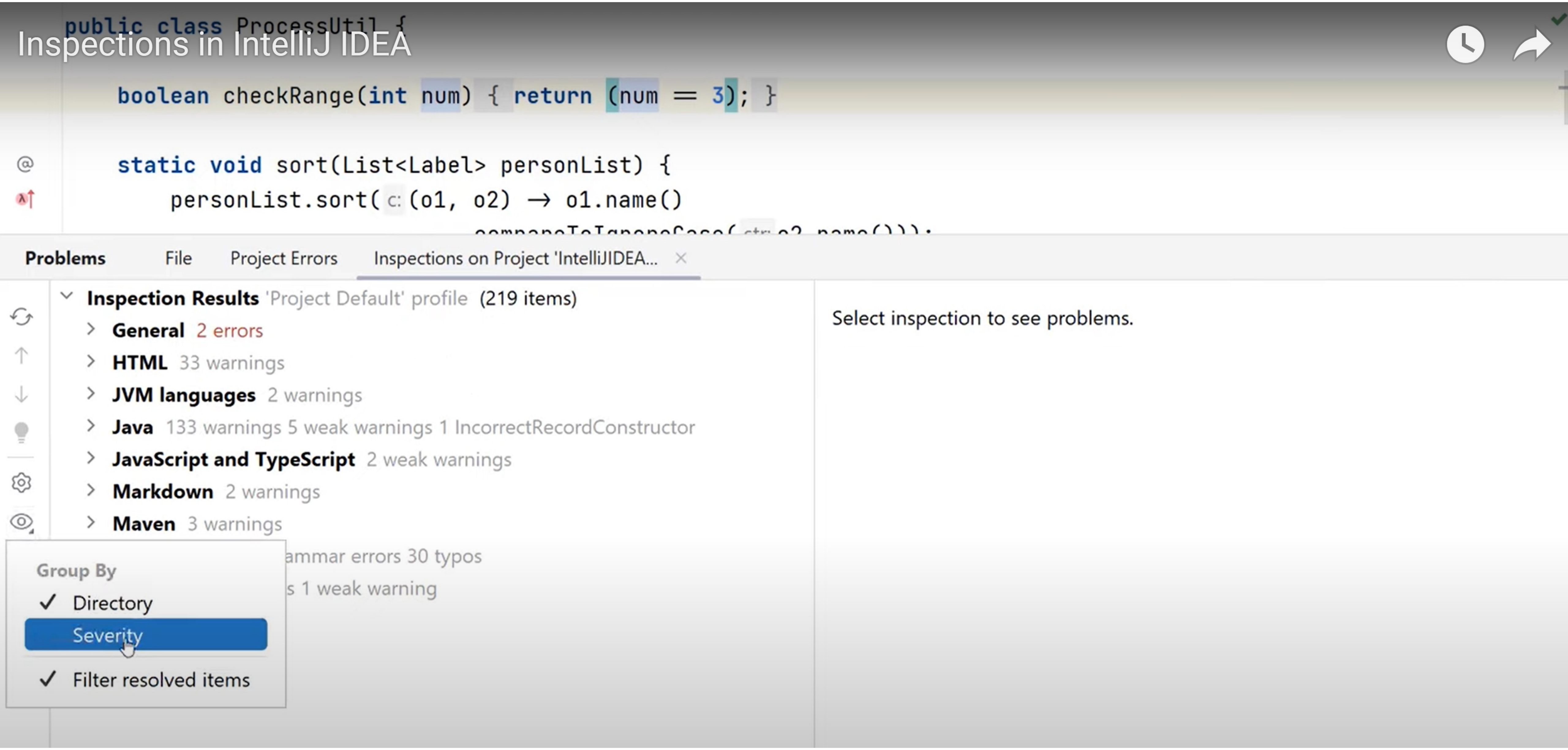
Project Default

Stored in IDE

Default

Migrate to Java 11

Migrate to Java 21


Inspections in IntelliJ IDEA

```
public class ProcessUtil {

    boolean checkRange(int num) { return (num == 3); }

    static void sort(List<Label> personList) {
        personList.sort( c: (o1, o2) -> o1.name()
```

Inspection Results 'Project Default' profile (219 items)
- General 2 errors
- HTML 33 warnings
- JVM languages 2 warnings
- Java 133 warnings 5 weak warnings 1 IncorrectRecordConstructor
- JavaScript and TypeScript 2 weak warnings
- Markdown 2 warnings
- Maven 3 warnings
- ammar errors 30 typos
- s 1 weak warning

Select inspection to see problems.

Thank you.