

# Pattern Matching for instanceof in Java 14

```
public boolean equals(Object o) {  
    if (this == o) return true;  
    return (o instanceof MyClass that  
            && abstractText.equals(that.abstractText)  
            && abstractURL.equals(that.abstractURL);  
}
```

# Pattern Matching for instanceof in Java 16

<https://openjdk.java.net/jeps/394>

instanceof-and-cast idiom

```
if (obj instanceof String) {  
    String s = (String) obj;  
}
```

# Usage

Pattern Matching for instanceof

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

*pattern variable*



## Pattern matching

**Pattern matching** allows common logic in a program, to be expressed more concisely and safely.

# Usage

Pattern Matching for instanceof


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

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




# flow sensitive analyses

A pattern variable is in scope where it has definitely matched

```
if (obj instanceof String s && s.length() > 5) { 
```

```
}
```

```
if (obj instanceof String s || s.length() > 5) { 
```

```
}
```

# Normal completion of statement

```
public String onlyForStrings(Object o){  
  
    if (!(o instanceof String str))  
        throw new RuntimeException();  
  
    // str is in scope  
    return str;  
  
}
```