Regression Failures

Unintended different behaviors
between the *base* and the *changed* version of the program

Program Slicing for Debugging

Program slicing computes a subset of program statements that **are related** to a particular variable in a statement [*]

Project	# buggy versions	Statements		
		# trace	# slice	% reduction
Chart	23	5,913	29	95.11
Closure	95	92,663	1,027	98.7
Lang	49	3,091	110	89.57
Math	633	8,801	148	94.49
Mockito	26	3,115	68	97.8
Time	22	13,038	75	99.03
Defects4J Avg.	**278**	**36,017**	**419**	**95.78**

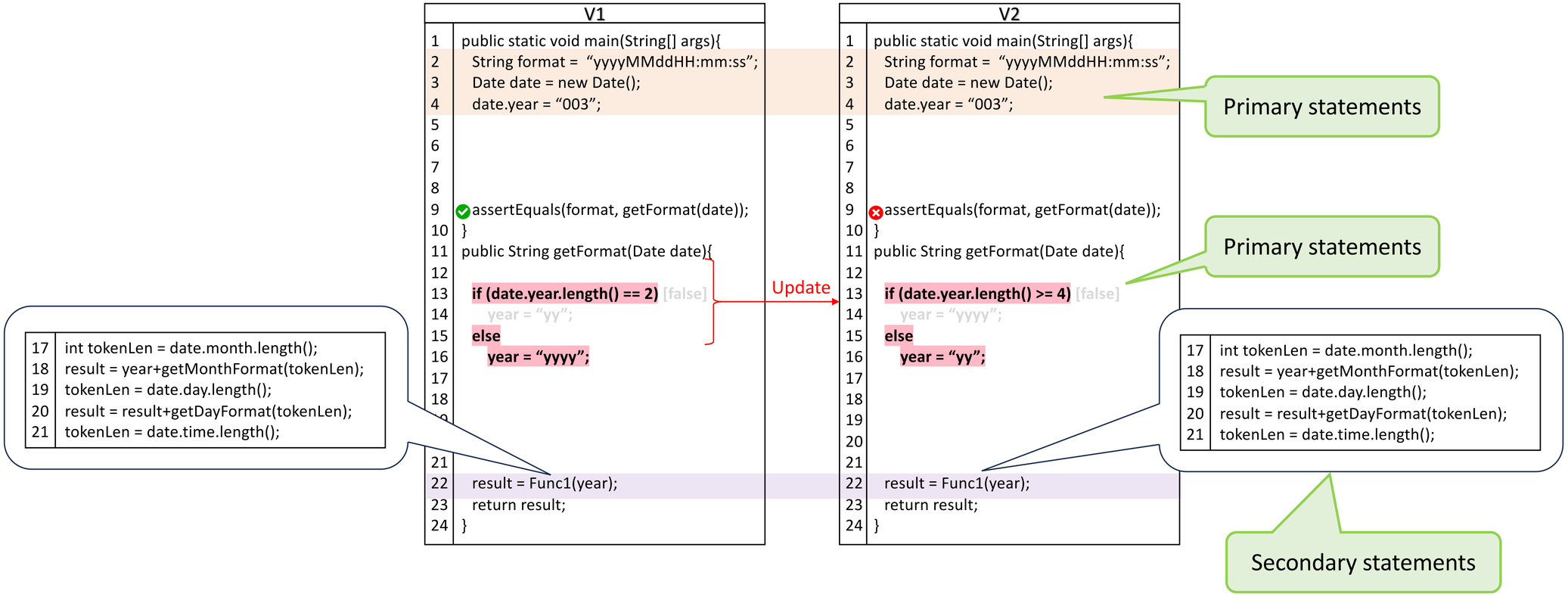
~35k statements

> 95% reduction of the code to inspect

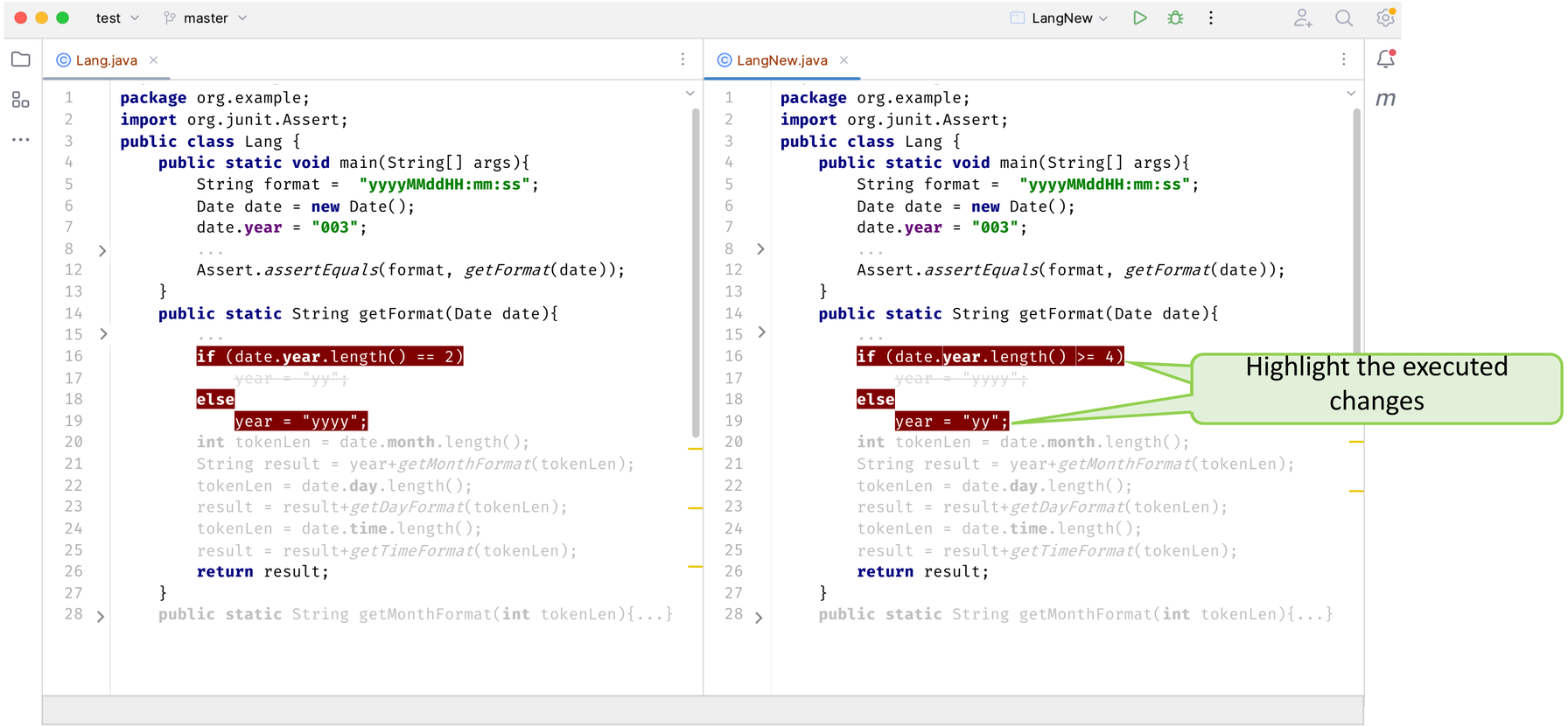
~500 statements

[*] Weiser M, Program slicing, IEEE Transactions on Software Engineering, 1984

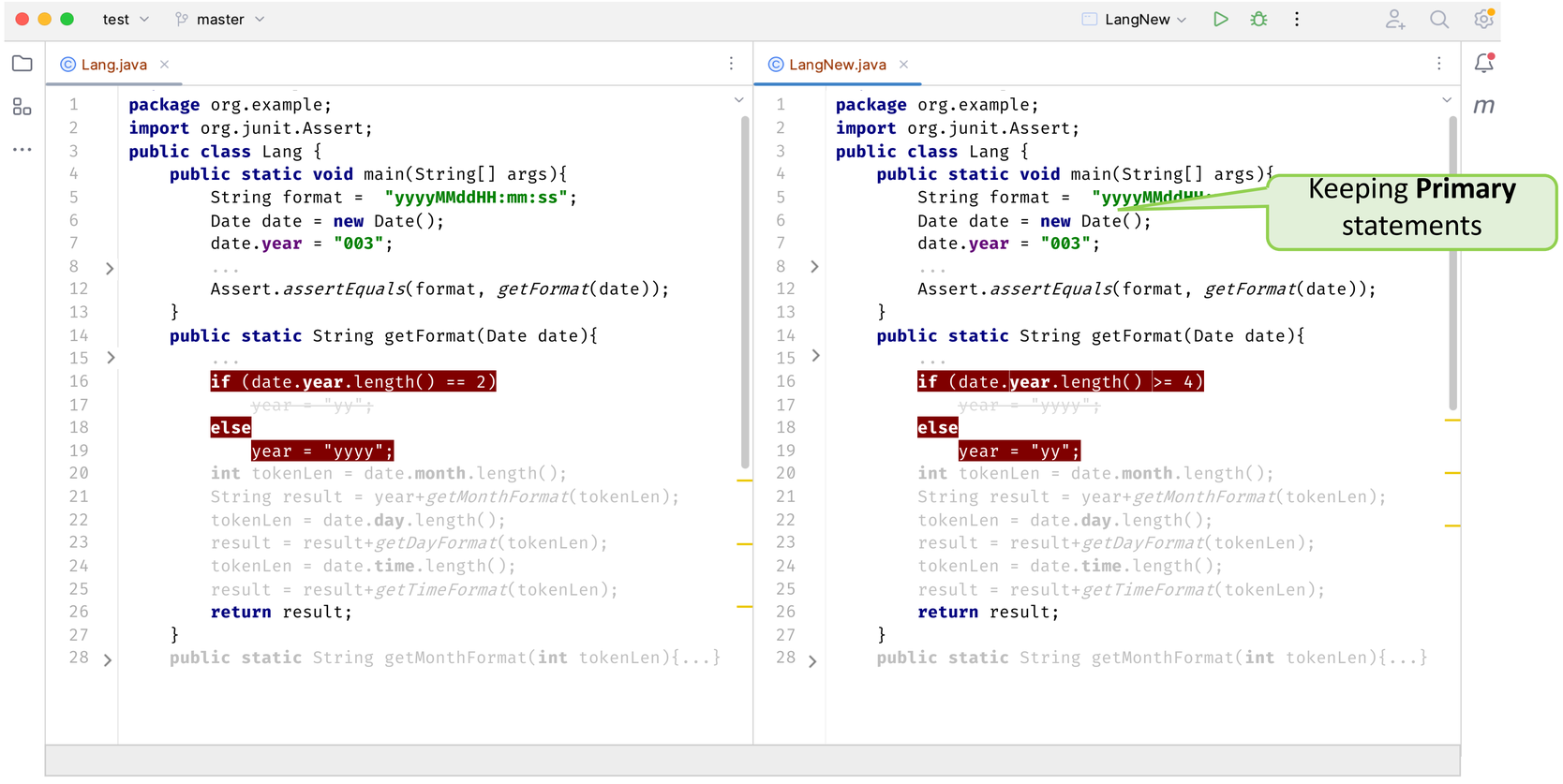
Context-Aware Refinement-Based Slicing for Debugging Regression Failures



Focusing on Necessary Information



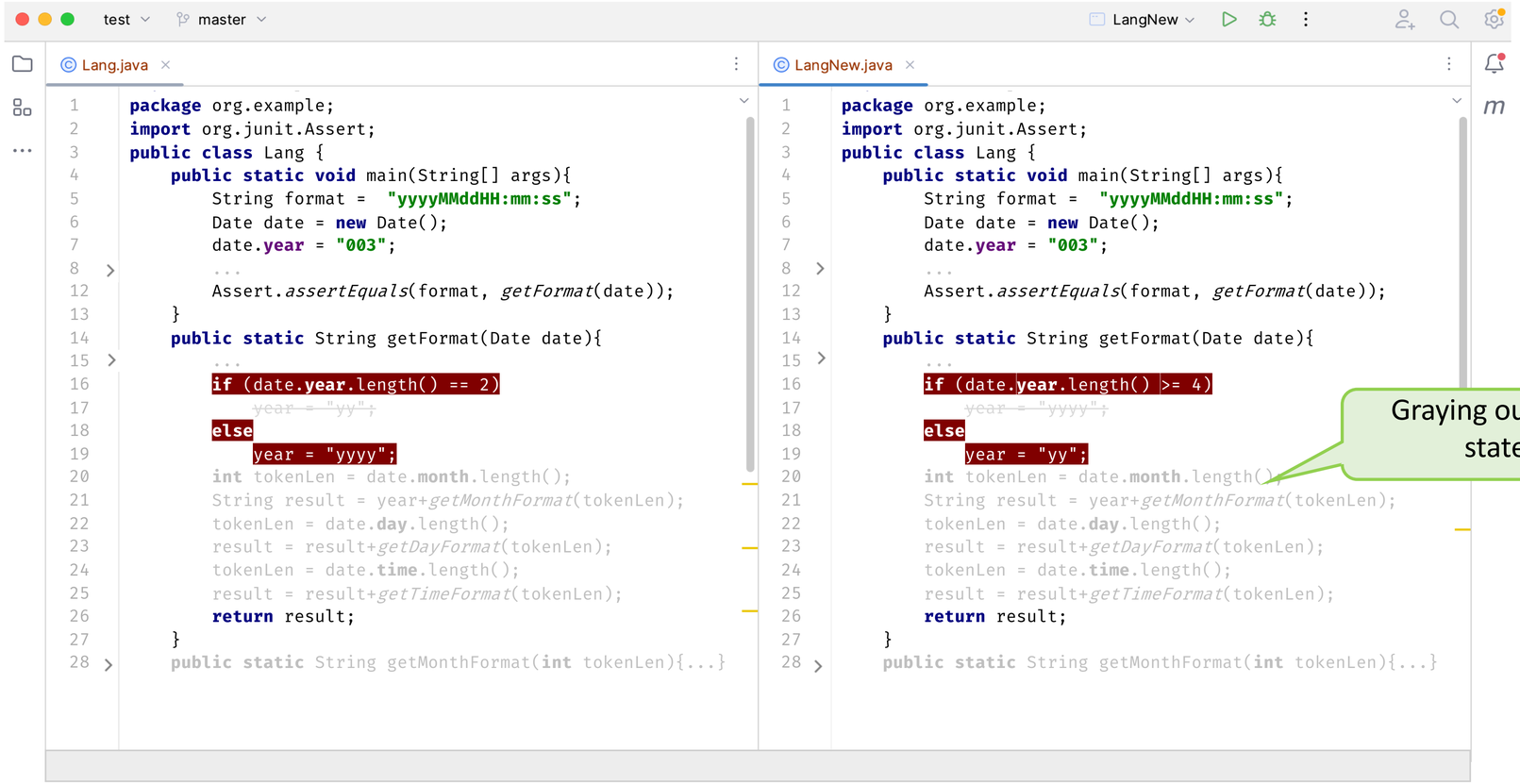
Focusing on Necessary Information




```
package org.example;
import org.junit.Assert;
public class Lang {
    public static void main(String[] args){
        String format =  "yyyyMMddHH:mm:ss";
        Date date = new Date();
        date.year = "003";
        ...
        Assert.assertEquals(format, getFormat(date));
    }
    public static String getFormat(Date date){
        ...
        if (date.year.length() == 2)
            year = "yy";
        else
            year = "yyyy";
        int tokenLen = date.month.length();
        String result = year+getMonthFormat(tokenLen);
        tokenLen = date.day.length();
        result = result+getDayFormat(tokenLen);
        tokenLen = date.time.length();
        result = result+getTimeFormat(tokenLen);
        return result;
    }
    public static String getMonthFormat(int tokenLen){...}
```

```
package org.example;
import org.junit.Assert;
public class Lang {
    public static void main(String[] args){
        String format =  "yyyyMMddHH:mm:ss";
        Date date = new Date();
        date.year = "003";
        ...
        Assert.assertEquals(format, getFormat(date));
    }
    public static String getFormat(Date date){
        ...
        if (date.year.length() >= 4)
            year = "yyyy";
        else
            year = "yy";
        int tokenLen = date.month.length();
        String result = year+getMonthFormat(tokenLen);
        tokenLen = date.day.length();
        result = result+getDayFormat(tokenLen);
        tokenLen = date.time.length();
        result = result+getTimeFormat(tokenLen);
        return result;
    }
    public static String getMonthFormat(int tokenLen){...}
```

Focusing on Necessary Information

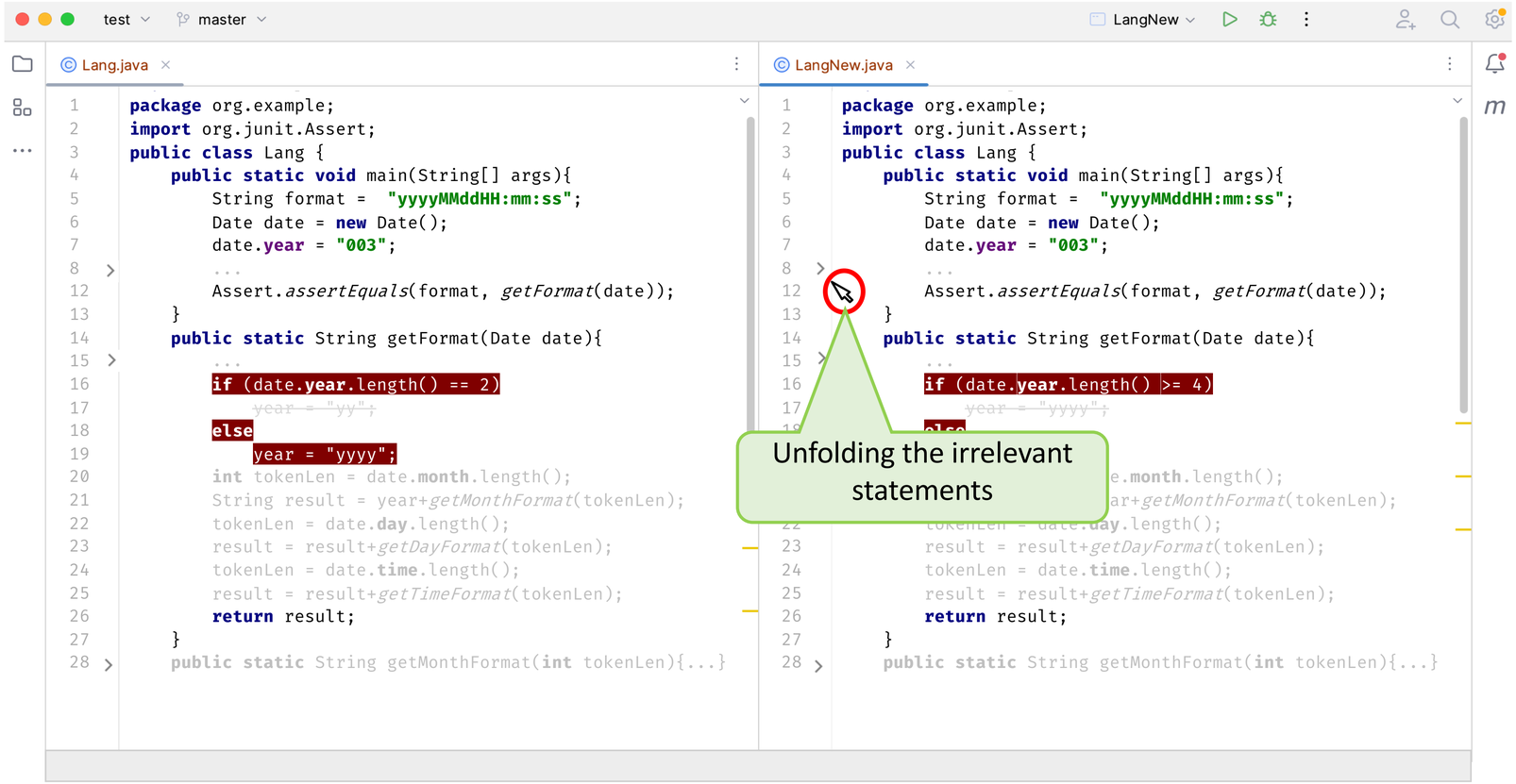



```
package org.example;
import org.junit.Assert;
public class Lang {
    public static void main(String[] args){
        String format =  "yyyyMMddHH:mm:ss";
        Date date = new Date();
        date.year = "003";
        ...
        Assert.assertEquals(format, getFormat(date));
    }
    public static String getFormat(Date date){
        ...
        if (date.year.length() == 2)
            year = "yy";
        else
            year = "yyyy";
        int tokenLen = date.month.length();
        String result = year+getMonthFormat(tokenLen);
        tokenLen = date.day.length();
        result = result+getDayFormat(tokenLen);
        tokenLen = date.time.length();
        result = result+getTimeFormat(tokenLen);
        return result;
    }
    public static String getMonthFormat(int tokenLen){...}
```

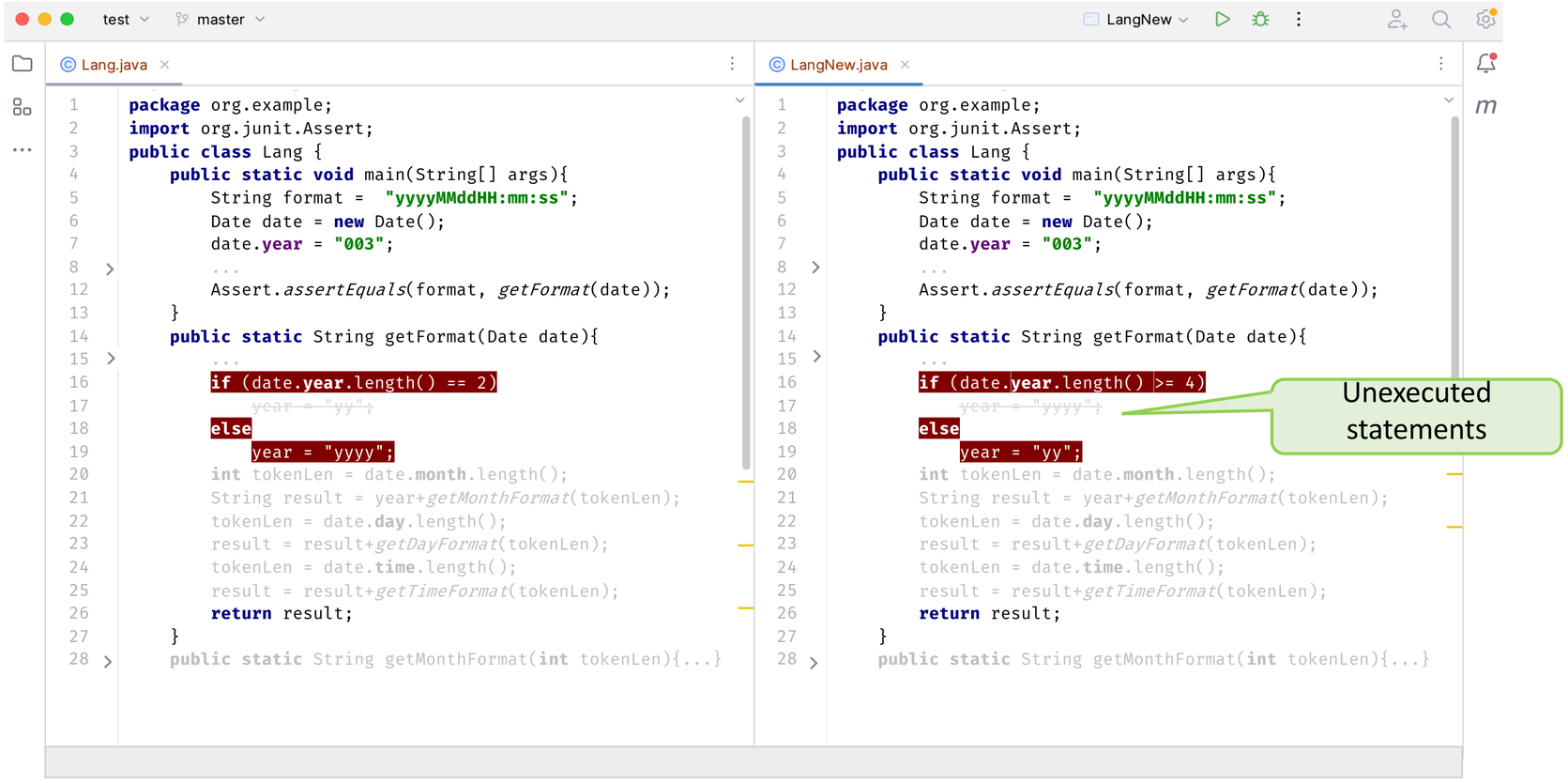
```
package org.example;
import org.junit.Assert;
public class Lang {
    public static void main(String[] args){
        String format =  "yyyyMMddHH:mm:ss";
        Date date = new Date();
        date.year = "003";
        ...
        Assert.assertEquals(format, getFormat(date));
    }
    public static String getFormat(Date date){
        ...
        if (date.year.length() >= 4)
            year = "yyyy";
        else
            year = "yy";
        int tokenLen = date.month.length();
        String result = year+getMonthFormat(tokenLen);
        tokenLen = date.day.length();
        result = result+getDayFormat(tokenLen);
        tokenLen = date.time.length();
        result = result+getTimeFormat(tokenLen);
        return result;
    }
    public static String getMonthFormat(int tokenLen){...}
```

Focusing on Necessary Information

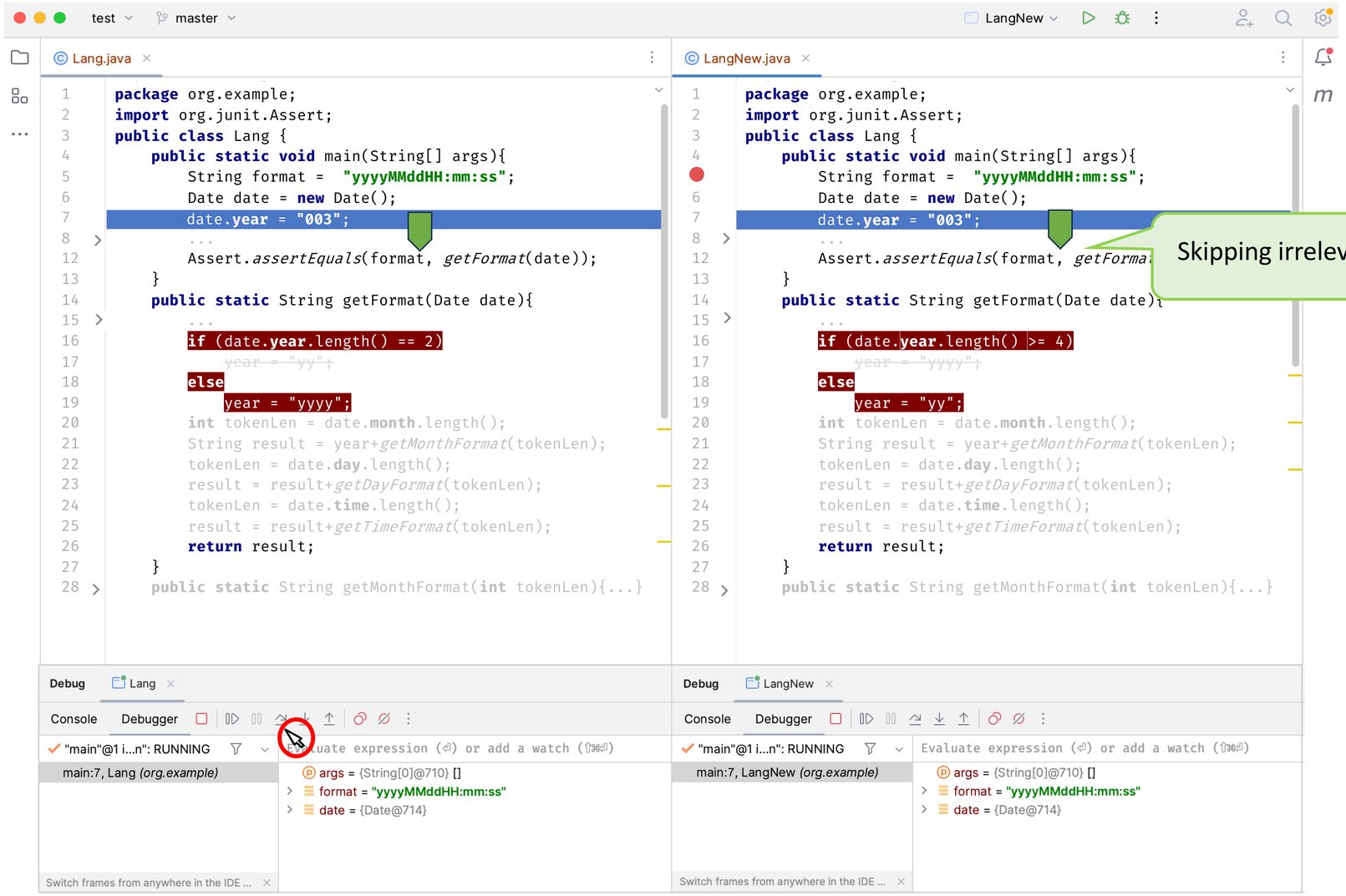
Focusing on Necessary Information



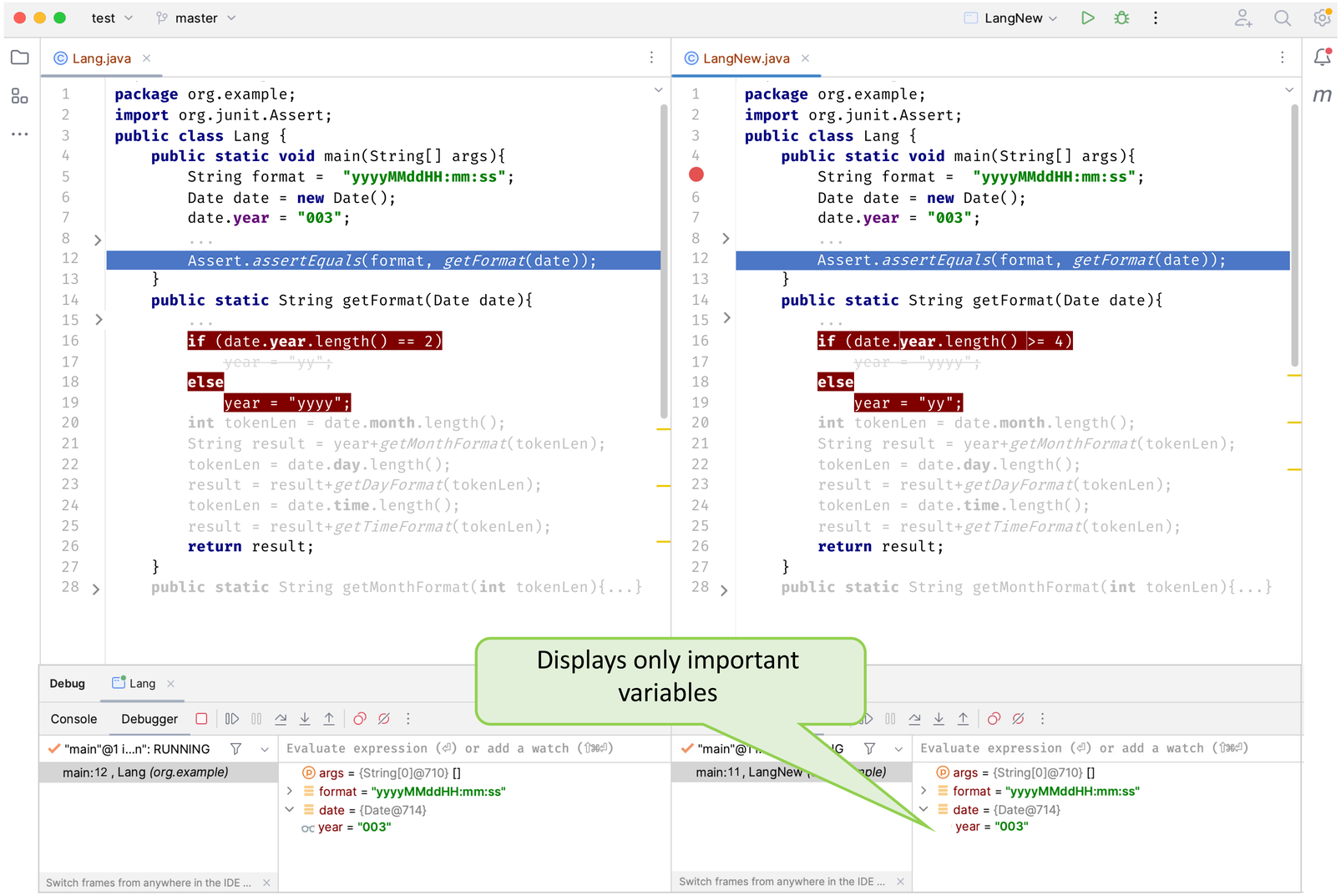
Focusing on Necessary Information



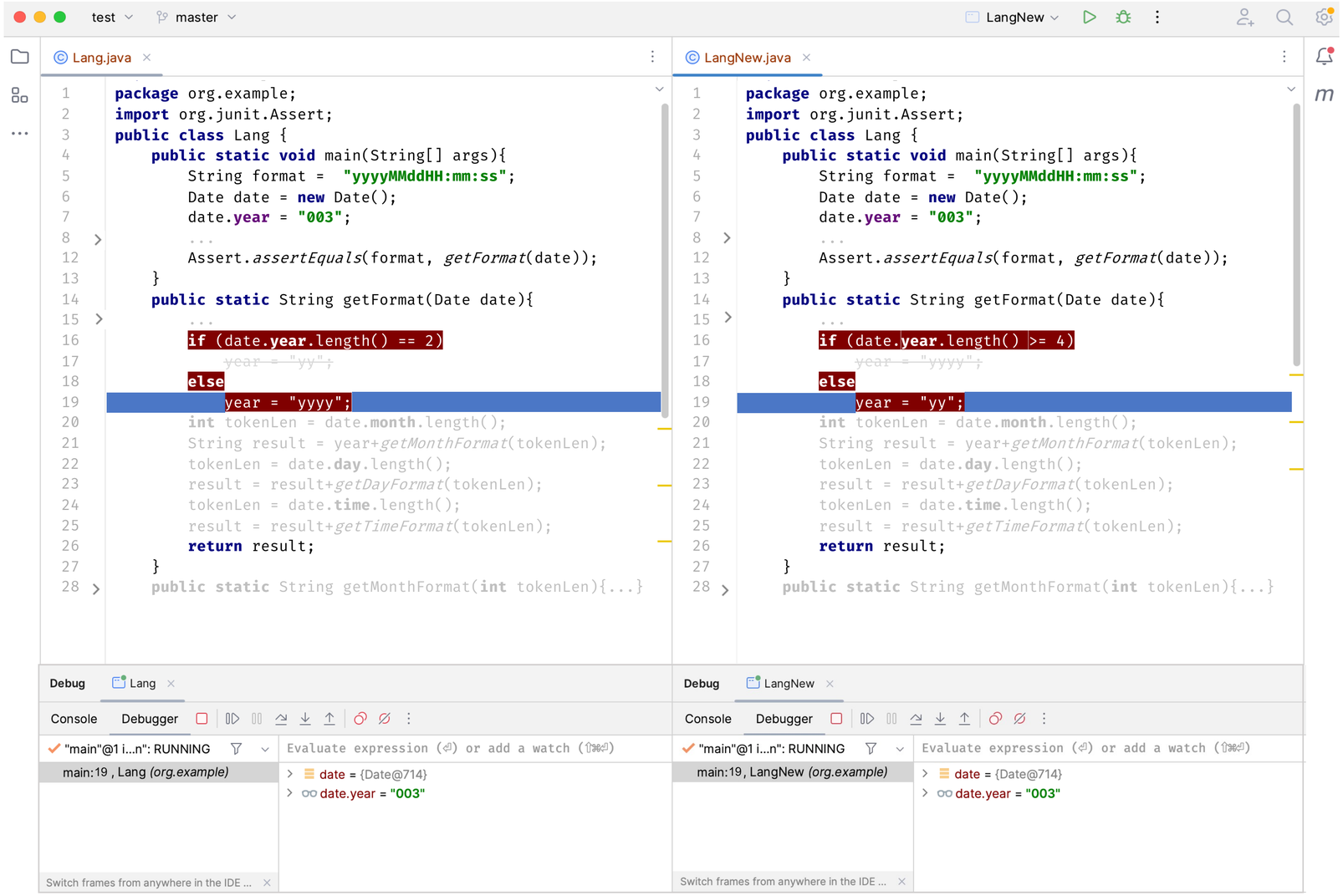
Debugging Essential Statements



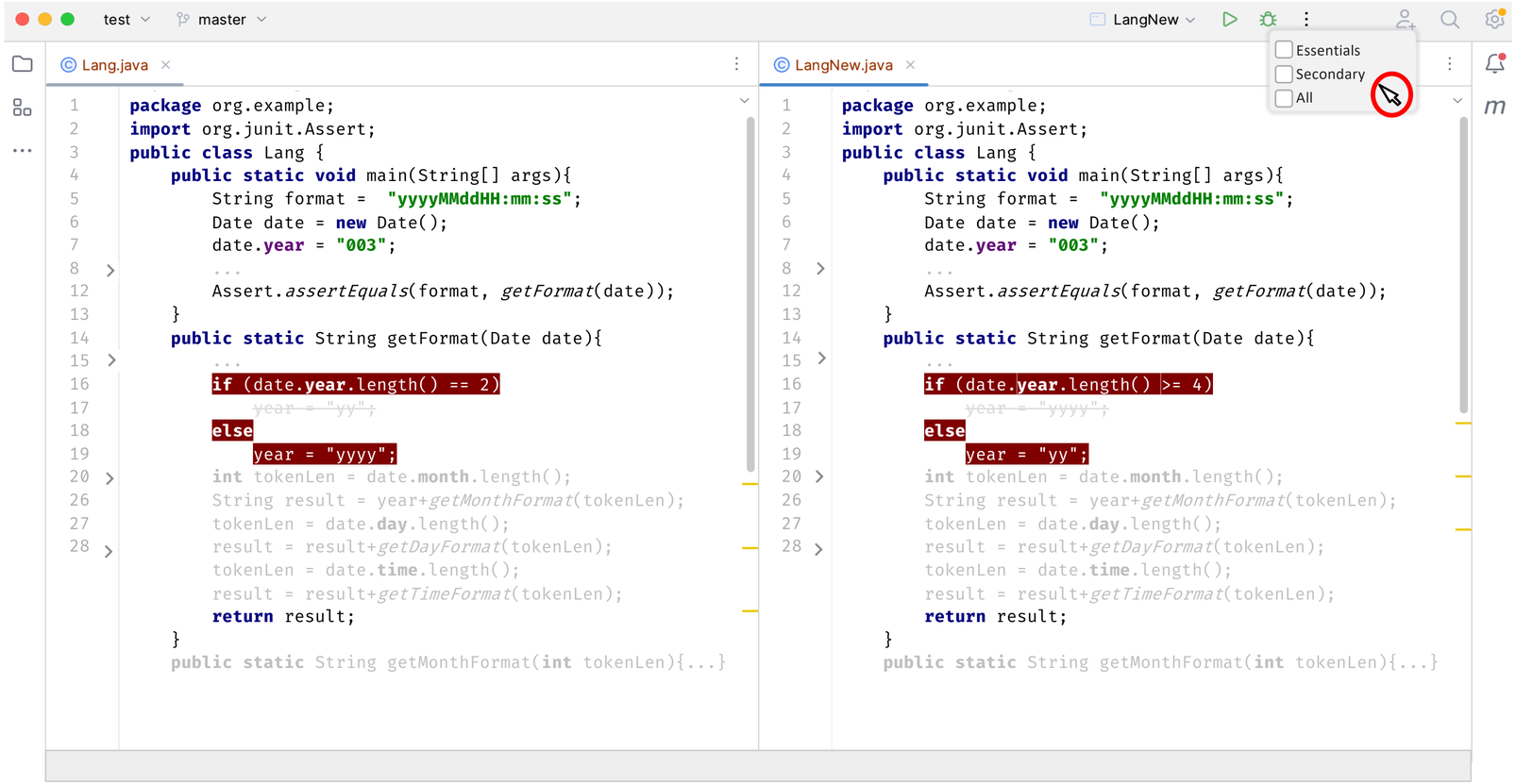
Debugging Essential Statements



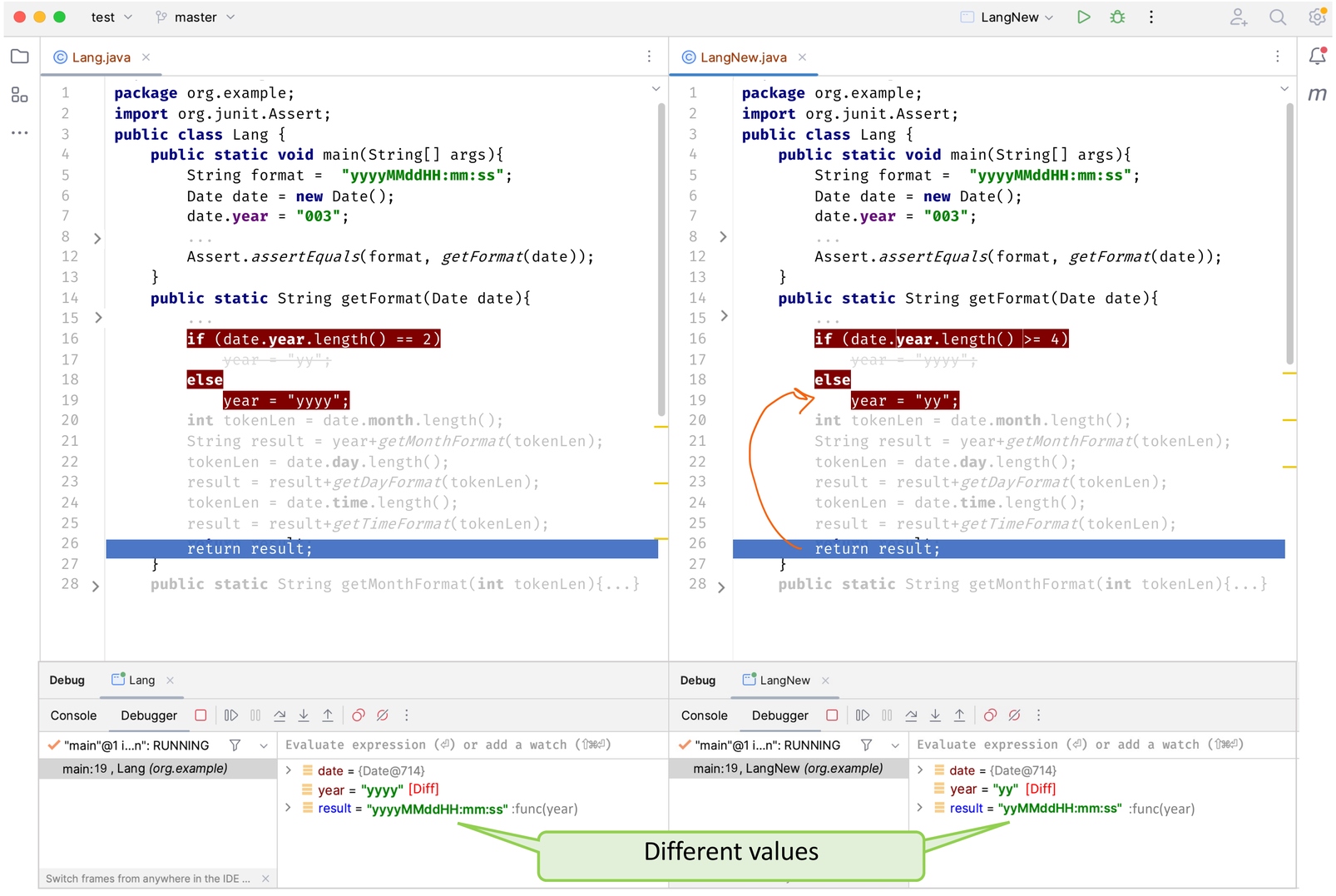
Debugging Essential Statements



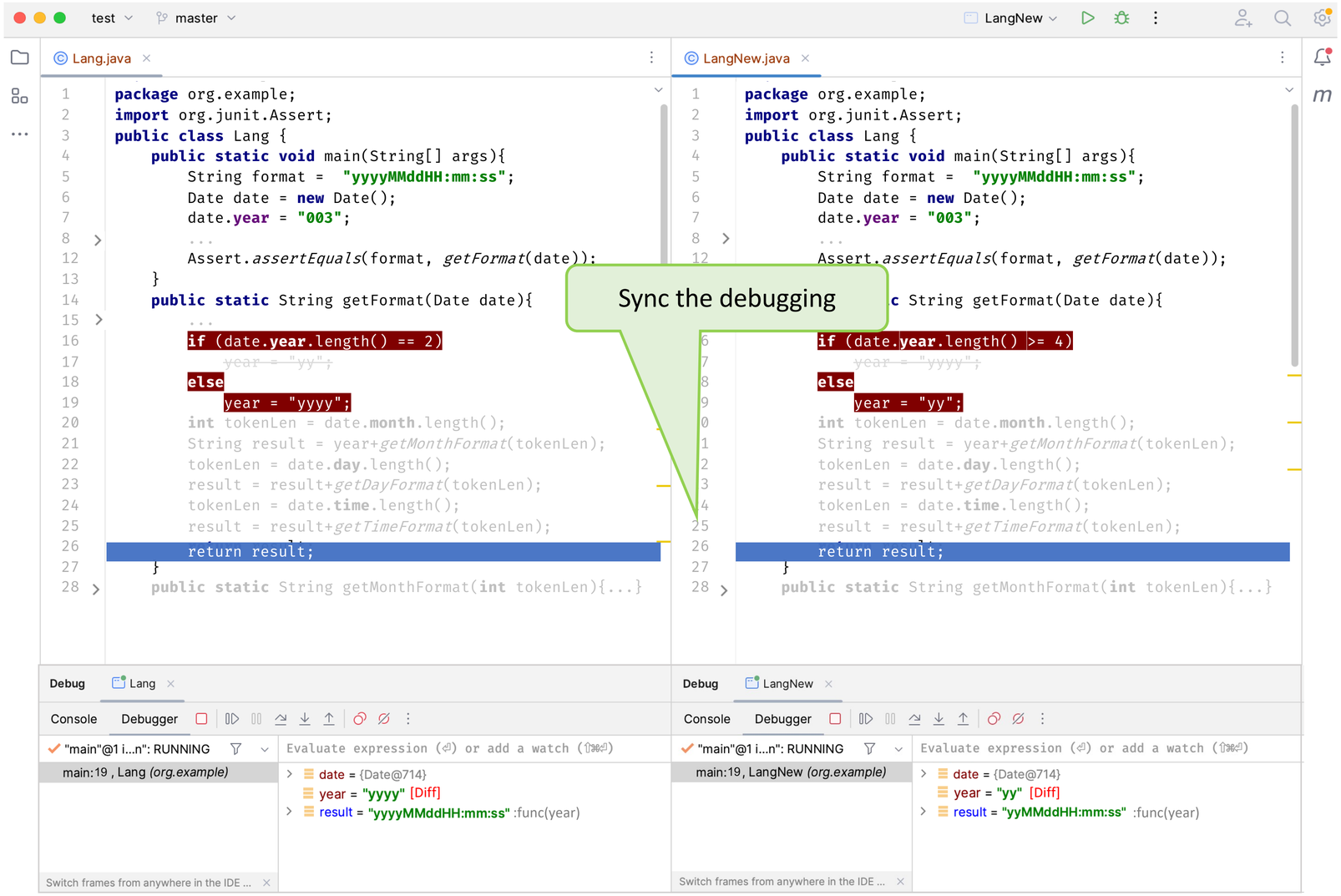
Selecting Which Statements to Debug on



Debugging Essential Statements



Side-by-Side Debugging



Textual Explanations