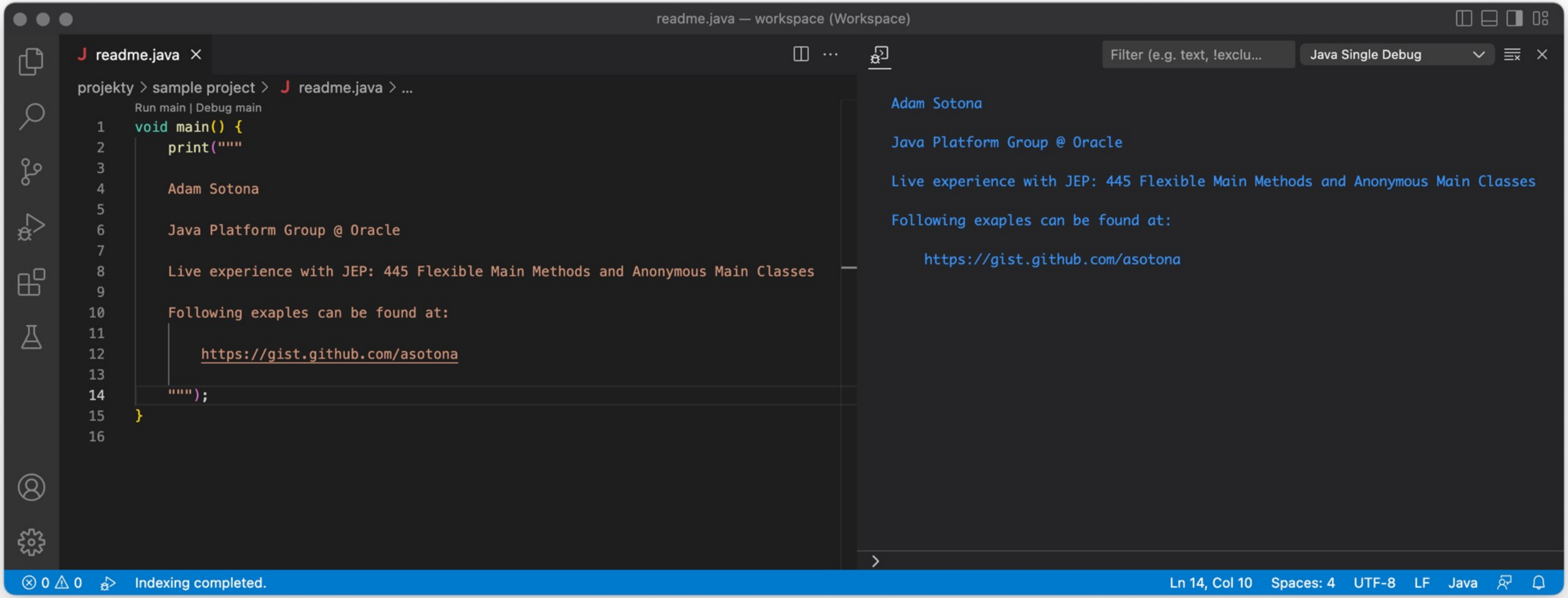
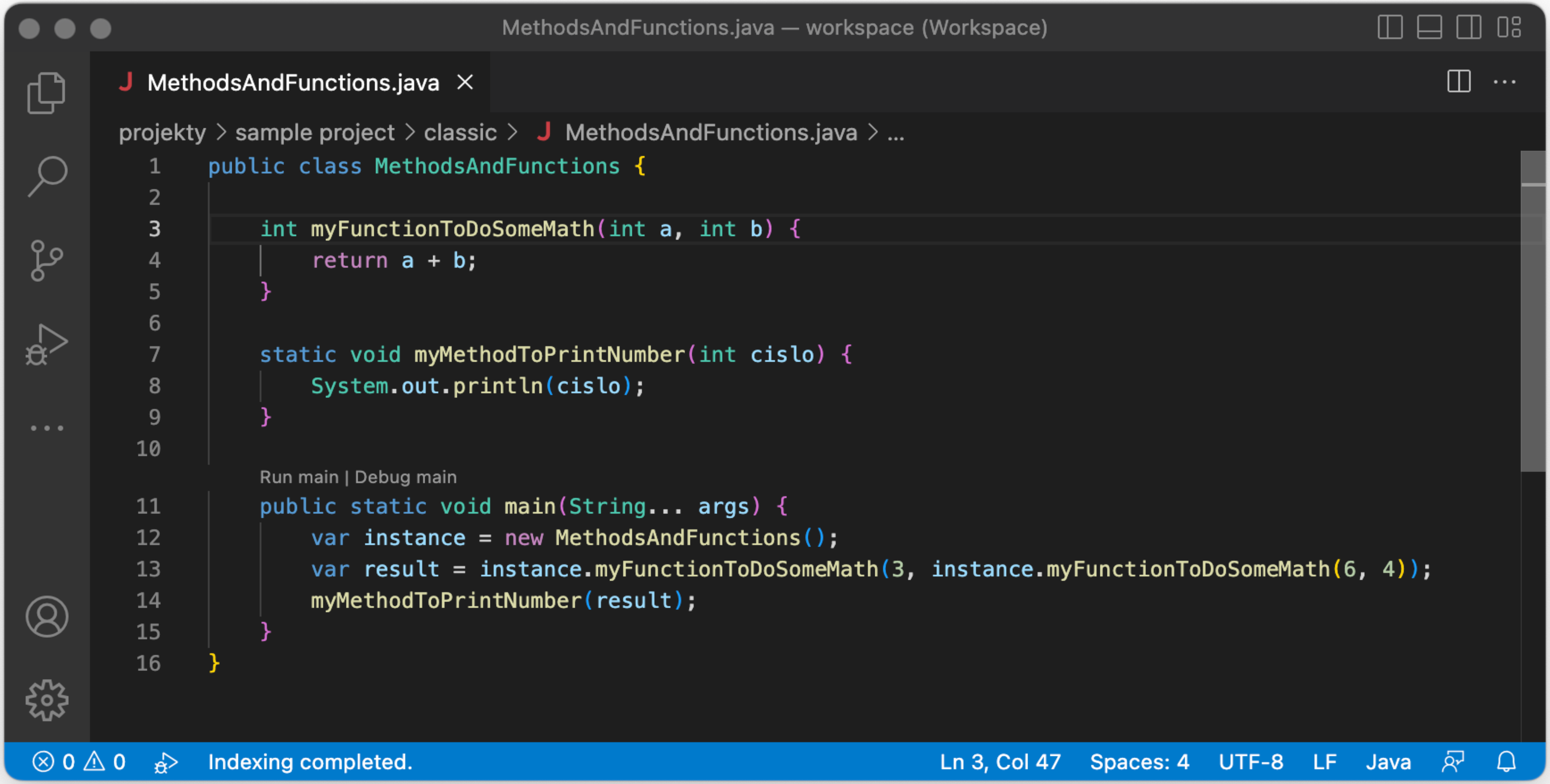
Demo by Adam Sotona



readme.java — workspace (Workspace)
readme.java
projekty > sample project > readme.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
Java Single Debug
0 0
Indexing completed.
Ln 14, Col 10
Spaces: 4
UTF-8
LF
Java

```java
void main() {
    print("""

    Adam Sotona

    Java Platform Group @ Oracle

    Live experience with JEP: 445 Flexible Main Methods and Anonymous Main Classes

    Following exaples can be found at:

        https://gist.github.com/asotona

    """);
}

```

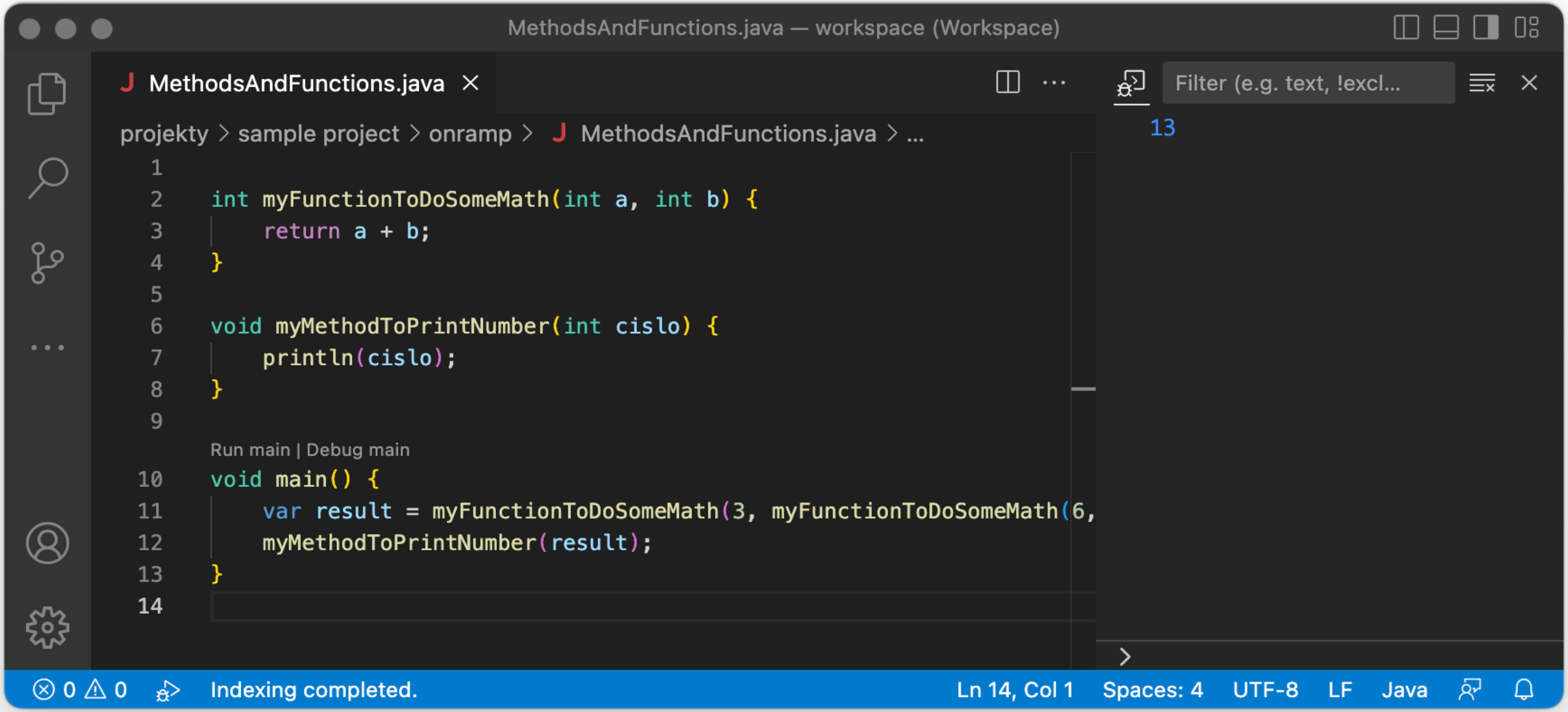
```
Adam Sotona

Java Platform Group @ Oracle

Live experience with JEP: 445 Flexible Main Methods and Anonymous Main Classes

Following exaples can be found at:

    https://gist.github.com/asotona
```



MethodsAndFunctions.java — workspace (Workspace)
MethodsAndFunctions.java
projekty > sample project > classic > MethodsAndFunctions.java > ...
Run main | Debug main
0 0
Ln 3, Col 47
Spaces: 4
UTF-8
LF
Java

```java
public class MethodsAndFunctions {

    int myFunctionToDoSomeMath(int a, int b) {
        return a + b;
    }

    static void myMethodToPrintNumber(int cislo) {
        System.out.println(cislo);
    }

    public static void main(String... args) {
        var instance = new MethodsAndFunctions();
        var result = instance.myFunctionToDoSomeMath(3, instance.myFunctionToDoSomeMath(6, 4));
        myMethodToPrintNumber(result);
    }
}
```

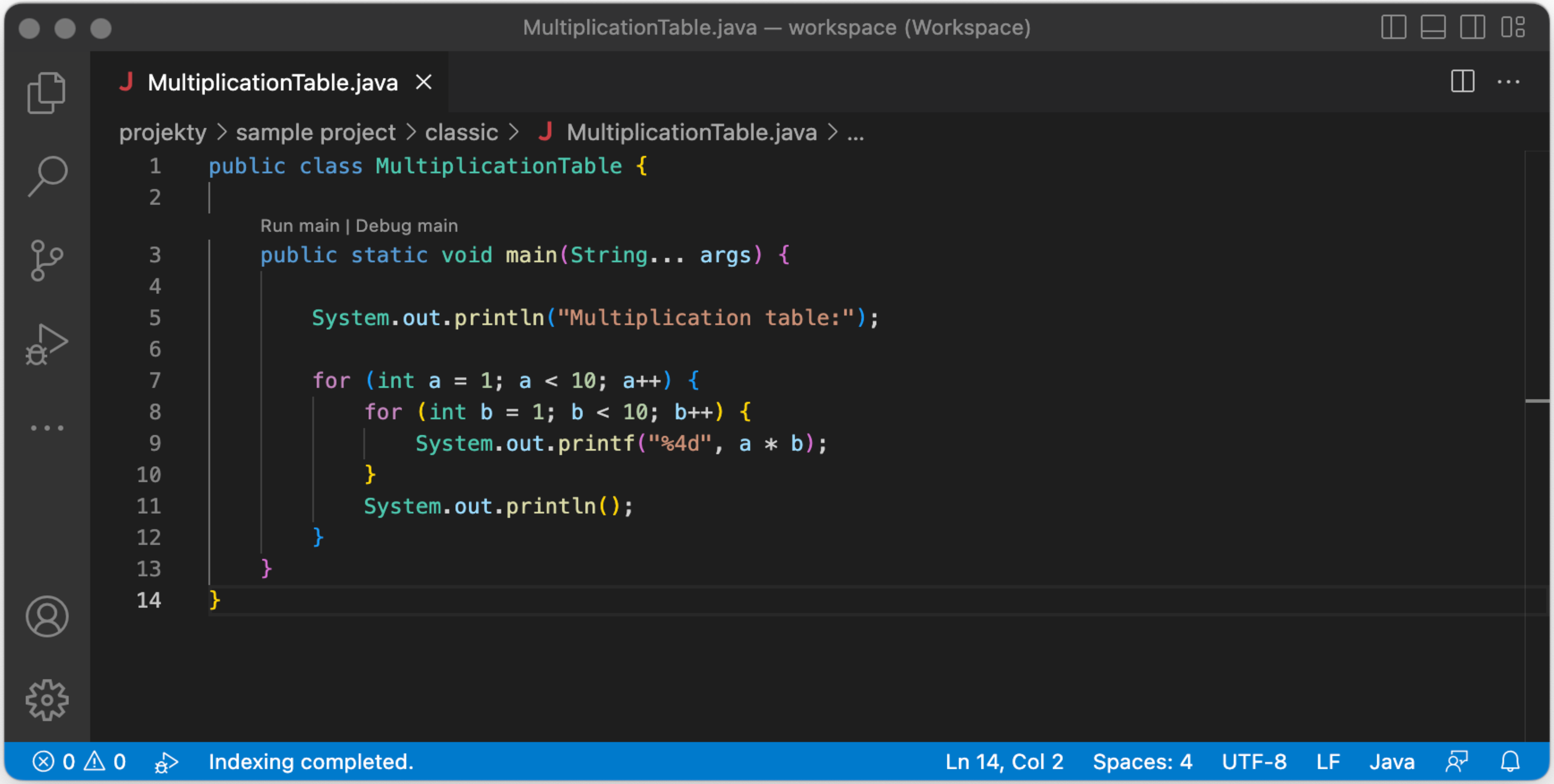
Indexing completed.



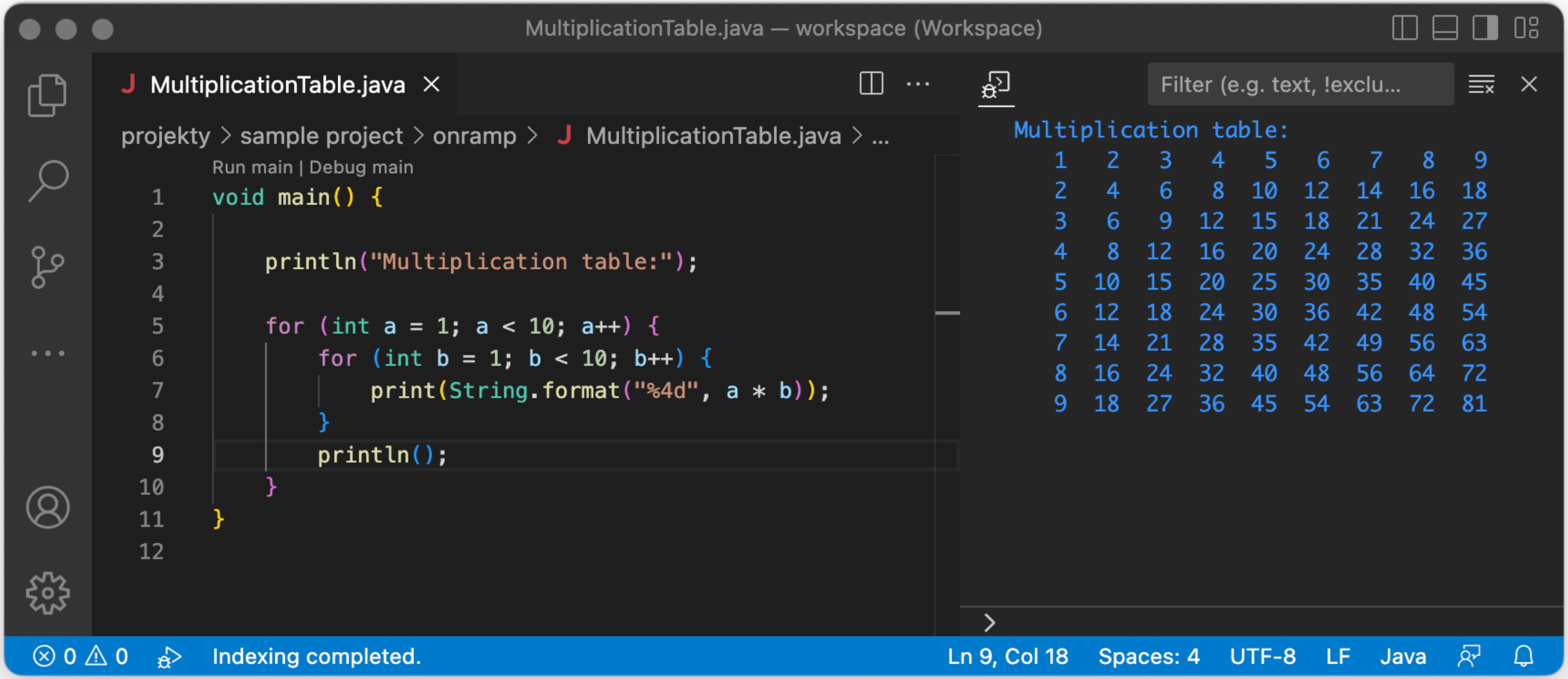
MethodsAndFunctions.java — workspace (Workspace)
MethodsAndFunctions.java
projekty > sample project > onramp > MethodsAndFunctions.java > ...
Filter (e.g. text, !excl...
Run main | Debug main
0 0
Indexing completed.
Ln 14, Col 1
Spaces: 4
UTF-8
LF
Java

```java

int myFunctionToDoSomeMath(int a, int b) {
    return a + b;
}

void myMethodToPrintNumber(int cislo) {
    println(cislo);
}

void main() {
    var result = myFunctionToDoSomeMath(3, myFunctionToDoSomeMath(6,
    myMethodToPrintNumber(result);
}

```

```
13
```

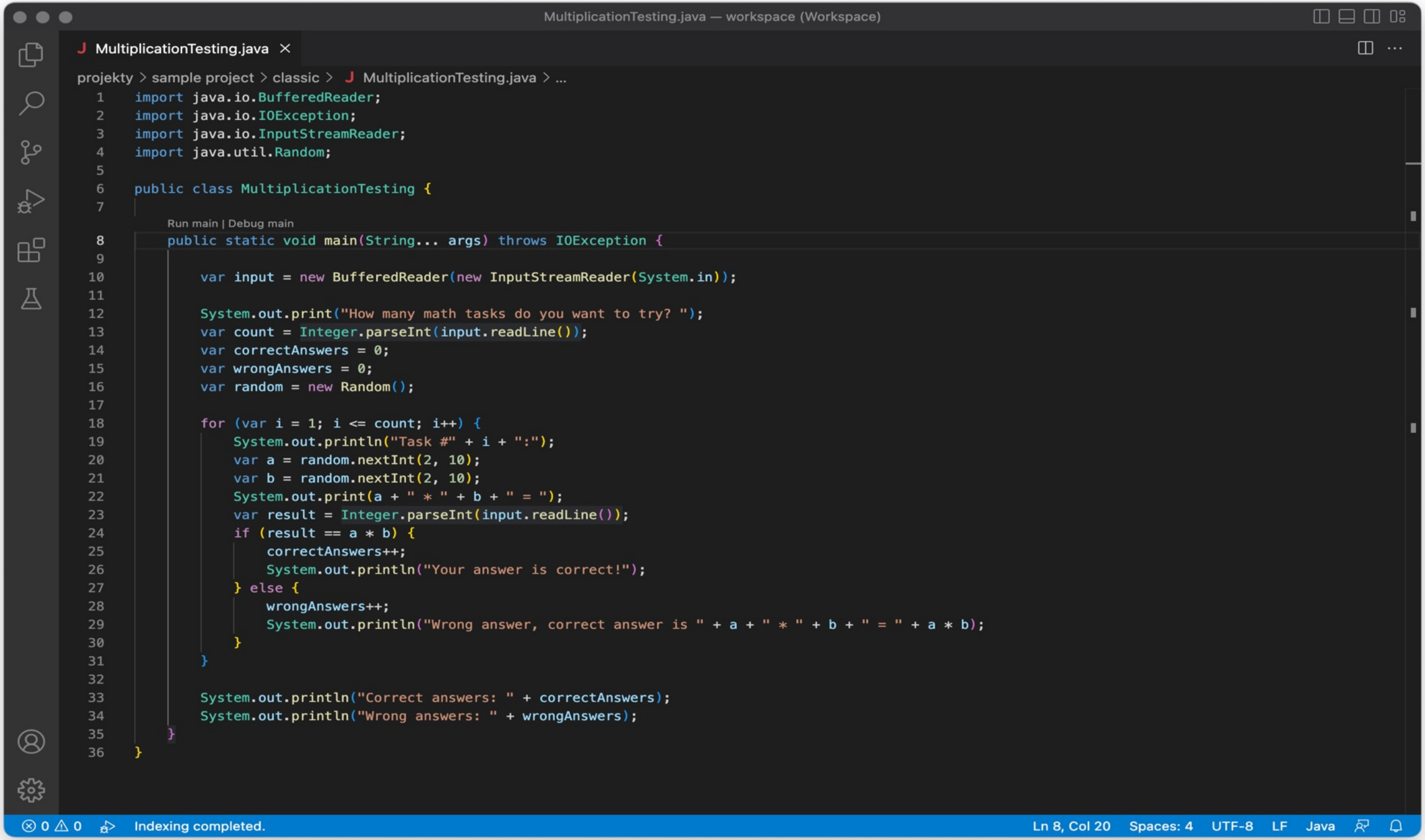

MultiplicationTable.java — workspace (Workspace)
MultiplicationTable.java
projekty > sample project > classic > MultiplicationTable.java > ...
Run main | Debug main
⊗ 0 ⚠ 0
Indexing completed.
Ln 14, Col 2
Spaces: 4
UTF-8
LF
Java

```java
public class MultiplicationTable {

    public static void main(String... args) {

        System.out.println("Multiplication table:");

        for (int a = 1; a < 10; a++) {
            for (int b = 1; b < 10; b++) {
                System.out.printf("%4d", a * b);
            }
            System.out.println();
        }
    }
}
```

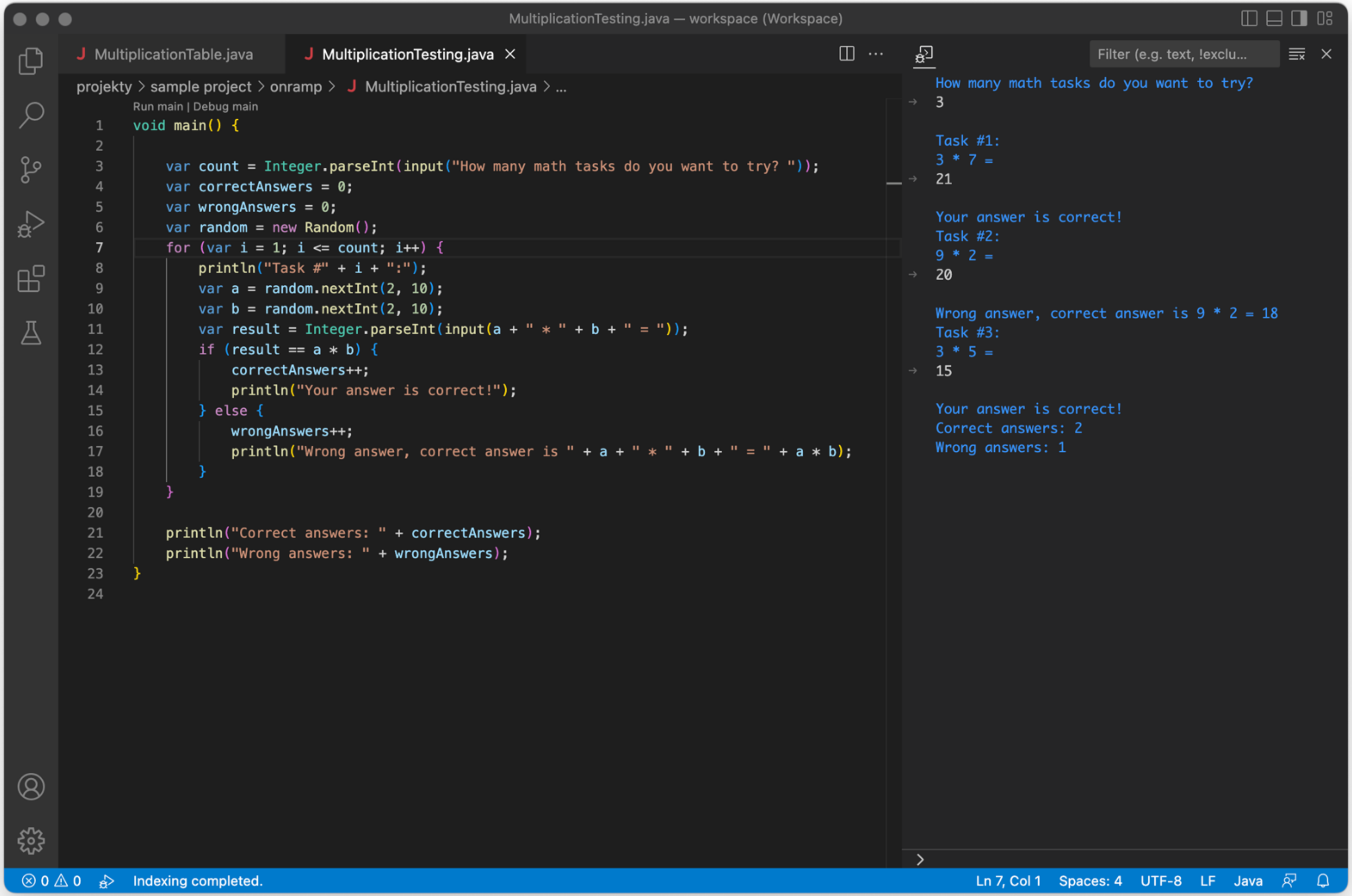

MultiplicationTable.java — workspace (Workspace)
MultiplicationTable.java
projekty > sample project > onramp > MultiplicationTable.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
0 0
Indexing completed.
Ln 9, Col 18
Spaces: 4
UTF-8
LF
Java

```java
void main() {

    println("Multiplication table:");

    for (int a = 1; a < 10; a++) {
        for (int b = 1; b < 10; b++) {
            print(String.format("%4d", a * b));
        }
        println();
    }
}
```

```
Multiplication table:
   1   2   3   4   5   6   7   8   9
   2   4   6   8  10  12  14  16  18
   3   6   9  12  15  18  21  24  27
   4   8  12  16  20  24  28  32  36
   5  10  15  20  25  30  35  40  45
   6  12  18  24  30  36  42  48  54
   7  14  21  28  35  42  49  56  63
   8  16  24  32  40  48  56  64  72
   9  18  27  36  45  54  63  72  81
```


MultiplicationTesting.java — workspace (Workspace)
MultiplicationTesting.java
projekty > sample project > classic > MultiplicationTesting.java > ...
Run main | Debug main
0 0
Indexing completed.
Ln 8, Col 20
Spaces: 4
UTF-8
LF
Java

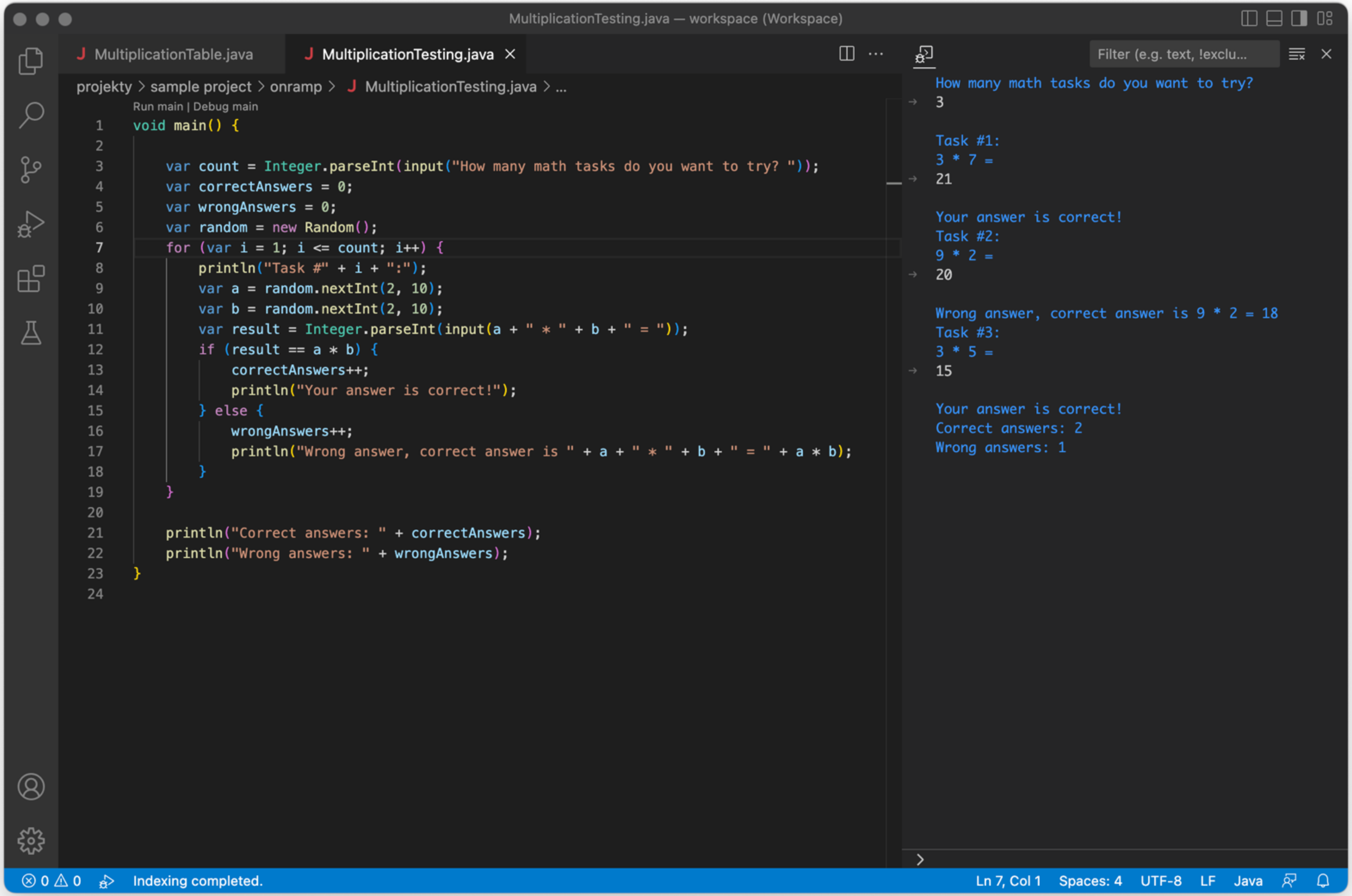
```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Random;

public class MultiplicationTesting {

    public static void main(String... args) throws IOException {

        var input = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("How many math tasks do you want to try? ");
        var count = Integer.parseInt(input.readLine());
        var correctAnswers = 0;
        var wrongAnswers = 0;
        var random = new Random();

        for (var i = 1; i <= count; i++) {
            System.out.println("Task #" + i + ":");
            var a = random.nextInt(2, 10);
            var b = random.nextInt(2, 10);
            System.out.print(a + " * " + b + " = ");
            var result = Integer.parseInt(input.readLine());
            if (result == a * b) {
                correctAnswers++;
                System.out.println("Your answer is correct!");
            } else {
                wrongAnswers++;
                System.out.println("Wrong answer, correct answer is " + a + " * " + b + " = " + a * b);
            }
        }

        System.out.println("Correct answers: " + correctAnswers);
        System.out.println("Wrong answers: " + wrongAnswers);
    }
}
```



MultiplicationTesting.java — workspace (Workspace)
MultiplicationTable.java
MultiplicationTesting.java
projekty > sample project > onramp > MultiplicationTesting.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
Indexing completed.
Ln 7, Col 1
Spaces: 4
UTF-8
LF
Java

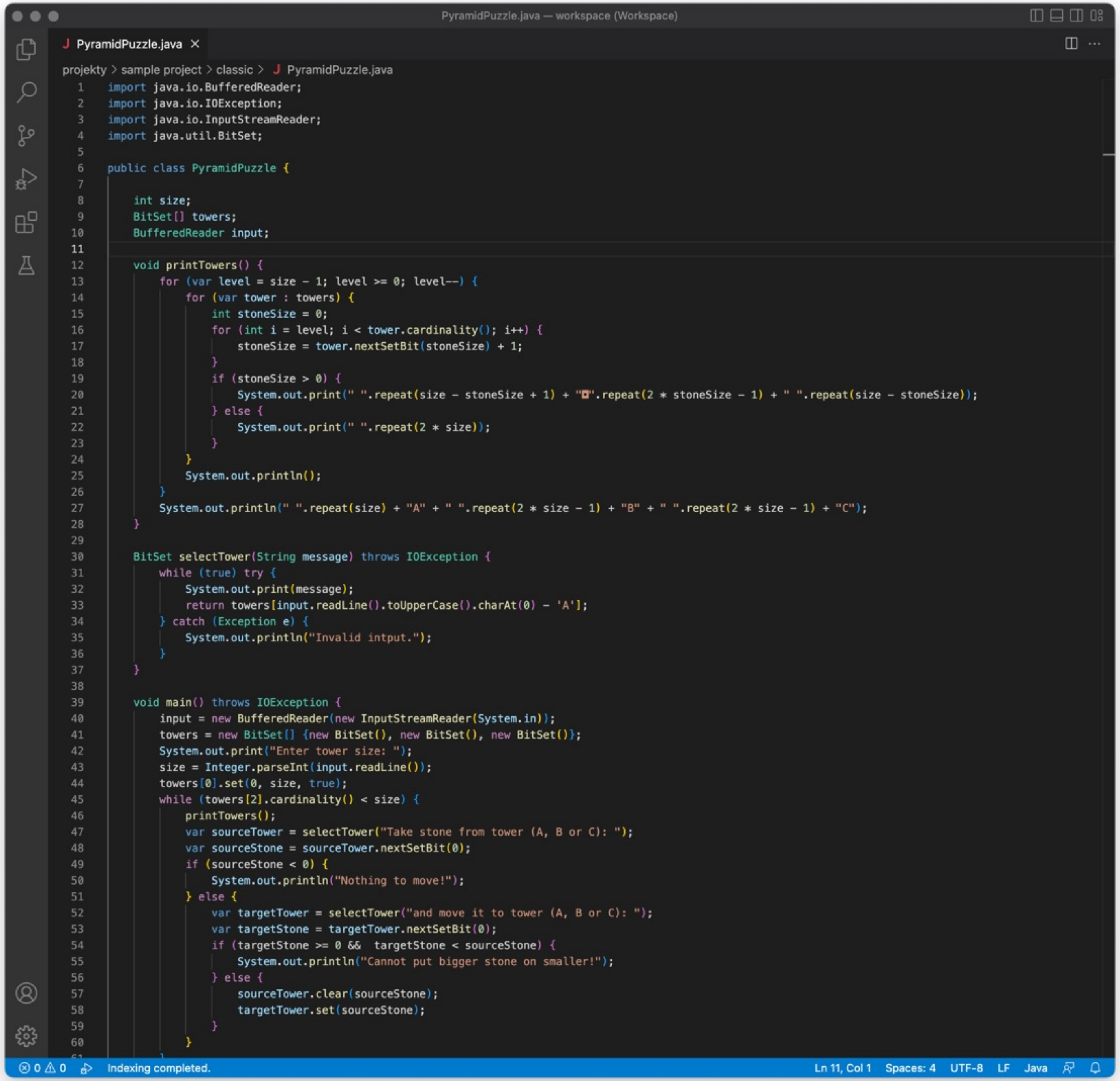
```java
void main() {

    var count = Integer.parseInt(input("How many math tasks do you want to try? "));
    var correctAnswers = 0;
    var wrongAnswers = 0;
    var random = new Random();
    for (var i = 1; i <= count; i++) {
        println("Task #" + i + ":");
        var a = random.nextInt(2, 10);
        var b = random.nextInt(2, 10);
        var result = Integer.parseInt(input(a + " * " + b + " = "));
        if (result == a * b) {
            correctAnswers++;
            println("Your answer is correct!");
        } else {
            wrongAnswers++;
            println("Wrong answer, correct answer is " + a + " * " + b + " = " + a * b);
        }
    }

    println("Correct answers: " + correctAnswers);
    println("Wrong answers: " + wrongAnswers);
}

```

```
How many math tasks do you want to try?
3

Task #1:
3 * 7 =
21

Your answer is correct!
Task #2:
9 * 2 =
20

Wrong answer, correct answer is 9 * 2 = 18
Task #3:
3 * 5 =
15

Your answer is correct!
Correct answers: 2
Wrong answers: 1
```


MultiplicationTesting.java — workspace (Workspace)
MultiplicationTable.java
MultiplicationTesting.java
projekty > sample project > onramp > MultiplicationTesting.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
Indexing completed.
Ln 7, Col 1
Spaces: 4
UTF-8
LF
Java

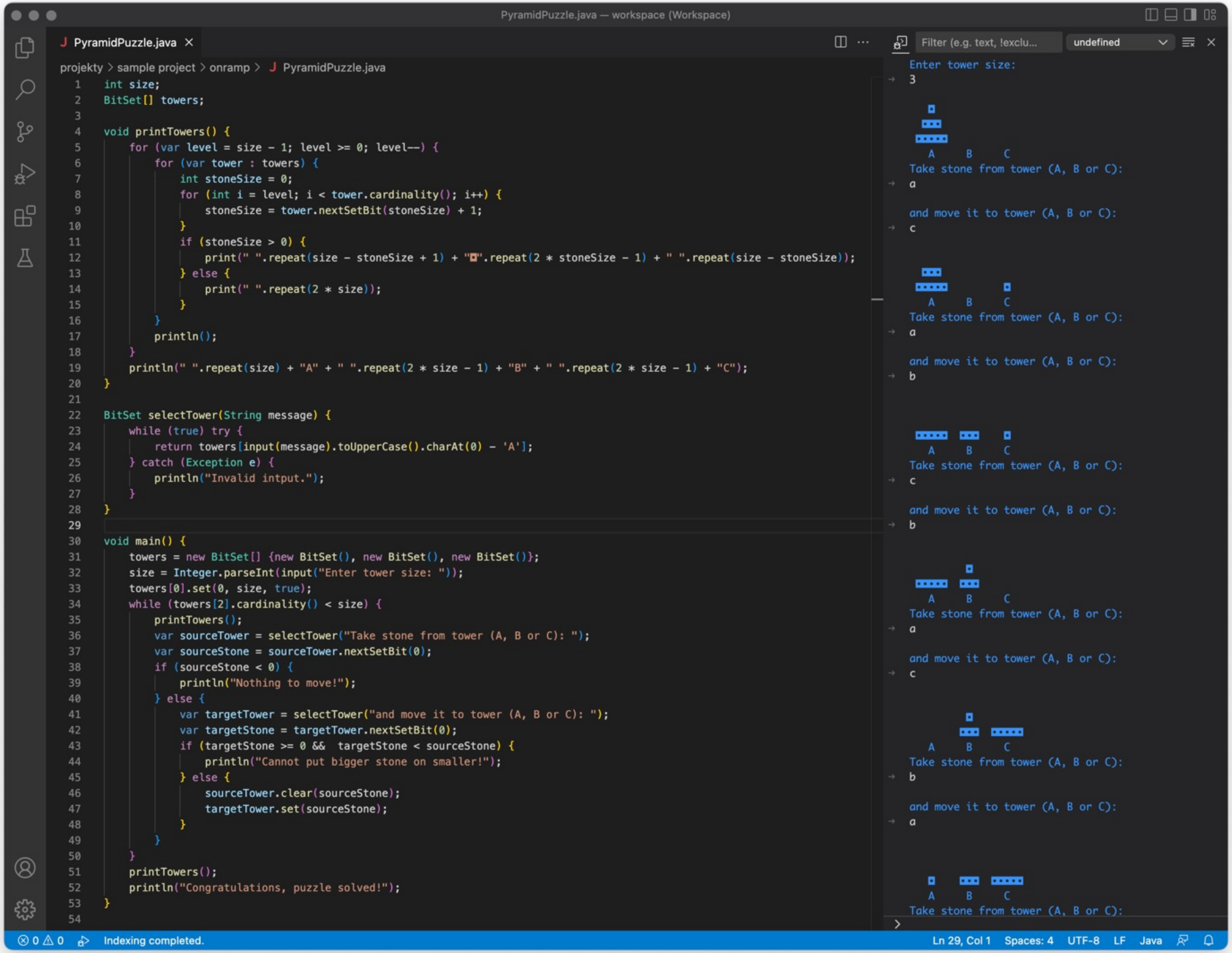
```java
void main() {

    var count = Integer.parseInt(input("How many math tasks do you want to try? "));
    var correctAnswers = 0;
    var wrongAnswers = 0;
    var random = new Random();
    for (var i = 1; i <= count; i++) {
        println("Task #" + i + ":");
        var a = random.nextInt(2, 10);
        var b = random.nextInt(2, 10);
        var result = Integer.parseInt(input(a + " * " + b + " = "));
        if (result == a * b) {
            correctAnswers++;
            println("Your answer is correct!");
        } else {
            wrongAnswers++;
            println("Wrong answer, correct answer is " + a + " * " + b + " = " + a * b);
        }
    }

    println("Correct answers: " + correctAnswers);
    println("Wrong answers: " + wrongAnswers);
}

```

```
How many math tasks do you want to try?
3

Task #1:
3 * 7 =
21

Your answer is correct!
Task #2:
9 * 2 =
20

Wrong answer, correct answer is 9 * 2 = 18
Task #3:
3 * 5 =
15

Your answer is correct!
Correct answers: 2
Wrong answers: 1
```



PyramidPuzzle.java — workspace (Workspace)
PyramidPuzzle.java
projekty > sample project > classic > PyramidPuzzle.java
0 0
Indexing completed.
Ln 11, Col 1
Spaces: 4
UTF-8
LF
Java

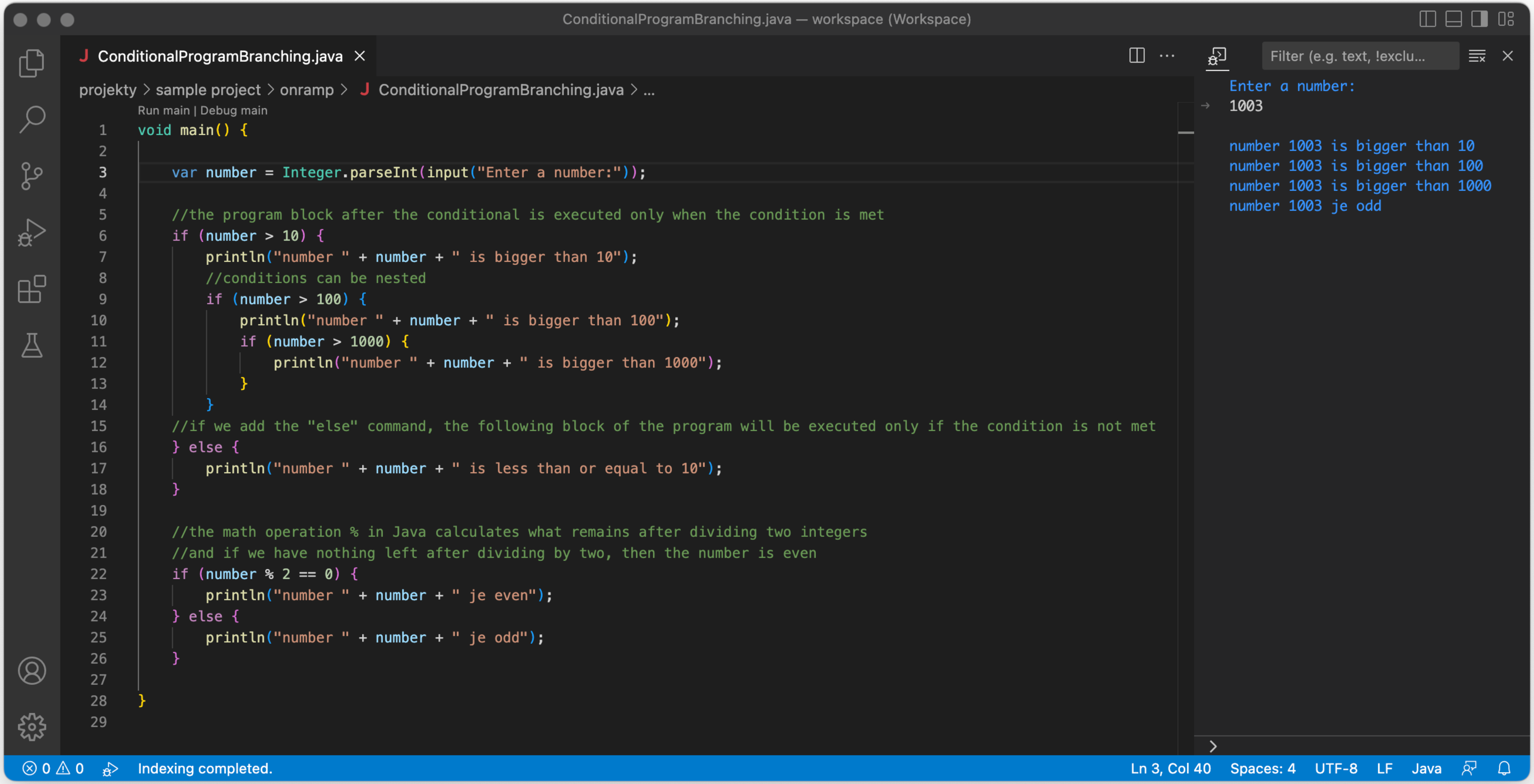
```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.BitSet;

public class PyramidPuzzle {

    int size;
    BitSet[] towers;
    BufferedReader input;

    void printTowers() {
        for (var level = size - 1; level >= 0; level--) {
            for (var tower : towers) {
                int stoneSize = 0;
                for (int i = level; i < tower.cardinality(); i++) {
                    stoneSize = tower.nextSetBit(stoneSize) + 1;
                }
                if (stoneSize > 0) {
                    System.out.print(" ".repeat(size - stoneSize + 1) + "■".repeat(2 * stoneSize - 1) + " ".repeat(size - stoneSize));
                } else {
                    System.out.print(" ".repeat(2 * size));
                }
            }
            System.out.println();
        }
        System.out.println(" ".repeat(size) + "A" + " ".repeat(2 * size - 1) + "B" + " ".repeat(2 * size - 1) + "C");
    }

    BitSet selectTower(String message) throws IOException {
        while (true) try {
            System.out.print(message);
            return towers[input.readLine().toUpperCase().charAt(0) - 'A'];
        } catch (Exception e) {
            System.out.println("Invalid intput.");
        }
    }

    void main() throws IOException {
        input = new BufferedReader(new InputStreamReader(System.in));
        towers = new BitSet[] {new BitSet(), new BitSet(), new BitSet()};
        System.out.print("Enter tower size: ");
        size = Integer.parseInt(input.readLine());
        towers[0].set(0, size, true);
        while (towers[2].cardinality() < size) {
            printTowers();
            var sourceTower = selectTower("Take stone from tower (A, B or C): ");
            var sourceStone = sourceTower.nextSetBit(0);
            if (sourceStone < 0) {
                System.out.println("Nothing to move!");
            } else {
                var targetTower = selectTower("and move it to tower (A, B or C): ");
                var targetStone = targetTower.nextSetBit(0);
                if (targetStone >= 0 &&  targetStone < sourceStone) {
                    System.out.println("Cannot put bigger stone on smaller!");
                } else {
                    sourceTower.clear(sourceStone);
                    targetTower.set(sourceStone);
                }
            }
```



PyramidPuzzle.java — workspace (Workspace)
PyramidPuzzle.java
projekty > sample project > onramp > PyramidPuzzle.java
Filter (e.g. text, !exclu...
undefined
0 0
Indexing completed.
Ln 29, Col 1
Spaces: 4
UTF-8
LF
Java

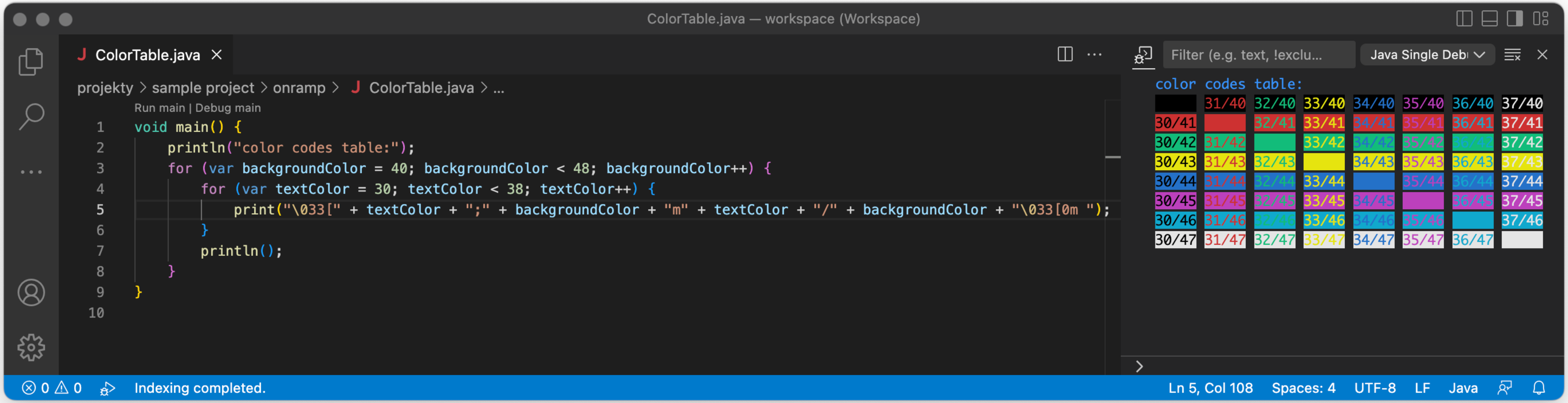
```java
int size;
BitSet[] towers;

void printTowers() {
    for (var level = size - 1; level >= 0; level--) {
        for (var tower : towers) {
            int stoneSize = 0;
            for (int i = level; i < tower.cardinality(); i++) {
                stoneSize = tower.nextSetBit(stoneSize) + 1;
            }
            if (stoneSize > 0) {
                print(" ".repeat(size - stoneSize + 1) + "◘".repeat(2 * stoneSize - 1) + " ".repeat(size - stoneSize));
            } else {
                print(" ".repeat(2 * size));
            }
        }
        println();
    }
    println(" ".repeat(size) + "A" + " ".repeat(2 * size - 1) + "B" + " ".repeat(2 * size - 1) + "C");
}

BitSet selectTower(String message) {
    while (true) try {
        return towers[input(message).toUpperCase().charAt(0) - 'A'];
    } catch (Exception e) {
        println("Invalid intput.");
    }
}

void main() {
    towers = new BitSet[] {new BitSet(), new BitSet(), new BitSet()};
    size = Integer.parseInt(input("Enter tower size: "));
    towers[0].set(0, size, true);
    while (towers[2].cardinality() < size) {
        printTowers();
        var sourceTower = selectTower("Take stone from tower (A, B or C): ");
        var sourceStone = sourceTower.nextSetBit(0);
        if (sourceStone < 0) {
            println("Nothing to move!");
        } else {
            var targetTower = selectTower("and move it to tower (A, B or C): ");
            var targetStone = targetTower.nextSetBit(0);
            if (targetStone >= 0 &&  targetStone < sourceStone) {
                println("Cannot put bigger stone on smaller!");
            } else {
                sourceTower.clear(sourceStone);
                targetTower.set(sourceStone);
            }
        }
    }
    printTowers();
    println("Congratulations, puzzle solved!");
}
```

```
Enter tower size:
3

   ◘
  ◘◘◘
 ◘◘◘◘◘
   A     B     C
Take stone from tower (A, B or C):
a

and move it to tower (A, B or C):
c


  ◘◘◘
 ◘◘◘◘◘         ◘
   A     B     C
Take stone from tower (A, B or C):
a

and move it to tower (A, B or C):
b



 ◘◘◘◘◘  ◘◘◘    ◘
   A     B     C
Take stone from tower (A, B or C):
c

and move it to tower (A, B or C):
b


         ◘
 ◘◘◘◘◘  ◘◘◘
   A     B     C
Take stone from tower (A, B or C):
a

and move it to tower (A, B or C):
c


         ◘
        ◘◘◘  ◘◘◘◘◘
   A     B     C
Take stone from tower (A, B or C):
b

and move it to tower (A, B or C):
a



   ◘    ◘◘◘  ◘◘◘◘◘
   A     B     C
Take stone from tower (A, B or C):
```


ConditionalProgramBranching.java — workspace (Workspace)
ConditionalProgramBranching.java
projekty > sample project > onramp > ConditionalProgramBranching.java > ...
Filter (e.g. text, !exclu...
Run main | Debug main
0 0
Indexing completed.
Ln 3, Col 40
Spaces: 4
UTF-8
LF
Java

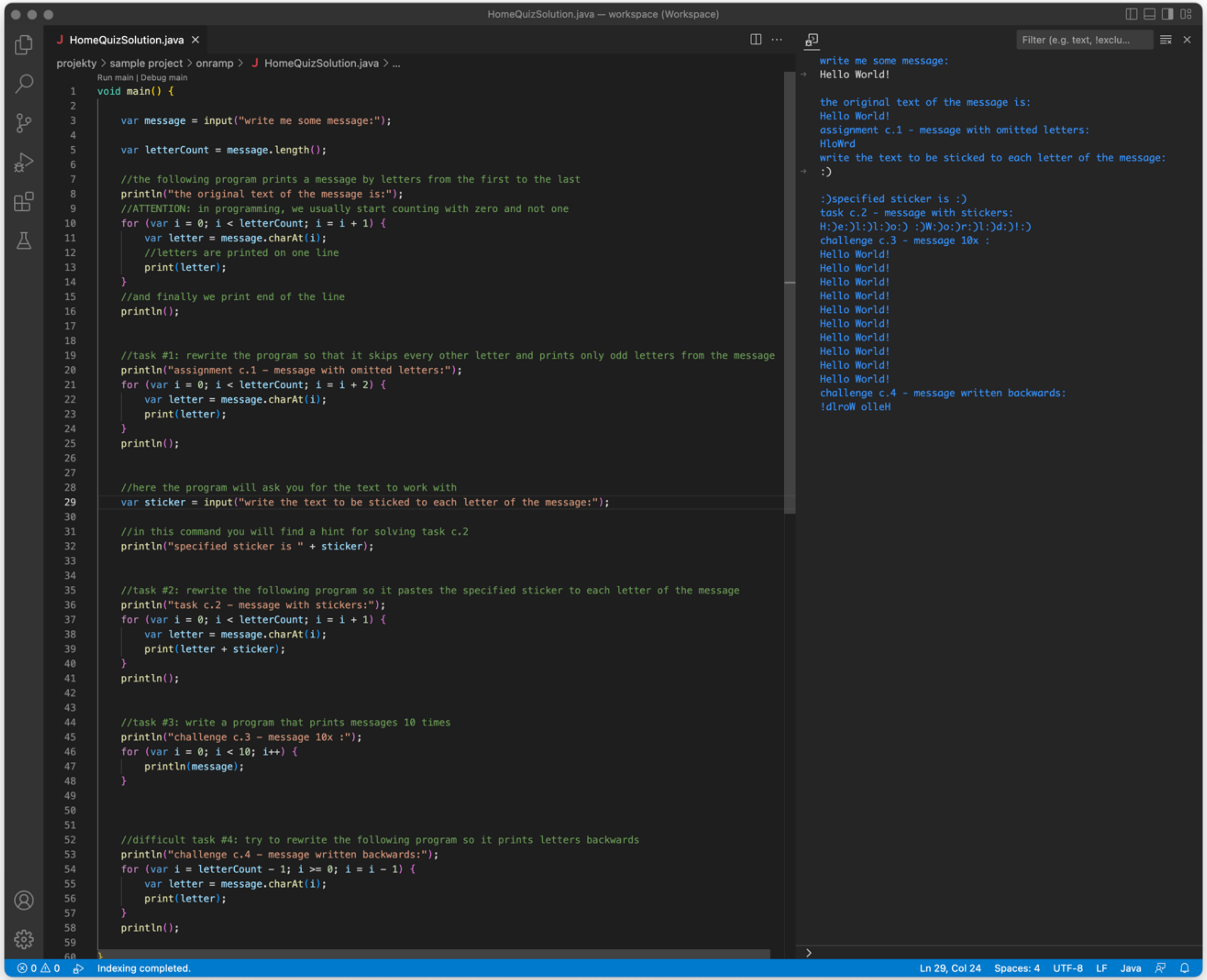
```java
void main() {

    var number = Integer.parseInt(input("Enter a number:"));

    //the program block after the conditional is executed only when the condition is met
    if (number > 10) {
        println("number " + number + " is bigger than 10");
        //conditions can be nested
        if (number > 100) {
            println("number " + number + " is bigger than 100");
            if (number > 1000) {
                println("number " + number + " is bigger than 1000");
            }
        }
    //if we add the "else" command, the following block of the program will be executed only if the condition is not met
    } else {
        println("number " + number + " is less than or equal to 10");
    }

    //the math operation % in Java calculates what remains after dividing two integers
    //and if we have nothing left after dividing by two, then the number is even
    if (number % 2 == 0) {
        println("number " + number + " je even");
    } else {
        println("number " + number + " je odd");
    }

}
```

```
Enter a number:
1003

number 1003 is bigger than 10
number 1003 is bigger than 100
number 1003 is bigger than 1000
number 1003 je odd
```


ColorTable.java — workspace (Workspace)
ColorTable.java
projekty > sample project > onramp > ColorTable.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
Java Single Debu
0 0
Indexing completed.
Ln 5, Col 108
Spaces: 4
UTF-8
LF
Java

```
void main() {
    println("color codes table:");
    for (var backgroundColor = 40; backgroundColor < 48; backgroundColor++) {
        for (var textColor = 30; textColor < 38; textColor++) {
            print("\033[" + textColor + ";" + backgroundColor + "m" + textColor + "/" + backgroundColor + "\033[0m ");
        }
        println();
    }
}
```

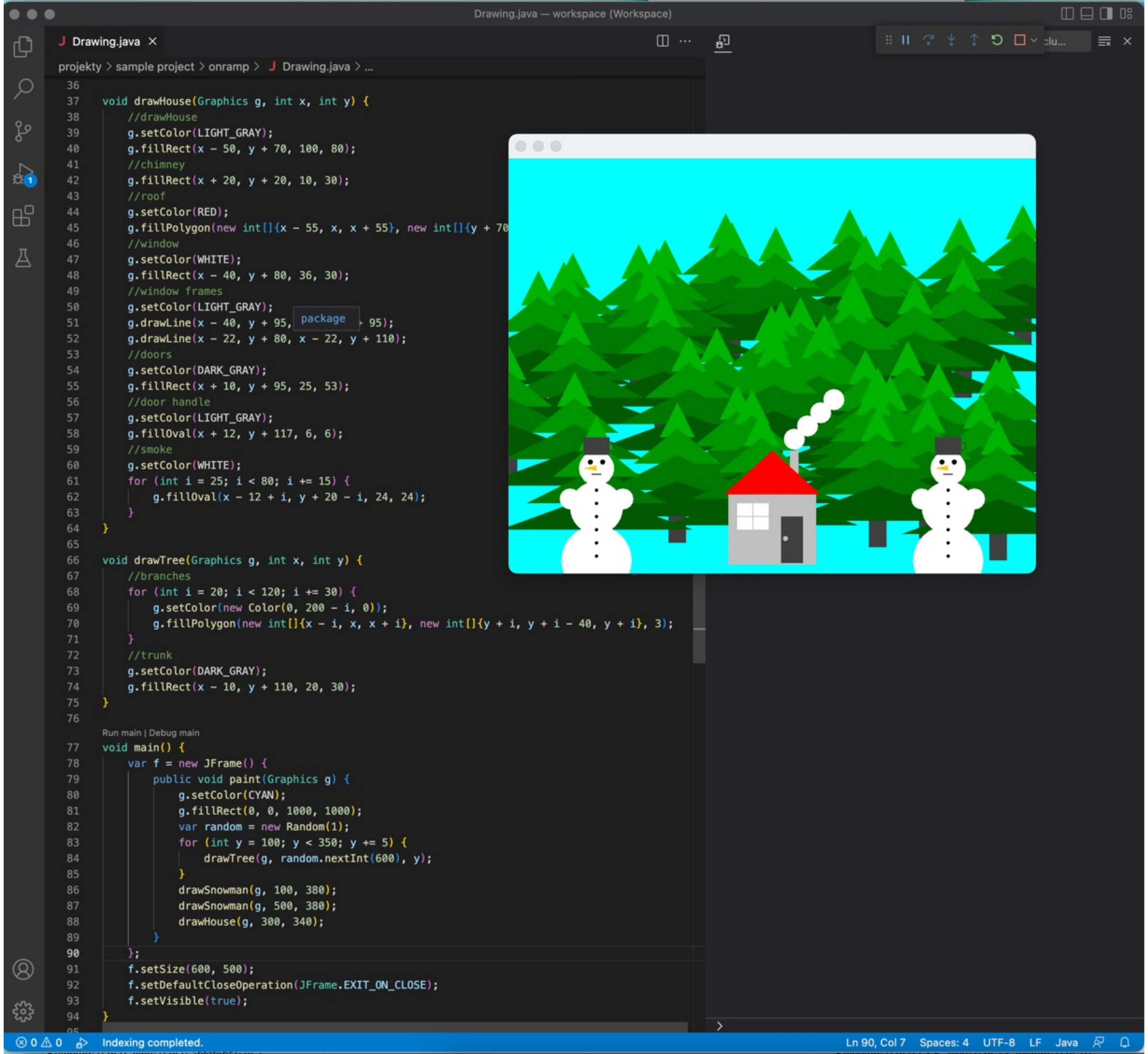
```
color codes table:
      31/40 32/40 33/40 34/40 35/40 36/40 37/40
30/41       32/41 33/41 34/41 35/41 36/41 37/41
30/42 31/42       33/42 34/42 35/42 36/42 37/42
30/43 31/43 32/43       34/43 35/43 36/43 37/43
30/44 31/44 32/44 33/44       35/44 36/44 37/44
30/45 31/45 32/45 33/45 34/45       36/45 37/45
30/46 31/46 32/46 33/46 34/46 35/46       37/46
30/47 31/47 32/47 33/47 34/47 35/47 36/47
```



HomeQuizSolution.java — workspace (Workspace)
HomeQuizSolution.java
projekty > sample project > onramp > HomeQuizSolution.java > ...
Run main | Debug main
Filter (e.g. text, !exclu...
0 0
Indexing completed.
Ln 29, Col 24
Spaces: 4
UTF-8
LF
Java

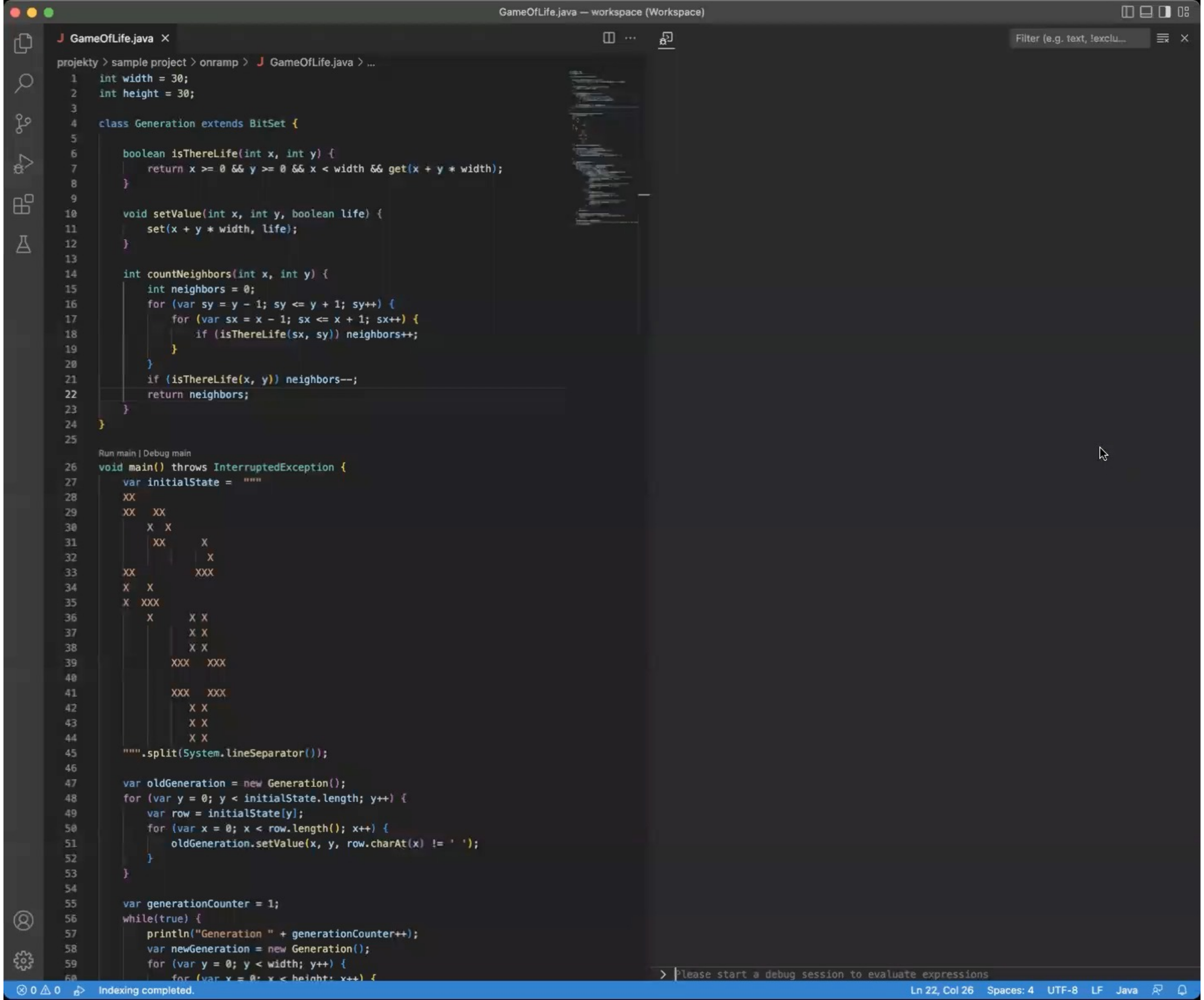
```java
void main() {

    var message = input("write me some message:");

    var letterCount = message.length();

    //the following program prints a message by letters from the first to the last
    println("the original text of the message is:");
    //ATTENTION: in programming, we usually start counting with zero and not one
    for (var i = 0; i < letterCount; i = i + 1) {
        var letter = message.charAt(i);
        //letters are printed on one line
        print(letter);
    }
    //and finally we print end of the line
    println();


    //task #1: rewrite the program so that it skips every other letter and prints only odd letters from the message
    println("assignment c.1 - message with omitted letters:");
    for (var i = 0; i < letterCount; i = i + 2) {
        var letter = message.charAt(i);
        print(letter);
    }
    println();


    //here the program will ask you for the text to work with
    var sticker = input("write the text to be sticked to each letter of the message:");

    //in this command you will find a hint for solving task c.2
    println("specified sticker is " + sticker);


    //task #2: rewrite the following program so it pastes the specified sticker to each letter of the message
    println("task c.2 - message with stickers:");
    for (var i = 0; i < letterCount; i = i + 1) {
        var letter = message.charAt(i);
        print(letter + sticker);
    }
    println();


    //task #3: write a program that prints messages 10 times
    println("challenge c.3 - message 10x :");
    for (var i = 0; i < 10; i++) {
        println(message);
    }



    //difficult task #4: try to rewrite the following program so it prints letters backwards
    println("challenge c.4 - message written backwards:");
    for (var i = letterCount - 1; i >= 0; i = i - 1) {
        var letter = message.charAt(i);
        print(letter);
    }
    println();

}
```

```
write me some message:
Hello World!

the original text of the message is:
Hello World!
assignment c.1 - message with omitted letters:
HloWrd
write the text to be sticked to each letter of the message:
:)

:)specified sticker is :)
task c.2 - message with stickers:
H:)e:)l:)l:)o:) :)W:)o:)r:)l:)d:)!:)
challenge c.3 - message 10x :
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
challenge c.4 - message written backwards:
!dlroW olleH
```

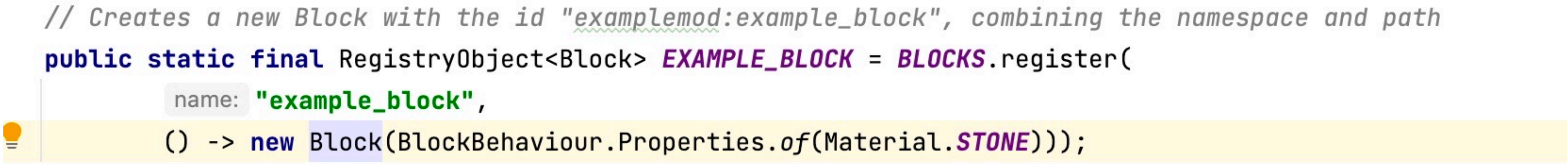


Drawing.java — workspace (Workspace)
Drawing.java
projekty > sample project > onramp > Drawing.java > ...
package
clu...
⊗ 0 ⚠ 0
Indexing completed.
Ln 90, Col 7
Spaces: 4
UTF-8
LF
Java

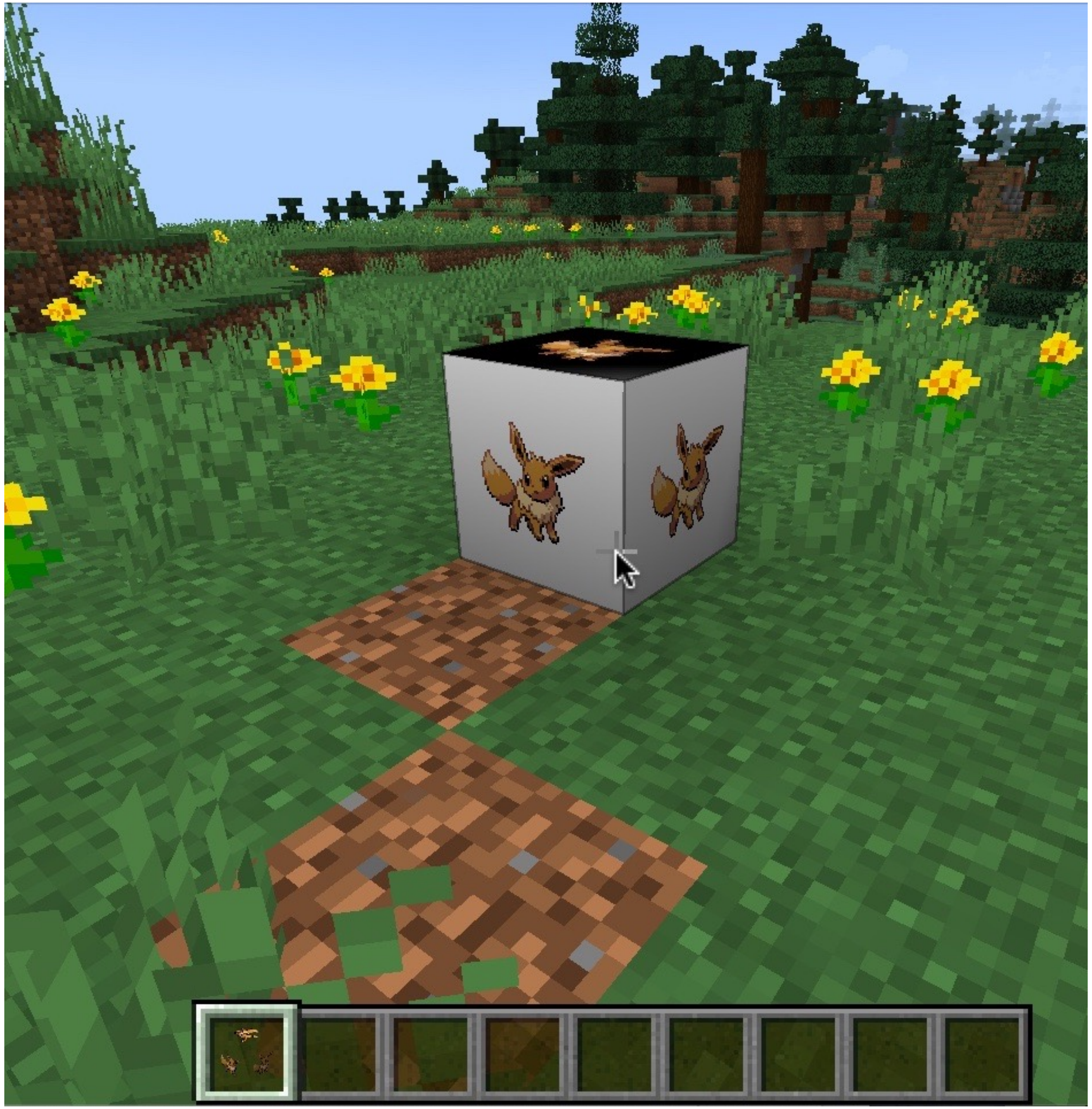
```

void drawHouse(Graphics g, int x, int y) {
    //drawHouse
    g.setColor(LIGHT_GRAY);
    g.fillRect(x - 50, y + 70, 100, 80);
    //chimney
    g.fillRect(x + 20, y + 20, 10, 30);
    //roof
    g.setColor(RED);
    g.fillPolygon(new int[]{x - 55, x, x + 55}, new int[]{y + 70
    //window
    g.setColor(WHITE);
    g.fillRect(x - 40, y + 80, 36, 30);
    //window frames
    g.setColor(LIGHT_GRAY);
    g.drawLine(x - 40, y + 95,          + 95);
    g.drawLine(x - 22, y + 80, x - 22, y + 110);
    //doors
    g.setColor(DARK_GRAY);
    g.fillRect(x + 10, y + 95, 25, 53);
    //door handle
    g.setColor(LIGHT_GRAY);
    g.fillOval(x + 12, y + 117, 6, 6);
    //smoke
    g.setColor(WHITE);
    for (int i = 25; i < 80; i += 15) {
        g.fillOval(x - 12 + i, y + 20 - i, 24, 24);
    }
}

void drawTree(Graphics g, int x, int y) {
    //branches
    for (int i = 20; i < 120; i += 30) {
        g.setColor(new Color(0, 200 - i, 0));
        g.fillPolygon(new int[]{x - i, x, x + i}, new int[]{y + i, y + i - 40, y + i}, 3);
    }
    //trunk
    g.setColor(DARK_GRAY);
    g.fillRect(x - 10, y + 110, 20, 30);
}

Run main | Debug main
void main() {
    var f = new JFrame() {
        public void paint(Graphics g) {
            g.setColor(CYAN);
            g.fillRect(0, 0, 1000, 1000);
            var random = new Random(1);
            for (int y = 100; y < 350; y += 5) {
                drawTree(g, random.nextInt(600), y);
            }
            drawSnowman(g, 100, 380);
            drawSnowman(g, 500, 380);
            drawHouse(g, 300, 340);
        }
    };
    f.setSize(600, 500);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setVisible(true);
}
```

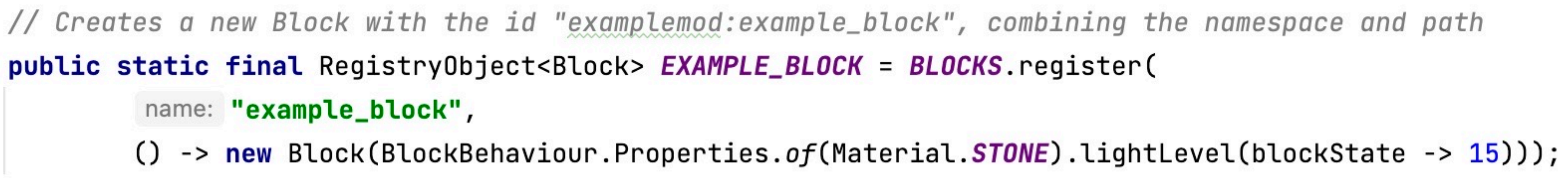

GameOfLife.java — workspace (Workspace)
GameOfLife.java
projekty > sample project > onramp > GameOfLife.java > ...
Filter (e.g. text, !exclu...
Run main | Debug main
Please start a debug session to evaluate expressions
0 0
Indexing completed.
Ln 22, Col 26
Spaces: 4
UTF-8
LF
Java

```java
int width = 30;
int height = 30;

class Generation extends BitSet {

    boolean isThereLife(int x, int y) {
        return x >= 0 && y >= 0 && x < width && get(x + y * width);
    }

    void setValue(int x, int y, boolean life) {
        set(x + y * width, life);
    }

    int countNeighbors(int x, int y) {
        int neighbors = 0;
        for (var sy = y - 1; sy <= y + 1; sy++) {
            for (var sx = x - 1; sx <= x + 1; sx++) {
                if (isThereLife(sx, sy)) neighbors++;
            }
        }
        if (isThereLife(x, y)) neighbors--;
        return neighbors;
    }
}

void main() throws InterruptedException {
    var initialState = """
    XX
    XX   XX
        X  X
         XX      X
                  X
    XX          XXX
    X   X
    X  XXX
        X      X X
               X X
               X X
            XXX   XXX

            XXX   XXX
               X X
               X X
               X X
    """.split(System.lineSeparator());

    var oldGeneration = new Generation();
    for (var y = 0; y < initialState.length; y++) {
        var row = initialState[y];
        for (var x = 0; x < row.length(); x++) {
            oldGeneration.setValue(x, y, row.charAt(x) != ' ');
        }
    }

    var generationCounter = 1;
    while(true) {
        println("Generation " + generationCounter++);
        var newGeneration = new Generation();
        for (var y = 0; y < width; y++) {
            for (var x = 0; x < height; x++) {
```

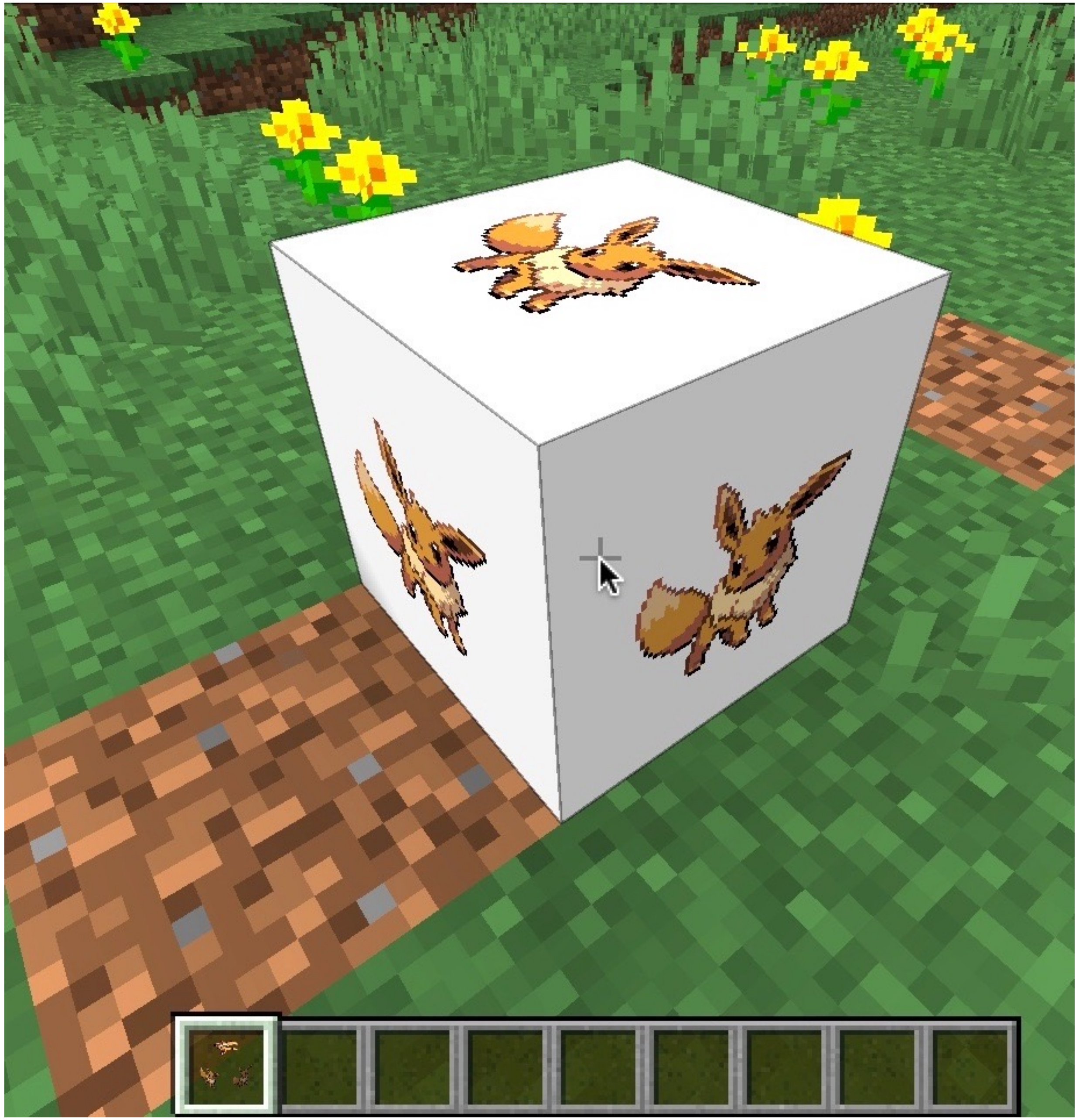


```java
// Creates a new Block with the id "examplemod:example_block", combining the namespace and path
public static final RegistryObject<Block> EXAMPLE_BLOCK = BLOCKS.register(
        name: "example_block",
        () -> new Block(BlockBehaviour.Properties.of(Material.STONE)));
```





```java
// Creates a new Block with the id "examplemod:example_block", combining the namespace and path
public static final RegistryObject<Block> EXAMPLE_BLOCK = BLOCKS.register(
        name: "example_block",
        () -> new Block(BlockBehaviour.Properties.of(Material.STONE).lightLevel(blockState -> 15)));
```



Some of the samples published
at https://gist.github.com/asotona