

## Patterns in Java

In Java, **patterns** refer to structured arrangements of characters (often stars **\***, numbers, or other symbols) printed using loops. These patterns are primarily created using **nested loops**, where:

- The **outer loop** controls the **rows**.
- The **inner loop** controls the **columns**.

Understanding how **i** and **j** move:

- **i** (outer loop) controls **which row** is being printed.
- **j** (inner loop) controls **what to print in each row**.

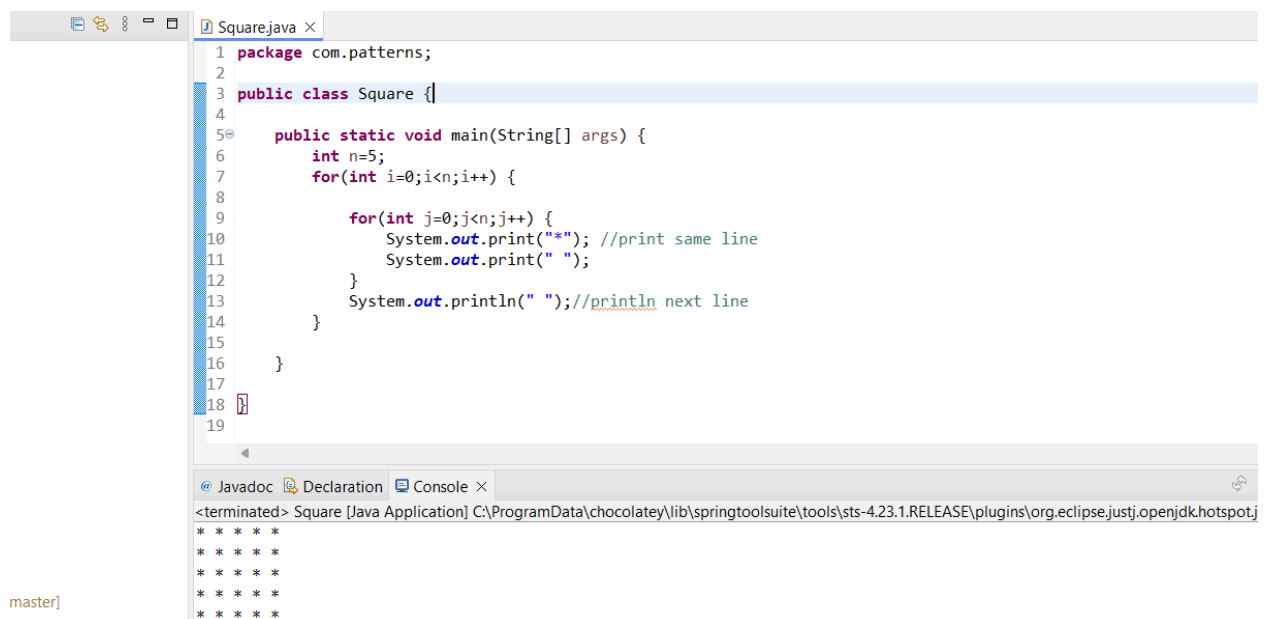
### 1. Square / Rectangle Pattern

A **simple square or rectangle** pattern consists of a fixed number of rows and columns.

How **i** and **j** move:

1. **i** = 0 → **j** = 0, 1, 2, 3, 4 → Print row 1
2. **i** = 1 → **j** = 0, 1, 2, 3, 4 → Print row 2
3. **i** = 2 → **j** = 0, 1, 2, 3, 4 → Print row 3
4. **i** = 3 → **j** = 0, 1, 2, 3, 4 → Print row 4
5. **i** = 4 → **j** = 0, 1, 2, 3, 4 → Print row 5

Each **j** loop completes before **i** moves to the next row.



```
1 package com.patterns;
2
3 public class Square {
4
5     public static void main(String[] args) {
6         int n=5;
7         for(int i=0;i<n;i++) {
8
9             for(int j=0;j<n;j++) {
10                 System.out.print("*"); //print same line
11                 System.out.print(" ");
12             }
13             System.out.println(" "); //println next line
14         }
15     }
16 }
17
18
19
```

master]

```
<terminated> Square [Java Application] C:\ProgramData\chocolatey\lib\springtoolsuite\tools\sts-4.23.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.j
```

## Ex2: Right Half Pyramid Pattern / Right-Angled Triangle Pattern

How **i** and **j** move:

1. **i** = 0 → **j** = 0 → \*
2. **i** = 1 → **j** = 0, 1 → \* \*
3. **i** = 2 → **j** = 0, 1, 2 → \* \* \*
4. **i** = 3 → **j** = 0, 1, 2, 3 → \* \* \* \*
5. **i** = 4 → **j** = 0, 1, 2, 3, 4 → \* \* \* \* \*

```
1 package com.patterns;
2
3 public class Square {
4
5     public static void main(String[] args) {
6         int n=5;
7         for(int i=0;i<n;i++) {
8
9             for(int j=0;j<=i;j++) {
10                 System.out.print("*"); //print same line
11                 System.out.print(" ");
12             }
13             System.out.println(" "); //println next line
14         }
15     }
16 }
17
18 }
```

@ Javadoc Declaration Console X

<terminated> Square [Java Application] C:\ProgramData\chocolatey\lib\springtoolsuite\tools\sts-4.23.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot

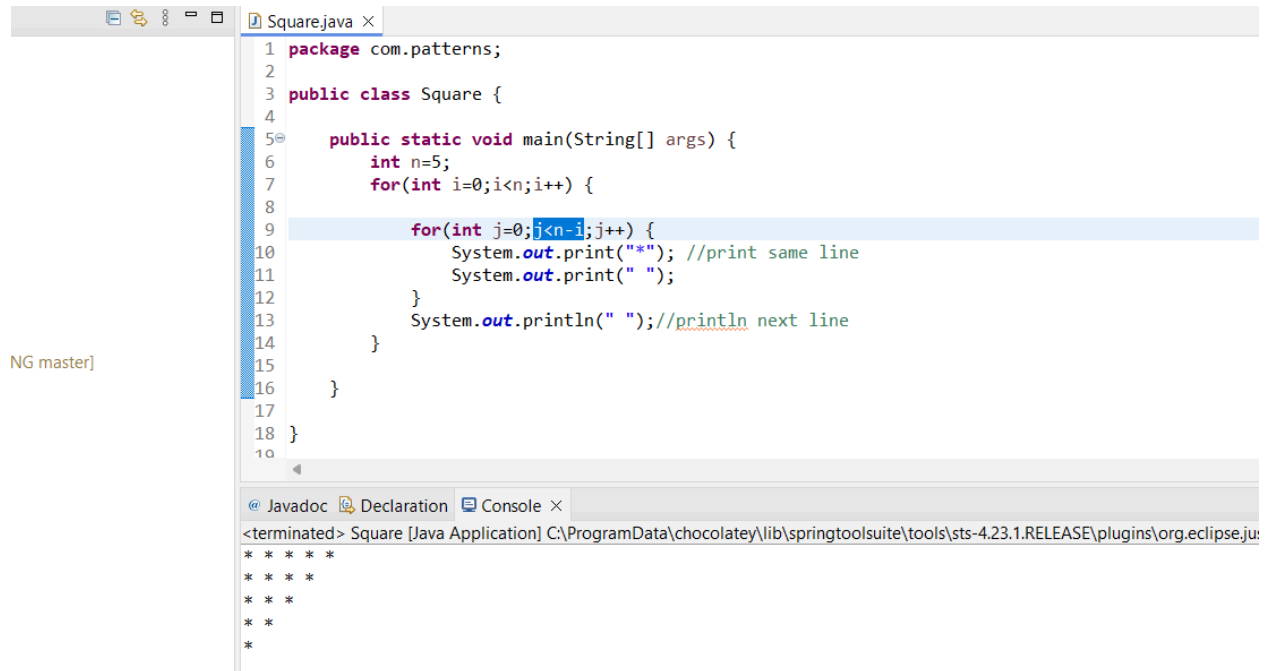
```
*
*
* *
* * *
* * * *
```

NG master]

## 3) Reverse Right Half Pyramid Pattern or Inverted Right-Angled Triangle

How **i** and **j** move:

1. **i** = 0 → **j** = 0, 1, 2, 3, 4 → \* \* \* \* \*
2. **i** = 1 → **j** = 0, 1, 2, 3 → \* \* \* \*
3. **i** = 2 → **j** = 0, 1, 2 → \* \* \*
4. **i** = 3 → **j** = 0, 1 → \* \*
5. **i** = 4 → **j** = 0 → \*



The screenshot shows the Eclipse IDE with a Java file named `Square.java`. The code defines a package `com.patterns` and a class `Square` with a `main` method. The `main` method sets `n=5` and uses nested loops to print a 5x5 square of asterisks. The console output shows the resulting pattern of asterisks.

```
1 package com.patterns;
2
3 public class Square {
4
5     public static void main(String[] args) {
6         int n=5;
7         for(int i=0;i<n;i++) {
8
9             for(int j=0;j<n-i;j++) {
10                 System.out.print("*"); //print same line
11                 System.out.print(" ");
12             }
13             System.out.println(" "); //println next line
14         }
15     }
16 }
17
18 }
19
```

NG master]

@ Javadoc Declaration Console ×

<terminated> Square [Java Application] C:\ProgramData\chocolatey\lib\springtoolsuite\tools\sts-4.23.1.RELEASE\plugins\org.eclipse.ju

```
* * * * *
* * * * *
* * *
* *
*
```

#### 4) Pyramid Pattern