This java program code will do something a bit unusual. Try to figure out what it will do before reading through the explanation and looking at the end which shows what the console would look like.

```java
int[] sizes = { 3, 5, 1, 0, 3, 6 }; for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}
```
int[] sizes = { 3, 5, 1, 0, 3, 6 }; 
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}

This program has NO user input. 
It has an arithmetic while loop INSIDE of for-in loop. 
It uses an array initialized at the very beginning. 
It only writes out the asterisk character (unless you count writing out end-of-lines)
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {
    int x = 1;  
    while(x<wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}  

This will create an array variable named sizes. It will make an integer array with 6 elements. That array will be kept in the variable named sizes. The numbers in that array will be:  
3 then 5 then 1 then 0 then 3 then 6.
int[] sizes = { 3, 5, 1, 0, 3, 6 }; 
for( int wide : sizes ) { 
    int x = 1; 
    while(x<=wide) { 
        out.print("*"); 
        x = x+1; 
    } 
    out.println(); 
} 

This "frames" a for-in loop. 
The loop variable is named wide and will hold an integer each time the loop code executes. 
The values (one at a time) for wide will come out of the array named sizes - namely they will be 3 then 5 then 1 then 0 then 3 and finally 6.
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {
    int x = 1;  
    while(x<=wide) {
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}
```

INSIDE THE for-in LOOP
there is the standard model for an arithmetic while loop. Its loop variable is x. The arithmetic sequence starts at 1 and ends with whatever number is kept in the variable wide. The arithmetic sequence counts "by ones". Thus it will repeat the out.print step as many times as one can count from 1 to wide. The variables wide and x ARE NOT printed on the console.
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
    int x = 1;  
    while(x<=wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}
```

The `out.print` command is repeated over and over by the surrounding `while` loop. However each repetition of the `out.print` command ONLY prints **ONE** asterisk. Since the while loop performs a number of repetitions equal to the number in the variable `wide` - this means that it will print out that many asterisks. Thus the number in `wide` determines how many asterisks are printed. When the while loop is completed - the `out.println()` command prints just the **end of the line**. Hence the next repetition of the `for-in` loop will start a new line.
Here is the final console output from running the program just explained:

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

***
******
```
int[] sizes = { 3, 5, 1, 0, 3, 6 }; 
for( int wide : sizes ) { 
    int x = 1; 
    while(x<=wide) { 
        out.print("*"); 
        x = x+1; 
    } 
    out.println(); 
} 

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
<th>wide</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Console
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}
```
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };
for (int wide : sizes) {
    int x = 1;
    while(x <= wide) {
        out.print("*");
        x = x + 1;
    }
    out.println();
}
```

```plaintext
Console
```

**Variables in Memory**

```
sizes
0 3
1 5
2 1
3 0
4 3
5 6

wide
3

x
1
```
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*" );
        x = x+1;
    }
    out.println();
}
```

Note that x=1 is indeed <= wide=3
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
    int x = 1;  
    while(x<=wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}
```

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
</tr>
<tr>
<td>1 5</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>3 0</td>
</tr>
<tr>
<td>4 3</td>
</tr>
<tr>
<td>5 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
int[] sizes = { 3, 5, 1, 0, 3, 6 }; 
for( int wide : sizes ) { 
    int x = 1; 
    while(x<=wide) { 
        out.print("*"); 
        x = x+1; 
    } 
    out.println(); 
}
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
    int x = 1;  
    while(x<=wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}  

Console  

*  

Note x=2 <= wide=3
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 }; // int x = 1;
for (int wide : sizes) {
    int x = 1;
    while (x <= wide) {
        out.print("*"); // x = x+1;
    }
    out.println(); //
}
```

**Variables in Memory**

<table>
<thead>
<tr>
<th>sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
</tr>
<tr>
<td>1 5</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>3 0</td>
</tr>
<tr>
<td>4 3</td>
</tr>
<tr>
<td>5 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
```java
int[] sizes = { 3, 5, 1, 0, 3, 6};
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}
```
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
    int x = 1;  
    while(x<=wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}  
```

Variables in Memory

```
sizes
[3] [5] [1] [0] [3] [6]  
wide
[3]  
x
[1] [2] [3] [4]  
```

Fast Forward 3 steps

```
x = 3 <= wide = 3
```
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 }; for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}
```

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
<th>wide</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note x=4 IS NOT <= wide=3

THUS the while loop EXITS and the next step will be here
int[] sizes = { 3, 5, 1, 0, 3, 6 };
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
</tr>
<tr>
<td>1 5</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>3 0</td>
</tr>
<tr>
<td>4 3</td>
</tr>
<tr>
<td>5 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

Nothing will show on the console screen but the next output to the console will move down to the next line.
int[] sizes = { 3, 5, 1, 0, 3, 6 };
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}

Variables in Memory

Console

***

Outer for-in loop advances to the next value in the for array named sizes.
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
  int x = 1;  
  while(x<=wide) {  
    out.print("*");  
    x = x+1;  
  }  
  out.println();  
}  

Console

***

Variables in Memory

sizes

wide

x

Outer for-in loop advances to the next value in the for array named sizes.
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 }; for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}
```

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
<th>widths</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
<td>5</td>
</tr>
<tr>
<td>1 5</td>
<td>1</td>
</tr>
<tr>
<td>2 1</td>
<td>3</td>
</tr>
<tr>
<td>3 0</td>
<td>4</td>
</tr>
<tr>
<td>4 3</td>
<td>5</td>
</tr>
<tr>
<td>5 6</td>
<td></td>
</tr>
</tbody>
</table>

```java
x
```

Fast forward through lots of steps - In fact, through the entire while loop until x exceeds wide and thus stops that while loop.

But because the while loop repeats 5 times, it prints out five asterisks.
int[] sizes = { 3, 5, 1, 0, 3, 6 };
for( int wide : sizes ) {
    int x = 1;
    while(x<=wide) {
        out.print("*");
        x = x+1;
    }
    out.println();
}

Variables in Memory

<table>
<thead>
<tr>
<th>sizes</th>
<th>wide</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3</td>
<td>5</td>
</tr>
<tr>
<td>1 5</td>
<td>1</td>
</tr>
<tr>
<td>2 1</td>
<td>3</td>
</tr>
<tr>
<td>3 0</td>
<td>4</td>
</tr>
<tr>
<td>4 3</td>
<td>5</td>
</tr>
<tr>
<td>5 6</td>
<td></td>
</tr>
</tbody>
</table>

Console

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

Moves next output position in console down to next line - just as before.
```java
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {     
    int x = 1;     
    while(x<=wide) {     
        out.print("*");     
        x = x+1;     
    }     
    out.println();     
}  
```

Variables in Memory

```plaintext
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

wide: 1

x:

```

Now fast forward through an entire repetition of the outer for-in loop
int[] sizes = { 3, 5, 1, 0, 3, 6 };  
for( int wide : sizes ) {  
    int x = 1;  
    while(x<=wide) {  
        out.print("*");  
        x = x+1;  
    }  
    out.println();  
}