

Code File Edit Selection View Go Run Terminal Window Help

Demo.java — Codes

TELUSKO

Demo.java 2 x Consumer.class

Demo.java > Demo > main(String[])

```
1 import java.util.ArrayList;
2 import java.util.Arrays;
3 import java.util.List;
4 import java.util.function.Consumer;
5 import java.util.stream.Stream;
6
7 public class Demo {
8     public static void main(String a[]) {
9
10         List<Integer> nums = Arrays.asList(...a: 4,5,7,3,2,6);
11
12         Consumer<Integer> con = n -> System.out.println(n);
13
14
15         nums.forEach(con);
16
17
18
19
20
21         //nums.forEach(n -> System.out.println(n));
22     }
```

TERMINAL ... zsh + v

```
navin@iMac Codes % javac Demo.java
navin@iMac Codes % java Demo
4
5
6
7
3
2
6
navin@iMac Codes %
```

Ln 12, Col 34 Spaces: 4 UTF-8 LF

Subscribe Telusko

Code File Edit Selection View Go Run Terminal Window Help

Demo.java — Codes

1 Demo.java 1 • Consumer.class

1 Demo.java > Demo > main(String[])

```
1 import java.util.ArrayList;
2 import java.util.Arrays;
3 import java.util.List;
4 import java.util.function.Consumer;
5 import java.util.stream.Stream;
6
7 public class Demo {
8     public static void main(String a[]) {
9
10         List<Integer> nums = Arrays.asList(...a: 4,5,7,3,2,6);
11
12
13
14         nums.forEach(n -> System.out.println(n));
15
16
17
18
19
20         //nums.forEach(n -> System.out.println(n));
21
22
```

TERMINAL ... zsh + - - - - -

```
navin@iMac Codes % javac Demo.java
navin@iMac Codes % java Demo
4
5
7
3
2
6
navin@iMac Codes % javac Demo.java
navin@iMac Codes % java Demo
4
5
7
3
2
6
navin@iMac Codes %
```

Ln 12, Col 6 Spaces: 4 UTF-8 LF

1 0

Subscribe Telusko

Code Editor (VS Code) showing a Java file named `Demo.java` with the following code:

```
1 import java.util.function.Consumer;
2 import java.util.stream.Stream;
3
4 public class Demo {
5     public static void main(String a[]) {
6
7         List<Integer> nums = Arrays.asList(4,5,7,3,2,6);
8
9         Consumer<Integer> con = new Consumer<Integer>() {
10
11             public void accept(Integer n)
12             {
13                 System.out.println(n);
14             }
15         };
16
17         nums.forEach(con);
18     }
19 }
```

The code is annotated with red circles and arrows. A red circle highlights the `Consumer<Integer>` interface, and a red arrow points to the `accept(Integer n)` method. Another red circle highlights the `con` variable, and a red arrow points to the `forEach` method call.

Terminal output:

```
navin@iMac Codes % javac Demo.java
navin@iMac Codes % java Demo
4
5
7
3
2
6
navin@iMac Codes %
```

A red text label "functional interface" is overlaid on the code, pointing to the `Consumer<Integer>` interface.

STATUS BAR: Ln 17, Col 14 Spaces: 4 UTF-8 LF

Subscribe  
Telusko