In Java, a stack is a data structure that follows the Last-In-First-Out (LIFO) principle. This means that the last element added to the stack is the first one to be removed. Java provides a Stack class that is part of the Java Collections Framework.

Here's an introduction to the Stack class in Java:

Importing the Stack Class

To use the Stack class, you need to import it from the java.util package:

```
import java.util.Stack;
```

Creating a Stack

You can create a stack as follows:

```
Stack<Integer> stack = new Stack<>();
```

Basic Operations

1. **Push**: Add an element to the top of the stack.

   ```
   stack.push(10);
   stack.push(20);
   stack.push(30);
   ```

2. **Pop**: Remove and return the top element of the stack.

   ```
   int top = stack.pop(); // top will be 30
   ```

3. **Peek**: Return the top element without removing it.

   ```
   java
   Copy code
   int top = stack.peek(); // top will be 20
   ```

4. **Check if the stack is empty**:

   ```
   boolean isEmpty = stack.isEmpty(); // returns false
   ```

5. **Search**: Determine the position of an element in the stack (1-based position from the top of the stack).

   int position = stack.search(10); // position will be 2

Example Code

Here is a complete example demonstrating the usage of a stack in Java:

```
import java.util.Stack;

public class StackExample {
    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();

        // Pushing elements onto the stack
        stack.push(10);
        stack.push(20);
        stack.push(30);

        // Printing the stack
        System.out.println("Stack: " + stack);

        // Popping an element from the stack
        int poppedElement = stack.pop();
        System.out.println("Popped Element: " + poppedElement);

        // Peeking at the top element of the stack
        int topElement = stack.peek();
        System.out.println("Top Element: " + topElement);

        // Checking if the stack is empty
        boolean isEmpty = stack.isEmpty();
        System.out.println("Is Stack Empty? " + isEmpty);

        // Searching for an element in the stack
        int position = stack.search(10);
        System.out.println("Position of 10: " + position);
```

```
    }
}
```

Output

```
mathematica
Stack: [10, 20, 30]
Popped Element: 30
Top Element: 20
Is Stack Empty? false
Position of 10: 2
```