Longest Increasing Subsequence Source Code

```
package longestsubsquence;
public class LongestIncreasingSubsequence {

    public static int[] findLongestIncreasingSubsequence(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            return new int[0];
        }

        int n = numbers.length;
        int[] lengths = new int[n];
        int[] previousIndices = new int[n];

        int maxLength = 1;
        int endIndex = 0;

        for (int i = 0; i < n; i++) {
            lengths[i] = 1;
            previousIndices[i] = -1;

            for (int j = 0; j < i; j++) {
                if (numbers[j] < numbers[i] && lengths[j] + 1 > lengths[i])
{
                    lengths[i] = lengths[j] + 1;
                    previousIndices[i] = j;
                }
            }

            if (lengths[i] > maxLength) {
                maxLength = lengths[i];
                endIndex = i;
            }
        }

        int[] longestIncreasingSubsequence = new int[maxLength];
        int index = maxLength - 1;
        while (endIndex >= 0) {
            longestIncreasingSubsequence[index] = numbers[endIndex];
            endIndex = previousIndices[endIndex];
            index--;
        }

        return longestIncreasingSubsequence;
    }

    public static void main(String[] args) {
        int[] numbers = {3, 10, 2, 1, 20, 15, 17, 18};

        int[] longestIncreasingSubsequence =
findLongestIncreasingSubsequence(numbers);

        System.out.print("Longest increasing subsequence: ");
        for (int num : longestIncreasingSubsequence) {
            System.out.print(num + " ");

        }
        System.out.println("\nLength of the the Longest increasing
subsequence: " +longestIncreasingSubsequence.length);
    }
}

/*
output
Longest increasing subsequence: 3 10 15 17 18
Length of the the Longest increasing subsequence: 5
*/
```