**package** dom;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** LongestIncreasingSubsequence {

**public** **static** **void** main(String[] args) {

List<Integer> numbers = **new** ArrayList<>();

**for** (**int** i = 0; i < 10; i++) {

numbers.add((**int**) (Math.*random*() \* 100));

}

System.***out***.println("Original list of numbers: " + numbers);

List<Integer> longestIncreasingSubsequence = *findLongestIncreasingSubsequence*(numbers);

System.***out***.println("Longest Increasing Subsequence: " + longestIncreasingSubsequence);

}

**static** List<Integer> findLongestIncreasingSubsequence(List<Integer> numbers) {

**if** (numbers == **null** || numbers.isEmpty()) {

**return** **new** ArrayList<>();

}

**int** n = numbers.size();

List<List<Integer>> lis = **new** ArrayList<>();

**for** (**int** i = 0; i < n; i++) {

lis.add(**new** ArrayList<>());

lis.get(i).add(numbers.get(i));

}

**for** (**int** i = 1; i < n; i++) {

**for** (**int** j = 0; j < i; j++) {

**if** (numbers.get(i) > numbers.get(j) && lis.get(i).size() < lis.get(j).size() + 1) {

lis.set(i, **new** ArrayList<>(lis.get(j)));

lis.get(i).add(numbers.get(i));

}

}

}

List<Integer> longestIncreasingSubsequence = **new** ArrayList<>();

**for** (List<Integer> subsequence : lis) {

**if** (subsequence.size() > longestIncreasingSubsequence.size()) {

longestIncreasingSubsequence = subsequence;

}

}

**return** longestIncreasingSubsequence;

}

}