

Sorting with Array

Sungchul Lee

Learning Object

- What is sorting?
- Sorting High to LOW
- Sorting Low to HIGH

Sorting

- Sorting is any process of arranging items systematically.
 - ❑ Ordering : sequence ordered
 - ❑ Categorizing : grouping

The image shows a screenshot of an Amazon web browser interface. The address bar displays the URL: `amazon.com/s?k=ms+surface+pro+7&crid=DPN...`. The Amazon header includes the logo, a search bar with the text "ms surface pro 7", and navigation links for "Hello, Sign in", "Account & Lists", "Orders", "Try Prime", and a shopping cart icon. Below the header, it says "1-16 of 829 results for 'ms surface pro 7'". On the left, there's a filter section for "Amazon Prime" with a checkbox and "Eligible for Free Shipping". On the right, a sorting dropdown menu is open, showing options: "Featured", "Price: Low to High", "Price: High to Low", "Avg. Customer Review", and "Newest Arrivals". The menu is highlighted with a red dashed border.

sorting

➤ Low to Hight

- ❑ Making ascending order
- ❑ Arranging elements from the smallest to the largest number

➤ Hight to Low

- ❑ Making descending order
- ❑ Arranging elements from the largest to the smallest number

LOW → HIGH	
BEER 1	\$4
BEER 2	\$4
BEER 3	\$4
BEER 4	\$5
BEER 5	\$6
BEER 6	\$7
BEER 7	\$7
BEER 8	\$7
BEER 9	\$7
BEER 10	\$8
BEER 11	\$8
BEER 12	\$9
BEER 13	\$10
AVG SALE \$5.78	

HIGH → LOW	
BEER 1	\$10
BEER 2	\$9
BEER 3	\$8
BEER 4	\$8
BEER 5	\$7
BEER 6	\$7
BEER 7	\$7
BEER 8	\$7
BEER 9	\$6
BEER 10	\$5
BEER 11	\$4
BEER 12	\$4
BEER 13	\$4
AVG SALE \$6.02	

Selection Sorts

- The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.
 - 1) The subarray which is already sorted.
 - 2) Remaining subarray which is unsorted.

Selection Sorts Algorithm

➤ Loop

- ❑ Find the index which has a max value
- ❑ **Swap** the value to rightmost elements
- ❑ Exclude rightmost element

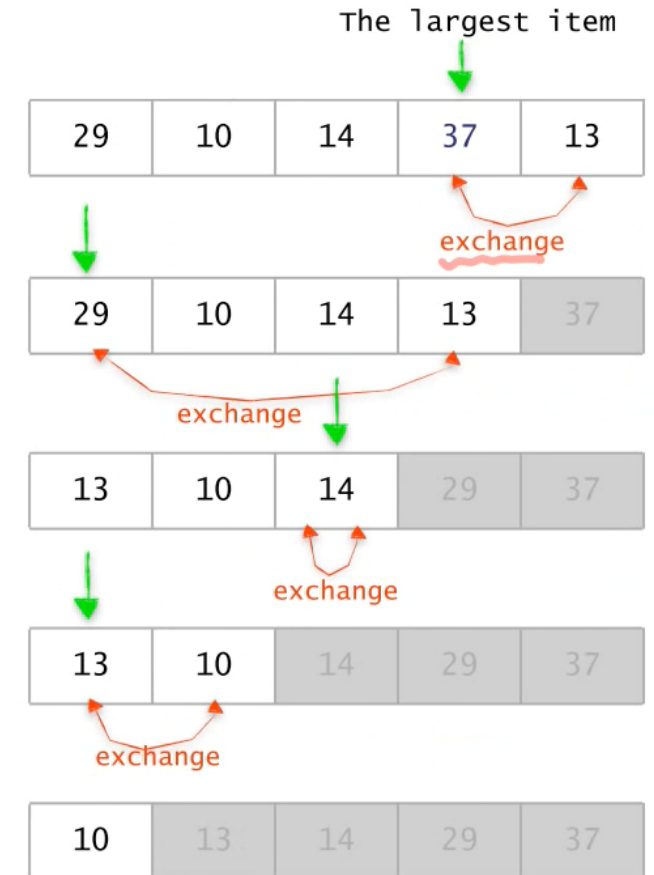
initial array

after 1st swap

after 2nd swap

after 3rd swap

after 4th swap

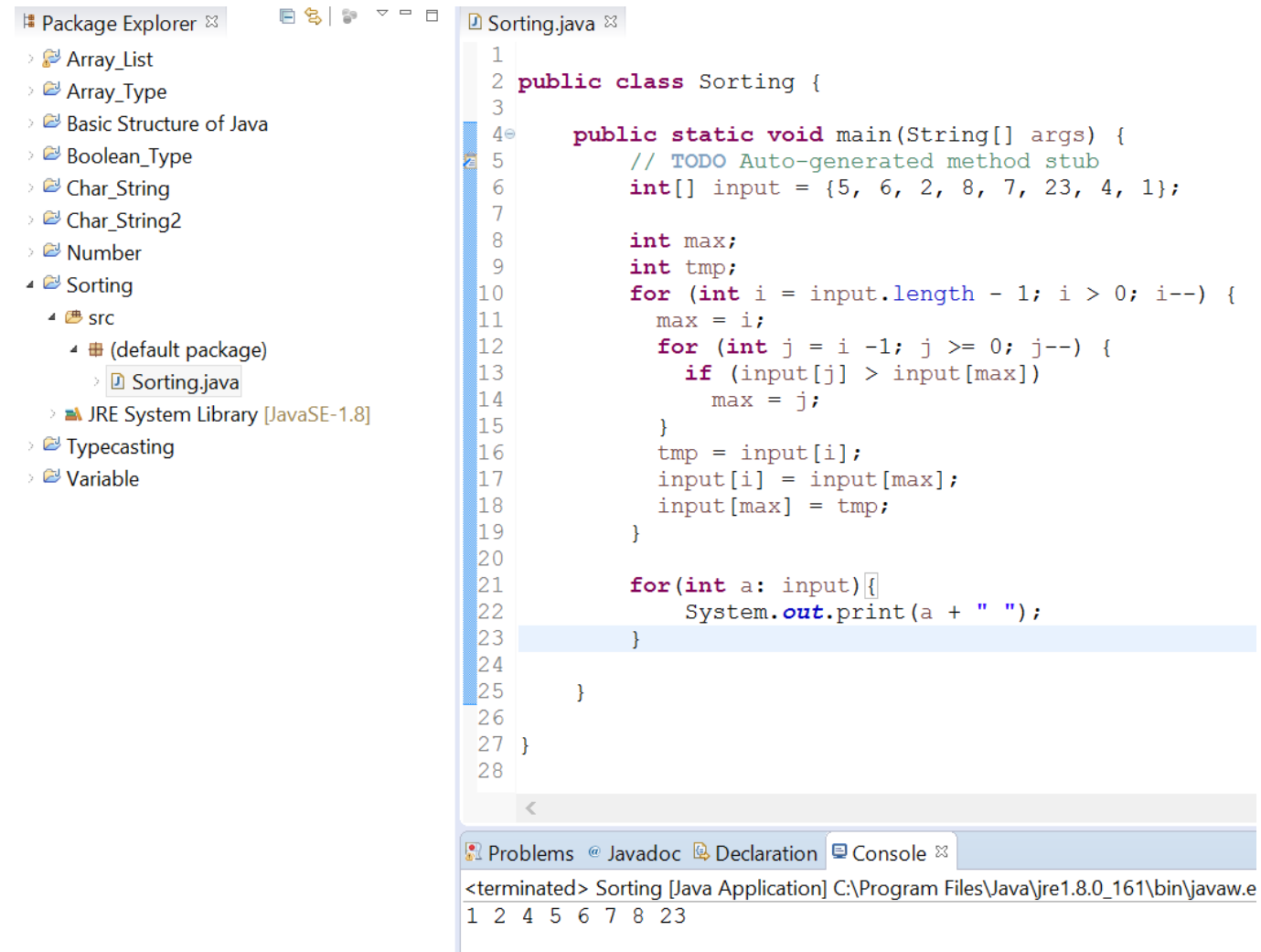


Practice

1. Make a new project
 - ☐ Project name: **Sorting**
 - ☐ Create a new Class File
 - ❖ Class name: **Sorting**
 - ☐ Coding:

```
public class Sorting {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int[] input = {5, 6, 2, 8, 7, 23, 4, 1};  
        int max;  
        int tmp;  
        for (int i = input.length - 1; i > 0; i--) {  
            max = i;  
            for (int j = i - 1; j >= 0; j--) {  
                if (input[j] > input[max])  
                    { max = j;}  
            }  
            tmp = input[i];  
            input[i] = input[max];  
            input[max] = tmp;  
        }  
        for(int a: input){  
            System.out.print(a + " ");  
        }  
    }  
}
```

Practice – Code and Result



The screenshot displays an IDE interface. On the left, the 'Package Explorer' shows a project structure with a 'Sorting' package containing a 'Sorting.java' file. The main editor window shows the code for 'Sorting.java', which implements a selection sort algorithm. The code initializes an array 'input' with the values {5, 6, 2, 8, 7, 23, 4, 1}. It then iterates through the array, finding the maximum element in each pass and swapping it with the element at the current index. Finally, it prints the sorted array. The bottom of the IDE shows the 'Console' tab with the output: '<terminated> Sorting [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.e 1 2 4 5 6 7 8 23'.

```
1
2 public class Sorting {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int[] input = {5, 6, 2, 8, 7, 23, 4, 1};
7
8         int max;
9         int tmp;
10        for (int i = input.length - 1; i > 0; i--) {
11            max = i;
12            for (int j = i - 1; j >= 0; j--) {
13                if (input[j] > input[max])
14                    max = j;
15            }
16            tmp = input[i];
17            input[i] = input[max];
18            input[max] = tmp;
19        }
20
21        for(int a: input){
22            System.out.print(a + " ");
23        }
24
25    }
26
27 }
28
```

<terminated> Sorting [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.e
1 2 4 5 6 7 8 23

Summary

➤ Sorting is any process of arranging items systematically.

□ Ordering

❖ sequence ordered

```
1
2 public class Sorting {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int[] input = {5, 6, 2, 8, 7, 23, 4, 1};
7
8         int max;
9         int tmp;
10        for (int i = input.length - 1; i > 0; i--) {
11            max = i;
12            for (int j = i - 1; j >= 0; j--) {
13                if (input[j] > input[max])
14                    max = j;
15            }
16            tmp = input[i];
17            input[i] = input[max];
18            input[max] = tmp;
19        }
20
21        for(int a: input){
22            System.out.print(a + " ");
23        }
24    }
25 }
```

initial array

after 1st swap

after 2nd swap

after 3rd swap

after 4th swap

