University of British Columbia
CPSC 111, Intro to Computation
2009W2: Jan-Apr 2010
Tamara Munzner

2D Arrays, Sorting

Lecture 23, Fri Mar 12 2010

borrowing from slides by Kurt Eiselt

http://www.cs.ubc.ca/~tmm/courses/111-10

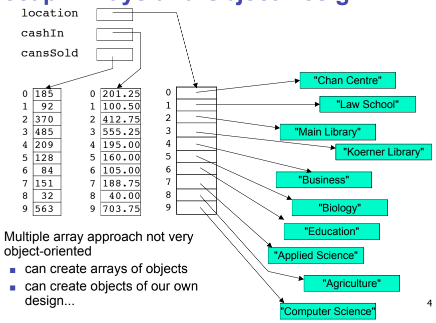
News

- Remember Learning Centre available!
  - Mon-Thu 10-6, Fri 10-4, x150 (near Reboot)
- Upcoming midterm
  - Mon 3/22, 6:30-8pm, FSC 1005

Reading

- Next week: no new reading
  - so no weekly question required

Recap: Arrays and Object Design

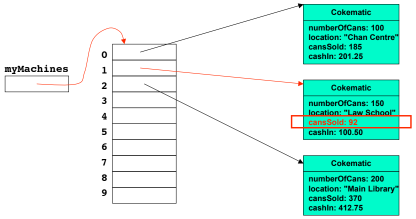


- Multiple array approach not very object-oriented
  - can create arrays of objects
  - can create objects of our own design...

Recap: CokeEmpire

- What does this return?

```
myMachines.getCokematic(1).getCansSold()
```

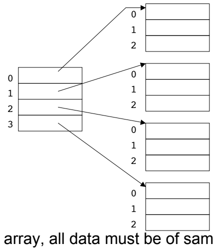


Cokematic
numberOfCans: 100
location: "Chan Centre"
cansSold: 185
cashIn: 201.25

Cokematic
numberOfCans: 150
location: "Law School"
cansSold: 92
cashIn: 100.50

Cokematic
numberOfCans: 200
location: "Main Library"
cansSold: 370
cashIn: 412.75

Recap: Arrays of Arrays



- In any given array, all data must be of same type
- All arrays in array of arrays must be of same type
- So easier to use a two-dimensional array!

Two-Dimensional Arrays

columns

rows	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4
3	0	3	6

- In Java, 2D array implemented internally as array of arrays
  - but externally syntax of 2D array may seem easier to use

Recap: Two-Dimensional Arrays

columns

rows	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4
3	0	3	6

- In Java, 2D array implemented internally as array of arrays
  - but externally syntax of 2D array may seem easier to use
- Typical control structure for computing with 2D array is nested loop
  - loop within another loop
- Let's write program to
  - load array with values shown
  - print contents of array

Recap: Two-Dimensional Arrays

columns

rows	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4
3	0	3	6

```
public class ArrayTest5 {
  public static void main(String[] args) {
    int[][] multTable = new int[4][3];

    for (int row = 0; row < multTable.length; row++){
      for (int col = 0; col < multTable[row].length; col++) {
        multTable[row][col] = row * col;
      }
    }

    for (int row = 0; row < multTable.length; row++){
      for (int col = 0; col < multTable[row].length; col++){
        System.out.print(multTable[row][col] + " ");
      }
      System.out.println();
    }
  }
}
```

Example: Per-Student Averages

scores

	0	1	2	3
0	95	82	13	96
1	51	68	63	57
2	73	71	84	78
3	50	50	50	50
4	99	70	32	12

```
average of row 0 is 71.5
average of row 1 is 59.75
average of row 2 is 76.5
average of row 3 is 50.0
average of row 4 is 53.25
```

- 2D array
  - each row is student in course
  - values in each row represent student's quiz scores in course
- Print average quiz score for each student
  - for each row of scores
    - add up scores
    - divide by number of quizzes in a row
  - approach: nested loop

Example: Per-Student Averages

```
public class ArrayEx4
{
  public static void main(String[] args)
  {
    double[][] scores = {{95, 82, 13, 96},
      {51, 68, 63, 57}, {73, 71, 84, 78}, {50, 50, 50, 50},
      {99, 70, 32, 12}};
    double average;

    // here's where we control looping row by row (student by student)
    for (int row = 0; row < scores.length; row++)
    {
      average = 0;
      // and here's where we control looping through the columns
      // (i.e., quiz scores) within each row
      for (int col = 0; col < scores[row].length; col++)
      {
        average = average + scores[row][col];
      }
      average = average / scores[row].length;
      System.out.println("average of row " + row + " is " + average);
    }
  }
}
```

Example: Per-Quiz Averages

scores

	0	1	2	3
0	95	82	13	96
1	51	68	63	57
2	73	71	84	78
3	50	50	50	50
4	99	70	32	12

```
average of column 0 is 73.6
average of column 1 is 68.2
average of column 2 is 48.4
average of column 3 is 58.6
```

- Print average score for each quiz
  - for each column of scores
    - add up all scores
    - divide by number of students
  - approach: again, nested loop
- Switch of outer loop with inner loop, vs. previous

Example: Per-Quiz Averages

```
public class ArrayEx5
{
  public static void main(String[] args)
  {
    double[][] scores = {{95, 82, 13, 96},
      {51, 68, 63, 57}, {73, 71, 84, 78}, {50, 50, 50, 50},
      {99, 70, 32, 12}};
    double average;

    // here's where we control looping column by column (quiz by quiz)
    for (int col = 0; col < scores[0].length; col++)
    {
      average = 0;
      // and here's where we control looping through the rows
      // (i.e., students) within each column
      for (int row = 0; row < scores.length; row++)
      {
        average = average + scores[row][col];
      }
      average = average / scores.length;
      System.out.println("average of column " + col + " is " + average);
    }
  }
}
```

Sorting

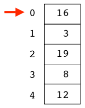
- Computers are essential for keeping track and finding large quantities of data
- Finding data when necessary is much easier when data is sorted in some way
  - computer people think a lot about how to sort things:
    - finding medical records
    - banking information
    - income tax returns
    - driver's license information...
    - even names in a phone book...
  - all depend on the information being sorted

Selection sort

0	16
1	3
2	19
3	8
4	12

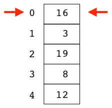
- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort

Selection sort



- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed

Selection sort

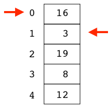


The smallest value so far is 16

Its index is 0

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value

Selection sort

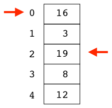


The smallest value so far is 3

Its index is 1

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value

Selection sort

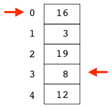


The smallest value so far is 3

Its index is 1

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value

Selection sort

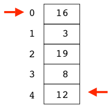


The smallest value so far is 3

Its index is 1

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value

Selection sort

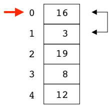


The smallest value so far is 3

Its index is 1

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value

Selection sort

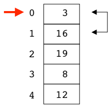


The smallest value so far is 3

Its index is 1

- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value
- Once we've found the minimum value
  - swap that value with the one we selected at beginning

Selection sort

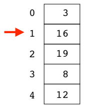


The smallest value so far is 3

Its index is 1

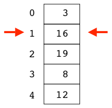
- Let's say want to sort array values in increasing order
  - one way to approach problem is to use algorithm called selection sort
- Start by setting pointer to first element in array
  - this is where smallest value in array will be placed
- Then look at every value in this unsorted array
  - find minimum value
- Once we've found the minimum value
  - swap that value with the one we selected at beginning

Selection sort



- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value

Selection sort

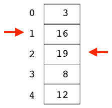


The smallest value so far is 16

Its index is 1

- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element

Selection sort

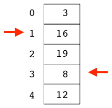


The smallest value so far is 16

Its index is 1

- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element

Selection sort

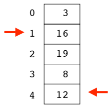


The smallest value so far is 8

Its index is 3

- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element

Selection sort

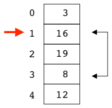


The smallest value so far is 8

Its index is 3

- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element

Selection sort

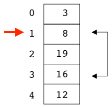


The smallest value so far is 8

Its index is 3

- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element
- Now swap minimum value with selected array value
  - in this case, second element

Selection sort

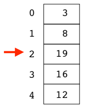


The smallest value so far is 8

Its index is 3

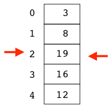
- At this point we know
  - smallest number in array is in first element (index 0)
  - first element is sorted
  - rest of array remains unsorted
- Now select second element of array to be location which will hold next smallest value
- In other words, do everything again to unsorted part of array
  - in this case, all but first element
- Now swap minimum value with selected array value
  - in this case, second element

Selection sort



- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before

Selection sort

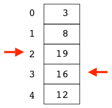


The smallest value so far is 19

Its index is 2

- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before

Selection sort

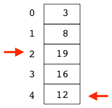


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before

The smallest value so far is 16

Its index is 3

Selection sort

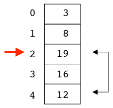


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before

The smallest value so far is 12

Its index is 4

Selection sort

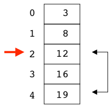


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before
- Again, swap values

The smallest value so far is 12

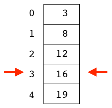
Its index is 4

Selection sort



- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before
- Again, swap values

Selection sort

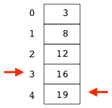


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before
- Again, swap values
  - then do whole thing again

The smallest value so far is 16

Its index is 3

Selection sort

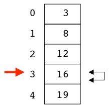


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before
- Again, swap values
  - then do whole thing again

The smallest value so far is 16

Its index is 3

Selection sort

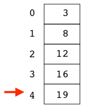


- Now first two elements of array are sorted
- Select third element of array to be location of next smallest value
  - Search unsorted portion of array for that value, just like before
- Again, swap values
  - then do whole thing again
- Swap again
  - not actually necessary in this case
  - but we follow algorithm

The smallest value so far is 16

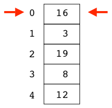
Its index is 3

Selection sort



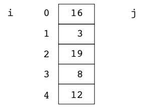
- Are we done?
  - could select last element of array
    - (index 4)
  - but all of array except for last element is already sorted
  - so last element is largest value in array
    - and that's the right place
- Yes, array is sorted, and we're done
  - no need to select last element

Selection sort



- Showed arrows moving down array
  - arrow on left represents one array index variable
  - arrow on right represents different one
- Consider variables being controlled by loop
  - red arrow shows outer loop
  - green arrow shows inner loop inside outer loop
- Nested loop structure again

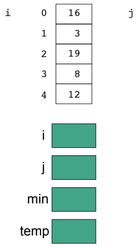
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

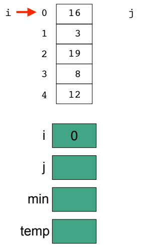
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

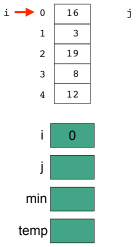
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

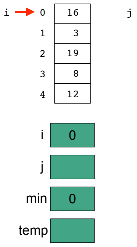
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

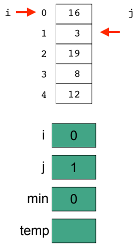
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

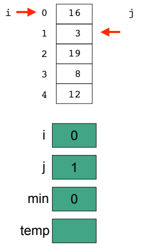
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

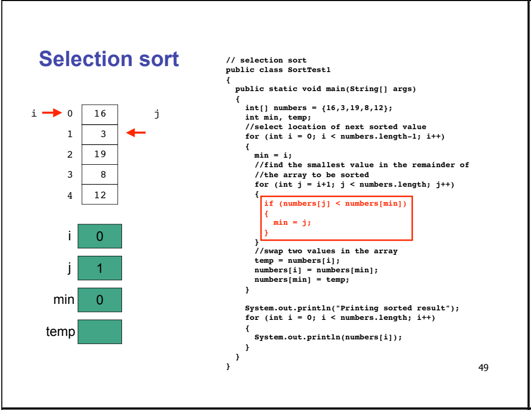
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

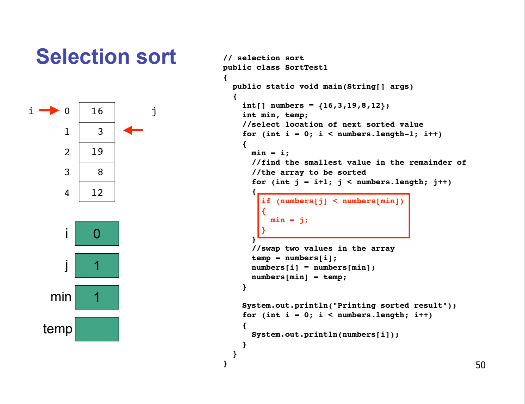
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

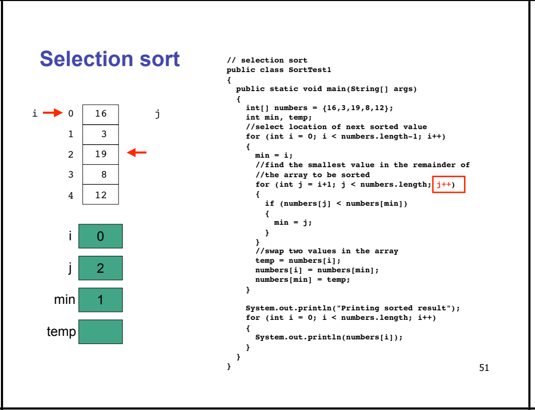
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

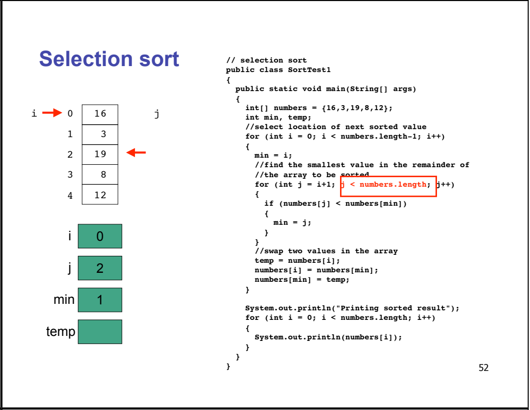
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

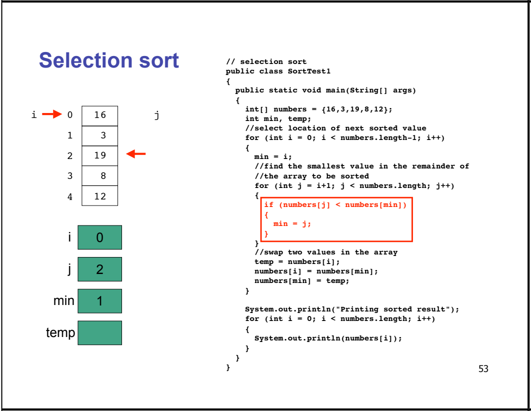
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

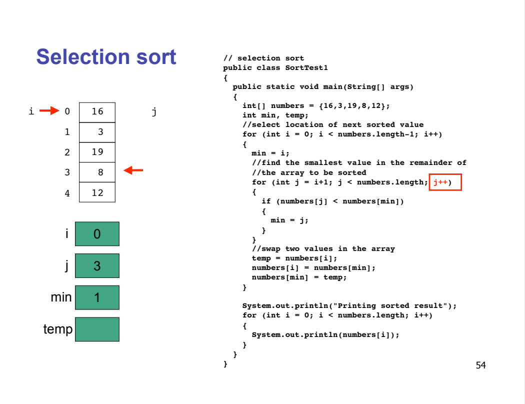
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

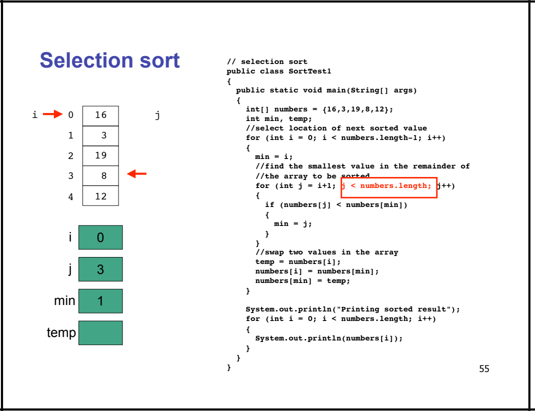
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

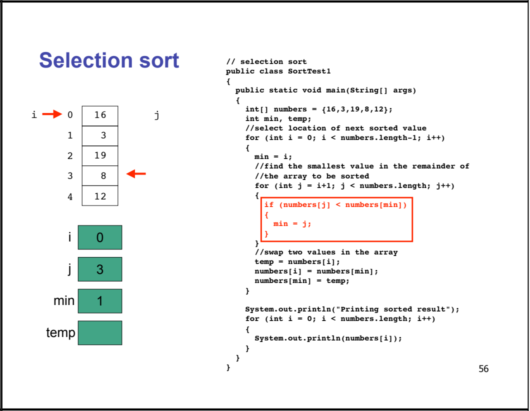
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

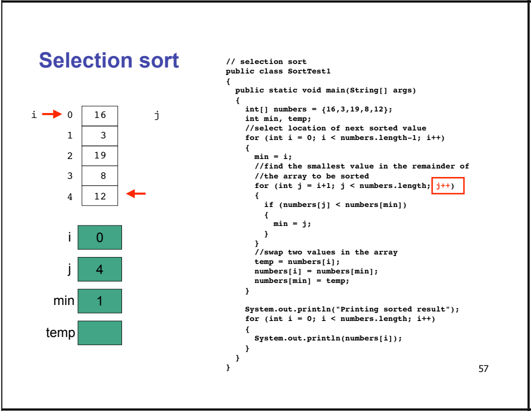
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

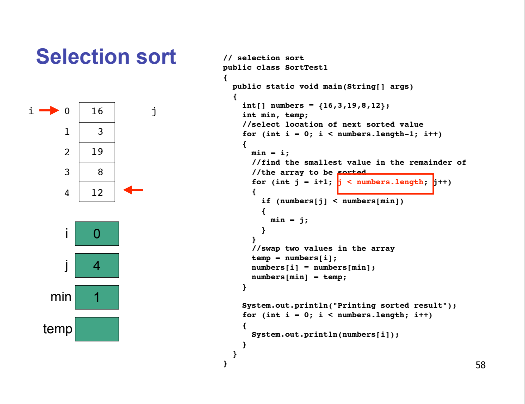
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

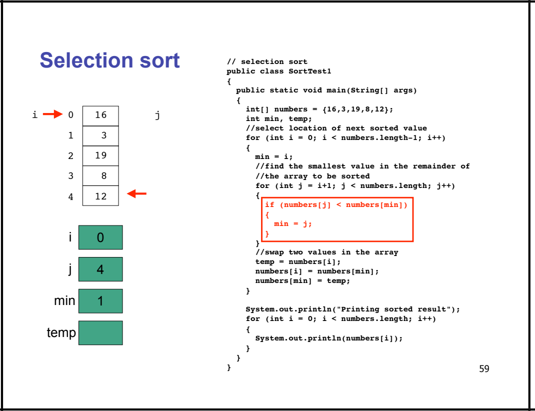
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

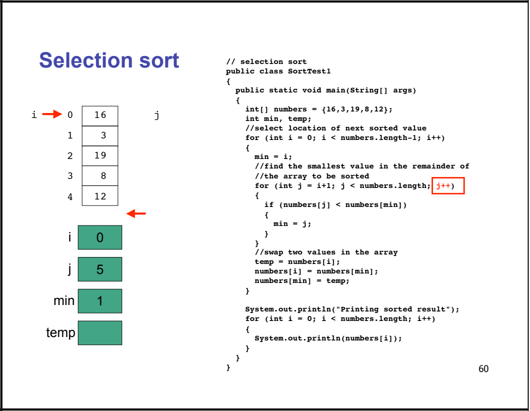
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

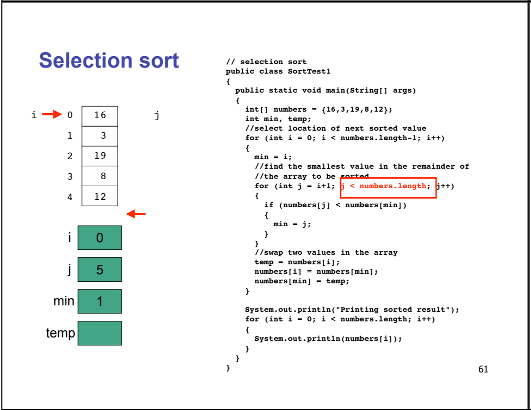
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

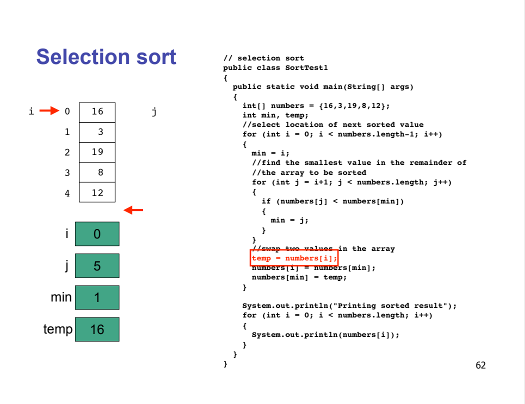
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

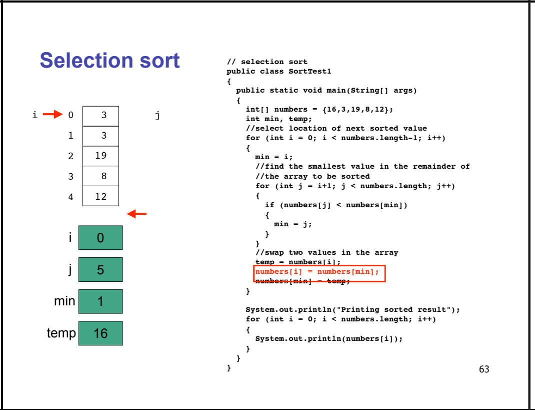
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

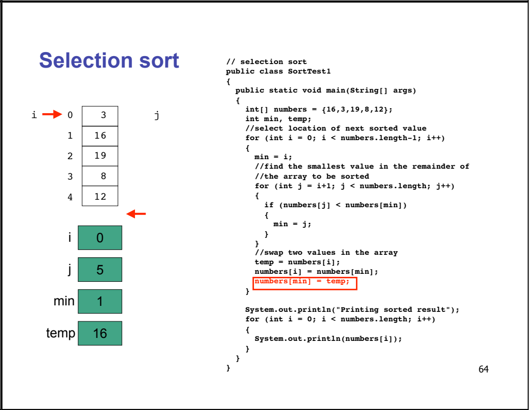
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

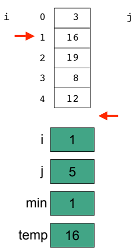
Selection sort



```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



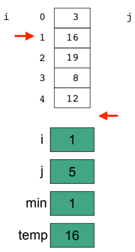
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



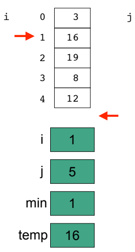
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



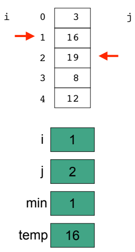
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



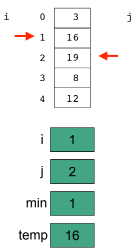
i			j
	0	3	
→	1	16	
	2	19	←
	3	8	
	4	12	

i	1
j	2
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



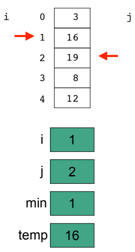
i			j
	0	3	
→	1	16	
	2	19	←
	3	8	
	4	12	

i	1
j	2
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



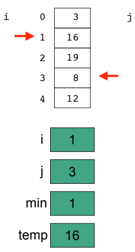
i			j
	0	3	
→	1	16	
	2	19	←
	3	8	
	4	12	

i	1
j	2
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



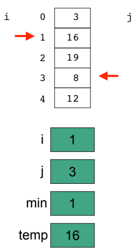
i			j
	0	3	
→	1	16	
	2	19	
	3	8	←
	4	12	

i	1
j	3
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



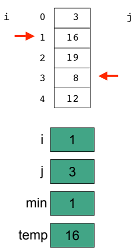
i			j
	0	3	
→	1	16	
	2	19	
	3	8	←
	4	12	

i	1
j	3
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



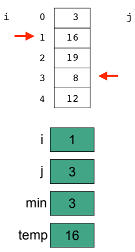
i			j
	0	3	
→	1	16	
	2	19	
	3	8	←
	4	12	

i	1
j	3
min	1
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



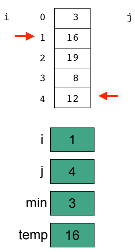
i			j
	0	3	
→	1	16	
	2	19	
	3	8	←
	4	12	

i	1
j	3
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



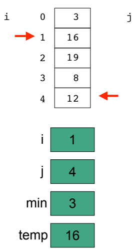
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	←

i	1
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



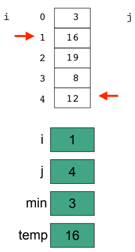
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	←

i	1
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



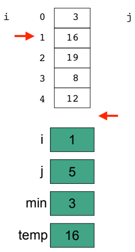
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	←

i	1
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



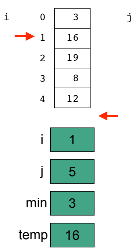
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



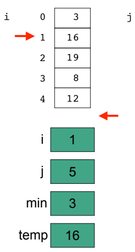
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



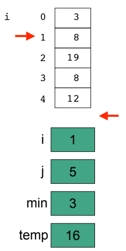
i			j
	0	3	
→	1	16	
	2	19	
	3	8	
	4	12	
			←

i	1
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



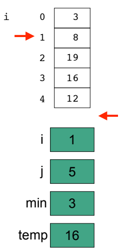
0	3
1	8
2	19
3	8
4	12

i	1
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



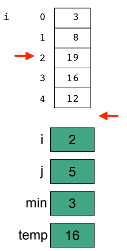
0	3
1	8
2	19
3	16
4	12

i	1
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



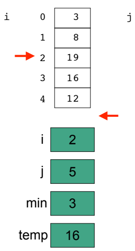
0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



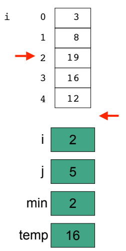
0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



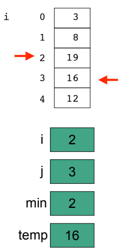
0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	2
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



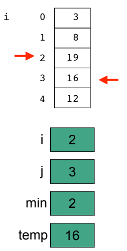
0	3
1	8
2	19
3	16
4	12

i	2
j	3
min	2
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



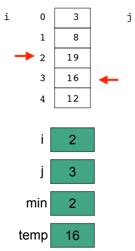
0	3
1	8
2	19
3	16
4	12

i	2
j	3
min	2
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



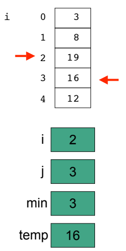
0	3
1	8
2	19
3	16
4	12

i	2
j	3
min	2
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



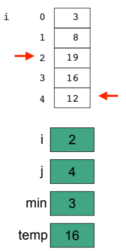
0	3
1	8
2	19
3	16
4	12

i	2
j	3
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



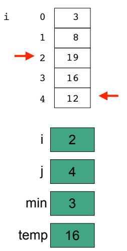
0	3
1	8
2	19
3	16
4	12

i	2
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



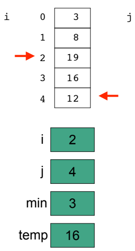
0	3
1	8
2	19
3	16
4	12

i	2
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



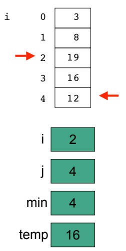
0	3
1	8
2	19
3	16
4	12

i	2
j	4
min	3
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



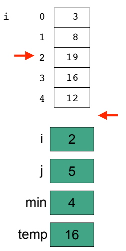
0	3
1	8
2	19
3	16
4	12

i	2
j	4
min	4
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



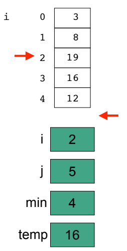
0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	4
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort



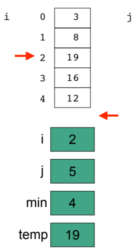
0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	4
temp	16

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Selection sort

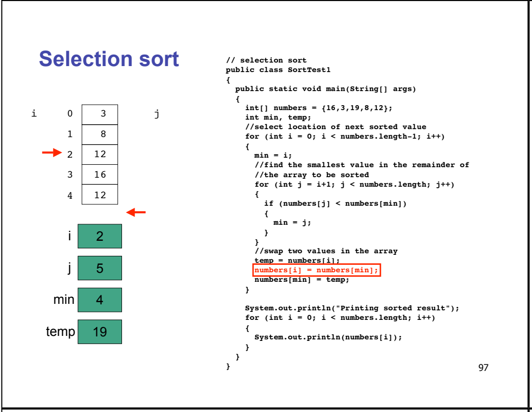


0	3
1	8
2	19
3	16
4	12

i	2
j	5
min	4
temp	19

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

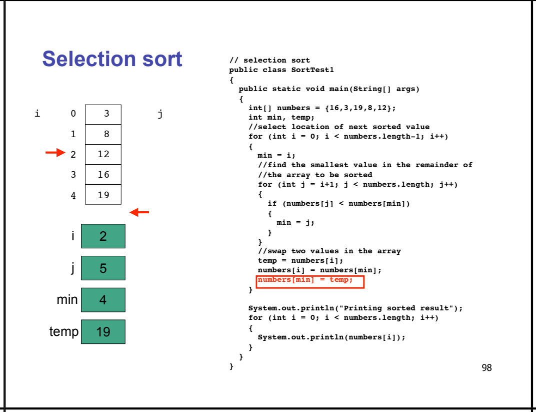


Selection sort

```
i     0   3    j
      1   8
   →  2   12
      3   16
      4   12
               ←
i     2
j     5
min   4
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

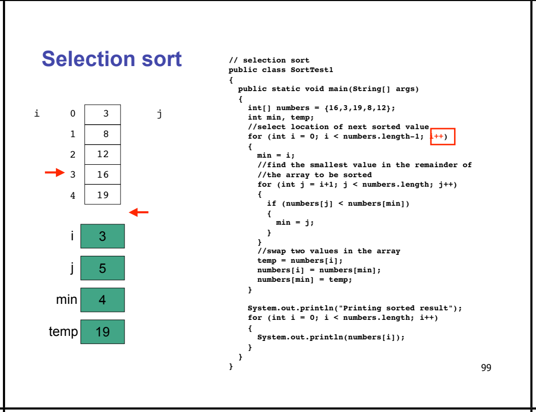


Selection sort

```
i     0   3    j
      1   8
   →  2   12
      3   16
      4   19
               ←
i     2
j     5
min   4
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

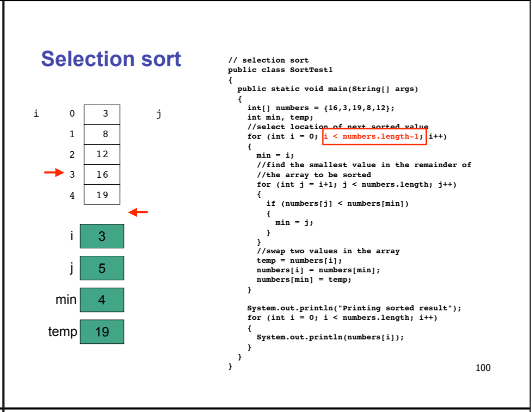


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   4
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

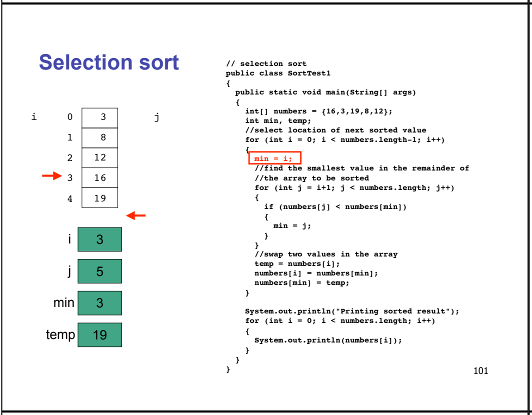


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   4
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

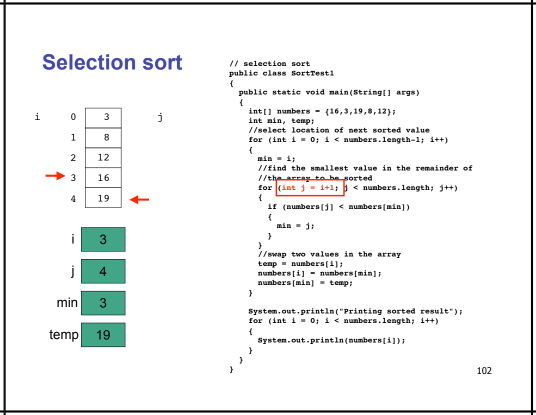


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

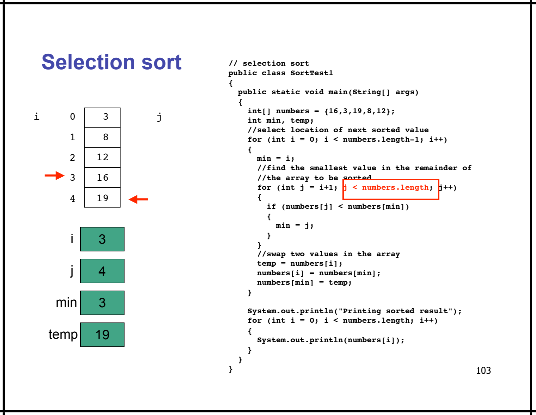


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19   ←
i     3
j     4
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

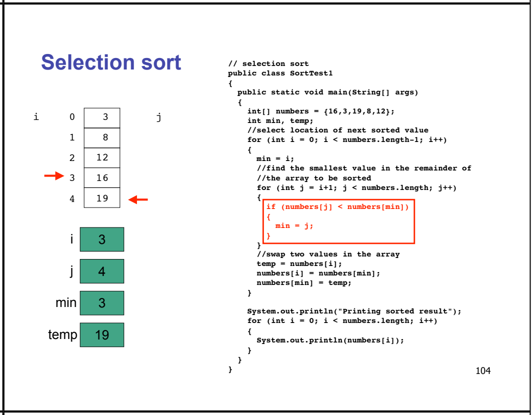


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19   ←
i     3
j     4
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

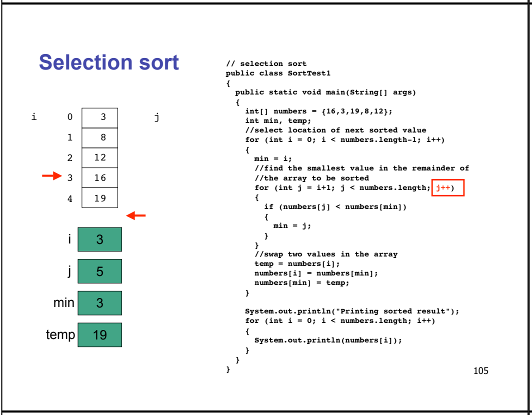


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19   ←
i     3
j     4
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

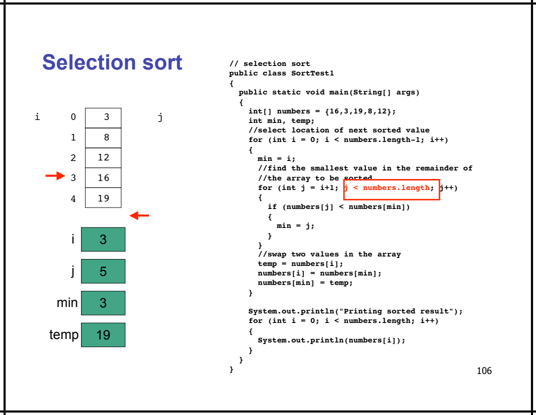


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

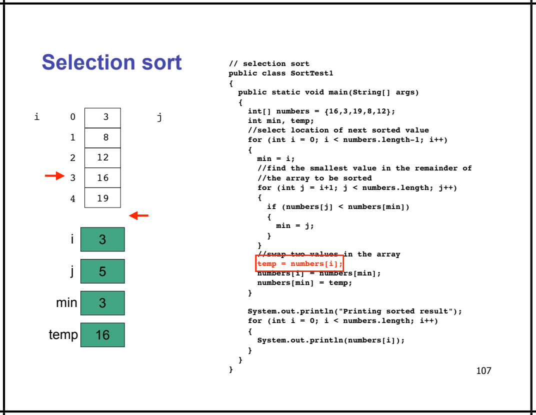


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  19
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

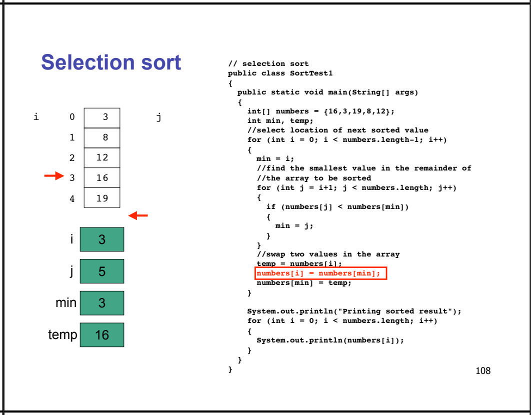


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

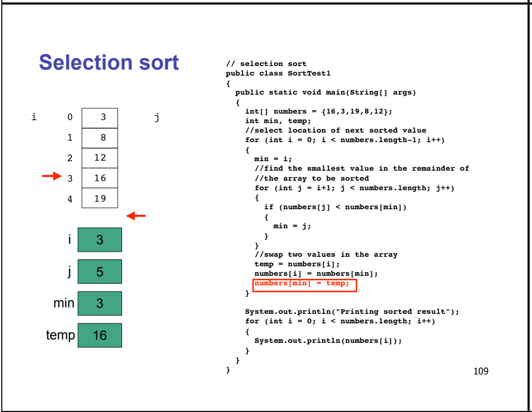


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

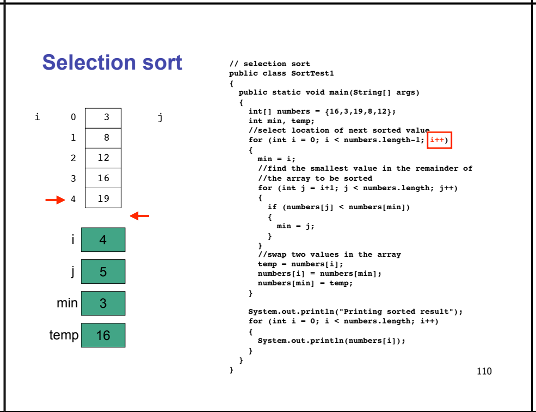


Selection sort

```
i     0   3    j
      1   8
      2   12
   →  3   16
      4   19
               ←
i     3
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

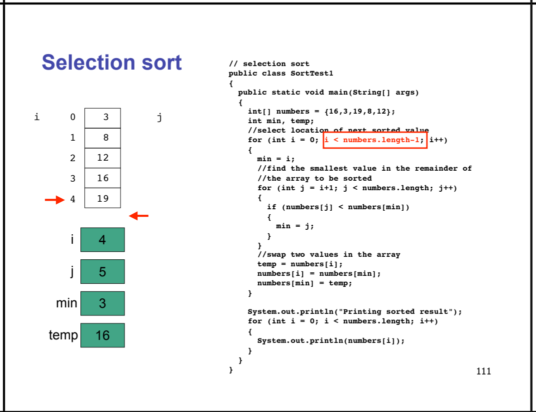


Selection sort

```
i     0   3    j
      1   8
      2   12
      3   16
   →  4   19
               ←
i     4
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

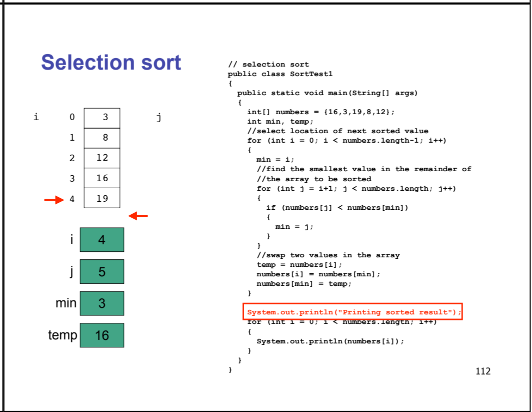


Selection sort

```
i     0   3    j
      1   8
      2   12
      3   16
   →  4   19
               ←
i     4
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```



Selection sort

```
i     0   3    j
      1   8
      2   12
      3   16
   →  4   19
               ←
i     4
j     5
min   3
temp  16
```

```
// selection sort
public class SortTest1
{
  public static void main(String[] args)
  {
    int[] numbers = {16,3,19,8,12};
    int min, temp;
    //select location of next sorted value
    for (int i = 0; i < numbers.length-1; i++)
    {
      min = i;
      //find the smallest value in the remainder of
      //the array to be sorted
      for (int j = i+1; j < numbers.length; j++)
      {
        if (numbers[j] < numbers[min])
        {
          min = j;
        }
      }
      //swap two values in the array
      temp = numbers[i];
      numbers[i] = numbers[min];
      numbers[min] = temp;
    }

    System.out.println("Printing sorted result");
    for (int i = 0; i < numbers.length; i++)
    {
      System.out.println(numbers[i]);
    }
  }
}
```

Tracing with the Debugger