Code Explanation for Selection Sort

In the attached SelectionSorter class, you can see the code for selection sort.

```
public void sort(int[] data) {
	for (int i = 0; i < data.length-1; i++) {
		int minIndex = i;
		for (int j = i+1; j < data.length; j++) {
			if (data[j] < data[minIndex]) {
				minIndex = j;
			}
		}
		int tmp = data[minIndex];
		data[minIndex] = data[i];
		data[i] = tmp;
	}
}
```

We can break this method down into parts. One method used just to find the index of the minimum number, another method just to swap the minimum number with the required index. Smaller methods also contribute to better, cleaner and more readable code.

So the sort method goes from the 1st element to the second last element. It first finds the index of the element which is the smallest within the array (NOT THE WHOLE ARRAY)

```
public void sort(int[] data) {
	for (int i = 0; i < data.length-1; i++) {
		int minIndex = findIndexOfMinElement(i, data);
		swapItems(i, minIndex, data);
	}
}
```

Let's look at the findIndexOfMinElement. The startIndex parameter tells this method to find the minimum number in the array starting from that index, and ignoring the elements before it. Of course we will need to pass the data, which is the second param.

```
private int findIndexOfMinElement(int startIndex, int[] data) {
	int minIndex = startIndex;
	for (int j = minIndex+1; j < data.length; j++) {
		if (data[j] < data[minIndex]) {
			minIndex = j;
		}
	}
	return minIndex;
}
```

The swap items method is quite simple. In the data array, we want to swap items at index1 and index2, which can be done using a temporary variable.

```
private void swapItems(int index1, int index2, int[] data) {
      int tmp = data[index1];
      data[index1] = data[index2];
      data[index2] = tmp;
}
```