

## Aufgabe – Sortieren

Es sei folgendes Array gegeben:

```
int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0, 22, 12, 16, 18, 11, 19, 13 };
```

1. Geben Sie das obige Array aufsteigend sortiert aus
2. Geben Sie das obige Array absteigend sortiert aus
3. Geben Sie aus dem obigen Array alle geraden Zahlen aufsteigend sortiert aus

Es sei folgendes Array gegeben:

```
string[] numbers =  
{ "zero", "one", "two", "three", "four", "five", "six", "seven", "eight",  
  "nine", "ten", "eleven", "twelve", "thirteen", "fourteen" };
```

1. Geben Sie das obige Array nach der Länge der Worte aufsteigend sortiert aus
2. Geben Sie das obige Array nach der Länge der Worte aufsteigend sortiert aus, bei gleicher Länge soll alphabetisch absteigend sortiert werden
3. Drehen Sie die Reihenfolge der Elemente im Array um

Erstellen Sie ein `DirectoryInfo`-Objekt für ein Verzeichnis Ihrer Wahl.

4. Listen Sie alle Dateien in dem Verzeichnis, absteigend nach Namen sortiert auf
5. Listen Sie alle Dateien in dem Verzeichnis, nach Größe aufsteigend sortiert auf
6. Listen Sie alle Dateien in dem Verzeichnis, nach dem Datum des letzten Zugriffs auf, jüngste Dateien zuerst

## Aufgabe – Partitionierung

Es sei folgendes Array gegeben:

```
int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0, 22, 12, 16, 18, 11, 19, 13 };
```

Ermitteln Sie mittels LINQ-Ausdrücken die folgenden Informationen:

1. Ermitteln Sie die ersten 5 Elemente im Array
2. Ermitteln Sie die letzten 5 Elemente im Array
3. Ermitteln Sie alle Elemente, bis auf die ersten und letzten drei Elemente
4. Ermitteln Sie alle Elemente, beginnend vom ersten Element, die größer als 0 sind
5. Ermitteln Sie alle Elemente, beginnend vom ersten Element, die nach der 12 im Array stehen

Erstellen Sie ein `DirectoryInfo`-Objekt für ein Verzeichnis Ihrer Wahl.

7. Listen Sie die fünf neuesten Dateien in dem Verzeichnis auf
8. Listen Sie alle Dateien in dem Verzeichnis in „Seiten“ zu je 5 Dateien auf