

### Lehrziele:

- Arrays
- Methoden
- UnitTest

### Aufgabenstellung

Ihr Programm errechnet aus zwei int-Arrays ein drittes **Ergebnisarray**, das alle Ziffern enthält, die sowohl im ersten Array als auch im zweiten Array enthalten sind.

Am Beginn der Methode Main sind zwei int-Arrays definiert, die als Eingabedaten für unser Programm dienen:

- Normalfall

```
int[] numbersA = { 1, 3, 5, 7, 9 };  
int[] numbersB = { 0, 1, 2, 3 };
```

```
Ziffern Und  
=====  
Feld A: 1 3 5 7 9  
Feld B: 0 1 2 3  
Ergebnis enthält 2 Ziffern  
Ziffern in Feld A und Feld B: 1 3
```

- Alle Ziffern gemeinsam

```
int[] numbersA = { 1, 3, 5 };  
int[] numbersB = { 1, 3, 5 };
```

```
Feld A: 1 3 5  
Feld B: 1 3 5  
Ergebnis enthält 3 Ziffern  
Ziffern in Feld A und Feld B: 1 3 5
```

- Keine gemeinsamen Ziffern

```
int[] numbersA = { 1, 3, 5, 7, 9 };  
int[] numbersB = { 0, 2, 4, 6, 8 };
```

```
Feld A: 1 3 5 7 9  
Feld B: 0 2 4 6 8  
Ergebnis enthält 0 Ziffern  
Ziffern in Feld A und Feld B:
```

Sie können die Definition der beiden Ausgangsarrays direkt in Ihr Programm kopieren.

Die Berechnung der „Intersect“ erfolgt in einer eigenen Methode (In der Datei IntersectTools.cs mit dem Namen Intersect). Dieser werden zwei Arrays übergeben und als Ergebnis liefert die Methode ebenfalls ein Int Array.

In dem bereitgestellten Programm-Template sind `UnitTests` vorhanden. Implementieren Sie die Methode so, dass **alle** Unittests erfolgreich ausgeführt werden können. Änderungen an den Unittests sind nicht erlaubt.