

### Lehrziele:

- Arrays
- Methoden
- UnitTest

### Aufgabenstellung

Ihr Programm errechnet aus zwei int-Arrays ein drittes **Ergebnisarray**.

Dabei wird unterschieden:

- 1.) Das Ergebnisarray enthält alle Zahlen aus dem ersten Array und alle Zahlen aus dem zweiten Array (Union).  
Beispiel: {1,2,3} mit {3,4,5} => {1,2,3,4,5}
- 2.) Das Ergebnisarray enthält nur die Zahlen aus dem ersten Array die nicht im zweiten Array vorkommen (Except).  
Beispiel: {1,2,3,4,5,2} mit {3,4,5,6} => {1,2}

Hinweis:

- Das Ergebnis enthält eine Zahl immer nur einmal (distinct).
- Das Ergebnisarray darf beliebig sortiert sein.

Zum Testen der Methoden implementieren Sie das folgendes Hauptprogramm (keine Eingabe, die Arrays sind fix im Programm hinterlegt).

### BildschirmAusgabe:

```
Union
=====
Normalfall
Feld A: 1 1 3 5 7 9 1 3
Feld B: 0 1 2 3
Ziffern in Feld A und Feld B: 1 3 5 7 9 0 2
=====
Alle gleich
Feld A: 1 1 3 5
Feld B: 1 3 3 5
Ziffern in Feld A und Feld B: 1 3 5
=====
Keine Überschneidung
Feld A: 1 3 5 7 9
Feld B: 0 2 4 6 8
Ziffern in Feld A und Feld B: 1 3 5 7 9 0 2 4 6 8
=====
Except
=====
Normalfall
Feld A: 1 1 3 5 7 9 1 3
Feld B: 0 1 2 3
Ziffern in Feld A und Feld B: 5 7 9
```

```
=====
Alle gleich
Feld A: 1 1 3 5
Feld B: 1 3 3 5
Ziffern in Feld A und Feld B:
=====
Keine Überschneidung
Feld A: 1 3 5 7 9
Feld B: 0 2 4 6 8
Ziffern in Feld A und Feld B: 1 3 5 7 9
=====
```

Die Berechnung der „Union“ und „Except“ erfolgt in je einer eigenen Methode (In der Datei SetTools.cs). Den Methoden werden zwei Arrays übergeben und als Ergebnis liefern die Methoden 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.