

Modul 24

Objektorientierte Programmierung



VITRUVIUS
HOCHSCHULE

Oliver Ziegler

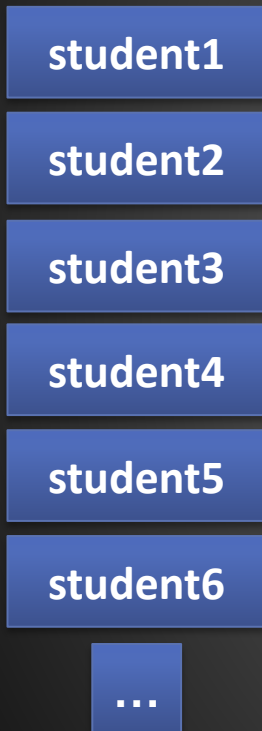
Felder und Schleifen

- Idee von Feldern (Arrays)
- Benutzung
- Arrays in Code
- Schleifen

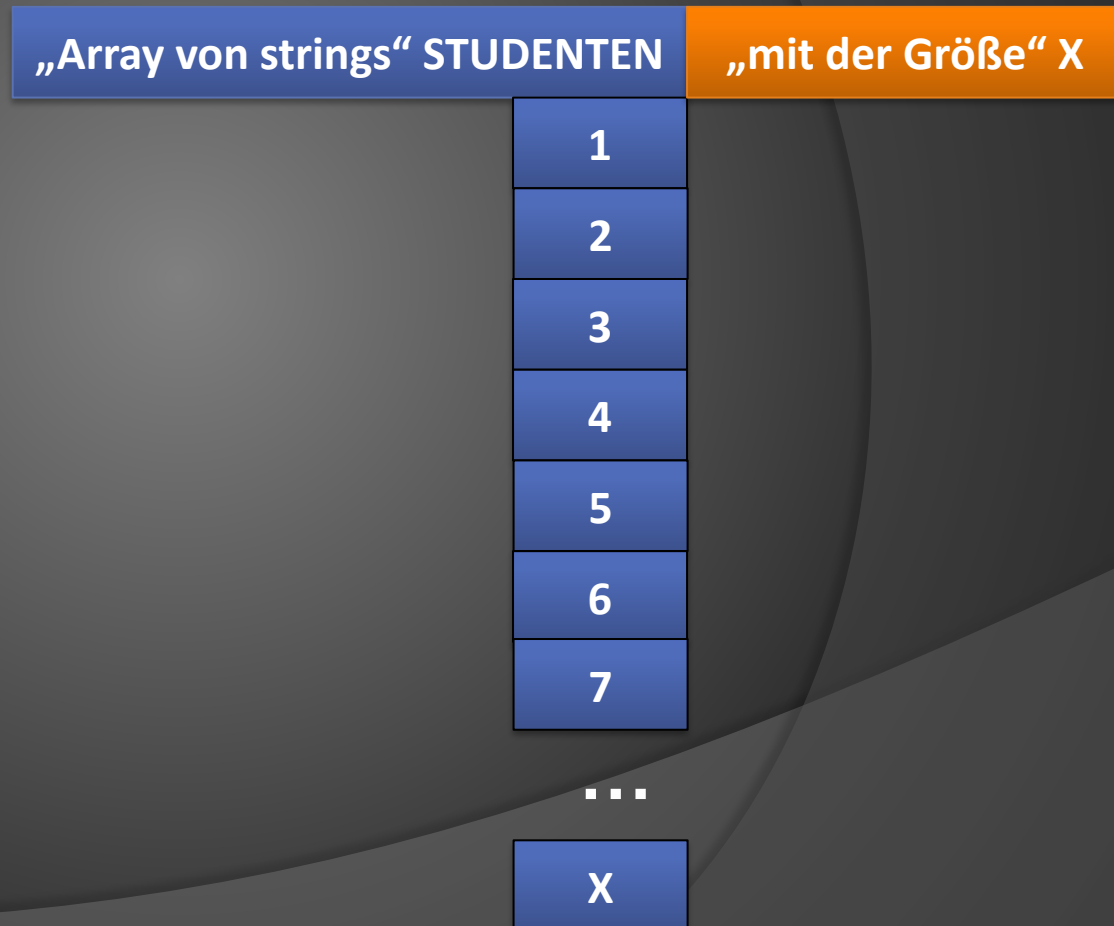
Felder (Arrays)

- Speichern mehrerer Variablen unter einer Adresse

Statt:



Mit Feldern:



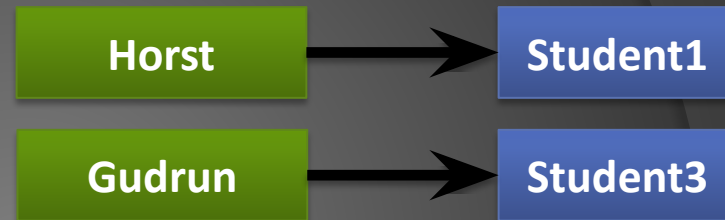
Felder (Arrays)

- Ändern von Daten durch Nummer (Index) im Feld

Statt:

Student1 = „Horst“

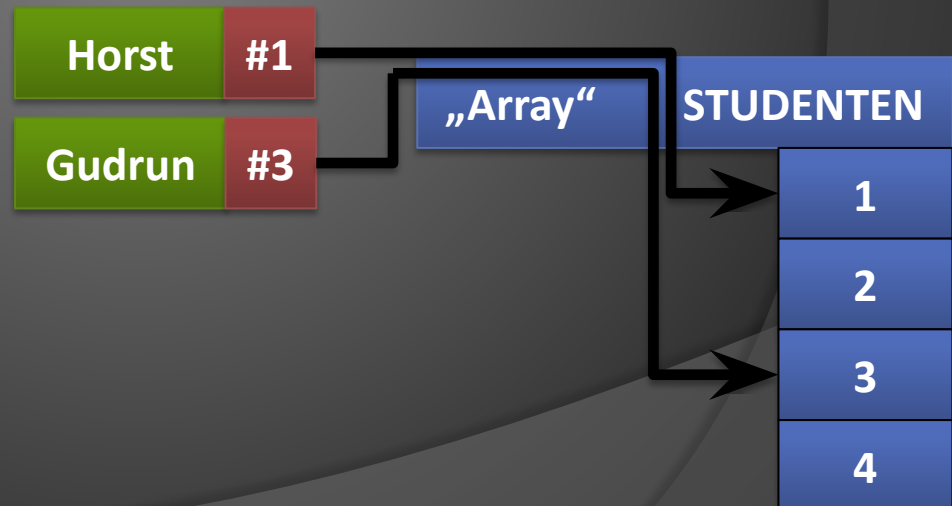
Student3 = „Gudrun“



Dann:

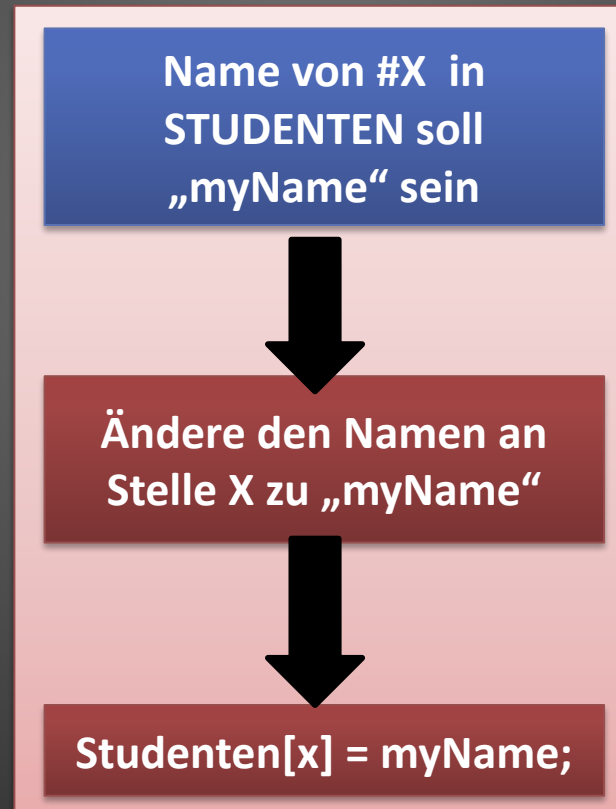
#1 in STUDENTEN = „Horst“

#3 in STUDENTEN = „Gudrun“



Arrays

- Allgemeineres Denken beim Setzen von Array-Werten



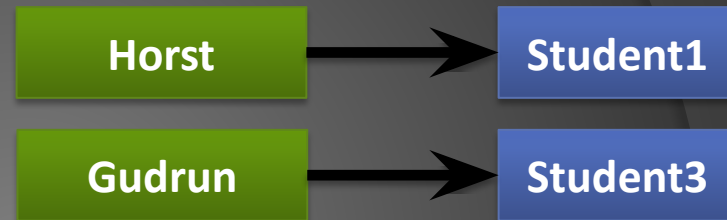
Felder (Arrays)

- Ändern von Daten durch Nummer (Index) im Feld

Statt:

Student1 = „Horst“

Student3 = „Gudrun“



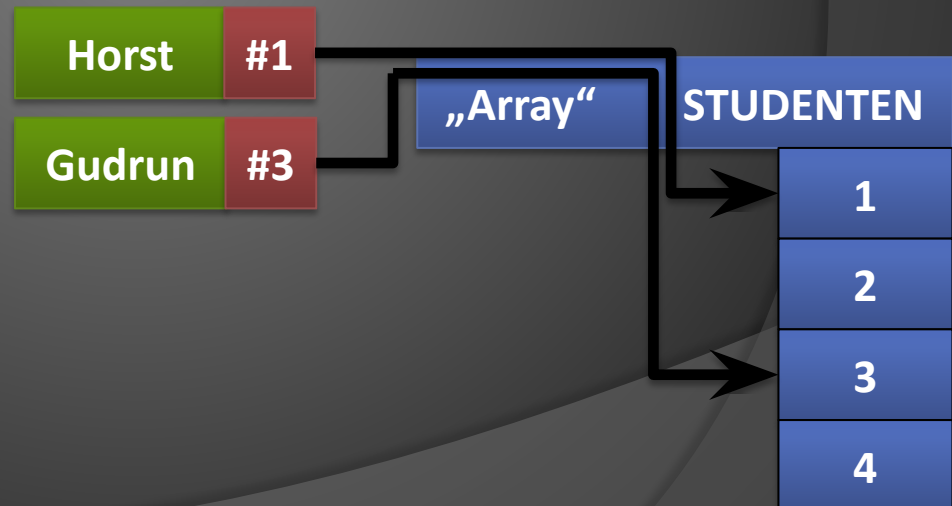
Dann:

#1 in STUDENTEN = „Horst“

#3 in STUDENTEN = „Gudrun“

Studenten[1] = „Horst“;

Studenten[3] = „Gudrun“;



Nutzung von Arrays

Allein als Speichermedium sind Arrays schon interessant

- **Einfaches Anlegen von vielen Variablen desselben Typs**
- **Große Datenmengen, können mit einer Variable an Funktionen gegeben werden (Parameter)**
- **Große Datenmengen können mit einer Zeile gelöscht werden (Lösche Array)**

Felder (Arrays)

- Befehle können an alle Einträge eines Arrays gegeben werden

Statt:

Student1 = „Horst“

Student3 = „Gudrun“

Dann:

#1 in STUDENTEN = „Horst“

#3 in STUDENTEN = „Gudrun“

Aber auch:

Statt:

E-Mail an Student1

E-Mail an Student2

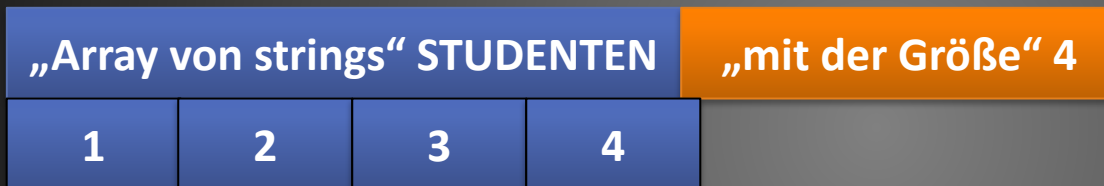
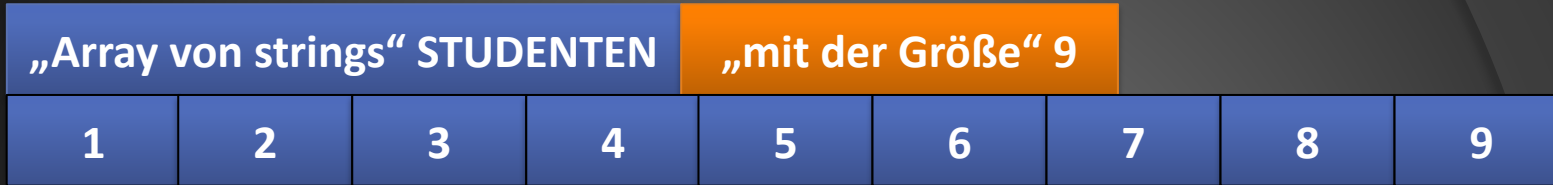
E-Mail an Student3

Schicke E-Mail an

ALLE in STUDENTEN

Felder (Arrays)

- Größe der Variable kann unterschiedlich sein



Liefert immer das
korrekte Ergebnis

Felder (Arrays)

- Spielrelevanz:
Quest – Töte alle Gegner

| „Array“ Gegner | | | |
|----------------|---|---|---|
| 1 | 2 | 3 | 4 |

Statt:

| |
|------------------------------|
| Wenn Gegner1.gesundheit <= 0 |
| UND |
| Wenn Gegner2.gesundheit <= 0 |
| UND |
| Wenn Gegner3.gesundheit <= 0 |
| UND |
| Wenn Gegner4.gesundheit <= 0 |

Jetzt:

| |
|----------------------|
| Wenn |
| Für ALLE in „Gegner“ |
| gesundheit <= 0 |

Felder (Arrays)

- Zählung beginnt bei 0!

| string[4] studenten | | | |
|---------------------|---|---|---|
| 0 | 1 | 2 | 3 |

Felder (Arrays)

```
public void Start()
{
    int[] zahlen = new int[5];    // Array-Größe „5“ definiert

    zahlen[0] = 6565;           // Erstes Element gesetzt auf 6565

    zahlen = new int[15];       // Array neu angelegt - Größe „15“ definiert

    Debug.Log( „Element an Stelle 0 = “ + zahlen[0] );
    //Ausgabe: „Element an Stelle 0 = 0“
}
```

Felder (Arrays)

```
public void Start()
{
    int[] zahlen = { 3,8,25,11 }    // Array der Länge 4 mit Werten

    Debug.Log( „Element an Stelle 3 = “ + zahlen[3] );
    //Ausgabe: „Element an Stelle 3 = 11“

    // Zählung beginnt bei 0 !!
}
```

Felder (Arrays)

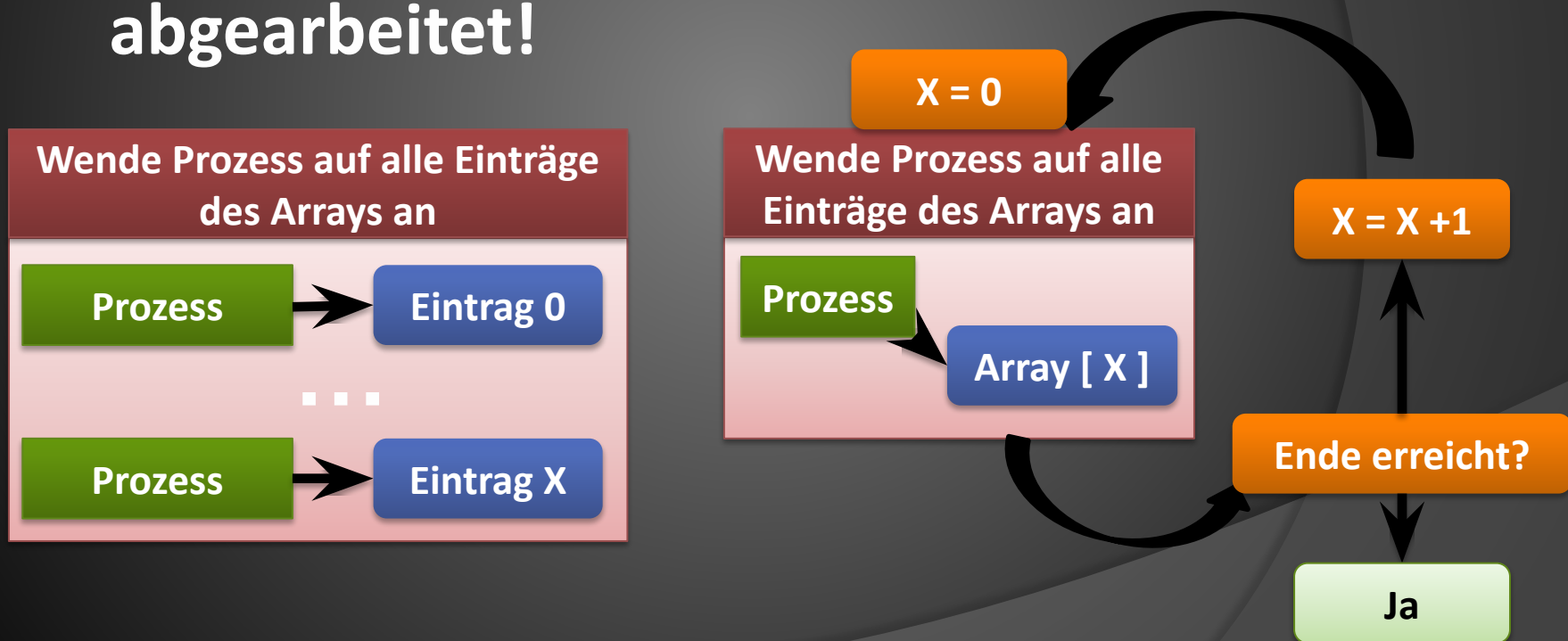
```
public void Start()
{
    string[] texte = { „Bla“ , „Keks“ , „Test“ }    // Array der Länge 3 mit Werten

    Debug.Log( „Element an Stelle 2 = “ + texte [2] );
    //Ausgabe: „Element an Stelle 2 = Test“

    // Zählung beginnt bei 0 !!
}
```

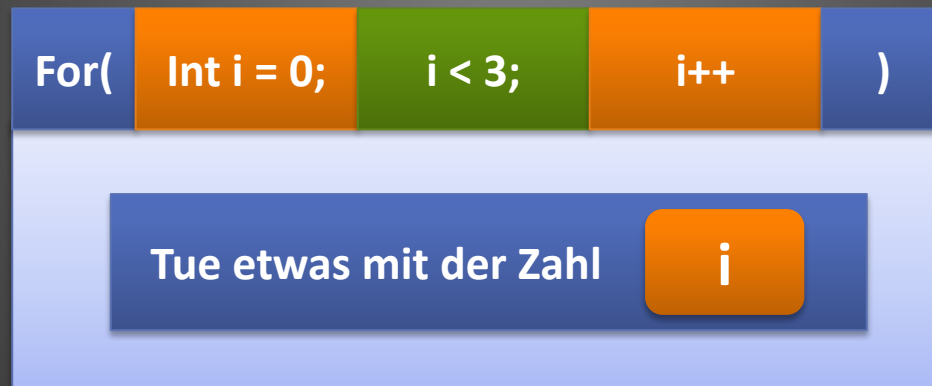
Schleifen

- Dienen dazu immer gleich Prozesse auf eine größere Datenmenge anzuwenden
- Werden in einem Frame abgearbeitet!

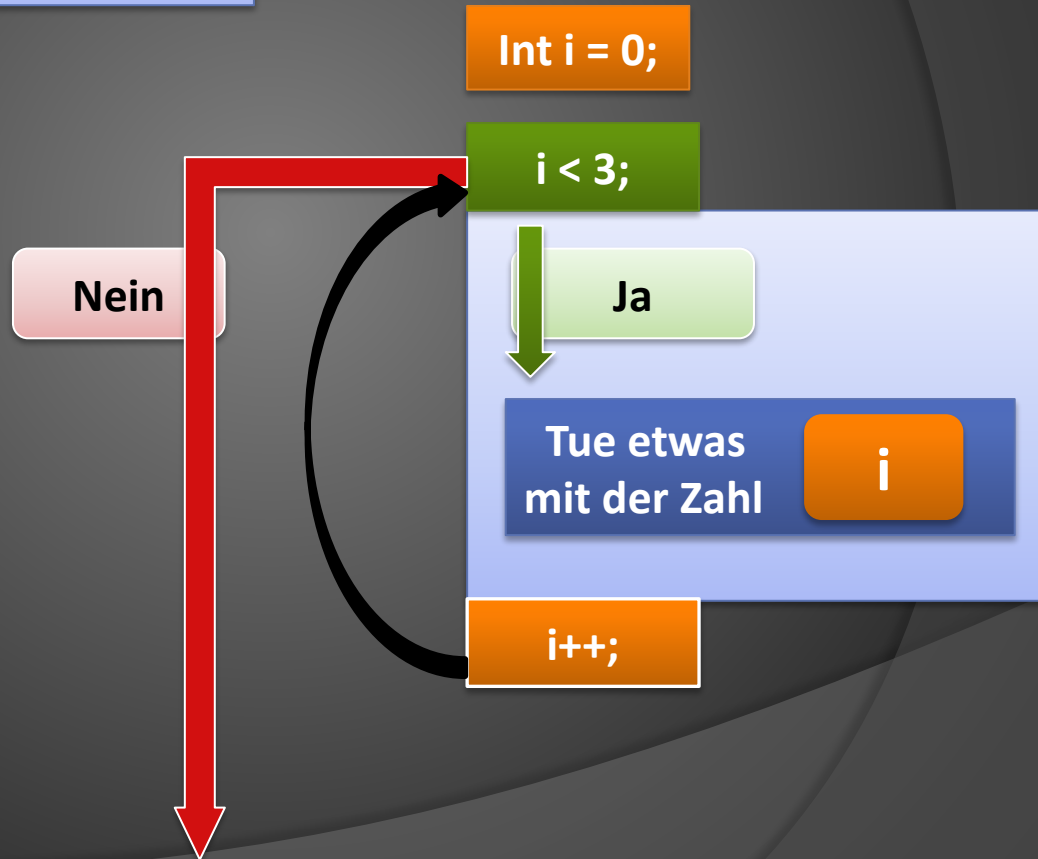
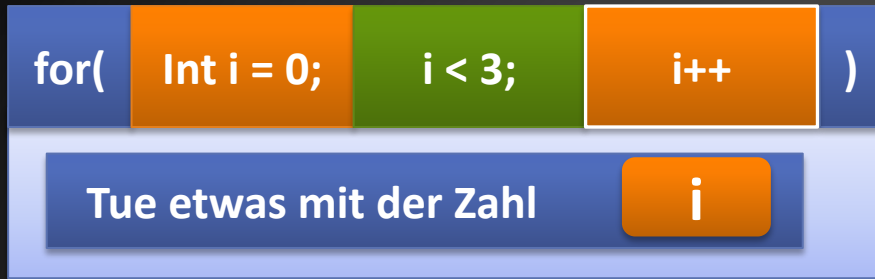


For-Schleife

- Beginnt bei einem Wert
- Prüft eine Bedingung
- Führt danach einen/mehrere Befehl(e) aus



For-Schleife



For-Schleife

```
public int[] zahlen;    // In Inspector gesetzt
public int summe;

public void Start()
{
    for (int i = 0; i < 3; i = i+1)
    {
        summe = summe + i;    // Addiert die ersten drei Zahlen
    }
    Debug.Log(„Summe = “ + x);
}
```

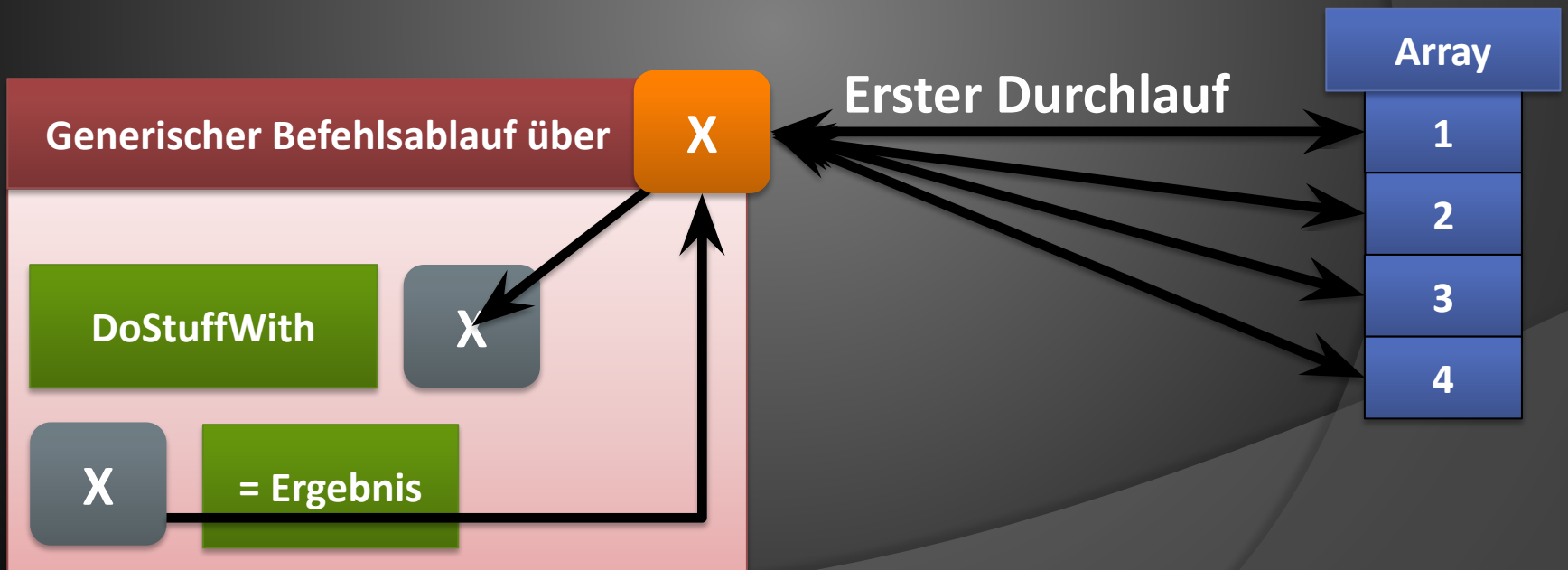
For-Schleife

```
public int[] zahlen;      // In Inspector gesetzt
public int summe;

public void Start()
{
    int lengthOfArray = zahlen.Length;
    for (int i = 0; i < lengthOfArray; i = i+1)
    {
        summe = summe + i;    // Addiert alle Zahlen
    }
    Debug.Log(„Summe = “ + x);
}
```

Foreach-Schleife

- Geht das Feld durch und speichert jeden Wert in einen generischen Wert
- Damit die Funktionalität allgemein geschrieben werden kann



Foreach-Schleifen

```
public int[] zahlen;      // In Inspector gesetzt
public int summe;

public void Start()
{
    foreach(int x in zahlen)
    {
        summe = summe + x;
    }
    Debug.Log(„Summe = “ + x);
}
```