

```
int GrootsteGemeneDeler(int getal1, int getal2)
{
    while (getal2 != 0)
    {
        int rest = getal1 % getal2;
        getal1 = getal2;
        getal2 = rest;
    }

    return getal1;
}

static void Main(string[] args)
{
    int getal1 = Int32.Parse(Console.ReadLine());
    int getal2 = Int32.Parse(Console.ReadLine());
    int ggd = GrootsteGemeneDeler(getal1, getal2);
    Console.WriteLine("{0} is de GGD van {1} en {2}.",
        ggd, getal1, getal2);
    Console.ReadKey();
}
```

CO
- ORIGIN
'warn'.

monitor

thing 2
duties 3
picture

Programmeren 1 (C#)

Gerwin van Dijken (gerwin.vandijken@inholland.nl)

Programma periode 1.1 (Programmeren 1)

01 (wk-36)	Inleiding / Visual Studio 2017/2019
02 (wk-37)	Sequentie
03 (wk-38)	Selectie
04 (wk-39)	Iteratie
05 (wk-40)	Array's
06 (wk-41)	Methoden
07 (wk-42)	Herhaling / oefententamen
08 (wk-43)	<i>roostervrije week</i>
09 (wk-44)	praktijktentamen (<i>computer opdrachten</i>)
10 (wk-45)	-

Selectie opdracht 4

- Lees drie getallen in. Toon som, gemiddelde, product, grootste en kleinste van de getallen.

Selectie opdracht 4

use 'sum'
here!

```
read number1, number2, number3
sum = number1 + number2 + number3
average = sum / 3
product = number1 * number2 * number3
highest = number1
if (number2 > highest)
    highest = number2
if (number3 > highest)
    highest = number3
lowest = number1
if (number2 < lowest)
    lowest = number2
if (number3 < lowest)
    lowest = number3
display sum, average, product, highest, lowest
```

Selectie opdracht 9

De hoogte van de contributie van een sportvereniging is afhankelijk van de sport die men beoefent, de leeftijd en het aantal jaren dat men lid is. Gegeven:

- de contributie voor voetbal bedraagt € 175,00 en voor handbal € 225,00;
- ouder dan 40 jaar? dan € 25,00 korting;
- langer dan 10 jaar lid? dan € 20,00 korting;

Gelezen wordt: sportsoort (voetbal of handbal), leeftijd en duur van lidmaatschap. Getoond wordt de te betalen contributie.

Selectie opdracht 9

```
read sport, age, duration

if sport = football
    fee = 175
else // if sport = handball
    fee = 225

if age > 40
    fee = fee - 25

if duration > 10
    fee = fee - 20

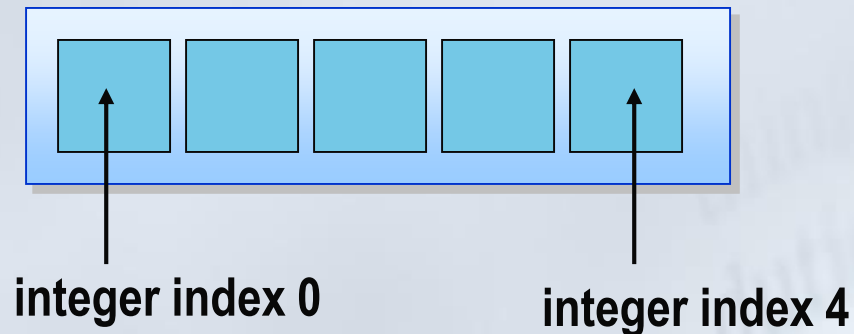
display fee
```

Array's

- int, double, float, bool variabelen kunnen één waarde bevatten
- Array's kunnen meerdere waarden bevatten (van hetzelfde type); reeks van waarden
- Vaak worden de waarden in een array via een iteratie (herhaling) gelezen/bewerkt

Array's

- Array bevat een reeks elementen die benaderd worden via een ('zero-based') index
- Eerste element van een array heeft index 0; laatste element van een array: aantal-1



Aanmaken van een array

- De declaratie van een array in C# bevat:
 - 1) Het element type van de array
 - 2) De rank van de array
 - 3) De naam van de array variabele

```
// declaration of a variable  
int number;
```

```
// declaration of an array  
int[] numbers;
```

De **naam** van de array variabele

De **rank** van de array

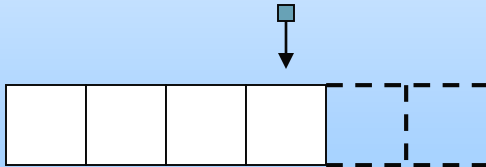
Het element **type** van de array

Array rank

- Rank is de dimensie van een array
- Het geeft aan hoeveel index-getallen je nodig hebt om een element te benaderen

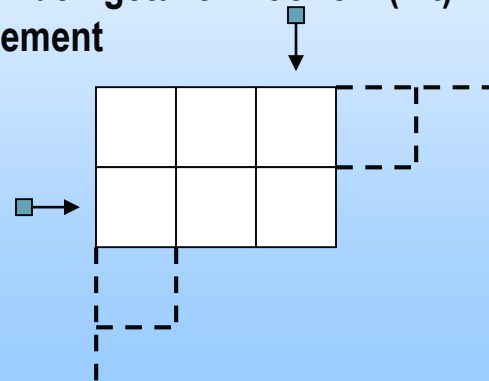
```
long[ ] row;
```

Rank 1: one-dimensional
1 index-getal voor elk (long)
element



```
int[,] grid;
```

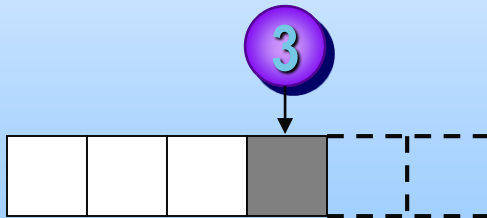
Rank 2: two-dimensional
2 index-getallen voor elk (int)
element



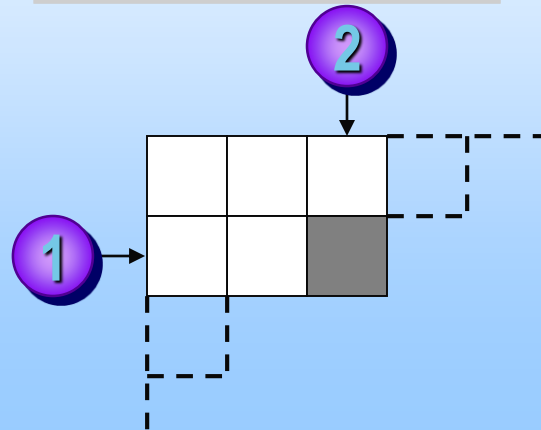
Array elementen adresseren

- Geef een integer index voor elke rank (zero based index-waarden)

```
long[ ] row;  
...  
row[3];
```

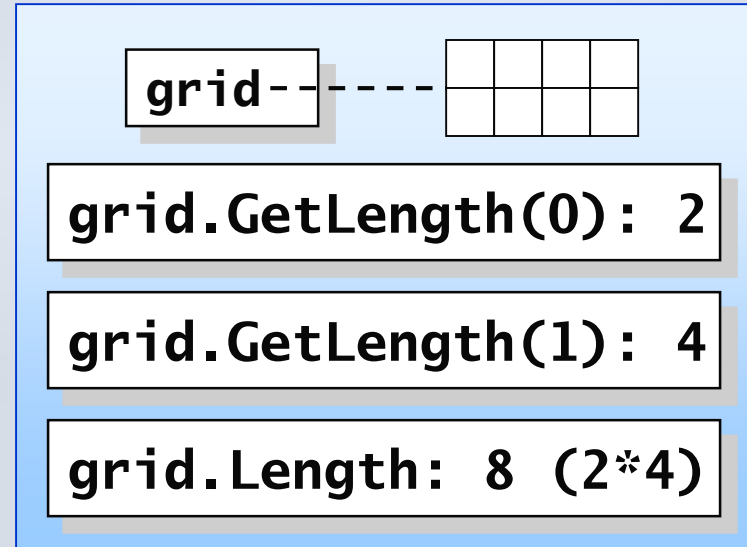
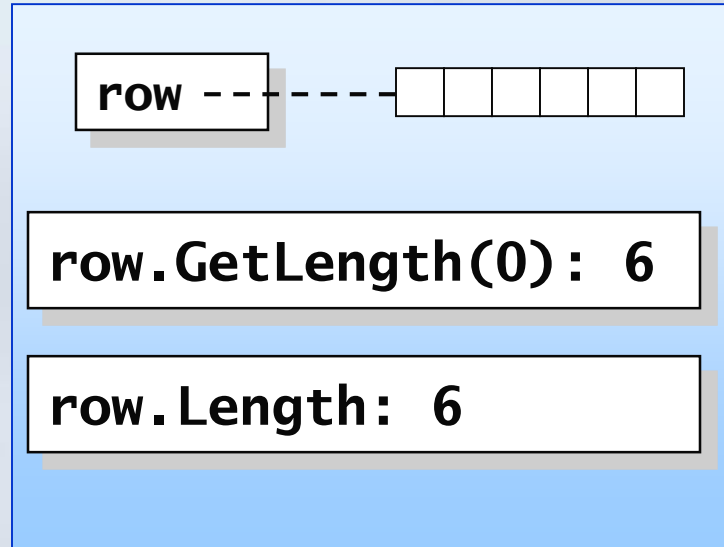


```
int[,] grid;  
...  
grid[1,2];
```



Array 'bounds' (grenzen)

- Bij een verkeerde adressering (index) volgt een 'IndexOutOfRangeException'
- Gebruik Length property en GetLength(...) methode



Array's aanmaken

- Declaratie van een array variabele maakt nog geen array aan!
- Gebruik *new* om een array aan te maken
- Default waarde van array elementen is 0

```
long[ ] row = new long[4];
```

row



0	0	0	0
---	---	---	---

```
int[,] grid = new int[2,3];
```



grid

0	0	0
0	0	0

Array's initialiseren

- Element waarden kunnen bij aanmaak array al opgegeven worden

```
long[ ] row = new long[4] {1, 1, 2, 3};
```

```
long[ ] row = {1, 1, 2, 3};
```

← equivalent



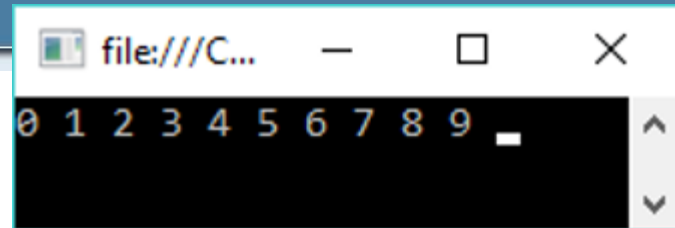
Array – an example

```
class Program
{
    static void Main(string[] args)
    {
        int[] numbers = new int[10];

        // fill array with numbers
        for (int i = 0; i < numbers.Length; i++)
        {
            numbers[i] = i;
        }

        // display array values
        for (int i = 0; i < numbers.Length; i++)
        {
            Console.Write(numbers[i] + " ");
        }

        Console.ReadKey();
    }
}
```



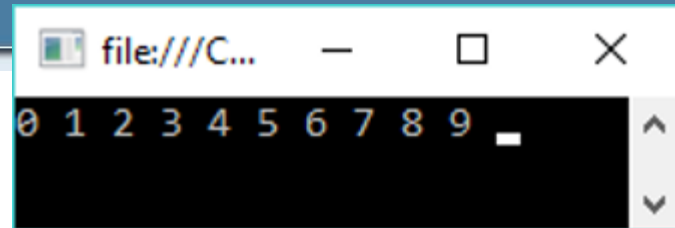
Array – foreach

```
class Program
{
    static void Main(string[] args)
    {
        int[] numbers = new int[10];

        // fill array with numbers
        for (int i = 0; i < numbers.Length; i++)
        {
            numbers[i] = i;
        }

        // display array values
        foreach (int number in numbers)
        {
            Console.Write(number + " ");
        }

        Console.ReadKey();
    }
}
```



Dynamische array lengte

- De grootte v/e array hoeft niet bekend te zijn tijdens compileren

lengte van array (4) bekend tijdens compileren

```
long[ ] row = new long[4];
```

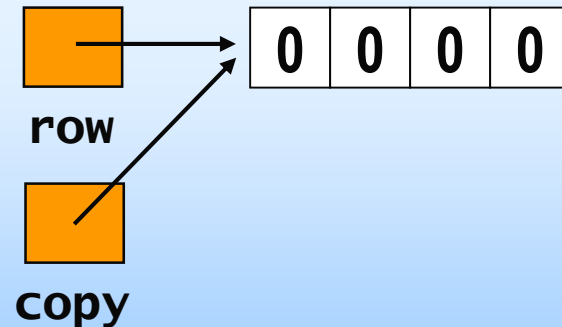
lengte van array niet bekend tijdens compileren (run-time value)

```
string s = Console.ReadLine();  
int size = int.Parse(s);  
long[ ] row = new long[size];
```

Array's kopiëren

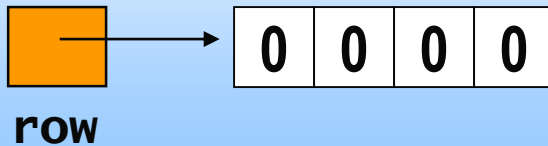
- Bij het kopiëren v/e array wordt alleen de referentie gekopieerd, niet de array zelf
- 2 array variabelen kunnen naar dezelfde array verwijzen

```
long[ ] row = new long[4];  
long[ ] copy = row;  
...  
row[0]++;  
long value = copy[0];  
Console.WriteLine(value);
```



Array properties

```
long[ ] row = new long[4];
```



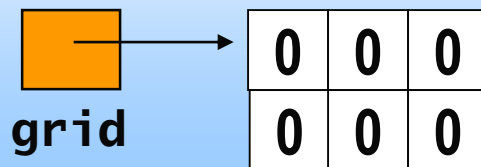
`row.Rank`

1

`row.Length`

4

```
int[,] grid = new int[2,3];
```



`grid.Rank`

2

`grid.Length`

6

Array's – oefening 1

- Lees de maandomzet van elke maand (12x). Bereken de gemiddelde omzet, en laat elke maandomzet zien die hoger is dan het gemiddelde.

Array's – oefening 1

```
profit ← array[12]
```

```
total = 0
```

```
for i = 0 to profit.Length - 1
```

```
    read profit[i]
```

```
    total = total + profit[i]
```

```
average = total / profit.Length
```

```
for i = 0 to profit.Length - 1
```

```
    if profit[i] > average
```

```
        display profit[i]
```

Array's – oefening 1

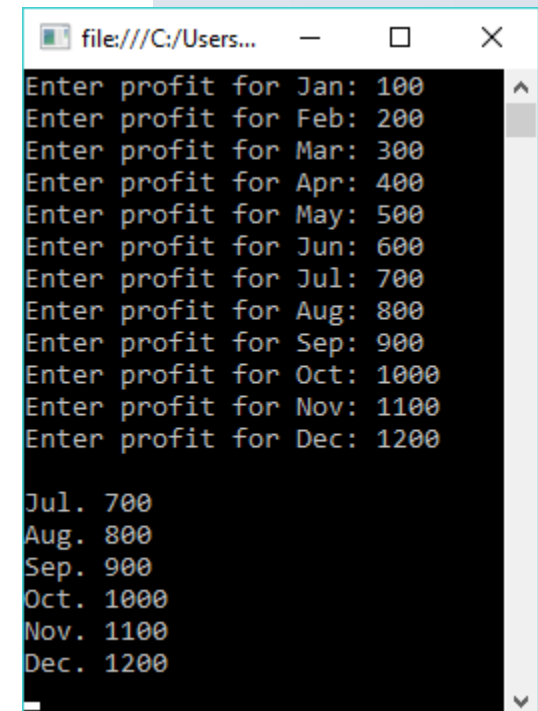
```
class Program
{
    const int NrOfMonths = 12;

    static void Main(string[] args)
    {
        string[] monthNames = { "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
        int[] profit = new int[NrOfMonths];

        int total = 0;
        for (int i = 0; i < profit.Length; i++)
        {
            Console.Write("Enter profit for " + monthNames[i] + ": ");
            profit[i] = int.Parse(Console.ReadLine());
            total += profit[i];
        }

        double average = (double)total / profit.Length;

        for (int i = 0; i < profit.Length; i++)
        {
            if (profit[i] > average)
            {
                string month = monthNames[i];
                Console.WriteLine($"{month}. {profit[i]}");
            }
        }
    }
}
```



```
file:///C:/Users...
Enter profit for Jan: 100
Enter profit for Feb: 200
Enter profit for Mar: 300
Enter profit for Apr: 400
Enter profit for May: 500
Enter profit for Jun: 600
Enter profit for Jul: 700
Enter profit for Aug: 800
Enter profit for Sep: 900
Enter profit for Oct: 1000
Enter profit for Nov: 1100
Enter profit for Dec: 1200

Jul. 700
Aug. 800
Sep. 900
Oct. 1000
Nov. 1100
Dec. 1200
```

Array's – oefening 2

- Zoek een getal in een array met 6 waarden en geef aan of deze aanwezig is

Array's – oefening 2

```
numbers ← array[6]
```

```
read searchValue
```

```
found = false
```

```
index = 0
```

```
while index < numbers.Length AND not found
```

```
    if numbers[index] = searchValue
```

```
        found = true
```

```
    else
```

```
        index = index + 1
```

```
display found
```


Array's – oefening 2

```
class Program
{
    const int NrOfItems = 6;

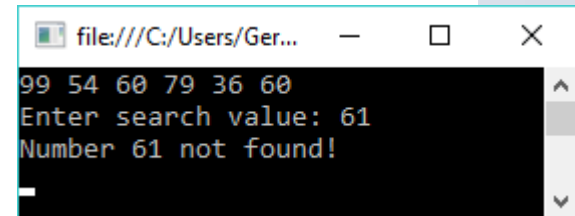
    static void Main(string[] args)
    {
        int[] numbers = new int[NrOfItems];

        Random rnd = new Random();
        for (int i = 0; i < numbers.Length; i++)
        {
            numbers[i] = rnd.Next(100);
        }

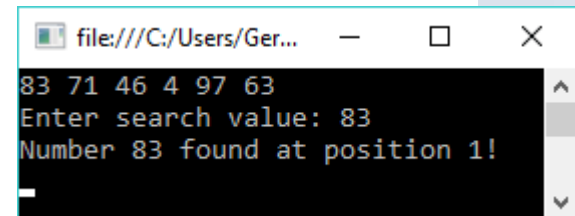
        Console.WriteLine("Enter search value: ");
        int searchValue = int.Parse(Console.ReadLine());

        bool found = false;
        int index = 0;
        while ((index < numbers.Length) && (!found))
        {
            if (numbers[index] == searchValue)
                found = true;
            else
                index++;
        }

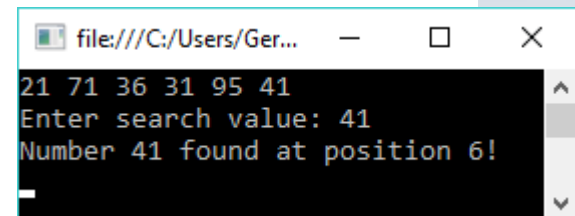
        if (found) Console.WriteLine($"Number {searchValue} found at position {index + 1}!");
        else Console.WriteLine($"Number {searchValue} not found!");
    }
}
```



```
file:///C:/Users/Ger...  —  □  ×
99 54 60 79 36 60
Enter search value: 61
Number 61 not found!
```



```
file:///C:/Users/Ger...  —  □  ×
83 71 46 4 97 63
Enter search value: 83
Number 83 found at position 1!
```



```
file:///C:/Users/Ger...  —  □  ×
21 71 36 31 95 41
Enter search value: 41
Number 41 found at position 6!
```

Wat is fout aan onderstaande code?

```
read searchValue
index = 0
while numbers[index] <> searchValue
    index = index + 1

display "element found at " + index
```

Als zoekwaarde niet voorkomt in de array, dan zal index op den duur 'wijzen' voorbij het laatste element v/d array; een (runtime) foutmelding zal dan optreden!

Oefening dobbelsteen

Maak een array 'doppelsteen' met 6 elementen van het type integer. Vul deze array met de waarde nul (6 keer dus).

Genereer in een herhaling 6000 keer een willekeurig getal met een gehele waarde tussen 1 en 6. Wanneer de waarde 1 is dan verhoog je element 0 van de array met 1, hetzelfde geldt voor de waarde 2, 3, 4, 5 en 6.

Oftewel: `doppelsteen[x-1]` geeft het aantal keren dat x is gegooid aan.

Na afloop van de herhaling toon je de inhoud van de array op het scherm.

Oefening dobbelsteen

```
dice ← array[6]
```

```
for i = 0 to 5  
    dice[i] = 0
```

```
for i = 1 to 6000  
    value = Random.Next(1, 7) // 1..6  
    dice[value - 1] = dice[value - 1] + 1
```

```
for i = 0 to 5  
    display dice[i]
```

Huiswerk

- Lezen
 - Yellow Book → zie Moodle
- (praktijk) Programmeren 1 *(deze week)*
 - week 5 opdrachten