Campus Nyköping

PROGRAMMERING I .NET C#1 -06

Arrayer

Förra gången

- Scope/Omfång
- Repetition och mer om metoder
- Code along - Gunnars Pizzeria
- Code along - Bankomaten

Idag

- Upprop
- Dice Game
- Arrayer

Demo – Dice Game

Arrayer (Vektorer)

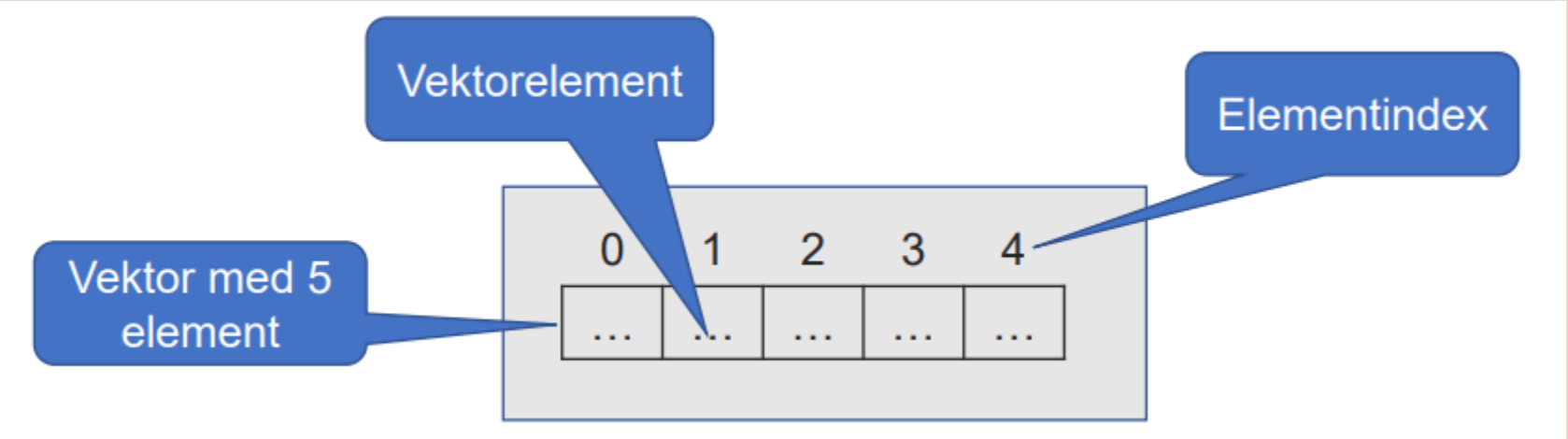
- Hittills har vi jobbat med enstaka variabler, t ex ett namn, ett tal eller en bool.
- Vanligast är dock att man har **många** värden, av samma typ.
- Såhär skulle vi kunna göra:

```
string name1 = "Micke";
string name2 = "Anita";
string name3 = "Stefan";
Console.WriteLine(name1);
Console.WriteLine(name2);
Console.WriteLine(name3);
```

- …men det är lite omständigt.

Arrayer

- En vektor (array) är en sekvens av element
  - Alla element är av samma typ
    - Tex int, string, bool eller double
    - …men också mer komplexa datatyper/objekt
  - Ordningsföljden är fast
  - Har en fast storlek (Array.Length)

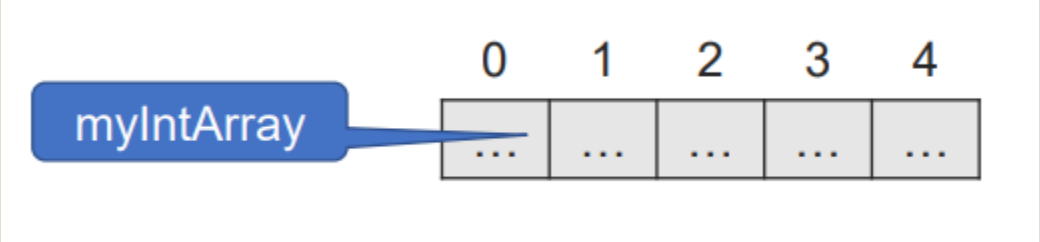


Deklarera arrayer

- Deklarationen talar om elementtypen
- Hakparenteser [ ] betyder “array" (“vektor")
- Exempel:
  - Deklarera en vektor av heltal: `int[] myIntArray;`
  - Deklarera en vektor av strängar: `string[] myStringArray;`

Skapa array

- Använd operatorn new •
  - Ange vektorns längd
    - Exempel: skapa en array med 5 heltal:
      - `myIntArray = new int[5];`



Tilldela värden

```
myIntArray[0] = 25;
myIntArray[1] = 345;
myIntArray[2] = 7;
myIntArray[3] = 47;
myIntArray[4] = 63;
```

Hela exemplet

```
// Deklarera array
int[] myIntArray;

// Skapa array
myIntArray = new int[5];

// Tilldela värden
myIntArray[0] = 25;
myIntArray[1] = 345;
myIntArray[2] = 7;
myIntArray[3] = 47;
myIntArray[4] = 63;
```

En kortare version

- `int[] myIntArray = {25, 345, 7, 47, 63};`

Hur ser man värdena i en array

- Åtkomst till array-element görs genom operatorn hakparentes **[ ]**
  - Kallas indexerare (indexer)
  - Indexeraren tar elementets index som parameter
  - Det första elementet har index 0 (noll)
  - Det sista elementet har index Length – 1
- Vektorelement kan läsas och ändras med [ ] operatorn

Skriva ut ett värde

- `int[] myIntArray = {25, 345, 7, 47, 63};`
- `Console.WriteLine(myIntArray[2]) // Kommer skriva ut siffran “7”`

Hur lång är en array?

- `int[] myIntArray = {25, 345, 7, 47, 63};`
- `Console.Writeline(myIntArray.Length)` // Skriver ut 5

- För att visa sista värdet skriver du:
  - `Console.Writeline(myIntArray[myIntArray.Length - 1]) // Skriver ut 63`

- Kom ihåg: alla arrayer har ett index som börjar från **0**

Skriva ut alla värden

```
int[] myIntArray = {25, 345, 7, 47, 63};

for(int i = 0; i < myIntArray.Length; i++)
{
        Console.WriteLine(myIntArray[i]);
}
```

Detta skriver ut:

```
25
345
7
47
63
```

Skriva in nytt värde i en array

```
int[] myIntArray = {25, 345, 7, 47, 63};

myIntArray[1] = 99;
myIntArray[3] = 16;



for(int i = 0; i < myIntArray.Length; i++)
{
       Console.WriteLine(myIntArray[i]);
}
```

Detta skriver ut:

```
25
99
7
16
63
```

for eller foreach

- En for-loop fungerar utmärkt för att lista innehållet I en array.
- Fördelarna är att man kan lista t ex bara varannat värde, och att man vet vilket index varje värde har.
- En enklare variant är **foreach**, som listar HELA arrayen, och inte håller reda på vilket index värdena har. Det går inte heller enkelt att ändra värdena.

foreach

- Hur fungerar en foreach-loop?
- `foreach(type value in array)`

- type – elementens typ
- value – lokalt namn på variabel med innehållet av ett element
- array – arrayen som hanteras

- Används när ingen indexering behövs
  - Alla element hanteras ett i taget
  - Element kan inte ändras (read-only)

Exempel - foreach

```
int[] myIntArray = {25, 345, 7, 47, 63};

foreach(int num in myIntArray)
{
        Console.Write(num);
}
```

Exempel - foreach

```
string[] capitals =
{
	"Stockholm",
	"Washington",
	"London",
	"Paris"
};
foreach (string capital in capitals)
{
	Console.WriteLine(capital);
}
```

Sortera en array

- Att sortera en array I bokstavsordning görs enkelt såhär:
  - `Array.Sort(capitals);`

- Mer om sortering, och andra inbyggda funktioner, senare I kursen…

Code along - arrayer

- Kod: Arraydemo.zip

Övning 1 – Vilken dag

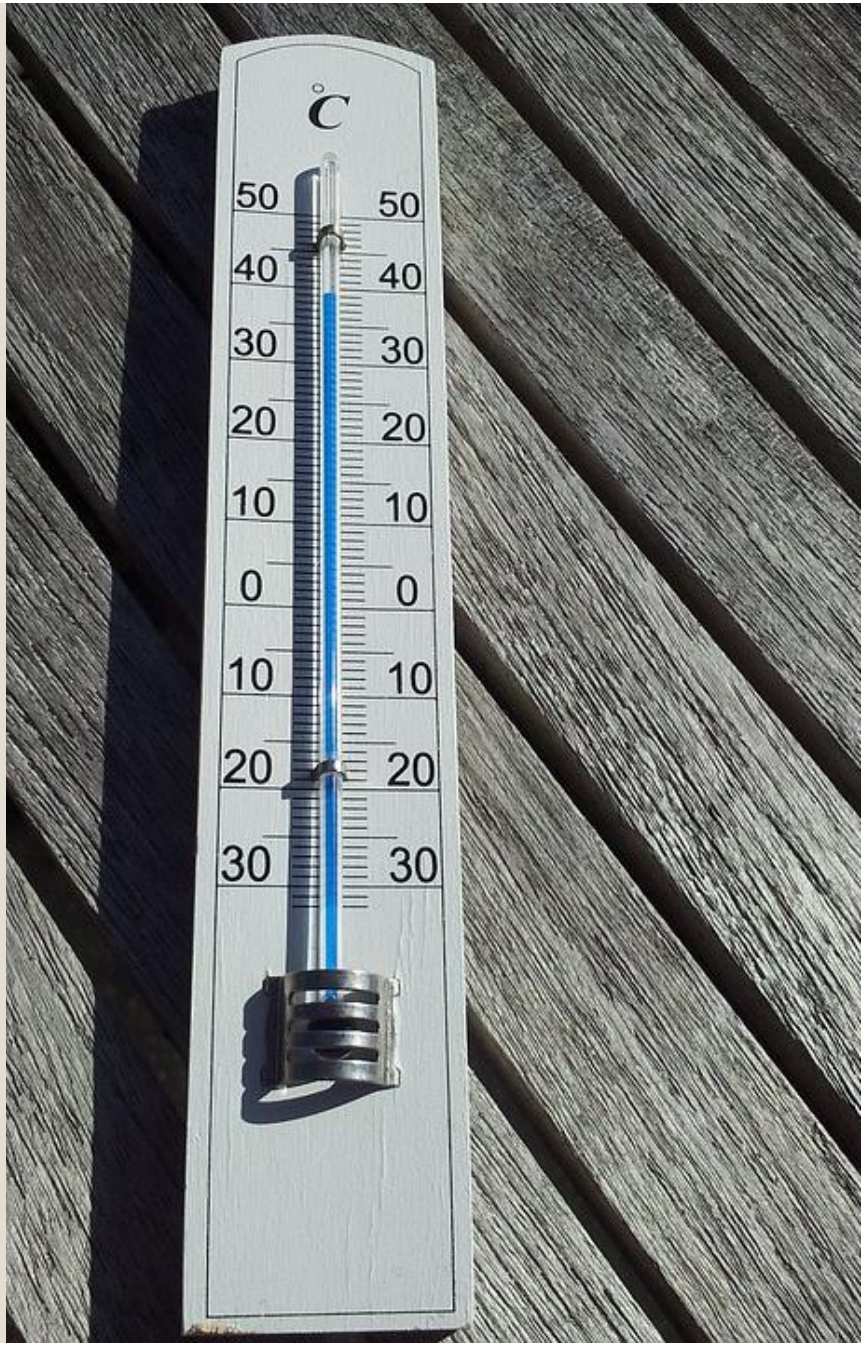
- Skapa en application där du kan skriva in en siffra mellan 1-12
- Skapa en sträng-array med årets alla månader
- Bygg så att om du t ex skriver in siffran 7, så visas månaden “Juli”

- Lägg till veckodagarna(från mitt exempel), och använd Datetime.Now()
  - DateTime.Day Property (System) | Microsoft Learn
- Se till att din application skriver ut t ex Idag är det tisdag den 15:e maj 2022

- Skapa en inmatning av år, månad och dag och skriv in ditt eget födelsedatum.
  - Svaret du ska visa är t ex: “Du föddes på en fredag”

Övning 2 - Medeltemperatur

- Skapa en app som läser in 5 temperaturvärden I en array
- Skriv sedan ut dessa värden, I omvänd ordning, och beräkna också medeltemperaturen
- Visa det högsta värdet (lite överkurs)



Övning 3 – Gruppindelningen

- Lägg in följande namn I en array:
  - **Karin**
  - Anders
  - Johan
  - Eva
  - Maria
  - Mikael
  - **Anna**
  - Sara
  - Erik
  - Per
  - Christina
  - Lena
  - **Lars**
  - Emma
  - Kerstin
  - Karl
  - Marie
  - Peter

- Skapa en application som plockar var tredje namn ur listan, och skapar grupper om tre personer, totalt 6 grupper.
- Namnen ska plockas ut så att grupperna inte har personen efter varandra, I samma grupp
- Sortera eleverna I bokstavsorning I respektive grupp
- Första Gruppen ska se ut såhär:
  - Anna, Karin, Lars
- Ändra koden så det går att ändra antalet personer I varje grupp, t ex 2, 3, 4, 5.. personer I varje grupp

Övning 4 – for eller foreach

- Fundera på om övning 1-3 skulle kunna göras med en foreach-loop istället.
- Om du tror att det går, prova gärna.
  - Skapa ett nytt project, så du inte ändrar I den gamla koden
- OBS! Det är troligtvis betydligt svårare att göra, än man kan tro.

Övning 5 – Öva på foreach

- Öva på att skriva foreach-loopar.
  - Skriv ut alla eleverna I Klassen från övning 3, bara uppifrån och ner.
    - Skriv ut alla namnen med bara stora bokstäver. Använd `name.ToUpper()`
    - Sortera dem I bokstavsorning
  - Skriv ut alla månader från övning 1, men låt sommarmånaderna vara markerade, med t ex en *

Länkar

- C# Arrays (w3schools.com)
- C# Loop Through an Array (w3schools.com)
- C# Sort Arrays (w3schools.com)