

Objektorienterad programmering

Föreläsning 10

© Copyright
Mahmud Al Hakim
mahmud@webacademy.se
www.webacademy.se

Agenda

- Fält fort.
 - Fält som parametrar
 - Parameterfält – **params**
 - Parametrar till Main
- Listor
 - ArrayList
- Sorteringsalgoritmer
 - Selection sort (sök minsta och byt)
 - Bubble sort (bubbelsortering)

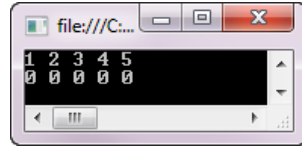
Fält som parametrar

```
static void Nollställ(int[] f)
{
    for (int i = 0; i < f.Length; i++)
    {
        f[i] = 0;
    }
}

static void Main(string[] args)
{
    int[] a = { 1,2,3,4,5};
    foreach (var item in a) { Console.Write(item + " "); }

    Nollställ(a);
    // Kom ihåg: a är en referensvariabel som refererar till ett fält

    Console.WriteLine("");
    foreach (var item in a) { Console.Write(item + " "); }
    Console.ReadKey();
}
```



Parameterfält - params

- Med hjälp av parameterfält (parameter array) kan man konstruera metoder som kan anropas med ett variabelt antal argument.
- I deklarationen av metoden skriver man det reserverade ordet **params**.
- Ordet params anger att man vid anrop av metoden som argument kan ge ett godtyckligt antal enkla värden.

Params – Exempel

```
public static double Medel(params double[] a)
{
    double summa = 0;
    foreach (double v in a)
    {
        summa += v;
    }
    return summa / a.Length;
}

static void Main(string[] args)
{
    double medelvärde = Medel(1.5,2.5);
    Console.WriteLine(medelvärde);
    Console.ReadKey();
}
```

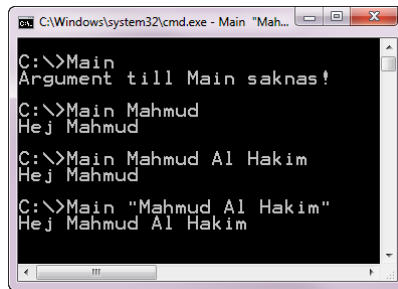
Du kan ge ett godtyckligt antal enkla värden vid anrop av metoden

Parametrar till Main

- Ett C# program måste ha en metod Main i vilken exekveringen börjar.
- Main kan ha en parameter av typen **string[]**
- Parametern är alltså en fältvariabel som refererar till ett fält där komponenterna är av typen string.
- Metoden Main anropas inte inifrån programmet utan från kommandotolken när programexekveringen startas.

Parametrar till Main - Exempel

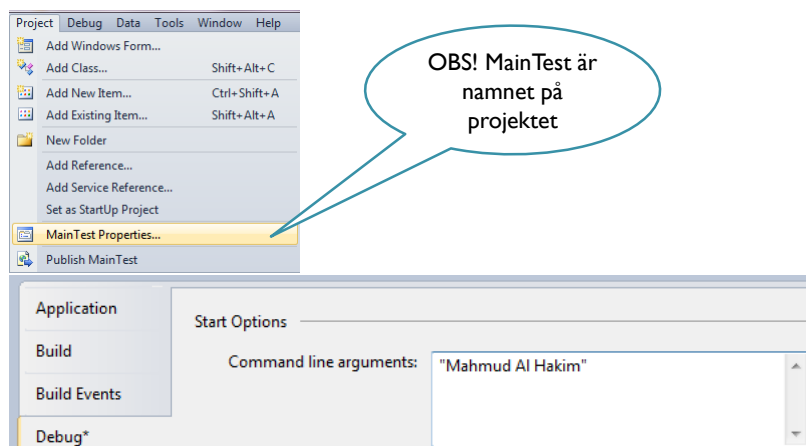
```
static void Main(string[] args)
{
    if( args.Length > 0 )
        Console.WriteLine("Hej " + args[0]);
    else
        Console.WriteLine("Argument till Main saknas!");
}
```



Copyright 2015 -Mahmud Al Hakim www.webacademy.se

7

Ange parametrar till Main i VS



Copyright 2015 -Mahmud Al Hakim www.webacademy.se

8

Listor

- I namnrymden **System.Collection.Generic** finns en mycket användbar standardklass med namnet **List**.
- List är en s.k. generisk klass (mer om detta senare).
- Med hjälp av denna kan man bygga upp listor som innehåller ett godtyckligt antal komponenter.
- Komponenterna brukar kallas **element**.
- Att skapa en tom lista:
List<typ> lista = new List<typ>();

Exempel

```
List<string> lista = new List<string >();
```

Skapa en kopia av ett fält eller en lista

```
// Ett fält  
int[] fält = { 1, 2, 3, 4, 5 };  
  
// En lista som är en kopia av fältet  
List<int> lista1 = new List<int>(fält);  
  
// En lista som är en kopia av en annan lista  
List<int> lista2 = new List<int>(lista1);
```

Att hantera listor – Några exempel

```
// En lista som innehåller 5 heltal
List<int> lista = new List<int>{ 1, 2, 3, 4, 5 };

// Lägg till ett element sist i listan
lista.Add(6);

// Ta bort heltalet 3.
lista.Remove(3);

// Ta bort det första elementet
lista.RemoveAt(0);

// Egenskapen Count ger antal element i listan
Console.WriteLine("Antal element i listan: " + lista.Count);

// Skriv ut listan
foreach (var item in lista)
{ Console.Write(item + " "); }
```

Copyright 2015 -Mahmud Al Hakim www.webacademy.se

11

ArrayList

- I namnrymden **System.Collections** finns en "gammal" standardklass med namnet **ArrayList**.
- När man skapar en lista av typen ArrayList kan man inte ange vilken typ elementen skall ha.
- Att skapa en ArrayList
ArrayList lista = new ArrayList();
- Listor av detta slag kan innehålla element av vilken referenstyp som helst.
- Man kan i samma lista ha element av olika slag!
- Elementen i listan blir referenser av typen **object**.

Copyright 2015 -Mahmud Al Hakim www.webacademy.se

12

ArrayList

```
// Skapa en lista av typen ArrayList
ArrayList lista = new ArrayList();

// Lägg till element av olika slag
lista.Add(5);      // int
lista.Add(2.5);   // double
lista.Add("Text"); // string

Console.WriteLine("Antal element i listan: " + lista.Count);
foreach (var item in lista)
{
    Console.WriteLine(item);
}
```

ArrayList användes innan de generiska klasserna fanns (innan version .NET 2.0) Använd om möjligt de generiska varianterna då de är säkrare.

Selection sort (sök minsta och byt)

- Algoritmen går ut på att man först letar reda på det minsta talet i fältet och placerar det på fältets första plats.
- Därefter letar man reda på det näst minsta talet och placerar det på fältets andra plats.
- Osv...

Selection sort – Pseudokod*

1. Sätt k till index för fältets första komponent.
2. Medan k är mindre än eller lika med index för fältets sista komponent:
 - 2.1. Leta reda på det minsta talet i den del av fältet som börjar med tal nummer k och som slutar med det sista talet i fältet.
 - 2.2. Låt det mista talet (från steg 2.1) och tal nummer k byta plats.
 - 2.3. Öka k med 1

*Tips: <http://sv.wikipedia.org/wiki/Pseudokod>

Selection sort

```
// Ett fält som innehåller 10 heltal (osorterat)
int[] f = { 0, 7, 4, 6, 8, 3, 6, 9, 1, 2, 5 };

Console.WriteLine("Talen osorterade: ");
foreach (var item in f) { Console.Write(item + " "); }

for (int k = 0; k < f.Length; k++)
{
    // Leta reda på minsta bland talen från k till fältets sista index
    int m = k;
    // Låt i löpa från k+1 till fältets sista index
    for (int i = k + 1; i < f.Length; i++)
    {
        if (f[i] < f[m])
        {
            m = i; // Det minsta talet finns nu på plats nr m
        }
    }
    // Låt det mista talet och tal nummer k byta plats
    int temp = f[k];
    f[k] = f[m];
    f[m] = temp;
}
```

```
Console.WriteLine("\nTalen sorterade: ");
foreach (var item in f) { Console.Write(item + " "); }
```


Bubble sort (bubbelsortering)

- Metoden går ut på att man upprepade gånger går igenom det område i listan som ska sorteras och gör parvis jämförelser av intilliggande element.
- När två intilliggande element ligger i fel ordning byter man plats på dem.
- Varje gång man gått igenom ett område kommer det sista talet att ha hamnat på rätt plats.
- Nästa gång reducerar man därför det område man går igenom med ett.
- Efter hand som man gör sorteringen kommer listan i botten bli alltmer korrekt och de överblivna talen "bubblar" uppåt, därav namnet på sorteringsalgoritmen.

Källa: <http://sv.wikipedia.org/wiki/Bubbelsortering>
Copyright 2015 -Mahmud Al Hakim www.webacademy.se

17

Bubble sort

```
int[] f = { 3, 2, 1 };
foreach (var item in f) { Console.Write(item + " "); }
Console.WriteLine("\n====");

int temp;
for (int j = 0; j < f.Length; j++)
{
    for (int i = 0, x = 1; x < f.Length ; i++, x++)
    {
        if (f[i] > f[x])
        {
            temp = f[i];
            f[i] = f[x];
            f[x] = temp;
            foreach (var item in f) { Console.Write(item + " "); }
            Console.WriteLine("\n-----");
        }
    }
}
```



Vilken sorteringsalgoritm är bäst?

Tips!

<http://www.sorting-algorithms.com/random-initial-order>

<https://www.youtube.com/watch?v=lyZQPjUT5B4>

Copyright 2015 -Mahmud Al Hakim www.webacademy.se

18