



Campus Nyköping

PROGRAMMERING I .NET C# 1 - 07

C# 5: Matriser och List

Förra gången

- Bankomaten
- Arrayer

Idag

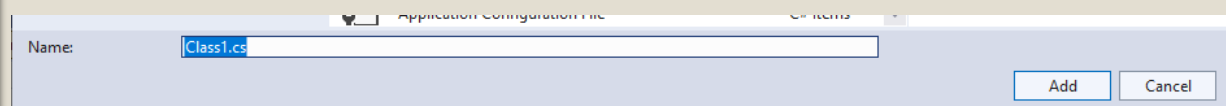
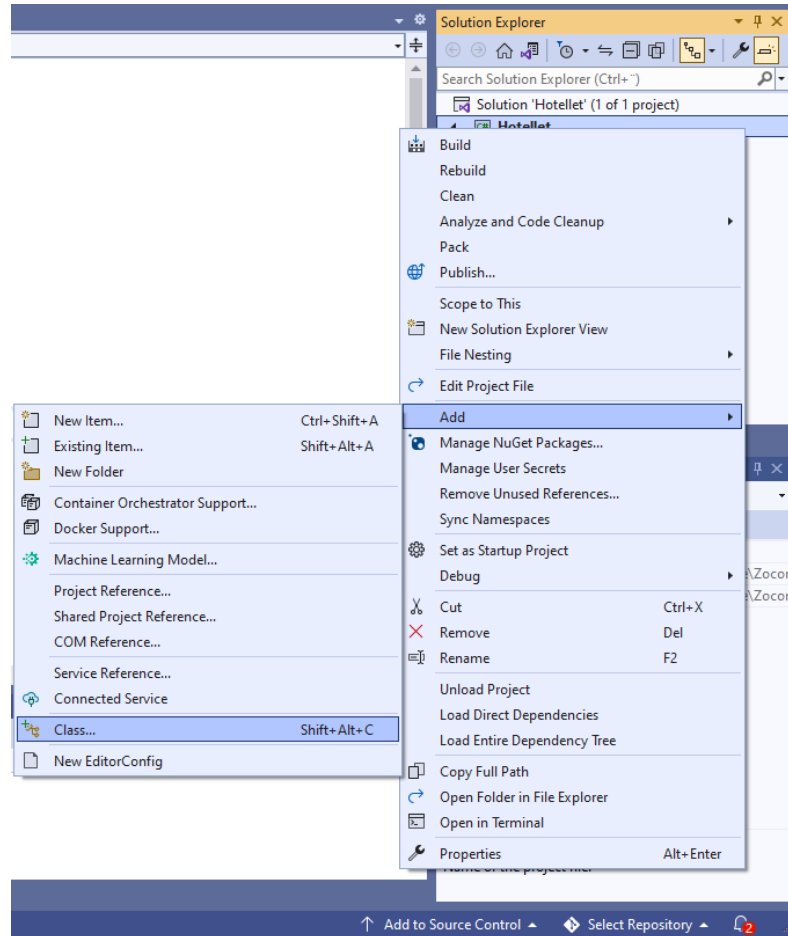
- Upprop
- Demo: Gruppindelning
- Fler filer/klasser i ett project
- Mer arrayer -> Matriser
- List

Flera filer I projektet

- Class
 - en klass är en programkodmall för att skapa objekt, tillhandahålla initialvärden för tillstånd (medlemsvariabler) och implementeringar av beteende (medlemsfunktioner eller metoder.
 - Ett sätt att dela upp kod
 - Ett sätt att gruppera saker som hör ihop
- Åtkomstmodifierare
 - public
 - All kod I hela programmet kommer åt klasser och metoder som är public
 - Mer om åtkomstmodifierare snart...
 - internal
 - Ungefär samma sak som public, men bara filerna I samma projekt kommer åt detta

Skapa en ny fil = class

- Högerklicka på Projektnamnet i Solution Explorer
- Välj Add
- Välj Class
- Namnge Klassen



En ny klass

- Du får en fil som ser ut ungefär såhär:
- `using System;`
- `using System.Collections.Generic;`
- `using System.Linq;`
- `using System.Text;`
- `using System.Threading.Tasks;`
- `namespace MyApp`
- `{`
- `internal class NyKlass`
- `{`
- `}`
- `}`

Jämför med Programklassen

- **NyKlass.cs:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace MyApp
{
    internal class NyKlass
    {
        // Ingen kod här ännu
    }
}
```

- **Program.cs:**

```
namespace MyApp
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello");
        }
    }
}
```

NyKlass

```
internal class NyKlass
{
    // Ingen kod här ännu
    // Här kan vi lägga till ny kod, t ex variabler, metoder m m.
}
```


Code Along – Klasser och metoder I andra filer

- Kod: MyApp.zip

Matriser – flerdimensionella arrayer

- Vanliga arrayer består av en följd av data, t ex
 - `int[] myIntArray = {25, 345, 7, 47, 63};`
 - `string[] capitals = {"Stockholm", "London", "Paris"};`
- Matriser har mer än en dimension (2, 3 eller flera)
 - Den viktigaste och vanligaste är de med två dimensioner
 - Kan ocks kallas tabell.
- Exempel på en matris av heltal med två rader och 4 kolumner.

| | 0 | 1 | 2 | 3 |
|---|---|---|----|---|
| 0 | 5 | 0 | -2 | 4 |
| 1 | 5 | 6 | 7 | 8 |

Deklarera och skapa matriser

- `int[,] intMatrix = new int[3, 4];`
- `float[,] floatMatrix = new float[8, 2];`
- `string[, ,] strCube = new string[5, 5, 5];`

Exempel

```
int[,] myTable = new int[2, 2];  
myTable[0, 0] = 25;  
myTable[0, 1] = 39;  
myTable[1, 0] = 15;  
myTable[1, 1] = 46;
```

Ger en matris som ser ut såhär:

| | 0 | 1 |
|---|----|----|
| 0 | 25 | 39 |
| 1 | 15 | 46 |

Skapa och initialisera en matris

- Det finns naturligtvis en kortare version:

```
int[,] myMatrix = {  
    {1, -2, 13, 47},    // rad 0 värden, kommatecken mellan raderna  
    {52, 60, -7, 888}   // rad 1 värden  
};  
  
// Matrisens storlek är 2 x 4 (2 rader, 4 kolumner)
```

- Ta fram på en utvald post i matrisen/tabellen

```
Console.WriteLine(myMatrix[1,2]); // Kommer ge -7
```

Visa en matris

```
int[,] myTable = { {1, -2, 13, 47}, {52, 60, -7, 888} };
```

```
Console.Write(myTable[0, 2]); // Ger 13
```

```
for (int i = 0; i < 2; i++)  
{  
    for(int j = 0; j < 4; j++)  
    {  
        Console.Write(myTable[i, j] + "\t");  
    }  
    Console.WriteLine();  
}
```

Antalet rader och kolumner?

```
int[,] myTable = { {1, -2, 13, 47}, {52, 60, -7, 888} };
```

```
for (int i = 0; i < myTable.GetLength(0); i++)  
{  
    for(int j = 0; j < myTable.GetLength(1); j++)  
    {  
        Console.Write(myTable[i, j] + "\t");  
    }  
    Console.WriteLine();  
}
```

Code Along - Matriser

- Kod: Matriser.zip

List – en annan sorts “array”

- En enkel form av array är en **List**.
- Den kommer vi använda mycket, då vi går in på objektorienterad programmering, men vi ska idag prova den.
- Justerbara arrayer – som kan ändra storlek

Listor – flexibla arrayer

- **List<T>** - vektor som kan ändra storlek dynamiskt
 - Vid tillägg eller borttag av element
 - Har också indexerare [] (som vektorer)
 - T anger vilken datatyp som listan kommer att innehålla
 - Exempel:
 - List<int> innehåller heltal
 - List<string> innehåller strängar
- Grundläggande funktioner och egenskaper:
 - Add (T element) – lägger till ett nytt element i slutet
 - Remove(element) – tar bort elementet
 - Count – returnerar listans aktuella storlek

Exempel

```
List<int> intList = new List<int>();  
for( int i=0; i<5; i++)  
{  
    intList.Add(i);  
}
```

Med vanliga arrayer skulle samma sak se ut såhär:

```
int[] intArray = new int[5];  
for( int i=0; i < 5; i++)  
{  
    intArray[i] = i  
}
```

- Huvudsakliga skillnaden:
 - När vi använder listor behöver vi inte veta (i förväg) hur många element som behövs

List<T>

- Definiera en List
 - `List<string> words = new List<string>();`
- Lägga till saker i en List:
 - `words.Add("Hejsan");`
- Lägga till många saker:
 - `string[] words = { "Hej", "på", "dig"};`
 - `words.AddRange(words2.ToList());`
- Ta bort saker i en List:
 - `words.RemoveAt(index);`

Code along - List

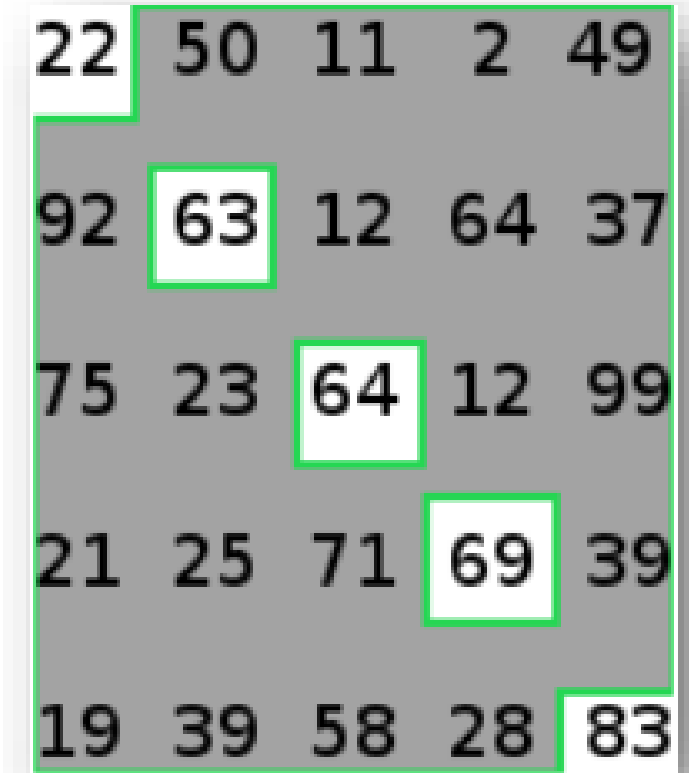
Övning 1 - Temperaturtabellen

- Skapa en **matris** av temperaturdata
- Varje rad är en stad, och varje kolumn är en månad.
- Tabellen ska se ut t ex såhär:

| | Jun | Jul | Aug |
|------------|-----|-----|-----|
| Eskilstuna | 23 | 18 | 12 |
| Stockholm | 19 | 22 | 11 |
| Nyköping | 21 | 18 | 10 |

Övning 2 – Räkna diagonalen

- Skriv ett program som beräknar summan av elementen på diagonalen av en **matrix**
- Svaret i exemplet ska bli 301
- Ange talen hårdkodat i koden.
- Prova också att skapa en app, som skapar alla talen slumpmässigt.



| | | | | |
|----|----|----|----|----|
| 22 | 50 | 11 | 2 | 49 |
| 92 | 63 | 12 | 64 | 37 |
| 75 | 23 | 64 | 12 | 99 |
| 21 | 25 | 71 | 69 | 39 |
| 19 | 39 | 58 | 28 | 83 |

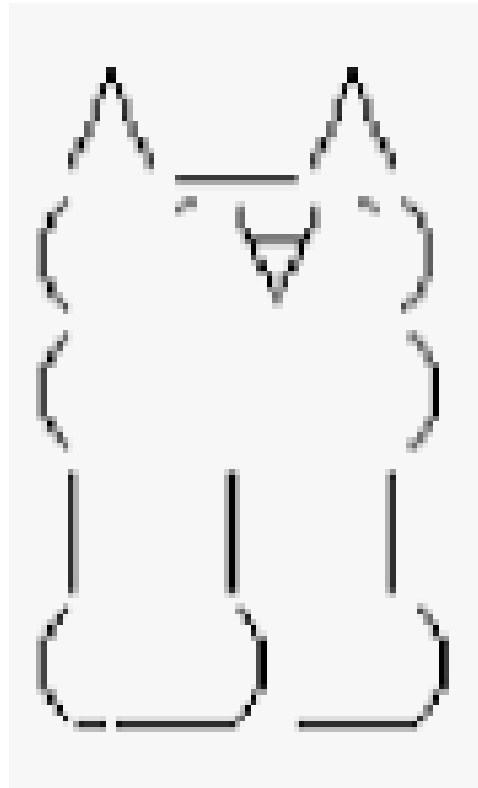
Övning 3 - Hotellbokningen

- Skapa en app som kan hantera ett hotel, **och bokningar**.
- Hotellet har 3 våningar, och fyra rum i varje
- Gästen ska kunna skriva in sitt namn, och vilket rum hen vill bo i.
- Därefter ska hotellet ritas upp i konsolen, och det ska synas vilka rum som är bokade, och vem som bor i rummet.

```
10:Tomt      11:Tomt    12:Nilsson 13:Tomt
00:Andersson 01:Tomt    02:Tomt   03:Tomt
Osv...
```

- Visa också vad rummet kostar, och vad hotellet totalt tjänar per dag.
- Prislista:
 - Våning 1: 650 kr/rum
 - Våning 2: 775 kr/rum
 - Våning 3: 995 kr/rum
- **Använd Matris-array(er)**





Övning 4 – “Ritprogrammet”

- Skapa en app som det går att “rita”
ascii-konst med.
- Skapa en **matrix** som är t ex 10x10
rutor stor.
- Med ASWD-tangenterna kan du
styra markören till en viss ruta
- Med t ex knappen E (Edit) kan du
sedan knappa in vilket tecken som
ska vara på den platsen.

- Skapa en app som det går att "rita" ascii-konst med.
- Skapa en **matris** som är t ex 10x10 rutor stor.
- Med ASWD-tangenterna kan du styra markören till en viss ruta
- Med t ex knappen E (Edit) kan du sedan knappa in vilket tecken som ska vara på den platsen.

Övning 5 – Kassaapparaten (List- övning)

- Använd List<double> för att bygga en enkel Kassaapparat
- Det ska gå att lägga till och ta bort olika belopp.
- Summan av alla belopp ska visas längst ner
- Såhär kan det se ut:
 - 1 – 29.90
 - 2 – 54.50
 - 3 – 199.00
- Summa: 283.40 kr
- Om man tar bort någon av posterna så ska summan räknas om.



Länkar

- [Multidimensional Arrays - C# Programming Guide | Microsoft Learn](#)
- [C# Multidimensional Array \(With Examples\) \(programiz.com\)](#)
- [C# List<T> Collection \(tutorialsteacher.com\)](#)