



UPPSALA  
UNIVERSITET

# . NET-programmering

Övrigt

# Ofullständiga klasser

- Alla delar måste markeras med nyckelordet "partial "
- Alla delar måste ha samma access modifier
- Om någon del ärver en annan klass så kommer även hela klassen att göra det

# Ofullständiga klasser, exempel

```
public partial class Car
{
    public void ChangeGear(int newGear)
    {
        //...
    }
}

public partial class Car : Vehicle
{
    public void Accelerate()
    {
        //...
    }
}
```



# Indexerare

- Ger objekt array-liknande syntax
- Har precis som properties ett get- och set-block
- Kan ha en eller flera parametrar

Exempel:

```
Classroom c = new Classroom();  
Student s = c[2];
```



UPPSALA  
UNIVERSITET

# Indexerare, exempel

```
class Classroom
{
    private Student[] mStudents;

    public Student this[int index]
    {
        get
        {
            return mStudents[index];
        }
    }
}
```



# Yield

- Används tillsammans med return för att skapa iteratorer
- En iterator kan sen användas i t.ex. foreach-loopar istället för en lista
- "Långsammare" än att använda listor
- "Mycket" mer minneseffektiva än listor



# Yield, exempel

```
public IEnumerable<int> GetEvenNumbers(int max)
{
    List<int> result = new List<int>();
    int i = 0;

    while(i < max)
    {
        result.Add(i);
        i += 2;
    }

    return result;
}
```



# Yield, forts exempel

```
public IEnumerable<int> GetEvenNumbers(int max)
{
    int i = 0;

    while(i < max)
    {
        yield return i;
        i += 2;
    }
}
```





# Yield, forts exempel

```
public static void Main()  
{  
    var numbers = GetEvenNumbers(10);  
  
    foreach(int n in numbers)  
    {  
        //0, 2, 4, 6, 8  
        Console.WriteLine(n);  
    }  
}
```



UPPSALA  
UNIVERSITET

# Yield - LINQ

```
int[] numbers = {1,2,3,4,5};
```

```
var evenNumbers =  
    from n in numbers  
    where n % 2 == 0  
    select n;
```

```
Console.WriteLine(evenNumbers.GetType());
```