

# Objektorienterad programmering

## Föreläsning 14

© Copyright  
Mahmud Al Hakim  
[mahmud@dynamicos.se](mailto:mahmud@dynamicos.se)  
[www.webacademy.se](http://www.webacademy.se)

## Agenda

- Exceptionella händelser
  - Vanliga Programfel
  - Exception-klasser
  - Automatiskt genererade exceptionella händelser
  - Exception-klasser
  - Throw-satsen
  - Try-satsen

# Vanliga Programfel

Programfel kan indelas i tre grupper:

## 1. Kompileringsfel

Fel som uppstår för att man inte har följt språkreglerna. Denna typ av fel upptäcks av kompilatorn (enklast att hitta).

## 2. Exekveringsfel

Fel som uppstår vid körningen! T.ex. vid försök att indexera utanför gränserna i ett fält eller försök att öppna en fil som inte finns.

## 3. Logiska fel

Man har tänkt fel när man har konstruerat programmet och har använt en felaktig algoritim (svårast att hitta).

Copyright 2015 - Mahmud Al Hakim [www.webacademy.se](http://www.webacademy.se)

# Exceptionella händelser

- Vid **exekvering** ett program uppstår ibland situationer som normalt inte skall eller brukar inträffa.
- Vi kallar en sådan situation för en **exceptionell händelse** (exception på engelska).
- I C# finns en mekanism med vars hjälp man kan hantera exceptionella händelser.
- En metod i vilken ett fel uppstår **genererar** en exceptionell händelse som sedan skickas vidare till andra metoder, där man kan **fånga** händelsen och vidta någon lämplig åtgärd.

Copyright 2015 - Mahmud Al Hakim [www.webacademy.se](http://www.webacademy.se)

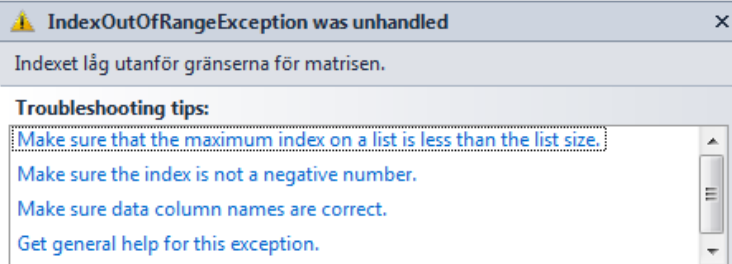
## Automatiskt genererade exceptionella händelser

- Fel som uppstår av **normala satser och uttryck** kan generera exceptionella händelser automatiskt.
- Några exempel
  1. Man försöker indexera utanför ett fält.
  2. Man försöker följa upp en referens som har värdet **null**.

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Exekveringsfel – Exempel I

```
int[] a = new int[100];  
for (int i = 0; i <= 100 ; i++)  
{  
    a[i] = 0;  
}
```



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Exekveringsfel Exempel 2

OBS!  
Detta händer om  
användaren avbryter  
programmet med  
kortkommandot  
CTRL+Z

```
Console.WriteLine("Vill du fortsätta, svara med J(Ja) eller N(Nej)");  
string s = Console.ReadLine();  
if (s.ToUpper().StartsWith("J"))  
{  
    Console.WriteLine("Ja");  
}  
else  
    Environment.Exit(0);
```

**NullReferenceException was unhandled**  
Objektreferensen har inte angetts till en instans av ett objekt.  
**Troubleshooting tips:**  
Use the "new" keyword to create an object instance.  
Check to determine if the object is null before calling the method.

**Tips!**  
Sådana fel kan man enkelt undvika genom att lägga till `s != null &&` alltså så här:  
`if (s != null && s.ToUpper().StartsWith("J"))`

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Exception-klasser

- När en exceptionell händelse inträffar sparas informationen om händelsen i ett speciellt objekt.
- Ett sådant objekt tillhör en subklass till standardklassen **Exception**.
- Exempel på klasser som används när fel uppstår vid normala satser och uttryck
  - **IndexOutOfRangeException**  
Vid indexering utanför ett fälts gränser
  - **NullReferenceException**  
När man försöker följa en null-referens
  - **DivideByZeroException**  
Vid försök att dividera ett heltal med noll

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

# Throw-satsen

- Exceptionella händelser kan genereras genom att en throw-sats exekveras.
- En sådan sats brukar ha någon av formerna
  - `throw new E();`
  - `throw new E(argument);`
- Där E är en klass som direkt eller indirekt är en subclass till standardklassen **Exception**.
- Man skapar ett nytt objekt av denna klass direkt i throw-satsen.

Copyright 2015 - Mahmud Al Hakim [www.webacademy.se](http://www.webacademy.se)

# Throw-satsen – Exempel

```
class Tidpunkt
{
    int t, m, s;
    public void SetTime(int t, int m, int s)
    {
        if (t < 0 || t > 23)
            throw new ArgumentOutOfRangeException("t");
        if (m < 0 || t > 60)
            throw new ArgumentOutOfRangeException("m");
        if (s < 0 || t > 60)
            throw new ArgumentOutOfRangeException("s");

        // Om alla argument är OK försätt
        // OBS! else-sats behövs inte

        this.t = t; this.m = m; this.s = s;
    }
}
```

Copyright 2015 - Mahmud Al Hakim [www.webacademy.se](http://www.webacademy.se)

## Throw-satsen – Exempel fort.

```
static void Main(string[] args)
{
    Tidpunkt tid = new Tidpunkt();
    tid.SetTime(24, 0, 0);
}
```

En exceptionell händelse visas vid körning, om man försöker skicka ett ogiltigt argument

```
if (t < 0 || t > 23)
```

```
throw new ArgumentOutOfRangeException("t");
```

```
if (
```

⚠ **ArgumentOutOfRangeException was unhandled**

```
if (
```

Det angivna argumentet hamnade utanför intervallet med giltiga värden.  
Parameternamn: t

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Try-satsen

- När man anropar en metod som kan generera exceptionella händelser, så kan man speciellt ange att man är beredd att fånga dessa händelser.
- Detta gör man genom att använda sig a en try-sats.

```
try {
    satser
}
catch (E e) {
    satser
}
finally {
    satser
}
```

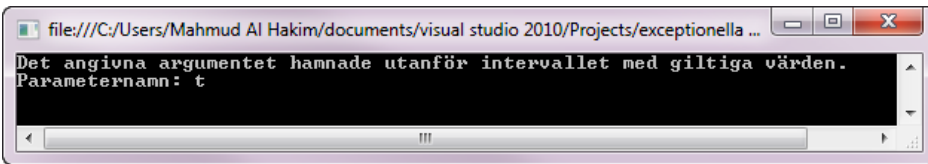
**E** betecknar namn på exceptionklass.  
**e** är godtycklig parameternamn.  
Man får ha hur många hanterare (catch-satser) som helst.  
Satser som finns i finally exekveras alltid allra sist (finally kan utelämnas)

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Try-satsen – Exempel

```
static void Main(string[] args)
{
    Tidpunkt tid = new Tidpunkt();
    try
    {
        tid.SetTime(24, 0, 0);
    }
    catch (ArgumentException e)
    {
        Console.WriteLine(e.Message);
    }
}
```

Här kan vi fånga en `ArgumentException`  
`e.Message` visar felmeddelandet!



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

## Try-satsen – Exempel 2

```
string indata;
int heltal;
try
{
    Console.Write("Ange ett heltal: ");
    indata = Console.ReadLine();
    heltal = int.Parse(indata);
}
catch (FormatException)
{
    Console.WriteLine("Felaktigt heltal! ");
}
catch (ArgumentNullException)
{
    Console.WriteLine("Värdet får inte vara null.");
}
```

`FormatException`  
fångas om `Parse` misslyckas

`ArgumentNullException`  
fångas om argumentet är null

Copyright 2015 - Mahmud Al Hakim www.webacademy.se