Klasser och objekt i C#

Från klassdiagram till C#-klass till objekt initierat av en konstruktor.

Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen Inledande programmering med C# vid Linnéuniversitetet.

Du får använda detta verk så här:

Det betyder att du i icke-kommersiella syften får:

- kopiera hela eller delar av innehållet
- sprida hela eller delar av innehållet
- visa hela eller delar av innehållet offentligt och digitalt
- konvertera innehållet till annat format
- du får även göra om innehållet

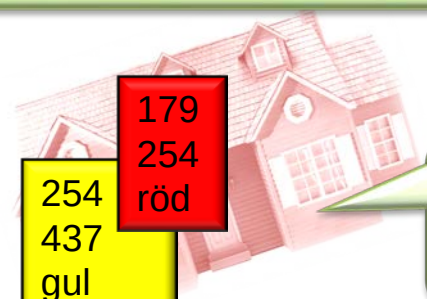
Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp och symbol samt fotografier i din nya version!

Vid all användning måste du ange källan: ”Linnéuniversitetet – Inledande programmering med C#” och en länk till https://coursepress.lnu.se/kurs/inledande-programmering-med-csharp och till Creative Common-licensen här ovan.

Klasser och objekt

En klass kan liknas med en ritning. Ritningen beskriver bl.a. de attribut som används för att beskriva ett objekt.

Rektangel
bredd
höjd
färg



254
437
gul

357
318
grön

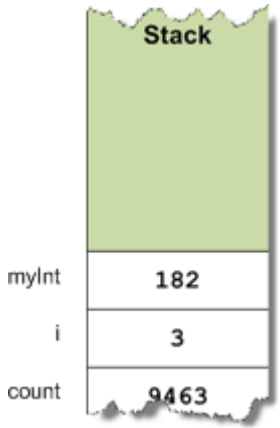
Varje objekt har sin uppsättning av värden för de olika attributen som beskrivs av klassen.

Jag tror jag förstår det här med klasser och objekt så långt. Klasser handlar om paketera ihop data på ett och samma ställe.

Ja, klassen talar vilka typer av data som ska samlas ihop, och objektet innehåller själva datat. Men hur får jag in ett objekt, en sådan där rektangel, i datorn?

Var lagras ett objekt?

```
int myInt = 182;
```

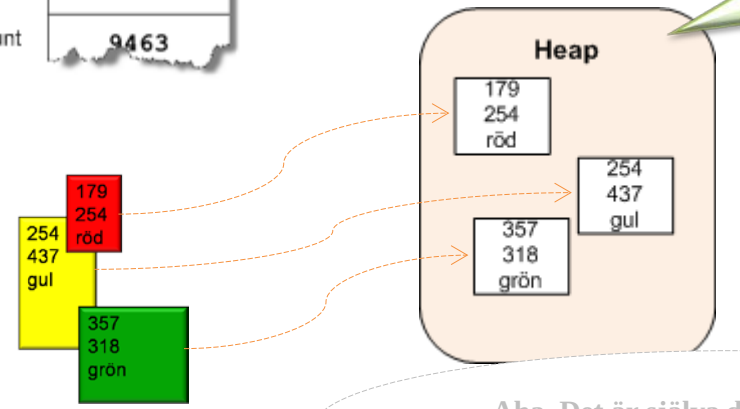


Värden som är av enkel typ, t.ex. `int` och `double`, lagras i den del av minnet som kallas stacken.

Typer som `int` och `double` kallas också värdetyper (*value types*).

Objekt, som är av klasstyp, lagras i den del av minnet som kallas heapen.

Typer som är av klasstyp kallas referenstyper (*reference types*).



Aha. Det är själva datat som beskriver rektangeln som sparas i datorns minne.

Ja, och det är datat, i det här fallet bredden, höjden och färgen, som tillsammans utgör det vi kallar objekt.

”Vem” skapar objektet?

1

Common Language Runtime (CLR)

CLR:en, som är en del av dotnetramverket, kör programmet och ansvarar bl.a. för att allokera minne på heapen till objekt. Hur mycket minne som ska allokeras…

2

ektangel
bredd
höjd
färg

…beskrivs av klassen. CLR:en läser klassdeklarationen och bestämmer på så sätt hur mycket minne som krävs, och…

3

Heap
179
254
röd

…allokera därefter tillräckligt med minne för ett objekt.

OK! Nu förstår jag mer till vad en klass bl.a. används till då programmet körs.

Hur skapar jag en klass?

nyckelord klassens namn

```
class Rectangle
{
    medlemmar
}
```

Du skapar en klass genom att skriva en klassdefinition. En klassdefinition består av:

- ✓ Klassens namn
- ✓ Klassens medlemmar

Fält (attribut) och metoder (operationer) är de viktigaste av klassens medlemmar. Fält är datamedlemmar och metoder är funktionsmedlemmar.

```
class Rectangle
{
    // Fält (fields)
    int _width;
    int _height;
    ConsoleColor _color;
}
```

Ett fält är en variabel som tillhör en klass. Här ser du en klass som har tre fält som beskriver datat som behöver lagras på heapen för ett objekt av typen `Rectangle`.

Ah! Nu förstår jag lite mer hur jag skapar en klass. Men det finns säkert mer att säga om detta.

Hur skapar jag ett objekt?

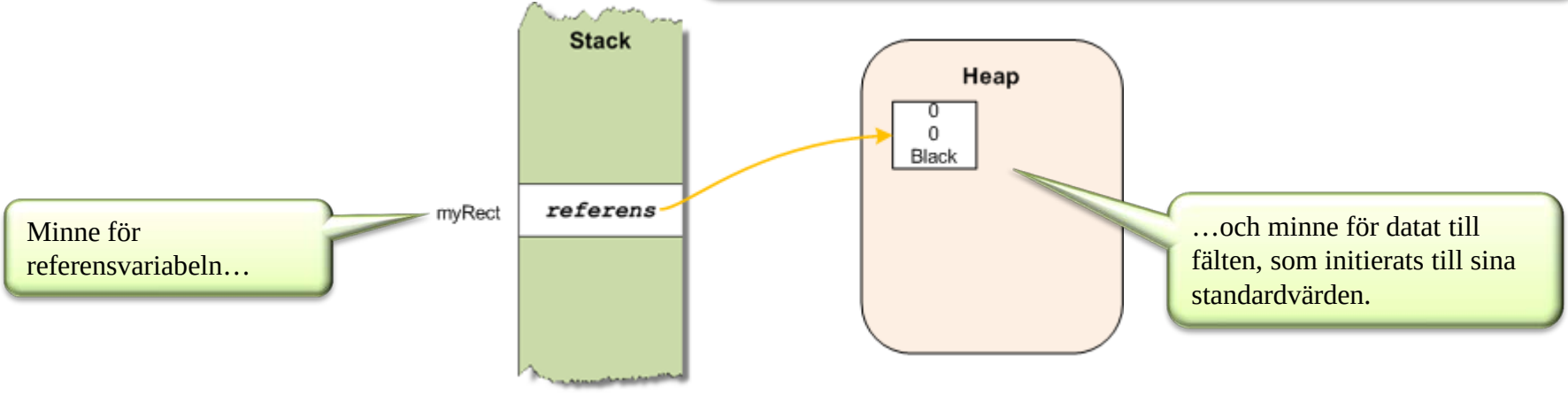
```
class Rectangle
{
    // Fält (fields)
    int _width;
    int _height;
    ConsoleColor _color;
}

class Program
{
    static void Main(string[] args)
    {
        Rectangle myRect;
        myRect = new Rectangle();
    }
}
```

standardkonstruktor

Klasser är referenstyper vilket innebär att skapade objekt kräver minne för såväl datat som en referens till datat (objektet).

Referensen till datat lagras i en referensvariabel av klassens typ, och…

…för att skapa själva objektet, allokera minne till det, måste du använda operatorn `new`, namnet på typen och efterföljande parenteser. Detta uttryck skapar objektet och returnerar en referens till det nya objektet som skapats och initierats på heapen.



Minne för referensvariabeln…

…och minne för datat till fälten, som initierats till sina standardvärden.

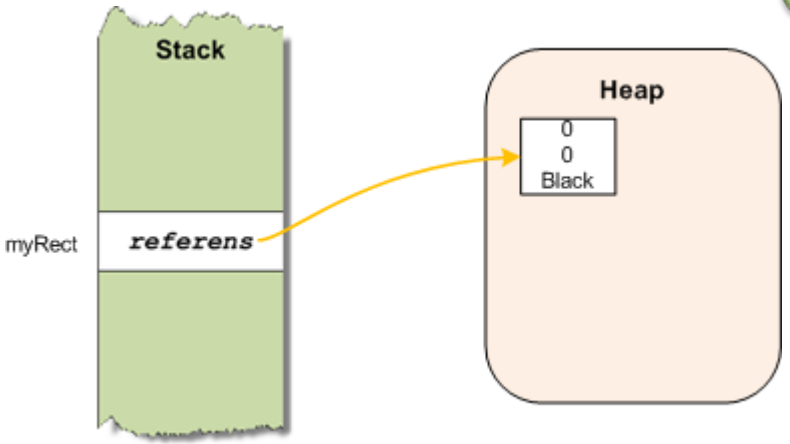
Standardkonstruktorn ”initierar” objektet

```
class Rectangle
{
    // Fält (fields)
    int _width;
    int _height;
    ConsoleColor _color;
}
```

Alla klasser måste ha en konstruktor och saknas en konstruktor skapas en sådan automatiskt. Konstruktorn som skapas saknar parameterlista och kallas därför standardkonstruktor (*default constructor*, *no-args constructor*).

```
Rectangle myRect = new Rectangle();
```

Då ett objekt skapas med hjälp av standardkonstruktorn får objektets fält standardvärden. Fält av typen `int` får t.ex. värdet `0` och fält av typen `double` får värdet `0.0`. Fältens standardvärden beror av vilken typ de är.



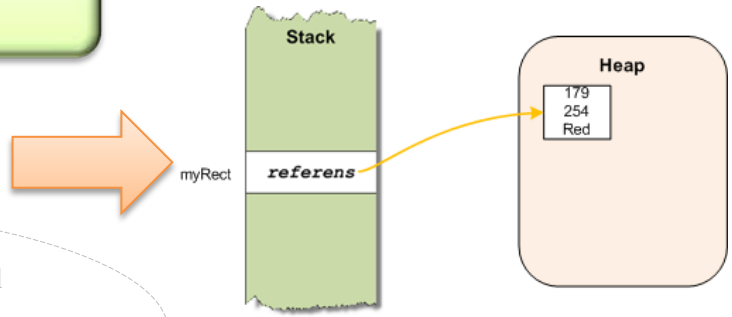
Din egna konstruktor

```
class Rectangle
{
    // Fält (fields)
    int _width;
    int _height;
    ConsoleColor _color;

    // Konstruktorer (constructors)
    public Rectangle(int width, int height, ConsoleColor color)
    {
        _width = width;
        _height = height;
        _color = color;
    }
}
static void Main(string[] args)
{
    Rectangle myRect = new Rectangle(179, 254, ConsoleColor.Red);
    ...
}
```

Konstruktorer är speciella metoder som används då objekt skapas.

En konstruktor är publik (i regel), har ingen returtyp, heter samma sak som klassen och har en parameterlista (som kan vara tom).

Konstruktorn har tre parametrar så alla fält i klassen kan tilldelas värden då ett objekt skapas och initieras.



Fungerar det så här? Värdet 179 kopieras till parametern `width` i konstruktorn och sedan kopieras värdet vidare från parametern `width` till fältet `_width`.

Ja! Du förstår precis hur det fungerar!

…nu vet du en hel del!

Nu förstår jag hjälpligt vad en klass är.

Jag vet hur jag skapar objekt.

Hur jag skriver en konstruktor vet jag också.

Jag vet ju en hel del!

Jag har ju kläm på datat, eller fälten heter det visst. Nu återstår bara vad jag kan göra med datat.

Dags att titta på det här med metoder och egenskaper. Det kan ju inte vara så svårt. De ”opererar” ju bara på datat.