C#- Array

Vad är en array

En *array* används för att lagra flera olika värden av samma typ. Detta kan jämföras med en bokhylla som har flera fack med en bok stående i. Om vi har massor av böcker som vi vill ska vara samlade så är det mer logiskt av oss att ställa dessa i en bokhylla än att ge dom en varsin hylla per bok. Samma sak gäller en *array*. Säg att vi har en uppgift att samla ihop 5 människor och sedan kolla medelvärdet på deras ålder, då är en *array* perfekt för denna uppgift. I det här exemplet har vi 5 personer vi har samlad ihop och frågat åldern på. Då vet vi att vi har fem heltal att arbeta med och vi väljer att skapa en *int array,* då ser det ut så här:
Exempel: int[] medel = new int[5];
Int[] berättar att vi vill skapa en variabel av typen *int array* där ”[]” pekar på att variabeln är av typen *array*. Därefter namnger vi vår variabel följt av tilldelningen ”= new int[5]” där vi säger att vi skapar en ny *int array* med storleken 5.

Olika sätt att tilldela värden till en array

Med exemplet ovan där vi skapade en *array* med namnet medel och storlek 5 kan vi nu tilldela värden och detta kan göras på olika sätt. Ett sätt är att fylla på med värden manuellt, som i detta fall är det bästa sättet då vi arbetar med olika åldrar.

Exempel:
```
medel[0] = 17;
medel[1] = 19;
medel[2] = 43;
medel[3] = 28;
medel[4] = 30;
```

Här tilldelar vi fem värden till vår *array* genom att säga till vår variabel vilken position vi vill fylla och sedan tilldelar vi ett värde. Viktigt att tänka på! Första siffran för en dator är inte 1, som för oss, utan för en dator är det första värdet alltid 0. Position 1 i vår *array* blir därför medel[0] och sista positionen blir medel[4]. Man kan även tänka att sista positionen är *storlek-1.*

Hur gör vi då om vi är osäkra på storleken på vår *array* då vi inte vet om fler än fem dyker upp efter att vi har börjat? Om vi inte vet hur många platser vi behöver så kan vi deklarera en *array* utan att ge den ett fast antal positioner.
Exempel: int[] nyMedel = new int[] {1, 2, 3, 4, 5, 6, 7};
Koden ovan tillåter oss att lägga till positioner genom att bara sätta ett kommatecken efter sista positionen och lägga till ett nytt värde så länge som vi tilldelar vår *array* ett eller flera värden vid deklarationen.

Multidimensionella arrayer

En *array* kan innehålla flera dimensioner, det vill säga flera rader. Detta kan man uppnå genom att vid deklarationen av din *array* sätter ett kommatecken mellan ”[]”.
Exempel: int[,] flerDimensioner = new int[2,3];
I exemplet har vi skapat en tvådimensionell *array* med två rader och tre kolumner. För att tilldela värden i dessa specificerar man vilken rad samt kolumn man tilldelar värdet till.

Exempel:
```
flerDimensioner[0,0] = 2;
flerDimensioner[0,1] = 5;
flerDimensioner[0,2] = 7;
flerDimensioner[1,0] = 4;
flerDimensioner[1,1] = 9;
flerDimensioner[1,2] = 1;
```

På första raden har vi värdena 2, 5 och 7 i våra 3 positioner och på andra raden har vi värdena 3, 9 och 1 i de tre positionerna.

För att skapa en tredimensionell *array* lägger man till ett ytterligare kommatecken ”[,,]” och då kan en tilldelning se ut så här:
Exempel: array3D[2,3,4] = 40;

Tvådimensionella och tredimensionella *arrayer* blir man oftast påmind om när man arbetar med vektorer för spel. *Exempel:* [x-värde, y-värde, z-värde].

Jagged Arrays

En så kallad *jagged array* är en *array* av *arrayer.* Dessa skapar man genom att tillägga ”[]” efter första ”[]” i deklarationen av variabeln.
Exempel: int[][] jaggedArray = new int[2][];
Och tilldelningen kan se ut så här:

Exempel:
```
jaggedArray[0][] = new int[2] {25, 30};
jaggedArray[1][] = new int[3] {10, 30, 50};
```

I exemplet ovan har vi skapat en *array* som innehåller två *arrayer*. Den första positionen innehåller en *array* med 2 positioner med värdena 25 och 30, och den andra positionen innehåller en *array* med 3 positioner med värdena 10, 30 och 50.