

Objektorienterad programmering

Föreläsning 9

© Copyright
Mahmud Al Hakim
mahmud@webacademy.se
www.webacademy.se

Agenda (halvdag)

- Fält
 - Grunderna
 - Fält med komponenter av struct-typ
 - Fält med referenser
 - Standardklassen Array
- Sökalgoritmer
 - Linjärsökning
 - Binärsökning

Fält (array på engelska)

- Ett fält är en sorts tabell som innehåller flera komponenter av samma slag.
- Komponenterna kan vara av vilken typ som helst, både av enkel typ eller av referenstyp.
- Exempel

```
int[ ] f;
```

Använd hakparenteser för att deklarera en fältvariabel

- **f** är en fältvariabel dvs. en referensvariabel som refererar till ett fält.
- Fältets komponenter är av typen int (heltal).

Att initiera en fältvariabel

1. Initiera ett fält med ordet new t.ex.

```
int[ ] i1 = new int[7];
```

Ett fält är ett objekt som nås via en referens

0	0	0	0	0	0	0
---	---	---	---	---	---	---

2. Initiera ett fält i samband med att det skapas t.ex.

```
int[ ] i2 = {2, 4, 6, 8, 10, 12, 14};
```

2	4	6	8	10	12	14
---	---	---	---	----	----	----

Kopiera ett fält

- Använd metoden Clone() för att göra en kopia av ett fält (kopiera innehållet från ett fält till ett annat fält).
- Metoden Clone returnerar ett värde av typen Object, därför behövs en explicit typomvandling (cast).

- Exempel

```
i1 = (int[ ]) i2.Clone();
```

- OBS!

i1 = i2 kopierar referensen. Variabeln i1 kommer att referera till samma fält som variabeln i2.

i1 == i2 undersöker om i1 och i2 refererar till samma fält.

Indexering

- Varje komponent i ett fält har ett unikt nummer som brukar kallas **index**.
- Den första komponenten har alltid index 0.
- index för den sista komponenten är alltid ett tal som är ett mindre än antalet komponenter i fältet.
- Med hjälp av index kan man komma åt en enskild komponent i ett fält t.ex.

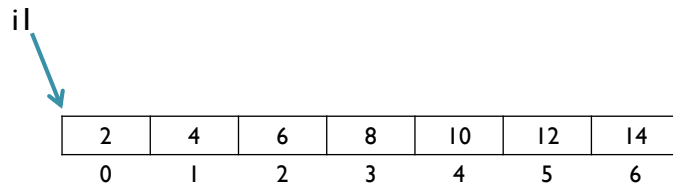
```
Console.WriteLine(i1[3]);
```

2	4	6	8	10	12	14
0	1	2	3	4	5	6

Egenskapen Length

- Med hjälp av egenskapen **Length** kan man få reda på antalet komponenter i ett fält t.ex.

```
Console.WriteLine(i1.Length);
```



i1.Length ger 7
(6+1)

Fält - Grunderna

```
static void Main(string[] args)
{
    // Skapa ett fält
    int[] i1 = new int[7];

    // Deklarera och initiera ett fält
    int[] i2 = { 2, 4, 6, 8, 10, 12, 14 };

    // Kopiera ett fält
    i1 = (int[])i2.Clone();

    // Indexering
    Console.WriteLine(i1[3]);

    // Egenskapen Length
    Console.WriteLine(i1.Length);

    Console.ReadKey();
}
```



Att löpa igenom ett fält

```
// Ett fält som innehåller kvadratrötterna ur talen 0-9

double[] kvadratrot = new double[10];

for (int i = 0; i < kvadratrot.Length; i++)
{
    kvadratrot[i] = Math.Sqrt(i);
}

// Skriv ut
for (int i = 0; i < kvadratrot.Length; i++)
{
    Console.WriteLine("Roten ur " + i + ": " + kvadratrot[i]);
}
```

Foreach-satsen

- Foreach-satsen används när man vill löpa igenom ett fält för att **avläsa** komponenterna.

Här brukar man skriva ordet **var** istället för att ange variabelns typ (implicit typade variabler)

```
foreach (double item in kvadratrot)
{
    Console.WriteLine(item);
}
```

"låt variabeln item löpa genom hela fältet kvadratrot"
item kallas iterationsvariabel (foreach iteration variable).

Fält med komponenter av struct-typ

```
struct Punkt
{
    public double x, y;
    public Punkt(int x, int y)
    {    this.x = x; this.y = y; }
}

static void Main(string[] args)
{
    Punkt[] femhörning = new Punkt[5];
    femhörning[2] = new Punkt(4, 3);
    foreach (var item in femhörning)
    {
        Console.WriteLine(item.x + "," + item.y);
    }
    Console.ReadKey();
}
```



Copyright 2015 - Mahmud Al Hakim www.webacademy.se

11

Fält med referenser

```
static void Main(string[] args)
{
    String[] meddelande = {
        "Stäng av datorn",
        "Starta om datorn",
        "Starta datorn i felsäkert läge"
    };

    Console.WriteLine(meddelande[2]);

    Console.ReadKey();
}
```

Komponenterna
är referenser till
objekt av typen
String

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

12

Standardklassen Array

- Standardklassen Array finns i namnrymden System och innehåller några statiska metoder som kan vara användbara när man arbetar med fält.
- **Sort (f)**
Sorterar komponenterna i fältet f i stigande ordning.
- **Reverse(f)**
Lägger komponenterna i fältet f i omvänd ordning
- **IndexOf (f , k)**
Ger index för första förekomsten av k i f
Ger -1 om k inte finns

Array - Exempel

```
int[] f = { 2 , 0 , 1 , 4 , 3 };

foreach (var item in f) { Console.WriteLine(item); }
Console.WriteLine(""); // En blank rad

// Lägg komponenterna i omvänd ordning
Array.Reverse(f);

foreach (var item in f) { Console.WriteLine(item); }
Console.WriteLine(""); // En blank rad

// Leta efter index för första förekomsten av 3
Console.WriteLine(Array.IndexOf(f, 3) );
Console.WriteLine(""); // En blank rad

// Sortera fältet
Array.Sort(f);

foreach (var item in f) { Console.WriteLine(item); }
```

Övning

Läs in heltal från konsolen och fyll i ett fält

```
static void Main(string[] args)
{
    // Läs in antal heltal först
    Console.Write("Hur många heltal? ");
    int antal = int.Parse(Console.ReadLine());

    // Skapa ett fält
    int[] f = new int[antal];

    // Läs in talen och spara i fältet
    for (int i = 0; i < f.Length ; i++)
    {
        Console.Write("Ange tal nr " + i + " ? ");
        f[i] = int.Parse(Console.ReadLine());
    }

    // Skriv ut
    foreach (var item in f) {Console.WriteLine(item); }
    Console.ReadKey();
}
```

Sökalgoritmer

- Det är mycket vanligt att man vill söka upp en viss information i tabeller och listor och använder man fält.
- De vanligaste sökalgoritmer är
 1. **Linjärsökning**
Man letar från början i ett fält tills man hittar den komponent som innehåller det sökta talet eller tills man sökt igenom hela fältet utan att finna det sökta talet.
 2. **Binärsökning**
Mycket effektivare än linjärsökning men kräver sorterade fält.

Linjärsökning (Skansholms algoritm)

```
// Ett fält som innehåller 15 heltal (osorterat)
int[] f = { 0, 7, 4, 6, 8, 10, 3, 6, 9, 1, 7, 2, 5, 4, 0 };

// Läs in ett heltal
Console.Write("Vilket tal söker du (mata in heltal) ? ");
int sökt = int.Parse(Console.ReadLine());

// Deklarera ett index utanför loopen (synlig)
int i;

// Kolla om talet finns i fältet
for (i = 0; i < f.Length && f[i] != sökt; i++)
    ; // OBS! enbart ett semikolon innebär fortsatt till nästa varv

if (i < f.Length)
    Console.WriteLine("Talet finns på plats nr " + i);
else
    Console.WriteLine("Talet finns ej");
```

Loopen fortsätter om man inte hittar talet

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

17

Linjärsökning (Mahmuds algoritm)

```
// Ett fält som innehåller 15 heltal (osorterat)
int[] f = { 0, 7, 4, 6, 8, 10, 3, 6, 9, 1, 7, 2, 5, 4, 0 };

// Läs in ett heltal
Console.Write("Vilket tal söker du (mata in heltal) ? ");
int sökt = int.Parse(Console.ReadLine());

// Deklarera ett index utanför loopen (synlig)
int i=0;

// Kolla om talet finns i fältet
while (i < f.Length)
{
    if (f[i] == sökt) break;
    i++;
}

if (i < f.Length)
    Console.WriteLine("Talet finns på plats nr " + i);
else
    Console.WriteLine("Talet finns ej");
```

Loopen avslutas så fort man hittar talet

Copyright 2015 - Mahmud Al Hakim www.webacademy.se

18

Binärsökning


- Om **fältet är sorterat** så kan använda en mycket effektivare sökalgoritm s.k. binärsökning.
 - Man tittar först på komponenten som ligger mitt i fältet.
 - Antag att denna innehåller värdet m .
 - Om det sökta talet är mindre än m , så vet vi att det sökta talet måste ligga i fältets vänstra del.
 - Om det sökta talet är större än m , så måste det sökta talet ligga i fältets högra del.
 - Vi kan nu betrakta den halva vi skall söka i som ett nytt fält, mindre än det ursprungliga.
 - Då använder vi samma idé som från början. Detta förfarande upprepas, med allt mindre delfält, tills vi har hittat det tal vi söker efter
 - Tips: <http://goo.gl/IUULrC>

Binärsökning – Del I

```
// Ett fält som innehåller 15 heltal (osorterat)
int[] f = { 0, 7, 4, 6, 8, 10, 3, 6, 9, 1, 7, 2, 5, 4, 8 };

// Läs in ett heltal
Console.WriteLine("Vilket tal söker du (mata in heltal) ? ");
int sökt = int.Parse(Console.ReadLine());

// Sortera fältet
Array.Sort(f);
```



Binärsökning
Fältet måste
vara sorterat

Binärsökning – Del 2

```
// första och sista håller reda på första och sista index
// i det delfält vi ska söka i
int första = 0, sista = f.Length-1;
int mitten = 0; // här sparar vi index för mitten i delfältet

while (första <= sista)
{
    mitten = (första + sista) / 2;

    if (sökt < f[mitten])
        sista = mitten - 1;
    else if (sökt > f[mitten])
        första = mitten + 1;
    else // Vi hittade talet
        break;
}
if (sökt == f[mitten])
    Console.WriteLine("Talet finns på plats nr " + mitten);
else
    Console.WriteLine("Talet finns ej");
```