



**Linnéuniversitetet**

Kalmar Vaxjö

## Övningsuppgift

---

# Repeterbara citat

Steg 2



*Författare:* Mats Looch

*Kurs:* Inledande programmering med C#

*Kurskod:* 1DV402

## 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:

Allt innehåll i verket Reperterbara citat av Mats Looock, förutom Linnéuniversitetets logotyp, symbol och kopparstick, är licensierad under:



Creative Commons Erkännande-IckeKommersiell-DelaLika 2.5 Sverige licens.  
<http://creativecommons.org/licenses/by-nc-sa/2.5/se/>

### 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, symbol och/eller kopparstick 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.

## Innehåll

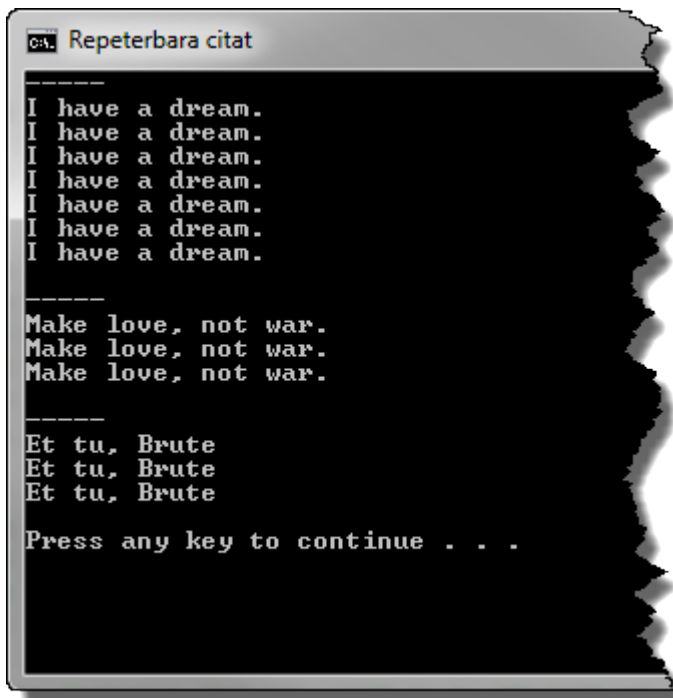
Uppgift	5
Problem	5
Mål	6
Tips	6
Lösning	8



## Uppgift

### Problem

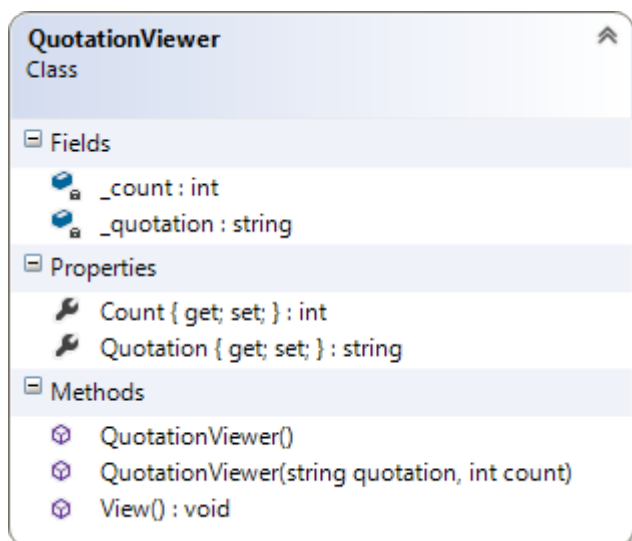
I projektet som hör till övningsuppgiften skapas två objekt av klassen `QuotationViewer` i metoden `Main`. Problemet är att klassen `QuotationViewer` saknas. Din uppgift är att skriva en klassdeklaration så att koden i `Main`-metoden kan köras utan att förändras och utan problem. Programmet ska ge utskriften enligt bilden nedan.



```
Repeterbara citat
-----
I have a dream.
I have a dream.
I have a dream.
I have a dream.
I have a dream.
I have a dream.
I have a dream.
-----
Make love, not war.
Make love, not war.
Make love, not war.
-----
Et tu, Brute
Et tu, Brute
Et tu, Brute
-----
Press any key to continue . . .
```

Figur 1. Utskrift efter att `QuotationViewer` har implementerats korrekt.

Du får inte redigera koden i klassen `Program` på något sätt. All kod du skriver ska skrivas i klassdeklarationen till klassen `QuotationViewer`. Genom att analysera figur 1, klassdiagrammet och metoden `Main` ska du komma fram till hur klassen `QuotationViewer` ska implementeras.



Figur 2. Klassdiagram över `QuotationViewer`.

```
Program.cs  [X]
RepeatableQuotes.Program

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace RepeatableQuotes
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             // Skapar ett nytt QuotationViewer-objekt med hjälp
14             // av standardkonstruktorn, använder egenskaper för
15             // att tilldela objektet värden och anropar en metod.
16             QuotationViewer qw = new QuotationViewer();
17             qw.Quotation = "I have a dream.";
18             qw.Count = 7;
19             qw.View();
20
21             // Skapar och initierar ännu ett QuotationViewer-objekt
22             // med hjälp av konstruktorn som har två parametrar och
23             // anropar en metod.
24             QuotationViewer anotherQw =
25                 new QuotationViewer("Make love, not war.", 3);
26             anotherQw.View();
27
28             // Ändrar på objektets data med hjälp av en egenskap
29             // och anropar en metod.
30             anotherQw.Quotation = "Et tu, Brute";
31             anotherQw.View();
32         }
33     }
34 }
```

Figur 3. I klassen Program, metoden Main, instansieras två QuotationViewer-objekt.

## Mål

Efter att ha gjort uppgiften ska du:

- Kunna implementera en klass genom att tolka ett klassdiagram och programkod som använder objekt instansierade av klassen.
- Förstå vad ett fält (*field*) är och hur du deklarerar ett.
- Kunna skriva en standardkonstruktör, och förstå vad konstruktörer har för roll.
- Förstå att en klass kan ha flera konstruktörer så länge de har olika parameterlistor. Kallas överlagring eller ”*overloading*”.
- Veta hur du skriver egenskaper (*properties*).
- Kunna skriva en metod som inte returnerar något värde (*void*) och som inte har någon parameterlista.

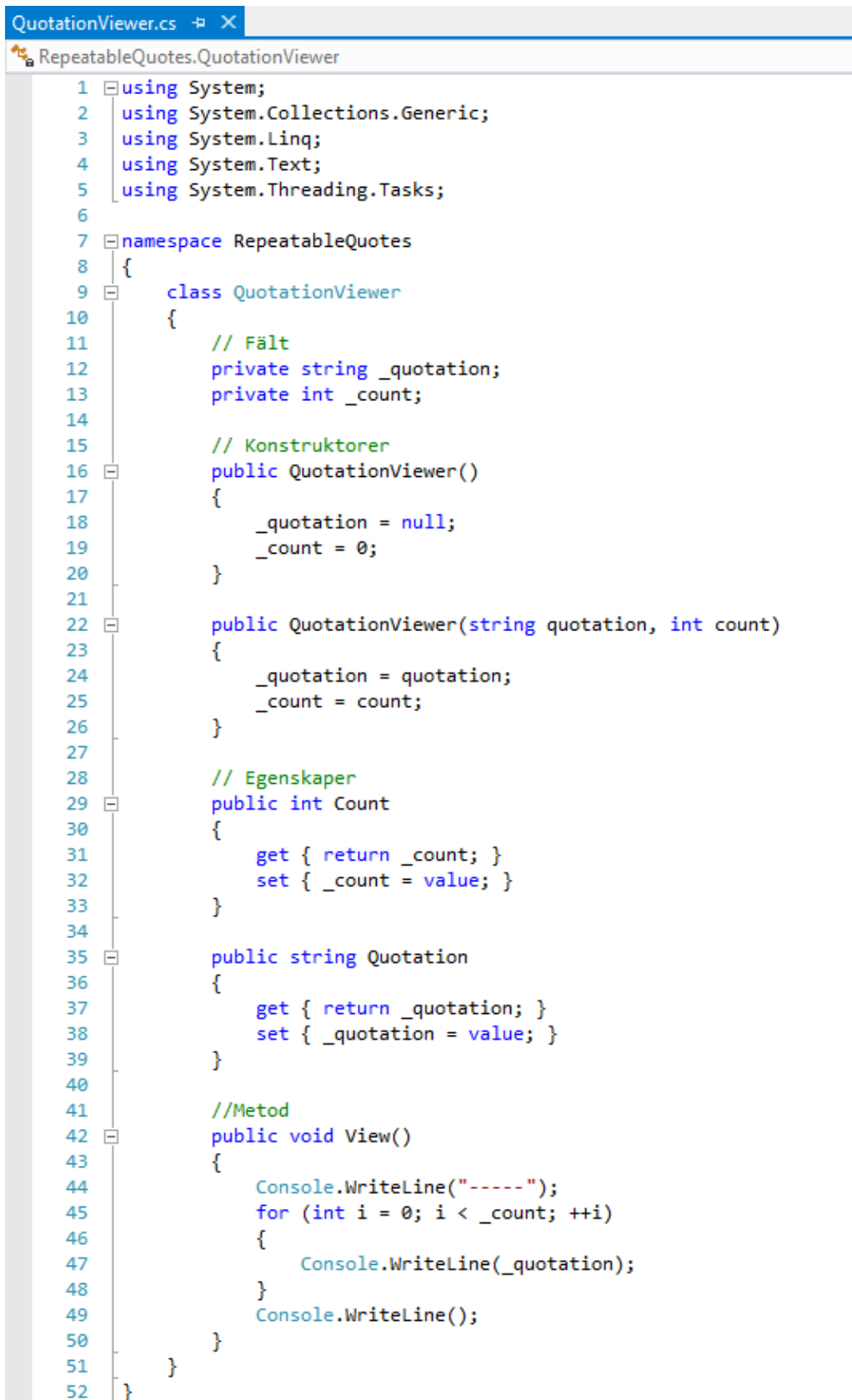
## Tips

Läs om klasser i kurslitteraturen, kapitel 5, ”*Classes*”:

- Fält under rubriken ”*Instance Fields*”.

- Metoder under rubriken "*Instance Methods*".
- Egenskaper under rubriken "*Properties*".
- Konstruktörer under rubriken "*Constructors*".

## Lösning



```
QuotationViewer.cs -# X
RepeatableQuotes.QuotationViewer

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace RepeatableQuotes
8 {
9     class QuotationViewer
10    {
11        // Fält
12        private string _quotation;
13        private int _count;
14
15        // Konstruktörer
16        public QuotationViewer()
17        {
18            _quotation = null;
19            _count = 0;
20        }
21
22        public QuotationViewer(string quotation, int count)
23        {
24            _quotation = quotation;
25            _count = count;
26        }
27
28        // Egenskaper
29        public int Count
30        {
31            get { return _count; }
32            set { _count = value; }
33        }
34
35        public string Quotation
36        {
37            get { return _quotation; }
38            set { _quotation = value; }
39        }
40
41        //Metod
42        public void View()
43        {
44            Console.WriteLine("-----");
45            for (int i = 0; i < _count; ++i)
46            {
47                Console.WriteLine(_quotation);
48            }
49            Console.WriteLine();
50        }
51    }
52 }
```

Figur 3.

Klassdiagrammet ger de två privata fälten `_count` och `_quotation` samt dess typer. Under rubriken *Properties* i klassdiagrammet hittar du de två publika egenskaperna som kapslar in respektive fält.



Under metoder finns de två konstruktörerna som ska implementeras. Observera att konstruktörer saknar returvärde och måste ha samma namn som klassen. Det som skiljer konstruktörerna åt är deras signaturer, d.v.s. namnet på konstruktorn och dess parameterlista. Titta gärna i metoden `Main` hur respektive konstruktor används.

Den publika metoden `View` skriver helt enkelt ut datat det privata fältet `_quotation` innehåller `_count` gånger. Hur utskriften ska ske torde framgå av figur 1.