



Linnéuniversitetet
Kalmar Vaxjö

Övningsuppgift

Statisk addition

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 Statisk addition 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	7

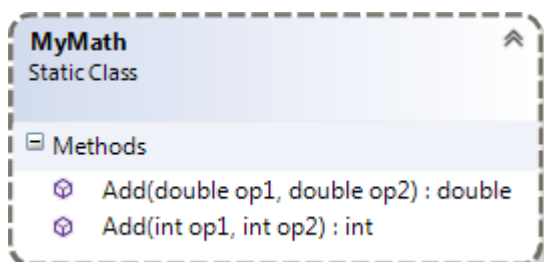
Uppgift

Problem

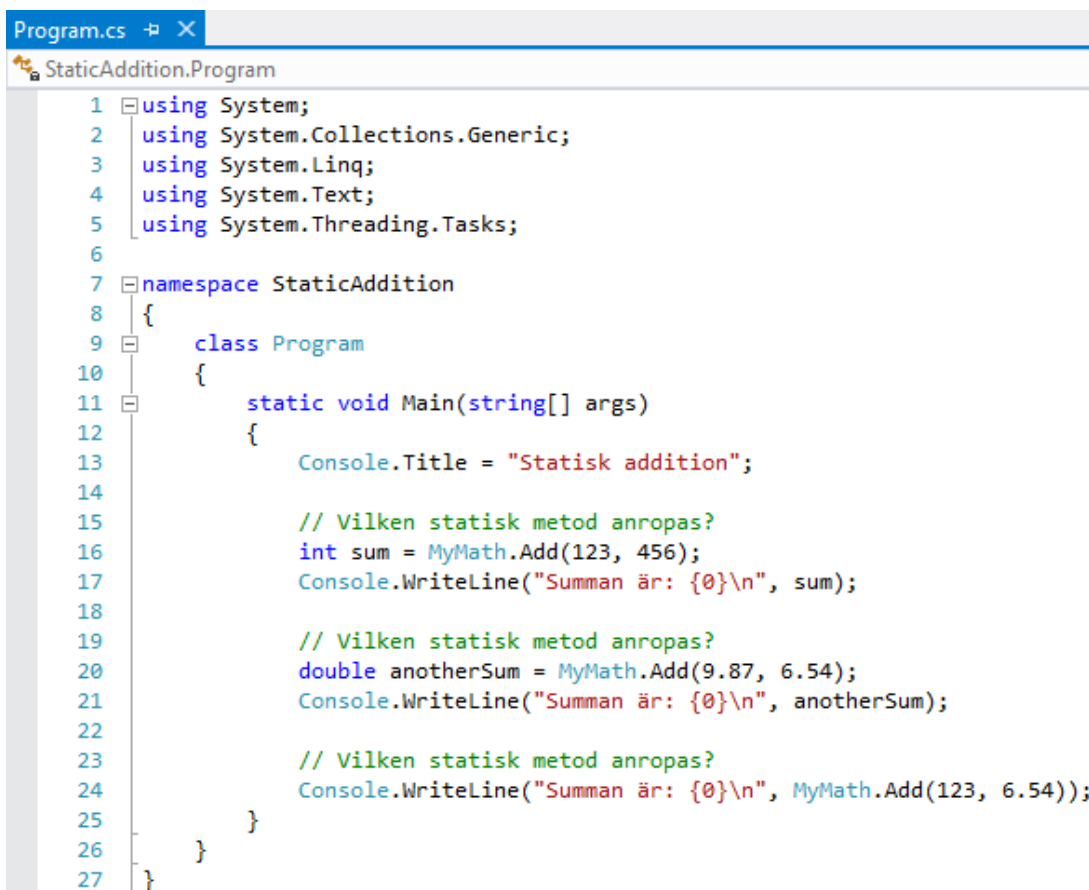
Komplettera det ofullständiga projektet med den statiska klassen `MyMath` så att metoden `Main` kan kompileras och exekveras utan problem.

Klassen `MyMath` ska ha två statiska metoder, som båda ska heta `Add` men ha olika parameterlistor och därmed olika signaturer. Den ena metoden ska returnera summan av två heltal av typen `int`, och den andra metoden ska returnera summan av två flyttal av typen `double`.

Metoden `Main` i klassen `Program` får du inte ändra på något sätt. Koden ska oförändrad kunna använda de statiska metoderna i klassen `MyMath` som du ska implementera (skriva).



Figur 1. Klassdiagram över den statiska klassen `MyMath` med de två statiska metoderna.



```
Program.cs
StaticAddition.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 StaticAddition
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.Title = "Statisk addition";
14
15            // Vilken statisk metod anropas?
16            int sum = MyMath.Add(123, 456);
17            Console.WriteLine("Summan är: {0}\n", sum);
18
19            // Vilken statisk metod anropas?
20            double anotherSum = MyMath.Add(9.87, 6.54);
21            Console.WriteLine("Summan är: {0}\n", anotherSum);
22
23            // Vilken statisk metod anropas?
24            Console.WriteLine("Summan är: {0}\n", MyMath.Add(123, 6.54));
25        }
26    }
27 }
```

Figur 2. Metoden `Main` som använder de statiska metoderna `Add`.

Fundera lite över vilken av de två statiska metoderna som kommer att anropas av de olika satserna. Vad är det som bestämmer det? Jo, typen som argumenten är av. Nog med ledtrådar...

Mål

Efter att ha gjort uppgiften ska du:

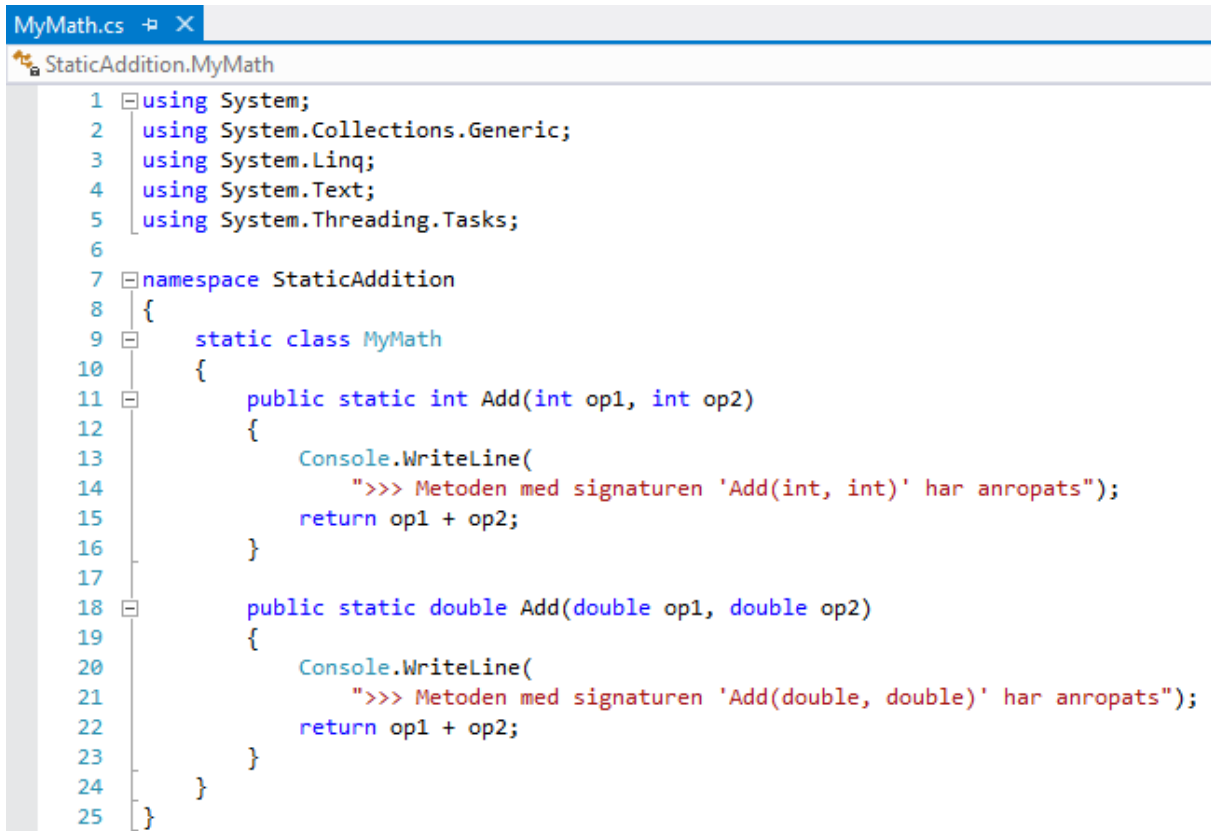
- Kunna skriva en statisk metod.
- Förstå vad signaturer och överlagring (*overloading*) innebär i samband med metoder.
- Kunna skriva en metod som har en parameterlista.
- Kunna skriva en metod som returnerar ett värde.

Tips

Läs om:

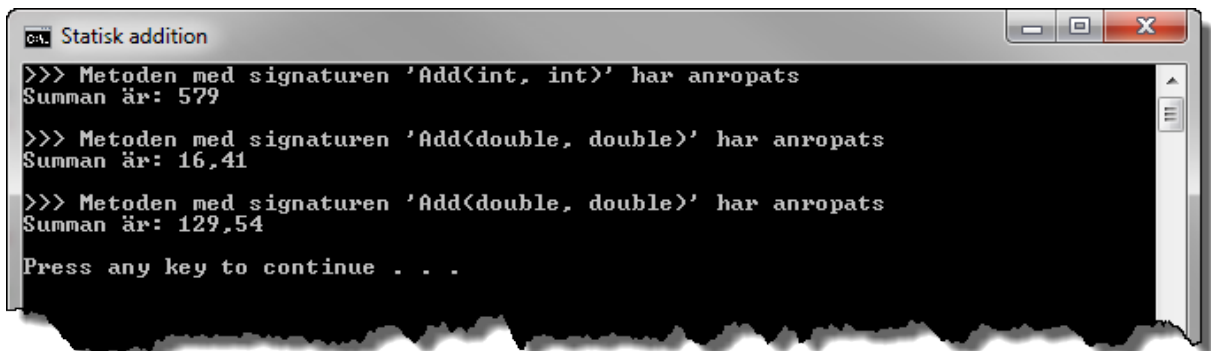
- Allmänt om metoder i kurslitteraturen, kapitel 4, främst under rubrikerna "*Calling a Method*" och "*Declaring a Method*".
- Överlagring av metoder i kurslitteraturen, kapitel 4, under rubriken "*Method Overloading*".
- Statiska metoder behandlas i kurslitteraturen, kapitel 5, under underrubriken "*Static Methods*". Online-hjälpen "*Static Classes and Static Class Members (C# Programming Guide)*" <http://msdn.microsoft.com/en-us/library/79b3xss3.aspx> får komplettera.

Lösning



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace StaticAddition
8 {
9     static class MyMath
10    {
11        public static int Add(int op1, int op2)
12        {
13            Console.WriteLine(
14                ">>> Metoden med signaturen 'Add(int, int)' har anropats");
15            return op1 + op2;
16        }
17
18        public static double Add(double op1, double op2)
19        {
20            Console.WriteLine(
21                ">>> Metoden med signaturen 'Add(double, double)' har anropats");
22            return op1 + op2;
23        }
24    }
25 }
```

Figur 3. Klassen MyMath implementerad.



Figur 4. Utskriften programmet ger med "debug-utskrifter" som visar vilken metod som anropats.

För att anropa en statisk metod använder du klassens namn, inte en referens till ett objekt. Statiska metoder kan faktiskt bara anropas med hjälp av klassens namn.

Klassens medlemmar begränsas till två statiska metoder varför hela klassen kan "märkas" som statisk.

Metoden som returnerar ett heltal av typen `int` har även två parametrar av typen. Summan av parameterarna returneras. Lagg märke till att du inte behöver lagra resultatet i en variabel vars värde returneras utan att det går alldeles utmärkt att returnera resultatet av uttrycket `op1 + op2`. Metoden inleds med en "debug-utskrift" som skriver ut vilken metod det är som anropats.

Den andra metoden, som också heter `Add`, returnerar ett flyttal av typen `double`. Fungerar i övrigt på samma sätt som metoden som returnerar ett heltal.

Det finns alltså inget som hindrar att metoder har samma namn så länge som dess signaturer skiljer sig åt. Observera att returtypen INTE är en del av signaturen.