



Linnéuniversitetet

Kalmar Vaxjö

Laborationsanvisning

Godtycklig lönerevision

Steg 1, laborationsuppgift 3



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 Godtycklig lönerrevision av Mats Loock, 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

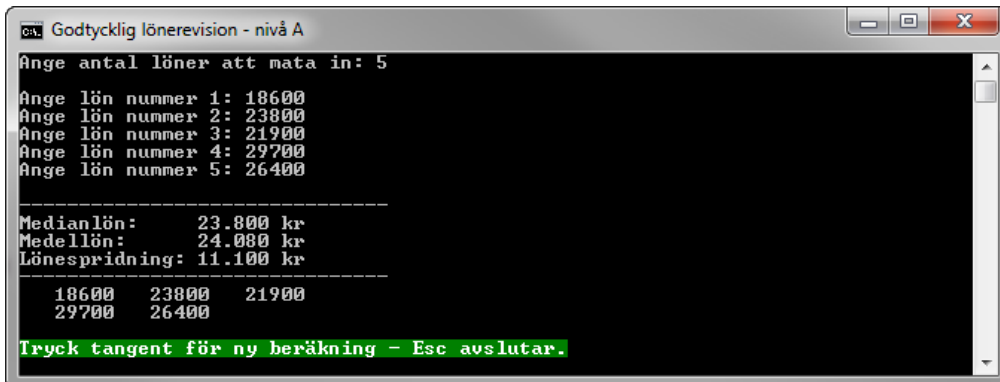
A. Uppgift	5
Problem	5
Beräkningar	5
Uppdelning av programmet	6
A-krav	7
Läsvärt	7
B. Uppgift	9
Beräkningar	9
Uppdelning av programmet	10
B-krav	12
Läsvärt	12
C. Uppgift	15
Beräkningar	15
Klassen Program	16
Klassen MyExtensions	17
C-krav	18
Läsvärt	18

A. Uppgift

Problem

Då en del fackförbund informerar om utfallet av löneförhandlingar redovisar de medianlönen (den mittersta lönen), genomsnittslönen och lönespridningen (skillnaden mellan den högsta och lägsta lönen). Skriv ett program som kan läsa in ett godtyckligt antal löner och sedan beräkna medianlön, medellön och lönespridning.

Programmet ska presentera, efter de beräknade värdena, lönerna i den ordning de matats in i med tre löner per rad.



```

Godtycklig lönerrevision - nivå A
Ange antal löner att mata in: 5
Ange lön nummer 1: 18600
Ange lön nummer 2: 23800
Ange lön nummer 3: 21900
Ange lön nummer 4: 29700
Ange lön nummer 5: 26400

-----
Medianlön:      23.800 kr
Medellön:       24.000 kr
Lönespridning:  11.100 kr
-----

18600  23800  21900
29700  26400

Tryck tangent för ny beräkning - Esc avslutar.
  
```

Figur A.1. Exempel på exekvering av programmet.

För att kunna bestämma medianlönen måste samtliga inmatade löner sparas i en array av typen `int[]`. Matar användaren in något annat än vad som kan tolkas som ett heltal ska inte programmet avslutas utan ett felmeddelande ska visas som uppmanar användaren att mata in ett heltal.

Då inmatning, beräkning och presentation är gjord ska användaren kunna välja att avsluta programmet genom att trycka på Escape-tangenten. Trycker användaren ner någon annan tangent ska användaren på nytt kunna ange antalet löner att mata in.

Beräkningar

Medellön och lönespridning

Medellönen och lönespridningen kan beräknas på traditionellt sätt genom att iterera igenom arrayen med löner. Med hjälp av aggregatfunktionerna `Average()`, `Max()` och `Min()` är det dock enklare att beräkna medelvärde, det högsta och lägsta värdet för en array innehållande numeriska värden.

Medianlön

Medianlönen är den mittersta lönen och för att kunna bestämma den måste lönerna sorteras. Enklast att sortera en array är att använda den statiska metoden `Sort()`.

Är det ett udda antal löner bestäms medianlönen av den mittersta lönen.

18600	21900	23800	26400	29700
		↑		
		medianlön		

Exempel A.1. Median vid udda antal löner.

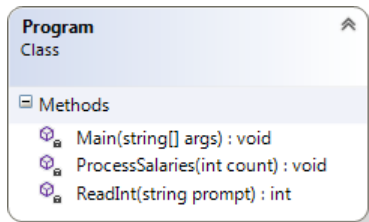
Vid jämt antal löner så bestäms medianlönen genom att medelvärde av de två mittersta lönerna beräknas.

18600	21900	23800	26400	29700	32100
		↑	↑		
		$(23800 + 26400) / 2 = 25100$			
			↑		
			medianlön		

Exempel A.2. Median vid jämt antal löner.

Uppdelning av programmet

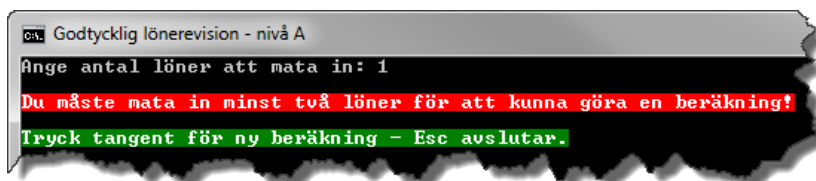
Programmet ska delas upp i tre privata statiska metoder som återfinns i Figur A.2.



Figur A.2. Klassdiagram över klassen Program.

Main

Metoden `Main` ska genom att anropa `ReadInt()` läsa in antalet löner som användaren vill mata in. Om antalet är minst två ska metoden `ProcessSalaries()` anropas och antalet löner att bearbeta skickas med som ett argument. Är antalet löner att bearbeta färre än två ska ett felmeddelande presenteras.



Figur A.3. Felmeddelande då antalet löner att mata in är för få.

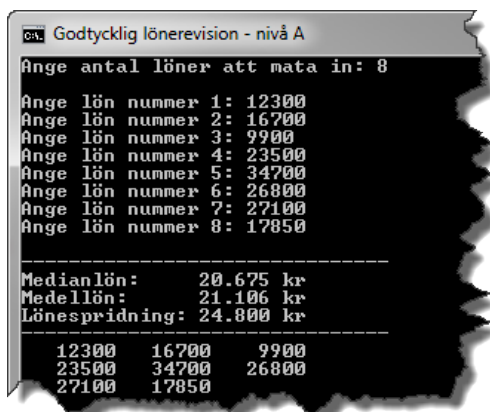
Vill du att användaren ska trycka vilken tangent som helst för att fortsätta, utan att behöva trycka på Enter-tangenten, använd dig då av `Console.ReadKey(true).Key != ConsoleKey.Escape` på lämpligt sätt.

ProcessSalaries

`ProcessSalaries()` uppgift är att läsa in lönerna till en lokal array av typen `int[]`. Efter att lönerna lästs in kan median, medellön och lönespridning beräknas. De beräkna resultaten ska presenteras på lämpligt sätt, formaterade som valuta, varefter de inmatade lönerna ska listas, med tre löner på varje rad, i den ordning de matades in.

För att medianlönen ska kunna beräknas måste arrayen med löner sorteras. En kopia av där nödvändig att skapa för att lönerna ska kunna presenteras i den ordning de matats in efter att median lönen beräknats.

Då lönerna ska listas med tre löner per rad kan modulusoperatoren (%) komma till användning. Tänk även på att det är lämpligt att högerjustera lönerna så att ental, tiotal osv. hamnar under varandra.



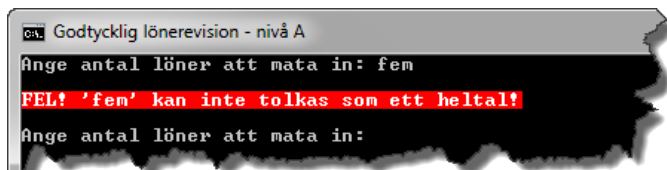
Figur A.4. Inmatade löner presenterade högerjusterade tre per rad i den ordning de matats in.

ReadInt

Metoden `ReadInt()` ska returnera ett värde av typen `int`.

Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning efter att ett tydligt felmeddelande presenterats (se Figur A.5).

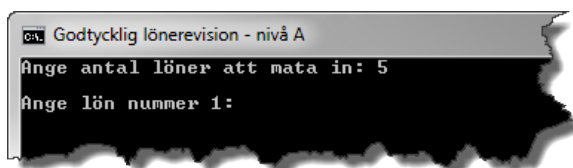
Till metoden ska det vara möjligt att skicka med ett argument. Argument ska vara en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. I Figur A.5 har argumenten "Ange antal löner att mata in: " anropet av metoden.



```
CS> Godtycklig lönerrevision - nivå A
Ange antal löner att mata in: fem
FEL! 'fem' kan inte tolkas som ett heltal!
Ange antal löner att mata in:
```

Figur A.5. Användaren får en ny chans att ange antalet löner att mata in efter att ett fel konstaterats.

Samma metod används för att läsa in en lön. I Figur A.6 har argumentet "Ange lön nummer 1: " anropet av metoden



```
CS> Godtycklig lönerrevision - nivå A
Ange antal löner att mata in: 5
Ange lön nummer 1:
```

Figur A.6. Metoden ReadInt har anropats med två olika argument.

A-krav

1. Indata till programmet ska vara antalet löner som ska matas in samt lönerna.
 - a) Antalet löner ska anges i form av ett heltal.
 - b) En lön ska anges i form av ett heltal.
2. Inläsning av antalet löner och själva lönerna ska ske via en och samma metod som returnerar ett heltal.
3. Eventuella fel i samband med inläsning av ett värde ska tas om hand med hjälp av en "try-catch"-sats och användaren ska få en ny möjlighet att mata in ett korrekt värde efter att ett relevant felmeddelande presenterats för användaren.
4. Inmatade löner ska lagras i en array av typen `int[]`.
5. Median-, medellön och lönespridning ska beräknas och presenteras formaterade som valuta utan decimaler.
6. Efter att beräkning och presentation av median- och medellön samt lönespridning gjorts ska lönerna presenteras i den ordning de matats in om tre löner på varje rad där ental, tiotal, osv. ska vara placerade ovanför varandra.
7. Flera beräkningar ska kunna göras efter varandra. Programmet ska avbrytas först då användaren trycker på Escape-tangenten.

Läsvärt

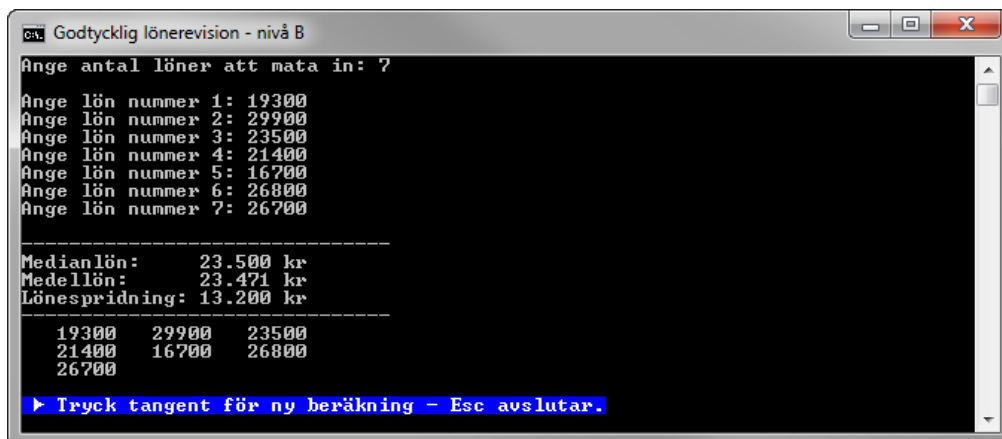
- variabler
 - Essential C# 5.0, 13-17.
 - [http://msdn.microsoft.com/en-us/library/hh147285\(VS.88\).aspx#Variables](http://msdn.microsoft.com/en-us/library/hh147285(VS.88).aspx#Variables)
- `String.Format`
 - Essential C# 5.0, 49.
 - <http://msdn.microsoft.com/en-us/library/fht0f5be.aspx>

- Arrayer
 - Essential C# 5.0, 67-82 (inte flerdimensionella arrayer).
 - <http://msdn.microsoft.com/en-us/library/system.array.aspx>
 - [http://msdn.microsoft.com/en-us/library/hh127989\(v=VS.88\).aspx#Complex](http://msdn.microsoft.com/en-us/library/hh127989(v=VS.88).aspx#Complex) (under rubriken Arrays)
- %-operatorn
 - Essential C# 5.0, 87-89.
 - <http://msdn.microsoft.com/en-us/library/0w4e0fzs.aspx>
- "if"-satsen
 - Essential C# 5.0, 107-114.
 - <http://msdn.microsoft.com/en-us/library/5011f09h.aspx>
- "while"-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/2aeyhxcd.aspx>
- "do-while"-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370s1zax.aspx>
- "for"-satsen
 - Essential C# 5.0, 130-133.
 - <http://msdn.microsoft.com/en-us/library/ttw7t8t6.aspx>
- Metoder
 - Essential C# 5.0, 155-167.
 - <http://msdn.microsoft.com/en-us/library/ms173114.aspx>
- Undantag
 - Essential C# 5.0, 194-201.
 - Essential C# 5.0, 423-430.
 - <http://msdn.microsoft.com/en-us/library/0yd65esw.aspx>
 - <http://msdn.microsoft.com/en-us/library/1ah5wsex.aspx>
 - <http://msdn.microsoft.com/en-us/library/ww58ded5.aspx>
- Aggregatfunktioner
 - Essential C# 5.0, 609.
- Hantering av färger i ett konsolfönster
 - <http://msdn.microsoft.com/en-us/library/yae1s0f9.aspx>
 - <http://msdn.microsoft.com/en-us/library/s66hf68a.aspx>
 - <http://msdn.microsoft.com/en-us/library/d3zkyxe.aspx>

B. Uppgift

Då en del fackförbund informerar om utfallet av löneförhandlingar redovisar de medianlönen (den mittersta lönen), genomsnittslönen och lönespridningen (skillnaden mellan den högsta och lägsta lönen). Skriv ett program som kan läsa in ett godtyckligt antal löner och sedan beräkna medianlön, medellön och lönespridning.

Programmet ska presentera, efter de beräknade värdena, lönerna i den ordning de matats in i med tre löner per rad.



Figur B.1. Exempel på exekvering av programmet.

För att kunna bestämma medianlönen måste samtliga inmatade löner sparas i en array av typen `int[]`. Matar användaren in något annat än vad som kan tolkas som ett heltal ska inte programmet avslutas utan ett felmeddelande ska visas som uppmanar användaren att mata in ett heltal.

Då inmatning, beräkning och presentation är gjord ska användaren kunna välja att avsluta programmet genom att trycka på Escape-tangenten. Trycker användaren ner någon annan tangent ska användaren på nytt kunna ange antalet löner att mata in.

Beräkningar

Medellön och lönespridning

Medellönen och lönespridningen kan beräknas på traditionellt sätt genom att iterera igenom arrayen med löner. Med hjälp av aggregatfunktionerna `Average()`, `Max()` och `Min()` är det dock enklare att beräkna medelvärdet, det högsta och lägsta värdet för en array innehållande numeriska värden.

Medianlön

Medianlönen är den mittersta lönen och för att kunna bestämma den måste lönerna sorteras. Enklarest att sortera en array är att använda den statiska metoden `Sort()`.

Är det ett udda antal löner bestäms medianlönen av den mittersta lönen.

18600	21900	23800	26400	29700
		↑		
		medianlön		

Exempel B.1. Median vid udda antal löner.

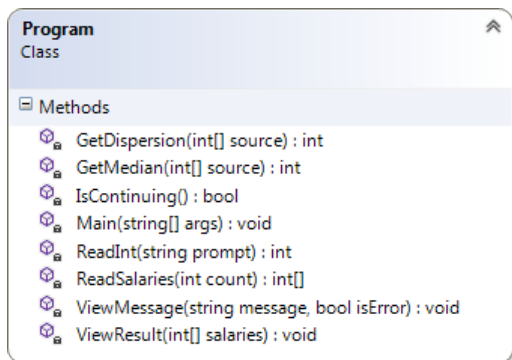
Vid jämt antal löner så bestäms medianlönen genom att medelvärdet av de två mittersta lönerna beräknas.

18600	21900	23800	26400	29700	32100
		↑	↑		
		$(23800 + 26400) / 2 = 25100$			
			↑		
			medianlön		

Exempel B.2. Median vid jämt antal löner.

Uppdelning av programmet

Programmet ska delas upp i åtta privata statiska metoder som återfinns i Figur B.2.

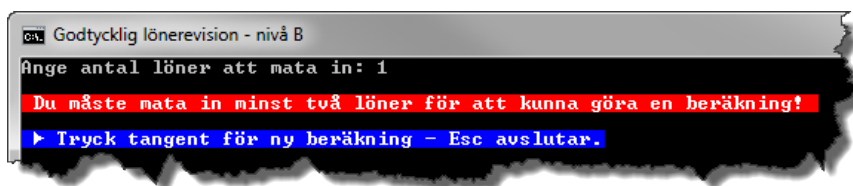


Figur B.2. Klassdiagram över klassen Program.

Main

Metoden ska anropa metoderna `ReadInt()`, `ReadSalaries()`, `ViewResult()` och `IsContinuing()`. Anropen ska placeras i en "do-while"-sats som avslutas då metoden `IsContinuing()` returnerar `false` vilket ska ske då användaren trycker på Escape-tangenten.

Metoden `ReadInt()` anropas för att ta reda på antalet löner som användaren vill mata in. Är antalet löner att bearbeta färre än två ska ett felmeddelande presenteras enligt Figur B.3.



Figur B.3. Felmeddelande då antalet löner att mata in är för få.

Om antalet löner att mata in är minst två ska metoden `ReadSalaries()` anropas och antalet löner att läsa in skickas med som ett argument och en array med lönerna returneras.

De inlästa lönerna skickas med som argument till metoden `ViewResult`, som ser till att beräkningar och presentation av resultat sker.

GetDispersion

Metoden ska beräkna lönespridningen på de löner som skickas som argument i form av en array till metoden.

GetMedian

Metoden ska beräkna medianlönen av de löner som skickas som argument i form av en array till metoden.

IsContinuing

Uppgiften `IsContinuing()` har är att presentera ett meddelande som uppmanar användaren att trycka på en tangent för att fortsätta, eller Escape-tangenten för att avsluta programmet. Trycker användaren på Escape-tangenten ska metoden returnera `false`; vilken annan tangent som helst ska leda till att `true` returneras.

ReadInt

Metoden `ReadInt()` ska returnera ett värde av typen `int`.

Metoden ska användas för att läsa in så väl antalet löner som en lön. Till metoden ska det därför vara möjligt att skicka med ett argument.

Argumentet ska vara en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. I Figur B.4 har argumentet "Ange antal löner att mata in: " respektive argumentet "Ange lön nummer 1: " används då metoden anropats.

```

C# Godtycklig lönerrevision - nivå B
Ange antal löner att mata in: 7
Ange lön nummer 1: för lite!
FEL! 'för lite!' kan inte tolkas som ett heltal!
Ange lön nummer 1:

```

Figur B.4 Metoden `ReadInt()` har anropats med två olika argument. Användaren får även en ny chans att mata in ett nytt värde efter att ett fel konstaterats.

Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning efter att ett tydligt felmeddelande presenterats.

ReadSalaries

Metodens uppgift är att läsa in lönerna, genom anrop av metoden `ReadInt()`, och lagra dessa i en lokal array av typen `int[]`, som sedan returneras.

ViewMessage

Metoden anropas för att presentera meddelanden för användaren. Två typer av meddelanden ska kunna presenteras, vanliga meddelanden och felmeddelanden. Andra parameterns (`isError`) värde bestämmer typen av meddelande.

```

19300 29900 23500
21400 16700 26800
26700
Tryck tangent för ny beräkning - Esc avslutar.

```

Figur B.5. Meddelande formaterat som ett vanligt meddelande då andra parameterns (`isError`) värde är `false` (dess standardvärde).

```

Ange lön nummer 1: 12300.50
FEL! '12300.50' kan inte tolkas som ett heltal!
Ange lön nummer 1:

```

Figur B.6. Meddelande formaterat som felmeddelande då andra parameterns (`isError`) värde är `true`.

ViewResult

Metodens parameter refererar till en array med inmatade löner. I samband med att medianlön, medellön och lönespridning presenteras på lämpligt sätt, formaterade som valuta, anropas metoderna som utför beräkningarna varefter de inmatade lönerna ska listas, med tre löner på varje rad, i den ordning de matades in.

Medianlönen fås genom ett anrop av metoden `GetMedian()`. Medellönen fås genom ett anrop av aggregatfunktionen `Average()`. Lönespridningen fås genom ett anrop av metoden `GetDispersion()`.

Då lönerna ska listas med tre löner per rad kan modulusoperatoren (%) komma till användning. Tänk även på att det är lämpligt att högerjustera lönerna så att ental, tiotal osv. hamnar under varandra.

```

C# Godtycklig lönerrevision - nivå B
Ange antal löner att mata in: 5
Ange lön nummer 1: 700
Ange lön nummer 2: 15900
Ange lön nummer 3: 12300
Ange lön nummer 4: 21600
Ange lön nummer 5: 9800

Medianlön:      12.300 kr
Medellön:       12.060 kr
Lönespridning:  20.900 kr

700  15900  12300
21600  9800
Tryck tangent för ny beräkning - Esc avslutar.

```

Figur B.7. Inmatade löner presenterade högerjusterade tre per rad i den ordning de matats in.

B-krav

1. Indata till programmet ska vara antalet löner som ska matas in samt lönerna.
 - a) Antalet löner ska anges i form av ett heltal.
 - b) En lön ska anges i form av ett heltal.
2. Inläsning av antalet löner och lönerna i sig ska ske via en och samma metod som returnerar ett heltal.
3. Eventuella fel i samband med inläsning av ett värde ska tas om hand med hjälp av en "try-catch"-sats och användaren ska få en ny möjlighet att mata in ett korrekt värde efter att ett relevant felmeddelande presenterats för användaren.
4. Inmatade löner ska lagras i en array av typen `int[]`.
5. Medianlön och lönespridning ska beräknas i separata metoder till vilka en array med löner skickas med som argument.
6. Median-, medellön och lönespridning ska presenteras formaterade som valuta utan decimaler.
7. Efter att beräkning och presentation av median- och medellön samt lönespridning gjorts ska lönerna presenteras i den ordning de matats in om tre löner på varje rad där ental, tiotal, osv. ska vara placerade ovanför varandra.
8. Flera beräkningar ska kunna göras efter varandra. Programmet ska avbrytas först då användaren trycker på Escape-tangenten.

Läsvärt

- variabler
 - Essential C# 5.0, 13-17.
 - [http://msdn.microsoft.com/en-us/library/47285\(VS.88\).aspx#Variables](http://msdn.microsoft.com/en-us/library/47285(VS.88).aspx#Variables)
- `String.Format`
 - Essential C# 5.0, 49.
 - <http://msdn.microsoft.com/en-us/library/fht0f5be.aspx>
- `Array`
 - Essential C# 5.0, 67-82 (inte flerdimensionella arrayer).
 - <http://msdn.microsoft.com/en-us/library/system.array.aspx>
 - [http://msdn.microsoft.com/en-us/library/47289\(v=VS.88\).aspx#Complex](http://msdn.microsoft.com/en-us/library/47289(v=VS.88).aspx#Complex) (under rubriken Arrays)
- %-operatör
 - Essential C# 5.0, 87-89.
 - <http://msdn.microsoft.com/en-us/library/0w4e0fzs.aspx>
- "if"-satsen
 - Essential C# 5.0, 107-114.
 - <http://msdn.microsoft.com/en-us/library/5011f09h.aspx>
- "while"-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/2aeyhxcd.aspx>

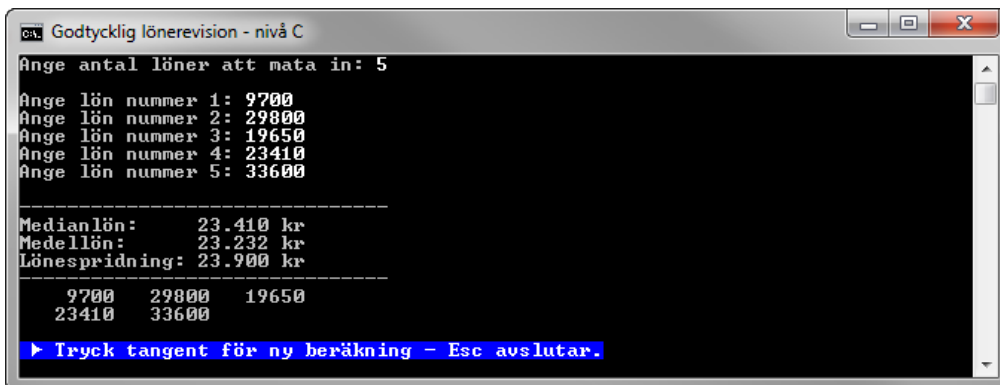
- ”do-while”-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370s1zax.aspx>
- ”for”-satsen
 - Essential C# 5.0, 130-133.
 - <http://msdn.microsoft.com/en-us/library/ttw7t8t6.aspx>
- Metoder
 - Essential C# 5.0, 155-167.
 - <http://msdn.microsoft.com/en-us/library/ms173114.aspx>
- Undantag
 - Essential C# 5.0, 194-201.
 - Essential C# 5.0, 423-430.
 - <http://msdn.microsoft.com/en-us/library/0yd65esw.aspx>
 - <http://msdn.microsoft.com/en-us/library/1ah5wsex.aspx>
 - <http://msdn.microsoft.com/en-us/library/ww58ded5.aspx>
- Aggregatfunktioner
 - Essential C# 5.0, 609.
- Hantering av färger i ett konsolfönster
 - <http://msdn.microsoft.com/en-us/library/yae1s0f9.aspx>
 - <http://msdn.microsoft.com/en-us/library/s66hf68a.aspx>
 - <http://msdn.microsoft.com/en-us/library/d3zkyxe.aspx>

C.Uppgift

Då en del fackförbund informerar om utfallet av löneförhandlingar redovisar de medianlönen (den mittersta lönen), genomsnittslönen och lönespridningen (skillnaden mellan den högsta och lägsta lönen). Skriv ett program som kan läsa in ett godtyckligt antal löner och sedan beräkna medianlön, medellön och lönespridning.

Programmet ska delas upp i två klasser, Program och MyExtensions, innehållande statistiska metoder. Klassen MyExtensions ska innehålla så kallade utökningsmetoder (*"extension methods"*) för beräkning av median och spridning, för vilka det saknas aggregatfunktioner.

Efter att programmet presenterat de beräknade värdena, ska lönerna presenteras i den ordning de matats in i med tre löner per rad.



```

Godtycklig lönerrevision - nivå C
Ange antal löner att mata in: 5
Ange lön nummer 1: 9700
Ange lön nummer 2: 29800
Ange lön nummer 3: 19650
Ange lön nummer 4: 23410
Ange lön nummer 5: 33600

-----
Medianlön:      23.410 kr
Medellön:       23.232 kr
Lönespridning:  23.900 kr
-----

  9700    29800    19650
 23410    33600

► Tryck tangent för ny beräkning - Esc avslutar.
  
```

Figur C.1. Exempel på exekvering av programmet.

Eventuella fel i samband med inmatning ska hanteras. Vid fel ska användaren få en ny chans att mata in ett korrekt värde.

Då inmatning, beräkning och presentation är gjord ska användaren kunna välja att avsluta programmet genom att trycka på Escape-tangenten. Trycker användaren ner någon annan tangent ska användaren på nytt kunna ange antalet löner att mata in.

Beräkningar

Medellön och lönespridning

Medellönen och lönespridningen kan beräknas på traditionellt sätt genom att iterera igenom arrayen med löner. Med hjälp av aggregatfunktionerna `Average()`, `Max()` och `Min()` är det dock enklare att beräkna medelvärdet, det högsta och lägsta värdet för en array innehållande numeriska värden.

Medianlön

Medianlönen är den mittersta lönen och för att kunna bestämma den måste lönerna sorteras. Enklarest att sortera en array är att använda den statistiska metoden `Sort()`, vilket dock kräver att en kopia av array med löner först skapas för att behålla lönerna i den ordning de matades in i.

Det kan vara mer intressant att använda `OrderBy()` tillsammans med `ToArray()`, vilka är så kallade *"standard query operators"*, eftersom då skapas en kopia innehållande de sorterade lönerna automatiskt.

Är det ett udda antal löner bestäms medianlönen av den mittersta lönen. Vid jämt antal löner så bestäms medianlönen genom att medelvärdet av de två mittersta lönerna beräknas.

18600	21900	23800	26400	29700	32100
		↑	↑		
		$(23800 + 26400) / 2 = 25100$			
			↑		
			medianlön		

Exempel C.2. Median vid jämt antal löner.

Klassen Program

Klassen Program ska innehålla sex privata statiska metoder enligt klassdiagrammet i Figur C.2.

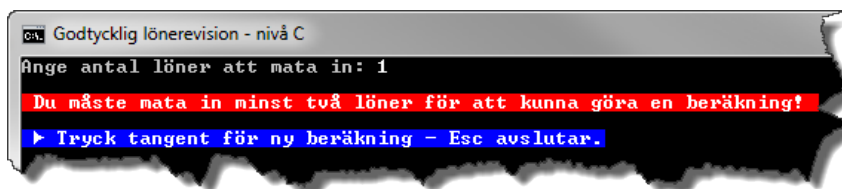


Figur C.2. Klassdiagram över klassen Program.

Main

Metoden ska anropa metoderna `ReadInt()`, `ReadSalaries()`, `ViewResult()` och `IsContinuing()`. Anropen ska placeras i en "do-while"-sats som avslutas då metoden `IsContinuing()` returnerar `false` vilket ska ske då användaren trycker på Escape-tangenten.

Metoden `ReadInt()` anropas för att ta reda på antalet löner som användaren vill mata in. Är antalet löner att bearbeta färre än två ska ett felmeddelande presenteras enligt Figur C.3.



Figur C.3. Felmeddelande då antalet löner att mata in är för få.

Om antalet löner att mata in är minst två ska metoden `ReadSalaries()` anropas och antalet löner att läsa in skickas med som ett argument och en array med lönerna returneras.

De inlästa lönerna skickas med som argument till metoden `ViewResult`, som ser till att beräkningar och presentation av resultat sker.

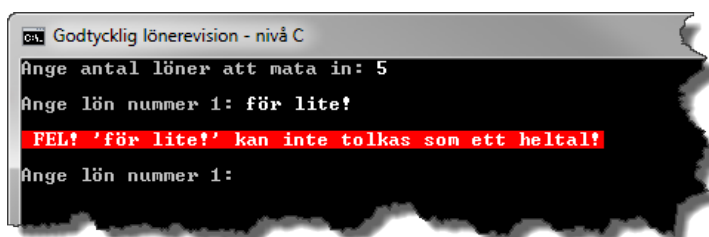
IsContinuing

Uppgiften `IsContinuing()` har är att presentera ett meddelande som uppmanar användaren att trycka på en tangent för att fortsätta, eller Escape-tangenten för att avsluta programmet. Trycker användaren på Escape-tangenten ska metoden returnera `false`; vilken annan tangent som helst ska leda till att `true` returneras.

ReadInt

Metoden `ReadInt()` ska returnera ett värde av typen `int`.

Metoden ska användas för att läsa in så väl antalet löner som en lön. Till metoden ska det därför vara möjligt att skicka med ett argument. Argumentet ska vara en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. I Figur C.4 har argumentet "Ange antal löner att mata in: " respektive argumentet "Ange lön nummer 1: " används då metoden anropas.



Figur C.4 Metoden `ReadInt()` har anropats med två olika argument. Användaren får även en ny chans att mata in ett nytt värde efter att ett fel konstaterats.

Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning efter att ett tydligt felmeddelande presenterats.

ReadSalaries

Metodens uppgift är att läsa in lönerna, genom anrop av metoden `ReadInt()`, och lagra dessa i en lokal array av typen `int[]`, som sedan returneras.

ViewMessage

Metoden anropas för att presentera meddelanden för användaren. Olika typer av meddelanden ska kunna presenteras, till exempel vanliga meddelanden och felmeddelanden. Första parametern är meddelandet, andra och tredje parametern bestämmer färgen på bakgrunden respektive texten.



Figur C.5. Meddelande formaterat som ett vanligt meddelande då andra och tredje parameterns värde är dess standardvärden.



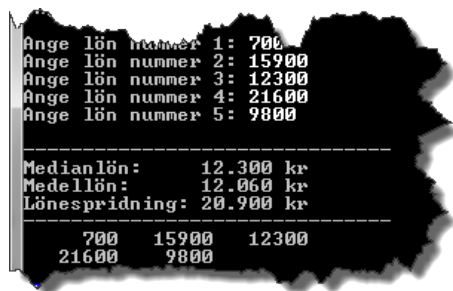
Figur C.6. Meddelande formaterat som felmeddelande då andra parameterns värde är `ConsoleColor.Red`.

ViewResult

Metodens parameter refererar till en array med inmatade löner. I samband med att medianlön, medellön och lönespridning presenteras på lämpligt sätt, formaterade som valuta, anropas metoderna som utför beräkningarna varefter de inmatade lönerna ska listas, med tre löner på varje rad, i den ordning de matades in.

Medianlönen fås genom ett anrop av utökningsmetoden `Median()`. Medellönen fås genom ett anrop av aggregatfunktionen `Average()`. Lönespridningen fås genom ett anrop av utökningsmetoden `Dispersion()`.

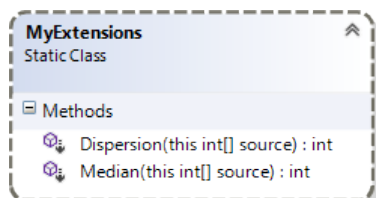
Då lönerna ska listas med tre löner per rad kan modulusoperatorn (%) komma till användning. Tänk även på att det är lämpligt att högerjustera lönerna så att ental, tiotal osv. hamnar under varandra.



Figur C.7. Inmatade löner presenterade högerjusterade tre per rad i den ordning de matats in.

Klassen `MyExtensions`

Den statiska klassen `MyExtensions` ska innehålla två publika statiska metoder enligt klassdiagrammet i Figur C.8.



Figur C.8. Klassdiagram över klassen `MyExtensions`.

Dispersion

Metoden ska beräkna lönespridningen på de löner som skickas som argument i form av en array till metoden.

Median

Metoden ska beräkna medianlönen av de löner som skickas som argument i form av en array till metoden.

C-krav

1. Indata till programmet ska vara antalet löner som ska matas in samt lönerna.
 - a) Antalet löner ska anges i form av ett heltal.
 - b) En lön ska anges i form av ett heltal.
2. Inläsning av antalet löner och lönerna i sig ska ske via en och samma metod som returnerar ett heltal.
3. Eventuella fel i samband med inläsning av ett värde ska tas om hand med hjälp av en "try-catch"-sats och användaren ska få en ny möjlighet att mata in ett korrekt värde efter att ett relevant felmeddelande presenterats för användaren.
4. Inmatade löner ska lagras i en array av typen `int[]`.
5. Medianlön och lönespridning ska beräknas i en separat klass av utökningsmetoder till vilka en array med löner skickas med som argument.
6. Median-, medellön och lönespridning ska presenteras formaterade som valuta utan decimaler.
7. Efter att beräkning och presentation av median- och medellön samt lönespridning gjorts ska lönerna presenteras i den ordning de matats in om tre löner på varje rad där ental, tiotal, osv. ska vara placerade ovanför varandra.
8. Flera beräkningar ska kunna göras efter varandra. Programmet ska avbrytas först då användaren trycker på Escape-tangenten.

Läsvärt

- variabler
 - Essential C# 5.0, 13-17.
 - [http://msdn.microsoft.com/en-us/library/55c427e2\(VS.88\).aspx#Variables](http://msdn.microsoft.com/en-us/library/55c427e2(VS.88).aspx#Variables)
- `String.Format`
 - Essential C# 5.0, 49.
 - [http://msdn.microsoft.com/en-us/library/55c427e2\(VS.88\).aspx#String.Format](http://msdn.microsoft.com/en-us/library/55c427e2(VS.88).aspx#String.Format)
- `Array`
 - Essential C# 5.0, 67-82 (inte flerdimensionella arrayer).
 - [http://msdn.microsoft.com/en-us/library/55c427e2\(VS.88\).aspx#Array](http://msdn.microsoft.com/en-us/library/55c427e2(VS.88).aspx#Array)
 - [http://msdn.microsoft.com/en-us/library/55c427e2\(VS.88\).aspx#Complex](http://msdn.microsoft.com/en-us/library/55c427e2(VS.88).aspx#Complex) (under rubriken Arrays)
- %-operatör
 - Essential C# 5.0, 87-89.
 - [http://msdn.microsoft.com/en-us/library/55c427e2\(VS.88\).aspx#Modulo](http://msdn.microsoft.com/en-us/library/55c427e2(VS.88).aspx#Modulo)

- ”if”-satsen
 - Essential C# 5.0, 107-114.
 - <http://msdn.microsoft.com/en-us/library/5011f09h.aspx>
- ”while”-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/2aeyhxcd.aspx>
- ”do-while”-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370s1zax.aspx>
- ”for”-satsen
 - Essential C# 4.0, 130-133.
 - <http://msdn.microsoft.com/en-us/library/ttw7t8t6.aspx>
- Metoder
 - Essential C# 5.0, 155-167.
 - <http://msdn.microsoft.com/en-us/library/ms173114.aspx>
- Undantag
 - Essential C# 5.0, 194-201.
 - Essential C# 5.0, 423-430.
 - <http://msdn.microsoft.com/en-us/library/0yd65esw.aspx>
 - <http://msdn.microsoft.com/en-us/library/1ah5wsex.aspx>
 - <http://msdn.microsoft.com/en-us/library/ww58ded5.aspx>
- OrderBy
 - Essential C# 5.0, 590-592.
- Aggregatfunktioner
 - Essential C# 5.0, 609.
- Hantering av färger i ett konsolfönster
 - <http://msdn.microsoft.com/en-us/library/yae1s0f9.aspx>
 - <http://msdn.microsoft.com/en-us/library/s66hf68a.aspx>
 - <http://msdn.microsoft.com/en-us/library/d3zkyxe.aspx>