



Linnéuniversitetet

Kalmar Vaxjö

Laborationsanvisning

Växelpengar

Steg 1, laborationsuppgift 1



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 Växelpengar 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
Öresavrundning	5
A-krav	5
Läsvärt	6
B. Uppgift	9
Problem	9
Öresavrundning	9
Uppdelning av programmet	9
B-krav	11
Läsvärt	12
C. Uppgift	15
Problem	15
Öresavrundning	15
Uppdelning av programmet	16
C-krav	17
Läsvärt	19

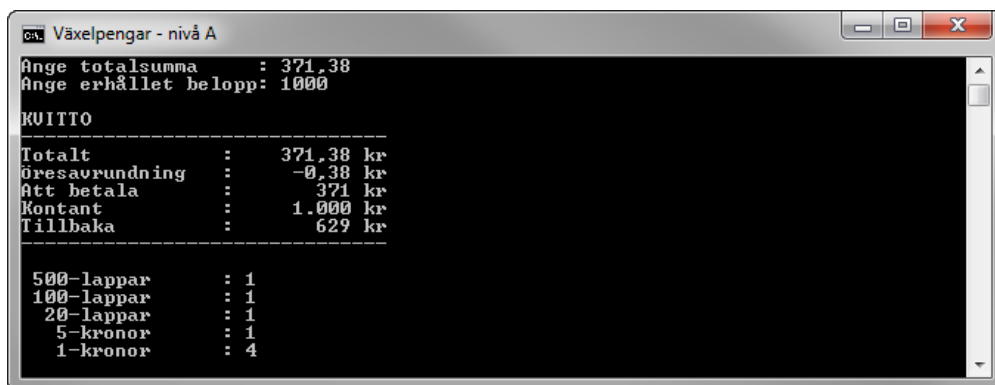
A. Uppgift

Problem

Då du handlar i en affär och betalar kontant får du kanske växel tillbaka. I regel får affärsbiträdet hjälp av kassaapparaten med att beräkna summan kunden ska få tillbaka. Vilka sedlar och mynt som ska lämnas tillbaka får affärsbiträdet själv räkna ut i huvudet. Kan ett program göra båda sakerna istället?

Skriv ett program som i samband med ett köp efterfrågar totalsumma och erhållet belopp, bestämmer öresavrundningen till närmsta hela krontal och beräknar den växel som ska lämnas tillbaka.

Eventuella fel i samband med inmatning av totalsumma och erhållet belopp ska hanteras. Det ska inte vara möjligt att ange en totalsumma mindre än en krona eller ett erhållet belopp mindre än totalsumman.



Figur A.1

Programmet ska, förutom att presentera beloppet kunden ska betala avrundat till närmsta hela krontal, även bestämma vilka, och antalet, sedlar och mynt som kunden ska få tillbaka så att så få sedlar och mynt används som möjligt.

Växel ska kunna ges tillbaka med sedlar av valörerna 500, 100, 50 och 20 samt mynten 10-, 5-, och 1-kronor. Du kan anta att det alltid finns tillräckligt antal av de sedlar och mynt som krävs. Skriv endast ut de sedlar och mynt som ska lämnas tillbaka.

Öresavrundning

För att avrunda totalsumman till närmsta hela krontal kan du t.ex. använda den statiska metoden `Round` i klassen `Math`. Genom att bestämma differensen mellan den avrundade totalsumman och totalsumman erhålls öresavrundningen:

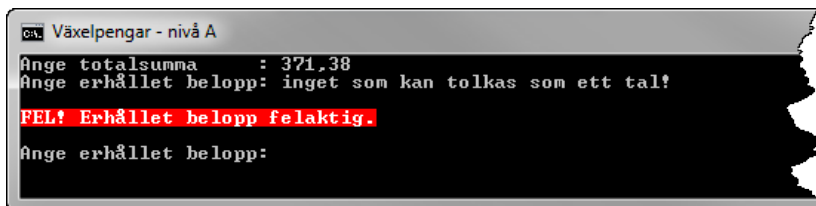
```
total = (uint)Math.Round(subtotal);
roundingOffAmount = total - subtotal;
```

Där `subtotal` och `roundingOffAmount` är variabler av typen `double` och `total` är en variabel av typen `uint`.

A-krav

1. Indata till programmet ska vara totalsumma respektive erhållet belopp.
 - a) Totalsumman ska kunna anges i kronor och ören.
 - b) Erhållet belopp ska enbart kunna anges i hela kronor.
2. Inmatad totalsumma avrundas till närmsta hela krontal.
3. Utdata från programmet ska vara ett kvitto innehållande:
 - a) Köpets totala summa.
 - b) Öresavrundningen.

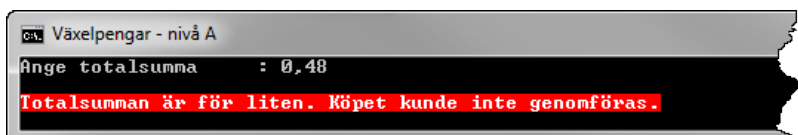
- c) Summa att betala efter öresavrundning.
 - d) Erhållet belopp (kontant).
 - e) Växel kunden ska ha tillbaka.
 - f) Växeln kunden ska ha tillbaka ska även delas upp i lämpligt antal 500-, 100, 50- och 20-lappar samt antal 10-, 5- och 1-kronor och bara presenteras om antalet sedlar och/eller mynt är större än 0.
- 4. Antalet sedlar och mynt ska bestämmas med hjälp av division- och modulusoperatorerna.
 - 5. Samlingar som t.ex. arrayer får inte användas på något sätt.
 - 6. Eventuella fel i samband med inmatningen ska tas om hand med hjälp av "try-catch"-satser och användaren ska få en ny möjlighet att mata in.



```
CA> Växelpengar - nivå A
Ange totalsumma      : 371,38
Ange erhållet belopp: inget som kan tolkas som ett tal!
FEL! Erhållet belopp felaktig.
Ange erhållet belopp:
```

Figur A.2.

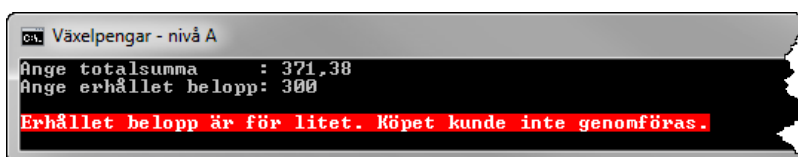
- 7. Om den inmatade totalsumman efter avrundning motsvarar ett belopp mindre än en krona är det att betrakta som ett fel varför programmet ska avslutas efter att ett felmeddelande presenterats.



```
CA> Växelpengar - nivå A
Ange totalsumma      : 0,48
Totalsumman är för liten. Köpet kunde inte genomföras.
```

Figur A.3.

- 8. Om beloppet att betala efter öresavrundning är större än det erhållna beloppet är det att betrakta som ett fel varför programmet ska avslutas efter att ett felmeddelande presenterats.



```
CA> Växelpengar - nivå A
Ange totalsumma      : 371,38
Ange erhållet belopp: 300
Erhållet belopp är för litet. Köpet kunde inte genomföras.
```

Figur A.4

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)
- %-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>

- ”do-while”-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370s1zax.aspx>
- “try-catch”-satsen
 - Essential C# 5.0, 194-201.
 - <http://msdn.microsoft.com/en-us/library/0yd65esw.aspx>
- 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/d3zkyxxe.aspx>

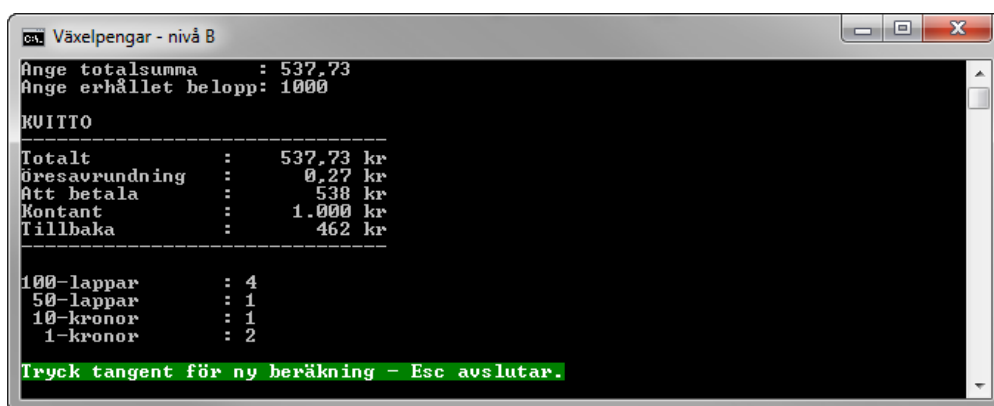
B. Uppgift

Problem

Då du handlar i en affär och betalar kontant får du kanske växel tillbaka. I regel får affärsbiträdet hjälp av kassaapparaten med att beräkna summan kunden ska få tillbaka. Vilka sedlar och mynt som ska lämnas tillbaka får affärsbiträdet själv räkna ut i huvudet. Kan ett program göra båda sakerna istället?

Skriv ett program som i samband med ett köp efterfrågar totalsumma och erhållet belopp, bestämmer öresavrundningen till närmsta hela krontal och beräknar den växel som ska lämnas tillbaka.

Eventuella fel i samband med inmatning av totalsumma och erhållet belopp ska hanteras. Det ska inte vara möjligt att ange en totalsumma mindre än en krona eller ett erhållet belopp mindre än totalsumman. Gör användaren en felaktig inmatning ska användaren erbjudas en ny möjlighet att mata in ett korrekt värde.



Figur B.1

Programmet ska, förutom att presentera beloppet kunden ska betala avrundat till närmsta hela krontal, även bestämma vilka, och antalet, sedlar och mynt som kunden ska få tillbaka så att så få sedlar och mynt används som möjligt.

Växel ska kunna ges tillbaka med sedlar av valörerna 500, 100, 50 och 20 samt mynten 10-, 5-, och 1-kronor. Du kan anta att det alltid finns tillräckligt antal av de sedlar och mynt som krävs. Skriv endast ut de sedlar och mynt som ska lämnas tillbaka.

Då en beräkning 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 mata in en ny totalsumma och nytt erhållet belopp.

Öresavrundning

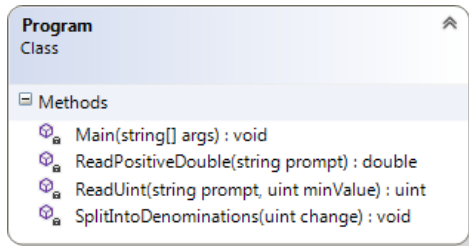
För att avrunda totalsumman till närmsta hela krontal kan du t.ex. använda den statiska metoden `Round` i klassen `Math`. Genom att bestämma differensen mellan den avrundade totalsumman och totalsumman erhålls öresavrundningen:

```
total = (uint)Math.Round(subtotal);  
roundingOffAmount = total - subtotal;
```

Där `subtotal` och `roundingOffAmount` är variabler av typen `double` och `total` är en variabel av typen `uint`.

Uppdelning av programmet

Programmet ska delas upp i de fyra privata metoder som återfinns i klassdiagrammet i Figur B.2.



Figur B.2

Main

Denna metod ska anropa metoderna `ReadPositiveDouble` och `ReadUInt` för att läsa in totalsumman respektive erhållet belopp. Efter att ha beräknat belopp att betala, öresavrundningen, växeln tillbaka och skrivit ut ett kvitto ska metoden `SplitIntoDenominations` anropas.

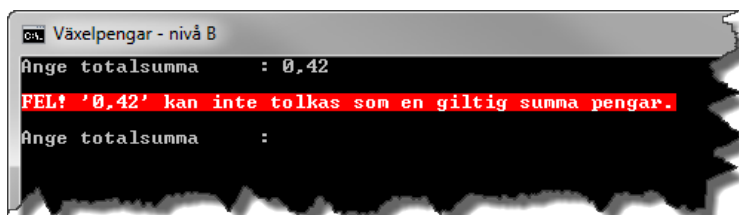
Satserna ska placeras i en "do-while"-sats som avslutas då användaren trycker ner Escape-tangenten.

ReadPositiveDouble

Metoden ska returnera ett värde av typen `double`.

Innan värdet returneras ska metoden säkerställa att användaren matat in ett värde som, efter avrundning, är större eller lika med 1. Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning.

Till metoden ska det vara möjligt att skicka en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. I Figur B.3 har argumentet `"Ange totalsumma : "` skickats med vid anropet av metoden.



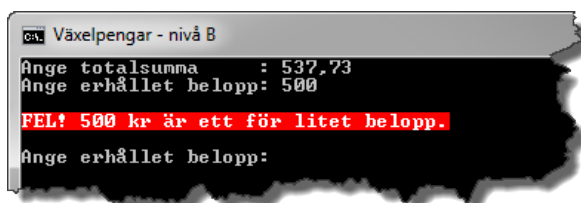
Figur B.3

ReadUInt

Metoden ska returnera ett värde av typen `uint`. (Datatypen `uint` passar i detta fall då endast hela kronor motsvarande ett värde större än 0 ska hanteras.)

Innan värdet returneras ska metoden säkerställa att användaren matat in ett värde som är större eller lika med angivet minsta värde. Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning.

Till metoden ska det vara möjligt att skicka med två argument. Det första argument ska vara en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. Det andra argumentet är det minsta värdet som är giltigt. I Figur B.4 har argumenten `"Ange erhållet belopp: "` och 538 skickats med vid anropet av metoden.



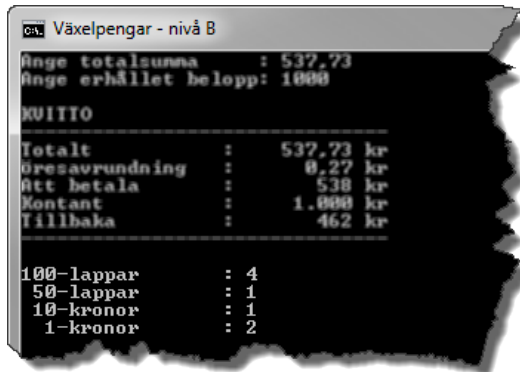
Figur B.4

SplitIntoDenominations

Metoden ska dela upp växeln och presentera vilka valörer som ska lämnas tillbaka.

Tillgängliga valörer måste lagras i en lokal array av typen `uint`, och metoden får endast använda sig av en `"if"`-sats, utan några `"else"`.

Till metoden ska det vara möjligt att skicka summan som ska lämnas tillbaka. I Figur B.5 har argumentet 462 skickats med vid anropet av metoden



Figur B.5

B-krav

1. Indata till programmet ska vara totalsumma respektive erhållet belopp.
 - a) Totalsumman ska kunna anges i kronor och ören.
 - b) Erhållet belopp ska enbart kunna anges i hela kronor.
2. Inläsning av totalsumma måste placeras i en separat statisk metod.
3. Inmatad totalsumma avrundas till närmsta hela krontal.
4. Inläsning av erhållet belopp måste placeras i en separat statisk metod.
5. Metoden `Main` ska skriva ut ett kvitto innehållande:
 - a) Köpets totala summa.
 - b) Öresavrundningen.
 - c) Summa att betala efter öresavrundning.
 - d) Erhållet belopp (kontant).
 - e) Växeln kunden ska ha tillbaka.
6. Metoden `SplitIntoDenominations` ska dela upp, och skriva ut, växeln i lämpligt antal 500-, 100, 50- och 20-lappar samt antal 10-, 5- och 1-kronor och bara presenteras om antalet sedlar och/eller mynt är större än 0.
7. Då antalet sedlar och mynt ska bestämmas måste följande uppfyllas:
 - a) Kod för beräkning och presentation måste placeras i en separat statisk metod.
 - b) Division- och modulusoperatorerna måste användas.
 - c) En array innehållande giltiga valörer måste användas.
 - d) Endast en `"if"`-sats får användas.
8. Eventuella fel i samband med inmatning av totalsumma och erhållet belopp ska tas om hand med hjälp av `"try-catch"`-satser och användaren ska få en möjlighet att mata in ett nytt värde.

Figureerna Figur B.6 till Figur B.9 visar exempel på några felaktiga inmatningar som måste hanteras och ge användaren en ny möjlighet att mata in ett korrekt värde.

```

C#. Växelpengar - nivå B
Ange totalsumma      : -200
FEL! '-200' kan inte tolkas som en giltig summa pengar.
Ange totalsumma      :
  
```

Figur B.6. Resultat av en inmatning av totalsumma mindre än 1 kr.

```

C#. Växelpengar - nivå B
Ange totalsumma      : ogiltigt värde
FEL! 'ogiltigt värde' kan inte tolkas som en giltig summa pengar.
Ange totalsumma      :
  
```

Figur B.7. Resultat av en inmatning som inte kan tolkas som en summa pengar.

```

C#. Växelpengar - nivå B
Ange totalsumma      : 537.73
Ange erhållet belopp: 537
FEL! 537 kr är ett för litet belopp.
Ange erhållet belopp:
  
```

Figur B.8. Resultat av en inmatning där det erhållna beloppet är mindre beloppet att betala efter öresavrundning.

```

C#. Växelpengar - nivå B
Ange totalsumma      : 537.73
Ange erhållet belopp: ogiltigt värde
FEL! 'ogiltigt värde' kan inte tolkas som en summa pengar.
Ange erhållet belopp:
  
```

Figur B.9. Resultat av en inmatning där det erhållna beloppet inte kan tolkas som en summa pengar.

Läsvärt

- variabler
 - Essential C# 5.0, 13-17.
 - [http://msdn.microsoft.com/en-us/library/55147285\(VS.88\).aspx#Variables](http://msdn.microsoft.com/en-us/library/55147285(VS.88).aspx#Variables)
- 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/55127989\(v=VS.88\).aspx#Complex](http://msdn.microsoft.com/en-us/library/55127989(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>

- ”do-while”-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370s1zax.aspx>
- ”foreach”-satsen
 - Essential C# 5.0, 133-135.
 - <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>
- 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>

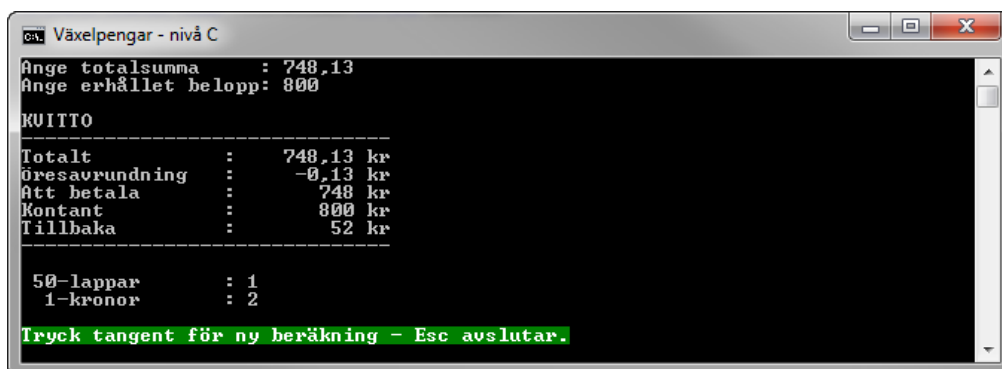
C. Uppgift

Problem

Då du handlar i en affär och betalar kontant får du kanske växel tillbaka. I regel får affärsbiträdet hjälp av kassaapparaten med att beräkna summan kunden ska få tillbaka. Vilka sedlar och mynt som ska lämnas tillbaka får affärsbiträdet själv räkna ut i huvudet. Kan ett program göra båda sakerna istället?

Skriv ett program som i samband med ett köp efterfrågar totalsumma och erhållet belopp, bestämmer öresavrundningen till närmsta hela krontal och beräknar den växel som ska lämnas tillbaka.

Eventuella fel i samband med inmatning av totalsumma och erhållet belopp ska hanteras. Det ska inte vara möjligt att ange en totalsumma mindre än en krona eller ett erhållet belopp mindre än totalsumman. Gör användaren en felaktig inmatning ska användaren erbjudas en ny möjlighet att mata in ett korrekt värde.



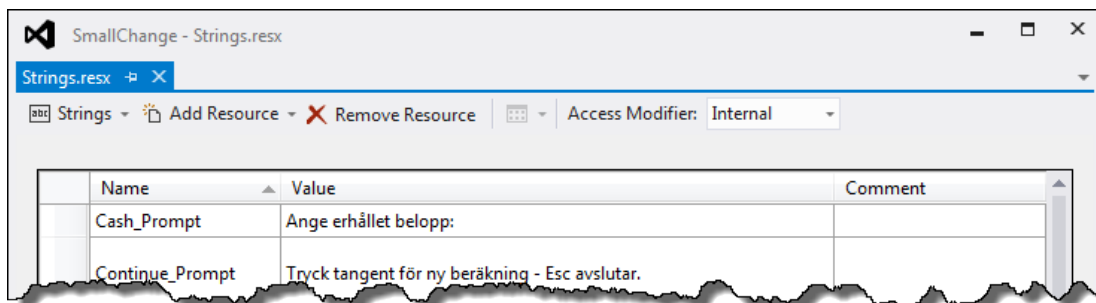
Figur C.1

Programmet ska, förutom att presentera beloppet kunden ska betala avrundat till närmsta hela krontal, även bestämma vilka, och antalet, sedlar och mynt som kunden ska få tillbaka så att så få sedlar och mynt används som möjligt.

Växel ska kunna ges tillbaka med sedlar av valörerna 500, 100, 50 och 20 samt mynten 10-, 5-, och 1-kronor. Du kan anta att det alltid finns tillräckligt antal av de sedlar och mynt som krävs. Skriv endast ut de sedlar och mynt som ska lämnas tillbaka.

Då en beräkning 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 mata in en ny totalsumma och nytt erhållet belopp.

Textmeddelanden till användaren ska hanteras med hjälp av en resursfil. Figur C.2 är ett exempel på en resursfil med strängar.



Figur C.2

Öresavrundning

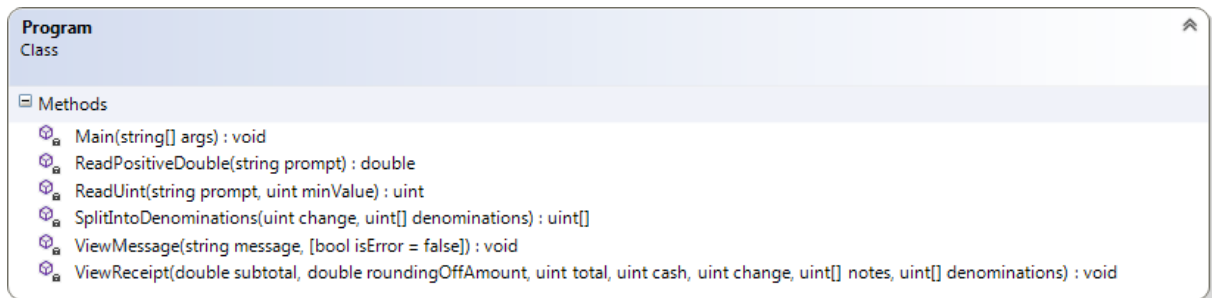
För att avrunda totalsumman till närmsta hela krontal kan du t.ex. använda den statiska metoden `Round` i klassen `Math`. Genom att bestämma differensen mellan den avrundade totalsumman och totalsumman erhålls öresavrundningen:

```
total = (uint)Math.Round(subtotal);  
roundingOffAmount = total - subtotal;
```

Där `subtotal` och `roundingOffAmount` är variabler av typen `double` och `total` är en variabel av typen `uint`.

Uppdelning av programmet

Programmet ska delas upp i de sex privata metoder som återfinns i klassdiagrammet i Figur C.3.



Figur C.3

Main

Denna metod ska anropa metoderna `ReadPositiveDouble` och `ReadUInt` för att läsa in totalsumman respektive erhållet belopp. Efter att ha beräknat belopp att betala, öresavrundningen, växeln tillbaka ska metoden `SplitIntoDenominations` anropas som returnerar en array med antalet av respektive valör som ska lämnas tillbaka.

Satserna ska placeras i en "do-while"-sats som avslutas då användaren trycker ner Escape-tangenten.

ReadPositiveDouble

Metoden ska returnera ett värde av typen `double`.

Innan värdet returneras ska metoden säkerställa att användaren matat in ett värde som, efter avrundning, är större eller lika med 1. Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning.

Till metoden ska det vara möjligt att skicka en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. I Figur C.4 har argumentet `"Ange totalsumma : "` skickats med vid anropet av metoden.



Figur C.4

ReadUInt

Metoden ska returnera ett värde av typen `uint`. (Datatypen `uint` passar i detta fall då endast hela kronor motsvarande ett värde större än 0 ska hanteras.)

Innan värdet returneras ska metoden säkerställa att användaren matat in ett värde som är större eller lika med angivet minsta värde (totalsumman). Om det inmatade inte kan tolkas som ett korrekt värde ska användaren få en chans att göra en ny inmatning.

Till metoden ska det vara möjligt att skicka med två argument. Det första argument ska vara en sträng med information som ska visas i anslutning till där inmatningen av värdet sker. Det andra argumentet är det minsta värdet som är giltigt. I Figur C.5 har argumenten `"Ange erhållet belopp: "` och 748 skickats med vid anropet av metoden.


```
cs Væxelpengar - nivå C
Ange totalsumma      : 748,13
Ange erhållet belopp: 700
FEL! 700 är ett för litet belopp.
Ange erhållet belopp:
```

Figur C.5

SplitIntoDenominations

Metoden ska dela upp växeln och returnera en array innehållande antalet av respektive valör som ska lämnas tillbaka.

Till metoden ska det vara möjligt att skicka summan som ska delas upp och tillgängliga valörer i form av en array av typen `uint`.

ViewMessage

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

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

Figur C.6. Meddelande formaterat som ett vanligt meddelande då andra parametrarnas (`isError`) värde är `false` (dess standardvärde).

```
FEL! 537 kr är ett för litet belopp.
```

Figur C.7. Meddelande formaterat som felmeddelande då andra parametrarnas (`isError`) värde är `true`.

ViewReceipt

Metoden ansvarar för att ett kvitto enligt Figur C.8 presenteras. Inga beräkningar får ske i metoden utan allt data som ska presenteras skickas som argument till metoden. Enbart de valörer som ska lämnas tillbaka till kunden ska presenteras.

```
KVITTO
-----
Totalt      : 748,13 kr
Öresavrundning : -0,13 kr
Att betala   : 748 kr
Kontant      : 800 kr
Tillbaka     : 52 kr
-----
50-lappar   : 1
1-kronor    : 2
```

Figur C.8

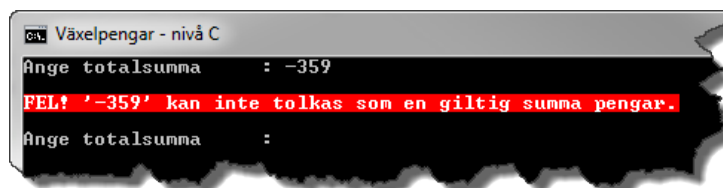
C-krav

1. Indata till programmet ska vara totalsumma respektive erhållet belopp.
 - a) Totalsumman ska kunna anges i kronor och ören.
 - b) Erhållet belopp ska enbart kunna anges i hela kronor.
2. Inläsning av totalsumma måste placeras i en separat statisk metod.
3. Inmatad totalsumma avrundas till närmsta hela krontal.
4. Inläsning av erhållet belopp måste placeras i en separat statisk metod.

5. Metoden `SplitIntoDenominations` ska dela upp växeln i lämpligt antal 500-, 100, 50- och 20-lappar samt antal 10-, 5- och 1-kronor. Följande måste vara uppfyllt:
 - a) En array innehållande giltiga valörer måste skickas som argument till metoden.
 - b) Division- och modulusoperatorerna måste användas.
 - c) Endast en `if`-sats får användas.
 - d) Antalet sedlar och mynt måste returneras i form av en array.
6. Metoden `ViewReciept` ska skriva ut ett kvitto innehållande:
 - a) Köpets totala summa.
 - b) Öresavrundningen.
 - c) Summa att betala efter öresavrundning.
 - d) Erhållet belopp (kontant).
 - e) Växeln kunden ska ha tillbaka.
 - f) Antal sedlar eller mynt ska bara presenteras om antalet sedlar eller mynt är större än 0.
7. Vanliga meddelanden till användaren ska presenteras genom att anropa metoden `ViewMessage`, som ska presentera ett formaterat meddelande.

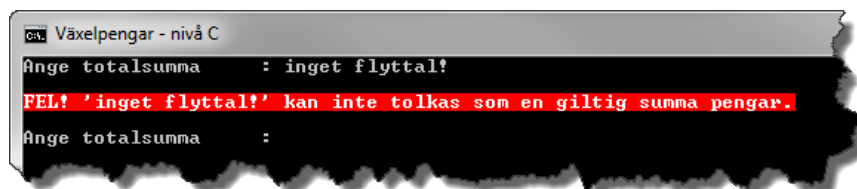
Metoden `ViewMessage` ska ha två parametrar. Första parametern ska vara av typen `string` och innehålla själva meddelandet. Andra parametern ska vara av typen `bool` och är dess värde `true` ska meddelandet formateras som ett felmeddelande; är den `false` ska meddelandet formateras som ett vanligt meddelande. Andra parametrarnas värde ska ha värdet `false` som standardvärde.
8. Eventuella fel i samband med inmatning av totalsumma och erhållet belopp ska tas om hand med hjälp av `try-catch`-satser och användaren ska få en möjlighet att mata in ett nytt värde. Användaren ska informeras om felet genom att metoden `ViewMessage` anropas som ska presentera ett formaterat felmeddelande.

Figuren Figur C.9 till Figur C.12 visar exempel på några felaktiga inmatningar som måste hanteras och ge användaren en ny möjlighet att mata in ett korrekt värde.



```
CA> Växelpengar - nivå C
Ange totalsumma : -359
FEL! '-359' kan inte tolkas som en giltig summa pengar.
Ange totalsumma :
```

Figur C.9. Resultat av en inmatning av totalsumma mindre än 1 kr.



```
CA> Växelpengar - nivå C
Ange totalsumma : inget flyttal!
FEL! 'inget flyttal!' kan inte tolkas som en giltig summa pengar.
Ange totalsumma :
```

Figur C.10. Resultat av en inmatning som inte kan tolkas som en summa pengar.

```
cs. Væxelpengar - nivå C
Ange totalsumma      : 748,13
Ange erhållet belopp: 700
FEL! 700 är ett för litet belopp.
Ange erhållet belopp:
```

Figur C.11. Resultat av en inmatning där det erhållna beloppet är mindre beloppet att betala efter öresavrundning.

```
cs. Væxelpengar - nivå C
Ange totalsumma      : 748,13
Ange erhållet belopp: inget positivt heltal!
FEL! 'inget positivt heltal!' kan inte tolkas som en giltig summa pengar.
Ange erhållet belopp:
```

Figur C.12. Resultat av en inmatning där det erhållna beloppet inte kan tolkas som en summa pengar.

Läsvärt

- variabler
 - Essential C# 5.0, 13-17.
 - [http://msdn.microsoft.com/en-us/library/447285\(VS.88\).aspx#Variables](http://msdn.microsoft.com/en-us/library/447285(VS.88).aspx#Variables)
- 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/447989\(v=VS.88\).aspx#Complex](http://msdn.microsoft.com/en-us/library/447989(v=VS.88).aspx#Complex) (under rubriken Arrays)
- %-operatörn
 - 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>
- "do-while"-satsen
 - Essential C# 5.0, 127-129.
 - <http://msdn.microsoft.com/en-us/library/370slzax.aspx>
- "foreach"-satsen
 - Essential C# 5.0, 133-135.
 - <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>
- Metoder med frivilliga argument
 - Essential C# 5.0, 189-193.

- <http://msdn.microsoft.com/en-us/library/dd264739.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>
- 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>