

# Examinerande uppgift

# Loggboken

Rapport

**Dokumentation för uppgiften "Loggboken"**

**Programmering 1 C#**

**NTI-Skolan**

**Johan Kämpe**

<https://www.linkedin.com/in/johankampe>

<https://github.com/GoblinDynamiteer>

**2017-07-17**

## Innehåll

<b>1 Inledning</b>	3
1.1 Syfte	3
1.2 Noteringar	3
1.3 Länkar	3
<b>2 Genomförande och resultat</b>	4
2.1 Använd programvara och litteratur	4
2.2 Avgränsningar och krav	4
2.2.1 Grundläggande funktionalitet	4
2.2.2 Utökad funktionalitet	4
2.3 Metod	5
2.4 Programmets funktion	5
2.5 Flödesschema	9
2.6 Pseudokod	10
2.7 Skillnader från uppgift 3	11
2.8 Programmets kod	11
2.8.1 Klassen Logbook	11
2.8.2 Klassen Menu	12
<b>3 Diskussion och slutsats</b>	13
3.1 Förslag på förbättringar	13
3.2 Val av algoritmer, variabler och datastrukturer	15
3.3 Utvärdering av insats	16

## 1 Inledning

### 1.1 Syfte

Syftet med uppgiften är att skapa en loggbok i programspråket C# där en användare kan skriva och hantera loggboksinlägg. Uppgiften används som examineringsunderlag i kursen Programmering 1 C#.

Som underlag för programmet används pseudokod som producerades in en tidigare inlämningsuppgift i kursen.

### 1.2 Noteringar

Skärmsklipp i rapporten visar ibland en äldre version av programmet, vissa skillnader kan förekomma i utseende och text i det inlämnade programmet.

ID/ID-nummer som nämns i rapporten innebär det ID-nummer som ett loggboksinlägg automatiskt får när det skapas.

Testinlägg kan genereras vid körning av programmet om så önskas, detta utförs genom att ändra skapandet av objektet logbook till

```
Logbook logbook = new Logbook(true);
```

från

```
Logbook logbook = new Logbook();
```

### 1.3 Länkar

Uppgiften på GitHub

[https://github.com/GoblinDynamiteer/P1CS/tree/master/exercise07\\_logbook](https://github.com/GoblinDynamiteer/P1CS/tree/master/exercise07_logbook)

Flödesdiagram på draw.io

<https://www.draw.io/?lightbox=1&highlight=0000ff&edit=blank&layers=1&nav=1&title=loggboken.html#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D0B0Da8QGBLf-0LXdwaFd0dVBpVnM%26export%3Ddownload>

## 2 Genomförande och resultat

### 2.1 Använd programvara och litteratur

#### **Programvara**

- Microsoft Visual Studio Community 2017 v4.7.02045
- Microsoft Word 2016 v1701
- Draw.io: <https://www.draw.io>

#### **Litteratur**

- Trangius, K *Programmering 1 C# Lärobok* (ISBN 978-91-7379-171-7)

### 2.2 Avgränsningar och krav

#### 2.2.1 Grundläggande funktionalitet

Enligt uppgiftens dokumentation ska följande grundläggande funktionalitet för programmet uppnås:

- Programmet ska ha en fungerande meny med val för programmets funktionalitet, använd switch-villkor ska användas.
- Det ska finnas åtgärder för att förhindra körtidsfel.
- Det ska finnas möjlighet att skriva ut innehållet i loggboken.
- Det ska finnas möjlighet att spara nya inlägg i loggboken, med åtminstone titel och meddelande. Loggboken ska vara en lista medans varje enskild logg ska vara en vektor (array).
- Det ska finnas möjlighet att söka inlägg i loggboken, på exempelvis postens titel. Använd pseudokod eller flödesschema för att planera logiken för sökningen. Utgå från en linjär sökning i kursboken.

Källkoden ska vara väl kommenterad. Val av datatyper, kontrollinstruktioner, metoder, datastrukturer och algoritmer ska motiveras.

Programmet ska köras i ett konsolfönster, i en Windows-miljö.

#### 2.2.2 Utökad funktionalitet

Enligt uppgiftens dokumentation kan programmer utökas med följande funktionalitet:

- Möjlighet att spara datum för loggboks-inlägg.
- Möjlighet att redigera eller ta bort loggboks-inlägg.
- Skapa en eller flera metoder för programmet.
- Implementera algoritmer för sökning och sortering av loggboks-inlägg.

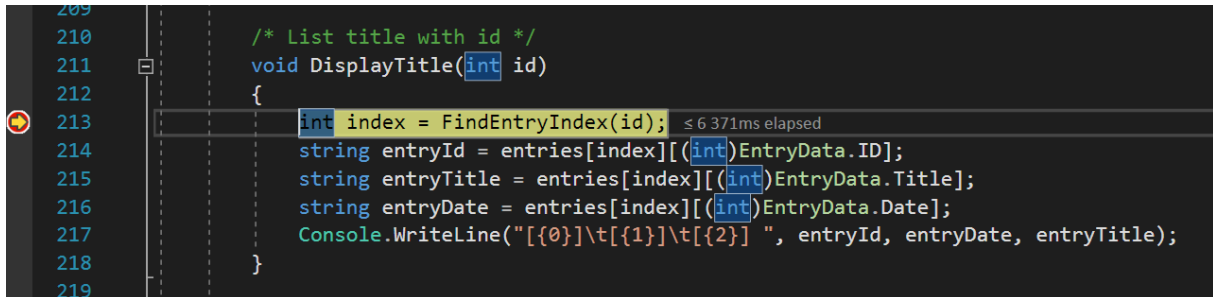
Programmets funktion, och förslag på förbättringar ska utvärderas i rapporten.

## 2.3 Metod

Programmets källkod skrevs uteslutande i utvecklingsmiljön *Visual Studio*. Som projektmall användes *Console App (.NET Framework)*.

I Visual Studio finns ett hjälpverktyg kallat *IntelliSense*, som underlättar vid kodskrivning.

Visual Studio har också stöd för debugging med breakpoints, vilket användes för att felsöka programmet.

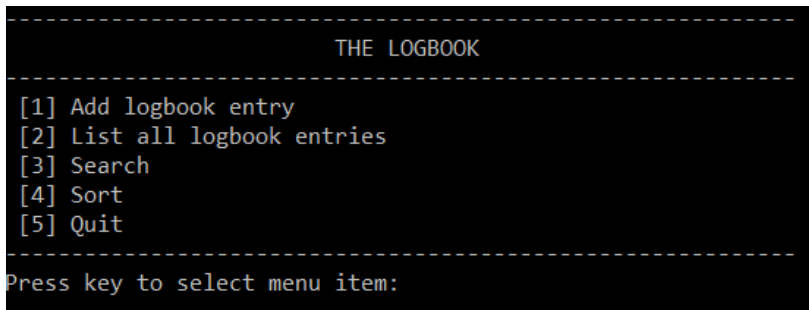


Figur 1 Kod i Visual Studio, debugläge med break point

Projektmallen skapar en första källkodsfil kallad *Program.cs* för projektet. Detta var den enda källkodsfil som användes för programmet.

## 2.4 Programmets funktion

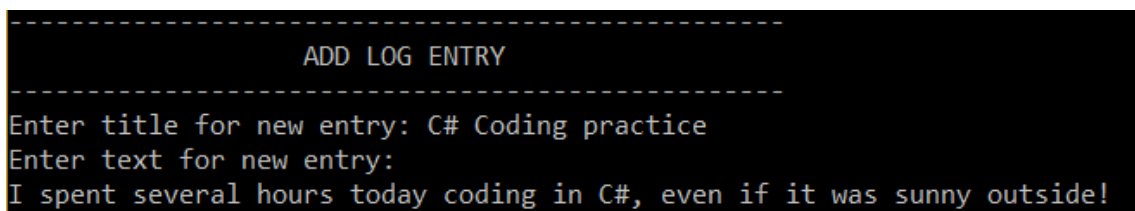
Vid programmets start presenteras en huvudmeny för användaren. Menyval utförs genom att trycka på en siffra på tangentbordet. I huvudmenyn behövs inget returslag efter sifferinmatning.



Figur 2 Programmets huvudmeny

Programmet kan användas som en loggbok eller dagbok. Användaren har möjlighet att skapa, visa, radera och ändra inlägg.

När ett inlägg skapas med menyalternativ "[1] Add logbook entry" uppmanas användaren att skriva in en titel och sedan texten (innehållet) för inlägget.



Figur 3 Skapande av ett nytt logginlägg

Ett inlägg kan skapas utan att skriva in en titel, det får då automatiskt titeln "Untitled entry". Ett inlägg kan dock inte ha tom text, om användaren inte skriver in någon text för inlägget visas ett felmeddelande som uppmanar användaren att försöka igen. Detta repeteras tills en text har matats in.

När ett inlägg skapas sätts också automatiskt ett ID-nummer och datum till inlägget.

Med menyalternativ "[2] List all logbook entries" listas alla logginlägg med titlar, datum och ID-nummer.

```

-----
                        ALL ENTRIES (6)
-----
ID      DATE                TITLE
[0]     [2017-07-16 13:25:12] [A day at the zoo!]
[4]     [2017-07-16 13:25:12] [C# Coding]
[2]     [2017-07-16 13:25:12] [New TV-Show]
[1]     [2017-07-16 13:25:12] [Today's lunch]
[5]     [2017-07-16 13:27:38] [Untitled entry]
[3]     [2017-07-16 13:25:12] [Weather Report]

Enter entry ID:
    
```

Figur 4 Lista med alla inlägg som finns i loggboken

Användaren ombeds skriva in ID-numret på det inlägg som ska visas. Om ett felaktigt ID-nummer matas in visas ett felmeddelande och programmet återgår till huvudmenyn.

Vid visning av ett inlägg kan användaren välja att redigera (ändra titel eller textinnehåll), radera, eller exportera inlägget till en textfil.

```

-----
                        A DAY AT THE ZOO!
                        2017-07-14 12:10:21
-----
I went to the zoo with my family today, we saw monkies and giraffes!!
-----
Options:
(E)xport | Change (T)itle | Change (C)ontent | (D)etele
Press any other key to return to main menu.
    
```

Figur 5 Visning att ett inlägg, med alternativ

Alternativen väljs genom att trycka på den bokstav på tangentbordet som visas inom parenteser, exempelvis tangent "T" för att välja alternativet "Change (T)itle".

Med menyalternativ "[3] Search" kan användaren skriva in en sträng för att söka i samtliga inlägg. Efter att sökningen har genomförts listas de inlägg där söksträngen förekommer. Likt menyalternativ 2 ombeds användaren också här att skriva in ett ID-nummer för att visa ett inlägg.

```

-----
                        SEARCH: FOUND 2 HITS FOR C#
-----
ID      DATE                TITLE
[4]     [2017-07-16 14:16:26] [C# Coding]
[5]     [2017-07-16 14:16:26] [C# Classes and Objects]

Enter entry ID: 4
    
```

Figur 6 Sökning av logginlägg

Notera att det går att skriva in ett ID-nummer för inlägg som inte visas i sök-listan, och detta ändå visas om det finns i loggboken.

Titlar, datum och inläggets text genomsöks för matchning.





Vid exportering av ett inlägg ombeds användaren att skriva in ett filnamn. Filen sparas till samma katalog där programmets exekverbara fil finns. Om filen inte går att exportera visas ett felmeddelande.

```

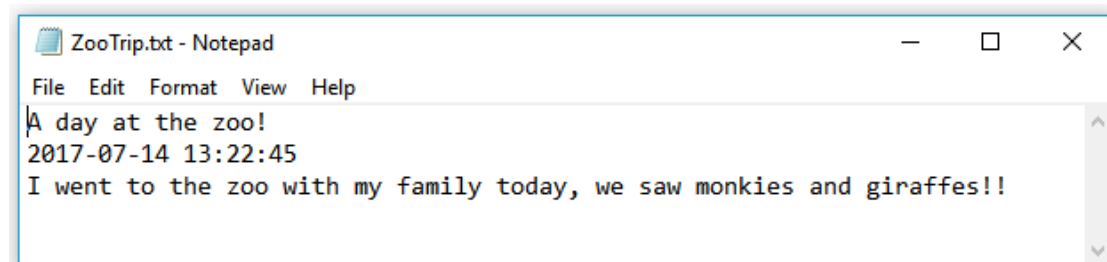
-----
                A DAY AT THE ZOO!
                2017-07-16 14:21:17
-----
I went to the zoo with my family today, we saw monkies and giraffes!!
-----
Options:
[1] Export to text file
[2] Change title
[3] Change content
[4] Delete entry
-----
Enter Option number
or press any other key to return to main menu.
-----
Enter filename: ZooTrip
File exported to ZooTrip.txt!
Press any key to continue!

```

Figur 7 Exportering av logginlägg

#	Name ^	Size	Modified	Comment
1	 exercise07_logbook.exe	12,00 KB	2 mins	
2	 exercise07_logbook.pdb	23,50 KB	2 mins	
3	 exercise07_logbook.exe.config	189 bytes	2 days	
4	 ZooTrip.txt	111 bytes	2 mins	

Figur 8 Exporterad textfil



Figur 9 Öppnad exporterad textfil

Med menyalternativ "[4] Sort" kan användaren sortera loggbokens inlägg. Sorteringsalternativ är efter titel, ID eller datum. Stigande eller fallande sortering.

```

-----
                        SORT ENTRIES
-----
Sort by:
[1] Entry ID, Ascending
[2] Entry ID, Descending
[3] Entry Title, Ascending
[4] Entry Title, Descending
[5] Entry Date, Ascending
[6] Entry Date Descending
-----
Enter menu item to sort entries,
or press any other key to return to main menu.

```

Figur 10 Sorteringsmenyn

```

-----
                        ALL ENTRIES (17)
-----
ID      DATE                TITLE
[0]     [2017-07-16 14:21:17] [A day at the zoo!]
[6]     [2017-07-16 14:34:39] [A day at the zoo!]
[12]    [2017-07-16 14:37:21] [A day at the zoo!]
[5]     [2017-07-16 14:21:17] [C# Classes and Objects]
[11]    [2017-07-16 14:34:39] [C# Classes and Objects]
[17]    [2017-07-16 14:37:21] [C# Classes and Objects]
[4]     [2017-07-16 14:21:17] [C# Coding]
[10]    [2017-07-16 14:34:39] [C# Coding]
[16]    [2017-07-16 14:37:21] [C# Coding]
[8]     [2017-07-16 14:34:39] [New TV-Show]
[14]    [2017-07-16 14:37:21] [New TV-Show]
[1]     [2017-07-16 14:21:17] [Today's lunch]
[7]     [2017-07-16 14:34:39] [Today's lunch]
[13]    [2017-07-16 14:37:21] [Today's lunch]
[3]     [2017-07-16 14:21:17] [Weather Report]
[9]     [2017-07-16 14:34:39] [Weather Report]
[15]    [2017-07-16 14:37:21] [Weather Report]

Enter ID to display entry.
Input blank to return to the main menu.
Enter ID:

```

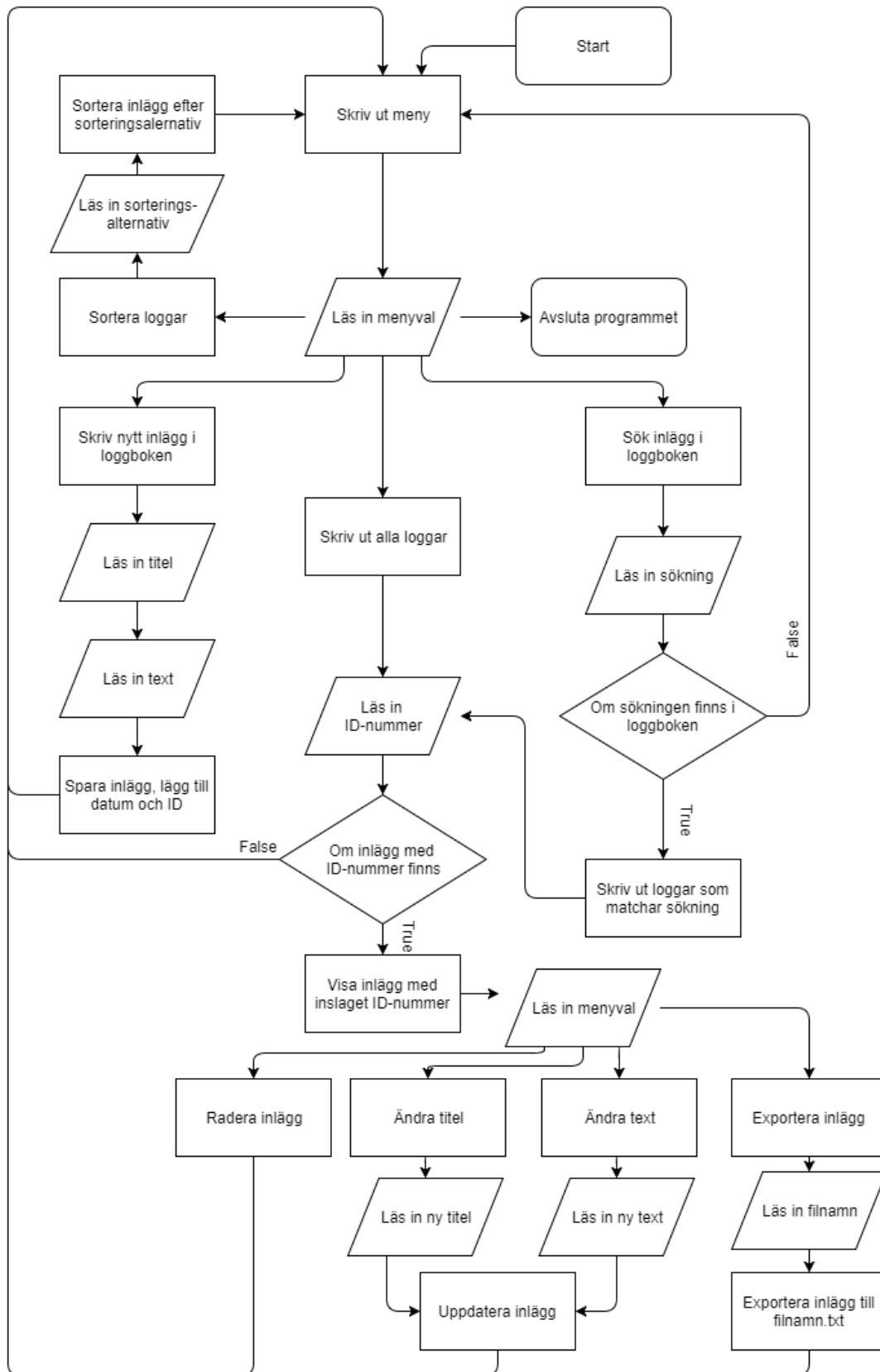
Figur 11 Logginlägg sorterade efter titelnamn

Loggbokens data sparas mellan programkörningar, till en datafil kallad *data.dat*. Filen sparas till samma katalog där programmets exekverbara fil finns.



## 2.5 Flödesschema

Här visas ett flödesschema för programmets funktion



Figur 12 Flödesschema

## 2.6 Pseudokod

Här visas den pseudokod som skrevs för programmet.

### START

**SÅ LÄNGE** *Tills användaren avslutar programmet*

Presentera meny för användaren

**VÄLJ** *fall ur*

*Menyval 1: Skriv nytt inlägg*

Läs in titel

Läs in text

Spara inlägg, lägg till datum och id

*Menyval 2: Skriv ut alla loggar*

Lista titlar på alla inlägg

Be användaren slå in ID för att visa inlägg

**OM** *inlägg finns med inslaget ID*

Visa inlägg

Visa alternativ för inlägg: radera, redigera, exportera

*Menyval 3: Sök inlägg i loggboken*

Läs in sökning

**OM** *sökning finns i loggboken*

Lista titlar på inlägg som matchar sökning

Be användaren slå in ID för att visa inlägg

**OM** *inlägg finns med inslaget ID*

Visa inlägg

Visa alternativ för inlägg: radera, redigera, exportera

*Menyval 4: Sortera*

Lista sorteringsalternativ

Be användaren välja ett sorteringsalternativ

**OM** *användaren väljer ett sorteringsalternativ*

Sortera inläggslistan efter vald sortering

*Menyval 5: Avsluta*

Avsluta programmet

### STOPP

## 2.7 Skillnader från uppgift 3

Pseudokoden från uppgift 3 har fått följande tillägg för detta program:

- Inlägg sparas med tillagt datum och ID-nummer.
- När inlägg listas med titlar från sök eller "lista alla loggar" kan användaren visa ett inlägg genom att skriva in dess ID-nummer.
- Nytt menyalternativ: Sortering

## 2.8 Programmetts kod

Källkoden beskrivs enbart övergripande i denna rapport, i den inlämnade källkodsfilen finns ytterligare kommentarer (på engelska) som beskriver koden.

### 2.8.1 Klassen Logbook

Klassen "`class Logbook`" innehåller en lista av string-arrayer "`List<string[]> entries`", som används för att hålla loggbokens inlägg. Varje inlägg består av en array med fyra string-variabler; Titel, ID-nummer, Text-innehåll, och datum/tid.

#### Metoder i klassen Logbook

Flera metoder i klassen Logbook finns som överlagrade, en variant där parametrar skickas till metoden direkt och en där användaren ombeds skriva in data manuellt.

Metod	Beskrivning
<b>Logbook</b>	Konstruktor för klassen. Laddar in sparad loggboksdata från fil om den existerar. Konstruktorn kan användas för att skapa generera testinlägg.
<b>Count</b>	Ger antal inlägg i loggboken.
<b>ClearSearchHits</b>	Nollställer sökresultat.
<b>Search</b>	Söker i logginlägg efter en angiven söksträng, finns som överlagrad. Metoden använder <i>linjär sökning</i> .
<b>DisplaySearchHits</b>	Listar titlarna på inlägg från det senaste sökresultatet.
<b>DisplayAllTitles</b>	Listar titlarna på samtliga inlägg i loggboken
<b>DisplayTitle</b>	Listar titeln på angivet inlägg, inläggets ID skickas som parameter.
<b>AddEntry</b>	Lägger till ett inlägg till loggboken, finns som överlagrad.
<b>DeleteEntry</b>	Raderar ett inlägg från loggboken.
<b>DisplayEntry</b>	Visar innehållet i ett inlägg, finns som överlagrad. Visar också en lista med alternativ för det aktuella inlägget: exportera, radera, redigera titel, redigera text.
<b>EditTitle</b>	Redigerar titel på ett inlägg.
<b>EditContent</b>	Redigerar texten i ett inlägg.
<b>ExportEntry</b>	Exporterar ett inlägg till en textfil.
<b>Save</b>	Sparar loggboken till en datafil.
<b>Load</b>	Laddar loggboken från en datafil.
<b>UpdateEntry</b>	Ändrar titel och/eller text för ett inlägg. Anropas av metoderna EditTitle och EditContent.
<b>Sort</b>	Sorterar loggboksinlägg. Finns som överlagrad. Metoden använder <i>insertion sort</i> .
<b>SortSwap</b>	Hjälpmetod för Sort, byter indexplats på två loggboksinlägg.
<b>FindEntryIndex</b>	Letar upp indexplats för ett inlägg, tar ID som parameter.
<b>AddSamples</b>	Lägger till några förskrivna logginlägg till loggboken, för testning.

### Variabler och datastrukturer i klassen Logbook:

Namn	Typ	Beskrivning
<code>entries</code>	<code>List&lt;String[]&gt;</code>	Håller loggboksinlägg med titel, text, ID-nummer och datum.
<code>searchHits</code>	<code>int[]</code>	Array som håller sökträffar, som ID-nummer.
<code>lastSearchString</code>	<code>string</code>	Senaste söksträngen som användes för sökning
<code>dataFileName</code>	<code>string</code>	Filnamn för datafil som sparar loggboken mellan programkörningar.
<code>id</code>	<code>int</code>	Används för att sätta ID-nummer på nya logginlägg.
<code>SearchData</code>	<code>enum</code>	Värden för sökningsmetoder, tom och max antal sökträffar.
<code>sortMenu</code>	<code>string[]</code>	Text för den meny som visas vid val av sortering.
<code>displayEntryMenu</code>	<code>string[]</code>	Text för de alternativ som kan göras när ett inlägg visas.
<code>SortBy</code>	<code>enum</code>	Sorteringsalternativ.
<code>EntryData</code>	<code>enum</code>	Indexnummer för text, ID-nummer, datum och titel i char-arrayen i listan <code>entries</code> .
<code>errorMsg</code>	<code>string[]</code>	Felmeddelanden.
<code>ErrorId</code>	<code>enum</code>	Index för felmeddelanden.

### 2.8.2 Klassen Menu

Klassen `class Menu` innehåller metoder för att visa menyer och diverse text, samt funktionalitet för användarinmatning.

### Metoder i klassen Menu:

Metod	Beskrivning
<code>DisplayMenu</code>	Visar programmets huvudmeny, ger i retur användarens inmatade menyval.
<code>DisplayTitle</code>	Genererar titlar för programmets olika menyer. Titeltexten anges som parameter. Texten centreras.
<code>DisplayLine</code>	Ritar ut en linje till konsolfönstret.
<code>Wait</code>	Uppmanar användaren att trycka på en valfri tangent för att fortsätta programmet. Meddelande kan ges som parameter.
<code>Confirm</code>	Ger i retur ett sanningsvärde, beroende på om användaren trycker in en viss tangent. Exempel: <i>"Tryck på 'J' för att ta bort filen"</i> .
<code>Error</code>	Visar ett felmeddelande, meddelandet anges som parameter.

### Variabler och datastrukturer i klassen Menu:

Namn	Typ	Beskrivning
<code>MenuItem</code>	<code>enum</code>	För navigering i huvudmenyn.
<code>lineLen</code>	<code>int</code>	Längd för linjer, används i <code>DisplayLine</code> och <code>DisplayTitle</code>

## 3 Diskussion och slutsats

### 3.1 Förslag på förbättringar

#### **Piltangentsnavigering**

Navigering och val i menyer hade kunnat utföras med tangentbordets piltangenter och retur-knapp. Det känns eventuellt mer naturligt och smidigare än att trycka in en siffra på tangentbordet för att välja ett menyalternativ.

På grund av att olika antal logginlägg kan listas vid sökning eller menyalternativ "[2] List all logbook entries" upplevdes det att det skulle bli komplext att implementera piltangentsnavigering.

#### **Bättre sortering**

I programmet kan användaren välja sex olika sorteringsalternativ.

Inlägg med samma titeltext sorteras inte sinsemellan när sortering efter titel har valts. Likaså med inlägg med samma datum, om sortering efter datum har valts.

```
[6]      [2017-07-16 14:34:39]    [A day at the zoo!]  
[18]     [2017-07-17 09:59:18]    [A day at the zoo!]  
[12]     [2017-07-16 14:54:19]    [A day at the zoo!]  
[0]      [2017-07-16 14:21:17]    [A day at the zoo!]
```

Figur 13 Nuvarande sortering av inlägg med samma titelnamn

En lösning skulle kunna vara att sortera inlägg med samma titel också sekundärt, efter ID-nummer eller datum.

Likaså med sortering efter datum, sekundärt efter ID-nummer eller titel.

#### **Inmatning av tecknen Å, Ä och Ö**

Vid skapande av nya inlägg blir det mellanrum efter de svenska bokstäverna Å, Ä och Ö, när de skrivs in, som versaler.

```
-----  
                        ADD LOG ENTRY  
-----  
Enter title for new entry: TEST:Å Ä Ö  
Enter text for new entry:  
Testtext åäö Å Ä Ö
```

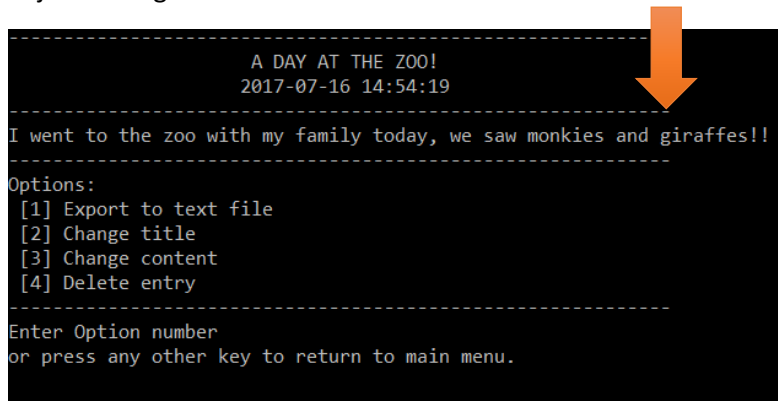
Figur 14 Inmatning av Å, Ä och Ö

Dock visas tecknen normalt efter att inlägget har skapats. Därför har detta problem inte försökt åtgärdas i nuvarande program.

**Förbättrat utseende**

Programmets utseende hade kunnat förbättrats, med till exempel bättre indentering av text, textfärg eller radbrytningar.

När logginlägg visas skulle det varit estetiskt trevligare om inläggets text/innehåll hade brutits innan linjernas längd.



```
-----
                A DAY AT THE ZOO!
                2017-07-16 14:54:19
-----
I went to the zoo with my family today, we saw monkieS and giraffes!!
-----
Options:
[1] Export to text file
[2] Change title
[3] Change content
[4] Delete entry
-----
Enter Option number
or press any other key to return to main menu.
```

Figur 15 Visning av inlägg, text går utanför linjelängd, markerat med orange pil

## 3.2 Val av algoritmer, variabler och datastrukturer

### Sökning och sortering

För sökning användes linjär sökning, då det valdes att alla inlägg som matchar söksträngen skulle visas. För linjär sökning behöver inte listan med inlägg vara sorterad innan.

För sortering av inlägg i loggboken användes *Insertion Sort*, koden baserades på färdig kod från en tidigare övning i programspråket C.

Modifiering av algoritmen gjordes så att den kan hantera stigande och fallande sortering, beroende på en sanningsvariabels värde. Arrayen som sorteras i C-koden byttes till Listan med logginlägg.

```
/* Byggt efter pseudokod på insertion sorts wikipedia-sida */
void insert_sort(int *array, int size){
    int ix, jx;
    for(ix = 0; ix < size; ix++){
        jx = ix;
        while(jx && (array[jx-1] > array[jx])){
            _swap(&array[jx], &array[jx-1]);
            jx--;
        }
    }
}

void _swap(int *n1, int *n2) {
    int temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}
```

Figur 16 Insertion sort i C

```
/* Insertion Sort */
int ix, jx;

for (ix = 0; ix < entries.Count; ix++)
{
    jx = ix;

    if (ascending)
    {
        while (jx != 0 &&
            string.Compare(entries[jx - 1][sortIndex], entries[jx][sortIndex]) > 0)
        {
            SortSwap(jx, jx - 1);
            jx--;
        }
    }
    else
    {
        while (jx != 0 &&
            string.Compare(entries[jx - 1][sortIndex], entries[jx][sortIndex]) < 0)
        {
            SortSwap(jx, jx - 1);
            jx--;
        }
    }
}
```

Figur 17 Modifierad Insertion sort för loggboken i C#

### ***Menytext och felmeddelanden***

Text för menyer och felmeddelanden lagras i string-arrayer, menyer kan lätt då skrivas ut med loopar och enum-variabler kan användas för att indexhantering av menyval och visning av felmeddelanden.

Ett mindre bekymmer med enum-värden är att de måste typecastas till int för att användas som indexvärde för arrayer.

Exempel av användning av string-array och enum för felmeddelande:

```
Menu.Error(errorMessage[(int)ErrorId.EmptyLog], true);
```

`errorMessage` är string-arrayen med text för de olika meddelandena, och `ErrorId` är enum-variabeln som håller index för de olika meddelandena. I detta fall visas meddelandet: "No entries found!"

### 3.3 Utvärdering av insats

Det upplevs att programmet har nått en tillfredsställande nivå i komplexitet och funktion.

Dock hade mer tid ägnats åt programmets utseende och menyer, om det hade varit möjligt.

Eventuellt skulle koden också kunnat snyggats till, metoderna i klassen Logbook upplevs som osorterade. Arbete hade kunnat lagts på att flytta om dem till en mer naturlig ordning.

Vissa kodrader upplevs också som för långa.

Programmet har ej testats i någon större utsträckning, vissa körtidsfel eller andra fel kan säkerligen uppkomma under användande av programmet.