**Examinerande uppgift Loggboken**

Rapport

**Dokumentation för uppgiften ”Loggboken”**

**Kurs Programmering 1 C#**

**NTI-Skolan**

**Johan Kämpe  
2017-07-16**

# 1 Inledning

## 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.

## Noteringar

Skärmdumpar 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.

## Länkar

FIXA

# 2 Genomförande och resultat

## 2.1 Använd programvara och dokumentation

SKRIV

DRAW IO, word, visual studio, etc boken

## 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 loggboksinlägg.
* Möjlighet att redigera eller ta bort loggboksinlägg.
* Skapa en eller flera metoder för programmet.
* Implementera algoritmer för sökning och sortering av loggboksinlägg.

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

## 2.3 Metod

Programmets källkod skrevs uteslutande i utvecklingsmiljön *Microsoft Visual Studio Community 2017.* Som projektmall användes *Console App (.NET Framework).*

BESKRIV VARFÖR VS ÄR BRA, INTELLISENSE ETC

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

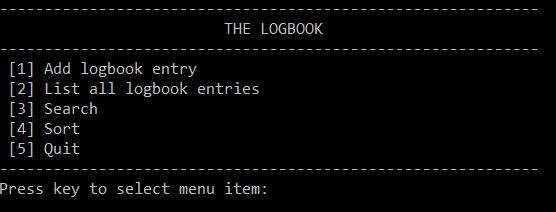


Figur Kod i Visual Studio, debugläge med break point

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

## 2.4 Programmets funktion

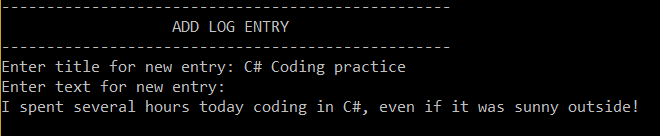
Vid programmets start presenteras en huvudmeny för användaren. Menyval utförs genom att trycka på en siffra på tangentbordet. I huvudmenyn behövs inget returslag efter sifferinmatning.



Figur 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.

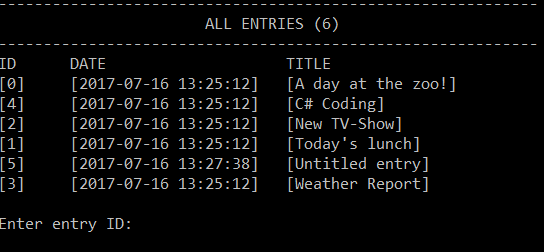


Figur Skapande av ett nytt logginlägg

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.

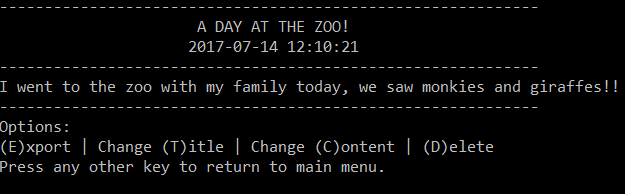
Med menyalternativ "[2] List all logbook entries" listas alla logginlägg med titlar, datum och ID-nummer.



Figur 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.

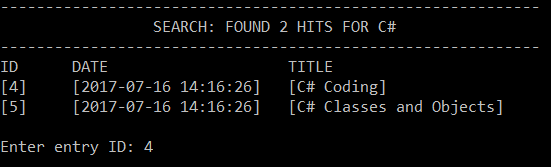
Vid visning av ett inlägg kan användaren välja att redigera (ändra titel eller textinnehåll), radera, eller exportera inlägget till en textfil.



Figur 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.

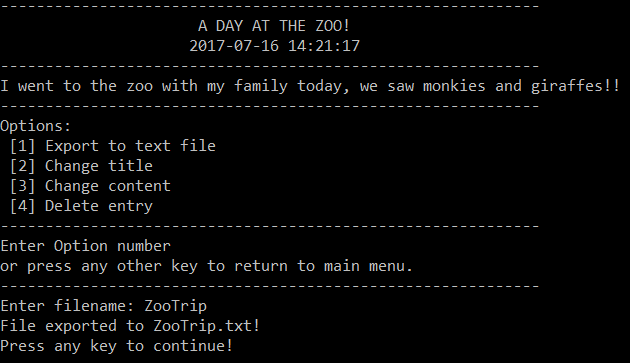


Figur Sökning av logginlögg

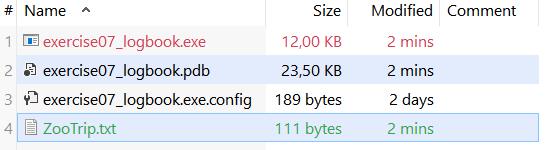
Notera att det går att skriva in ett ID-nummer för inlägg som inte visas i sök-listan, och detta ändå visas om det finns i loggboken.

Titlar, datum och inläggens text genomsöks för matchning.

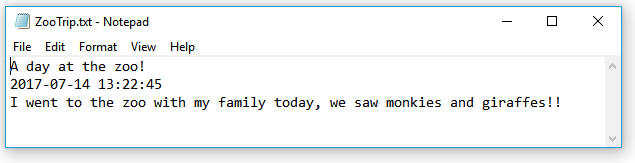
Vid exportering av ett inlägg ombeds användaren att skriva in ett filnamn. Filen sparas till samma katalog där programmets exekverbara fil finns. Om filen inte går att exportera visas ett felmeddelande.



Figur Exportering av logginlägg

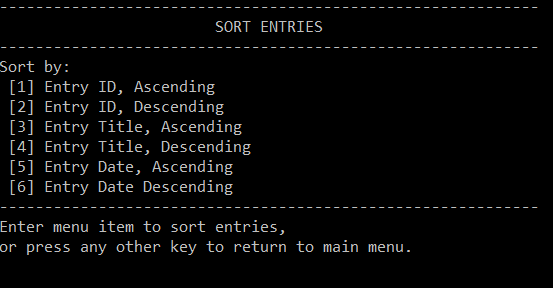


Figur Exporterad textfil

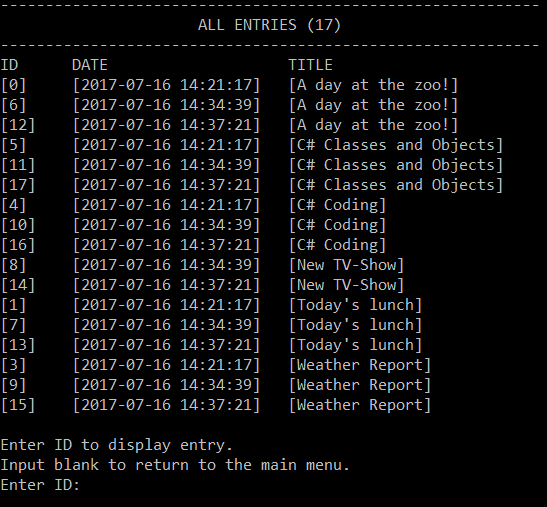


Figur Öppnad exporterad textfil

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



Figur Sorteringsmenyn

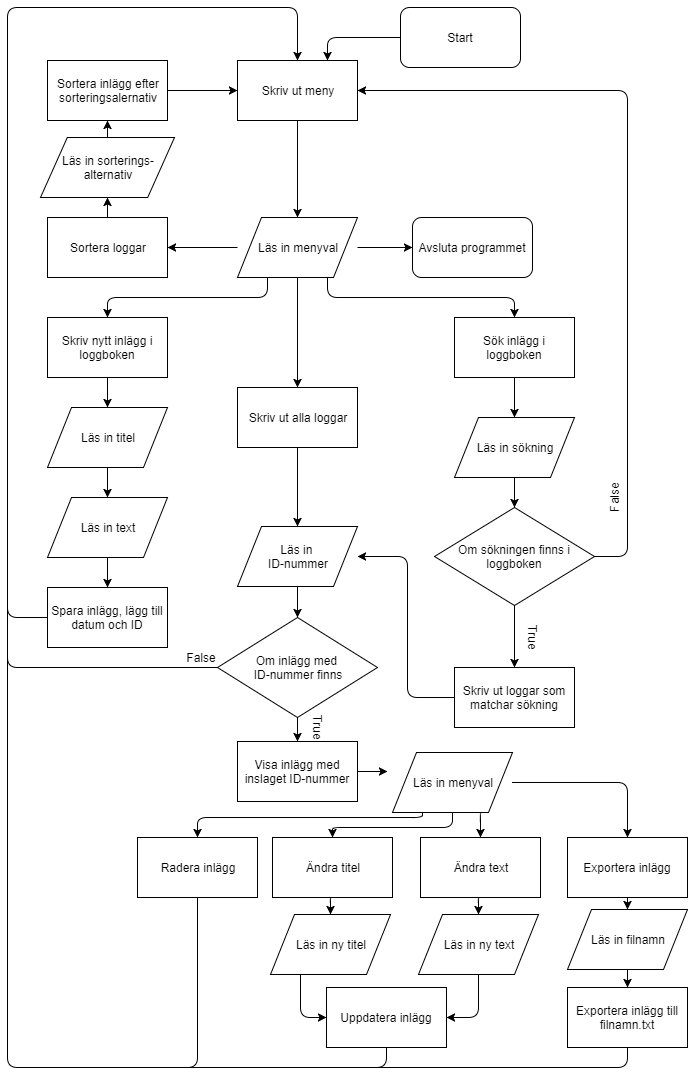


Figur Logginlägg sorterade efter titelnamn

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 Flödesschema

## 2.6 Pseudokod

Här visas den pseudokod som skrevs för programmet.

**START**

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

Variabler och datastrukturer i klassen Logbook:

|  |  |  |
| --- | --- | --- |
| Namn | Typ | Beskrivning |
| entries | List<String[]> | Håller loggboksinlägg med titel, text, ID-nummer och datum. |
| searchHits | int[] | Array som håller sökträffar, som ID-nummer. |
| lastSearchString | string | Senaste söksträngen som användes för sökning |
| dataFileName | string | Filnamn för datafil som sparar loggboken mellan programkörningar. |
| id | int | Används för att sätta ID-nummer på nya logginlägg. |
| SearchData | enum | Värden för sökningsmetoder, tom och max antal sökträffar. |
| sortMenu | string[] | Text för den meny som visas vid val av sortering. |
| displayEntryMenu | string[] | Text för de alternativ som kan göras när ett inlägg visas. |
| SortBy | enum | Sorteringsalternativ. |
| EntryData | enum | Indexnummer för text, ID-nummer, datum och titel i char-arrayen i listan *entries*. |
| errorMsg | string[] | Felmeddelanden. |
| ErrorId | enum | 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 |
| DisplayMenu | Visar programmets huvudmeny, ger i retur användarens inmatade menyval. |
| DisplayTitle | Genererar titlar för programmets olika menyer. Titeltexten anges som parameter. Texten centreras. |
| DisplayLine | Ritar ut en linje till konsollfönstret. |
| Wait | Uppmanar användaren att trycka på en valfri tangent för att fortsätta programmet. Meddelande kan ges som parameter. |
| Confirm | Ger i retur ett sanningsvärde, beroende på om användaren trycker in en viss tangent. Exempel: *”Tryck på ’J’ för att ta bort filen”.* |
| Error | Visar ett felmeddelande, meddelandet anges som parameter. |

# 3 Diskussion och slutsats

HOPPA ÖVER PILNAVIGERING – jobbigt vid listande av titlar

ÅÄÖ input konstigt…

Val av algoritmer, etc etc

Sortering med samma namn/datum – jobbigt att fixa