**TRF**

**Tigerälskarnas Riksförbund**

Rapport

**Programmering 2 C#**

**NTI-Skolan**

**Johan Kämpe**<https://www.linkedin.com/in/johankampe><https://github.com/GoblinDynamiteer>

**2017-07-17**

Innehåll

[***1 Inledning*** 3](#_Toc488052648)

[**1.1 Syfte** 3](#_Toc488052649)

[**1.2 Noteringar** 3](#_Toc488052650)

[**1.3 Länkar** 3](#_Toc488052651)

[***2 Genomförande och resultat*** 4](#_Toc488052652)

[**2.1 Använd programvara och litteratur** 4](#_Toc488052653)

[**2.2 Avgränsningar och krav** 4](#_Toc488052654)

[2.2.1 Grundläggande funktionalitet 4](#_Toc488052655)

[2.2.2 Utökad funktionalitet 4](#_Toc488052656)

[**2.3 Metod** 5](#_Toc488052657)

[**2.4 Programmets funktion** 5](#_Toc488052658)

[**2.5 Flödesschema** 9](#_Toc488052659)

[**2.6 Pseudokod** 10](#_Toc488052660)

[**2.7 Skillnader från uppgift 3** 11](#_Toc488052661)

[**2.8 Programmets kod** 11](#_Toc488052662)

[2.8.1 Klassen Logbook 11](#_Toc488052663)

[2.8.2 Klassen Menu 12](#_Toc488052664)

[***3 Diskussion och slutsats*** 13](#_Toc488052665)

[**3.1 Förslag på förbättringar** 13](#_Toc488052666)

[**3.2 Val av algoritmer, variabler och datastrukturer** 15](#_Toc488052667)

[**3.3 Utvärdering av insats** 16](#_Toc488052668)

# 1 Inledning

## Syfte

Syftet med uppgiften är att skapa ett program i programspråket C#, som underlag finns en projektspecifikation med vissa krav som måste uppfyllas.

Utöver dessa krav får programmet skapas med fria händer.

## Noteringar

Skärmklipp 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();**

## Länkar

Uppgiften på GitHub  
<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](https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=loggboken.html%23Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D0B0Da8QGBLf-0LXdwaFd0dVBpVnM%26export%3Ddownload)

Ikoner som används i programmet  
<https://www.iconfinder.com/icons/381599/error_icon>  
<https://www.iconfinder.com/icons/285654/cat_icon>

# 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 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 *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

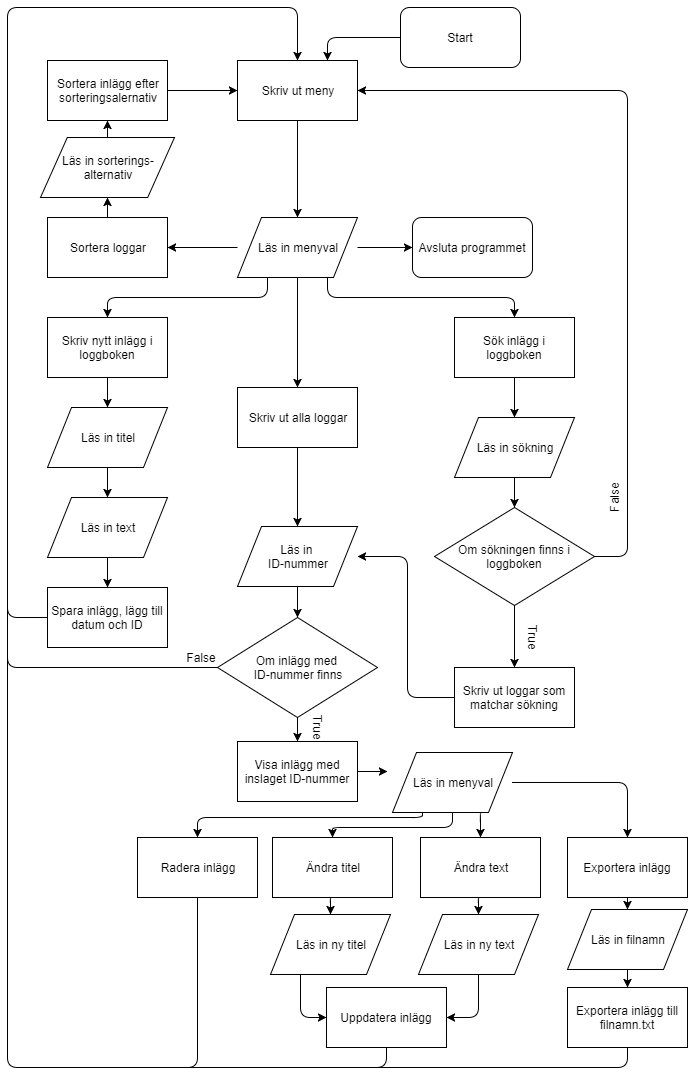
Projektmallen skapar en första källkodsfil kallad *Program.cs* för projektet. Detta var den enda källkodsfil som användes för programmet.

## 2.4 Programmets funktion

BESKRIV PROGRAMMET

## 2.5 Flödesschema

Här visas ett flödesschema för programmets funktion



Figur 12 Flödesschema

DATABASEN/SQL/ADAPTER ETC

## 2.8 Programmets kod

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

Kod som är automatiskt genererad av Visual Studio beskrivs eller kommenteras ej.

### 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. |
| AddSamples | Lägger till några förskrivna logginägg till loggboken, för testning. |

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. |

Variabler och datastrukturer i klassen Menu:

|  |  |  |
| --- | --- | --- |
| Namn | Typ | Beskrivning |
| MenuItem | enum | För navigering i huvudmenyn. |
| lineLen | int | Längd för linjer, används i DisplayLine och DisplayTitle |

# Diskussion och slutsats

## 3.1 Förslag på förbättringar

ETC