

# Visual Studio

## Indholdsfortegnelse:

<b>Indholdsfortegnelse:</b>	1
<b>Indledning:</b>	1
<b>Dokumentation</b>	1
Spørgsmål:	1
Opsætning:	4
Libary kode:	4
Kode for Fodbold opgave:	8
Kode for Danse opgave:	9
Kode for Password opgave	10
<b>Konklusion</b>	12
<b>Opgaveforløb</b>	12

## Indledning:

Vi har fået til opgave at løse 3 opgaver i C# som omhandler class og objekts. Første Opgaver er en fodbold opgave som går ud på at finde ud af hvad publikum skal råbe ud fra antal afleveringer og mål. Anden opgave omhandler en danse konkurrence og at lave et program som holder styr på points som danse partner får. Sidste opgave er at skabe et program som holder styr på at man oprette et sikkert kodeord.

## Dokumentation

### Spørgsmål:

***Hvad er et Entry point og hvor mange kan man have i koden?***

Entry points lad dig navngive en funktion som noget andet end den var originalt. Du kan have så mange af dem som du vil.

**Hvad kalder man en variable der er erklæret ved hjælp af ordet "var"?  
Og hvornår skal man initialiser variablen?**

Var er en variable som først bliver initialiseret når den får en værdi. Så hvis du sætter en var til en string værdi laver compileren den om til en string.

**Dokumenter og forklare hvad hver del gør i initialisering af et loop og hvordan det afvikles. Gør dette for alle loop funktioner.**

While():{}

Et while loop læser koden i {} indtil det i () ikke længere passer.

Do{}While()

Et do while loop læser koden i {} en gang og hvis det i () stadige passer læser den det igennem igen, indtill det i () ikke passer længere.

While() Do{}

Ser ikke ud til at eksistere, men hvis den gør så må den gøre det samme som en while løkke bare mere opdelt. (Jeg kan ikke finde information om den)

Foreach(){}

Læser koden i igennem {} for hver gang (). Det kan for eksempel være foreach

For(statement 1 , statement 2 , statemet 3){}

Bliver brugt når man ved hvor mange gange man ville køre et loop. Statement 1 bliver læst en gang før {} bliver læst. Statement 2 siger noget om hvad der skal til for at loopet bliver ved. Statement 3 bliver læst igennem hver gang loopet bliver kørt igennem. For loops kan se sådan her ud: for (int i = 0; i < 5, i++){}

**Hvad er forskelle på ++x og x++?**

X++ plusser x med en, efter det er blevet læst igennem, hvorimod ++x plusser x med en men det bliver læst igennem. Så hvis x er 1 og du skriver Console.WriteLine(x++) så kommer x stadige ud som 1, fordi den først bliver plusset når koden er læst.

**Hvad betyder det når noget er null, og hvorfor kan vi ikke lide det?**

Null er når noget ikke er givet en værdi, det kan for eksempel være et string som ikke er sat til noget, eller et objekt der ikke har fået nogen værdi. Det generelt set en dårlig ting at prøve at anvende de ting der sat til null, da de kommer ud som null og det kan programmet ikke forstå så det crasher.

***Hvornår er string interpolation operator (\$) en fordel at bruge? Og hvornår er StringBuilder en fordel at bruge? Hvorfor skal man være opmærksom på det?***

String interpolation lad dig printe en masse variabler på en overskuelig måde. Tilgængelig lad StringBuilder dig ændre på en masse ting i string'en som om tekst skal tilføjes foran eller bagved. Så jeg ville sige til ikke så kompliceret ting ville jeg nok bruge interpolation, men til lange og kompliceret ting ville jeg bruge StringBuilder.

***Hvad er forskellen på Value types og reference types? Hvordan fungerer en Value type og hvordan fungerer en reference type? Herunder hvad er det der sker når du kopier et objekt( reference types) og når du kopiere en værdi (Value type)***

Value types holder data i deres eget sted, hvorimod reference types referer til et andet sted dataet ligger. Reference types er ting som en string, og value types er ting som int og bool. Når du kopiere en reference type skaber programmet et nyt sted i memory til det nye objekt til at pege på den samme værdi som det gamle objekt. Når du kopiere en value type så skaber bliver den nye værdi gemt separat fra den gamle.

***Hvad er encapsulation / Information hiding og hvorfor ønsker vi at gøre det?***

Encapsulation er når vi for eksempel laver en privat værdi eller protected. Vi gør det primært når der er ting vi ikke ville have brugeren til at røre ved eller se.

***Forklar de forskellige dele der til sammen skaber en metode. Først generelt og derefter med udgangspunkt i denne metode.***

Det første ord bestemmer om metoden skal være private, public osv. Det var det med encapsulation som skrevet om tidligere. Det næste bestemmer om metoden skal være static altså om dens værdier kan ændres på. Bagefter bestemmer man hvilken type metoden skal være, så som string og int.

***Private static int?[] MyMethod(int x, int y = 20)***

```
{  
  
Return new int?[3] { 5 + x, y ,null};  
  
}
```

***Forklar hvad der returneres?***

Der returnes en ny int array hvor [1] er x + 5 og [2] er 20 og [3] er null.

### ***Hvad ligger der bag begrebet "Operator overloading"?***

Operatorer er matematiske tegn som + eller -. Operator overloading er når vi selv laver regler for hvad der skal ske når vi + eller - ting sammen, som normalt ikke indgår i matematik. Dette er ting som objekter.

### ***Hvad gør en Destructor og hvorfor anvendes den? Hvorfor bruger udvikler den ikke særligt ofte i C# kode?***

En destructor køres automatisk når programmet lukkes. Man kan ikke selv styre når destructoren kører, men den bruges til at fjerne ting som objekter, og værdier der ikke skal gemmes.

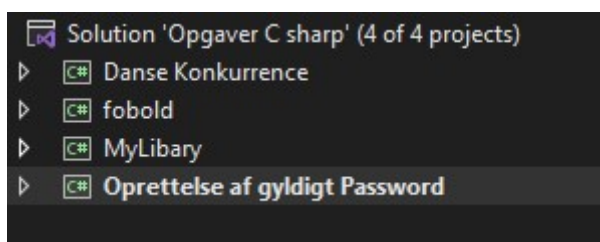
### ***Hvad er Regular Expressions (RegEx) og hvordan kan man bruge regex?***

RegEx kan bruges til at læse strings efter specifikke ting, så som om strings starter med samme bogstav eller om de indeholder mellemrum.

### ***Hvis du har flere projekter, så skal du forklare hvad en cirkulær reference er?***

Cirkulær reference er når 2 ressourcer der ikke er afhængige af hinanden stopper hinanden for at virke. Det kan være 2 classes der bruger samme værdier og stopper hinanden for at virke.

## **Opsætning:**



Figur 1 Opsætning

Opgaverne er delt ind under en solution med et samlet Library til alle 3.

## **Library kode:**

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
namespace MyLibrary
{
    public class Afleveringer //dette er vores class jeg har valgt at kald afleveringer
    {
        private readonly int _pass; //her sætter vi nogle variabler som kun høre til
        denne class, altså det ændre ikke på vores originale variabler fra programmet
        private readonly string _mål;
        public Afleveringer(int pass, string mål) //her siger vi så hvad et objekt i
        denne class skal have af argumenter
        {
```

```
        _pass = pass; //her sætter vi de variabler fra linje 12-13 til at være det
samme som vi for fra objektet
        _mål = mål;
    }
    public string Shout() //her skriver vi en metode i vores class, som vi gerne ville
bruge til at finde ud af hvad vi skal printe ud
    {
        StringBuilder udskrivning = new StringBuilder("", 50); //her skaber vi så en
stringbuilder, som er en form for string vi kan tilføje mere på. 50 er bare at den
skal kunne holde op til 50 tegn.

        if (_mål.ToLower() == "mål") //her checkker vi så om brugeren har skrevet mål,
ligegyldigt med stort eller småt
        {
            udskrivning.Append("Olé olé olé");//her tilføjer vi så svarnde tekst
hvis der mål til vores stringbuilder
        }
        else //else betyder at det næste i koden skal ske hvis den første if
statement ikke bliver opfyldt
        {
            if (_pass < 10 & _pass > 1)//denne if statement chekker om der er over 1
aflevering og under 10
            {
                for (int i = 0; i < _pass; i++) //denne løkke gør koden inden i den
flere gange baseret på antal afleveringer
                {
                    udskrivning.Append("Huh!");//koden her tilføjer Huh! til vores
stringbuilder
                }
            }
            //her siger vi hvad der skal ske hvis der er over 10 aflevering
            else if (_pass > 10) { udskrivning.Append("High Five - Jubel!!!"); }
            //her siger vi hvad der skal ske hvis der er under 1 aflevering
            else if (_pass < 1) { udskrivning.Append("Shh"); }
        }
        //her skal metoden så returnere hvad end udskrivning nu er blevet sat til, og
den skal selvfølgelig være et string.
        return Convert.ToString(udskrivning);
    }
}

public class DancerPoints //i dancerpoints classen skriver vi igen hvilke 2 værdier
vi skal bruge.
{
    private readonly int _points; //her sætter vi nogle variabler som kun høre til
denne class, altså det ændre ikke på vores originale variabler fra programmet
    private readonly String _navn;
    public DancerPoints(string navn,int points) //her skriver jeg hvad et objekt i
klassen skal have af værdier
    {
        _points = points; //her sætter vi de variabler fra linje 50-56 til at være
det samme som vi for fra objektet
        _navn = navn;
    }

    public string I_alt() //i alt metoden er her for at ligge navn og points
sammen så programmet nemmere kan print det
    {
```

```
        return (_navn + _points); //hvad i alt skal returnere når vi kalder den
    }

    public static DancePoints operator+ (DancePoints a, DancePoints b) //her
    bruger jeg en operator til at sige hvad der sker når 2 dancepoint objekter plusses
    sammen
    {
        //den skal lave et nyt objekt som svarer til navne og pointsne plusset med
        hinanden fra objekterne
        DancePoints c = new DancePoints(a._navn + "&" + b._navn , a._points +
        b._points);
        return(c); //til sidst skal den så returnere det nye objekts værdier.
    }
}

public class Password_check //her skabes en class
{
    private string _password; //her sætter vi nogle variabler som kun hører til denne
    class, altså det ændrer ikke på vores originale variabler fra programmet
    private string _navn;
    public Password_check(string navn, string password)
    {
        _password = password; //her sætter vi de variabler fra linje 12-13 til at
        være det samme som vi får fra objektet
        _navn = navn;
    }
    public bool Oprt_txt() //en metode som opretter en tekst baseret på objektets
    værdier
    {
        string path = @"C:\Users\rassør\Desktop\Visuel studio
        opgaver\teks.txt"; //sætter path til .txt filen
        File.WriteAllText(path, _navn + Environment.NewLine + _password);
        return true;
    }
    public bool Check(string input1, string input2) //Check om password møder
    requirements
    {
        _password = input2; //input 2 er brugerens input på password
        _navn = input1; //input 1 er brugerens input på brugernavn

        string path = @"C:\Users\rassør\Desktop\Visuel studio opgaver\teks.txt"; //her
        sætter vi vores path til .txt filen

        string[] readText = File.ReadAllLines(path); //her læser programmet txt filen
        og gemmer hver linje i en string
        if (_navn == _password) { return false; } //check om passwordet ikke er det
        samme som brugernavn

        if (readText[0] == _navn) //check om brugernavn indtastet er det samme som
        brugernavn gemt i .txt filen
        {
            if (readText[1] == "n") { return (true); } //check om passwordet i .txt
            filen er n, dette er fordi den kun kan blive sat til n hvis der ikke er noget password
        }
    }
}
```

```
        if (readText[1] == _password) //chekker at password indtastet er det samme
som password gemt i .txt filen
        {
            return (true); //når Check() bliver returned true betyder det at brugeren
har skrevet rigtigt brugernavn og password
        }
        else { return (false); } //hvis brugen ikke har tastet rigtigt bliver
Check() returned false
    } else { return (false); }
}

public bool Create_Pass(string createText , string Brugernavn) //chekker om det
skabte password er oppe til standard
{
    bool lowercase = false;
    bool uppercase = false;
    string path = @"C:\Users\rassør\Desktop\Visuel studio
opgaver\teks.txt"; //sætter en path til en .txt fil
    if (createText.Contains("123")) { Console.WriteLine("fejl password må ikke
indeholde 123"); return false; } //chekker password ikke har 123 i sig
    if (createText.Length < 12) { Console.WriteLine("fejl password er for kort");
return false; } //chekker om password er for kort
    for (int i = 0; i < createText.Length; i++) //laver en for løkke som køre hver
karakter i det nye password igennem
    {
        uppercase = Char.IsUpper(createText, i);
        if (uppercase == true) i = createText.Length + 1; //stopper løkken hvis
der er et stort bogstav i
    }

    for (int i = 0; i < createText.Length; i++) //nye løkke
    {
        lowercase = Char.IsLower(createText, i);
        if (lowercase == true) i = createText.Length + 1; //stopper løkken hvis
der er et lille bogstav i
    }

    if (lowercase == false) { Console.WriteLine("fejl password skal havde både
store og små bogstaver"); return false; }
    if (uppercase == false) { Console.WriteLine("fejl password skal havde både
store og små bogstaver"); return false; }

    bool isNumber = false;
    bool hasSpechialChars = false;
    Regex rgx = new Regex("[^A-Za-z0-9]"); //chekker om der er tal i det nye
password
    for (int i = 0; i < createText.Length; i++)
    {
        isNumber = int.TryParse(createText[i].ToString(), out int n);
        if (isNumber == true) i = createText.Length + 1;
    }
    for (int i = 0; i < createText.Length; i++) //chekker om der er spechiale tegn
i det nye password
    {
        hasSpechialChars = rgx.IsMatch(createText[i].ToString());
        if (hasSpechialChars == true) i = createText.Length + 1;
    }

    if (isNumber == false) { Console.WriteLine("fejl password skal indeholde
tal"); return false; }
```

```
        if (hasSpecialChars == false) { Console.WriteLine("fejl password skal indeholde special tegn"); return false; }

        isNumber = int.TryParse(createText[0].ToString(), out int a); //chekker om første tegn er et tal
        if (isNumber == true) { Console.WriteLine("fejl første tegn i password må ikke være et tal"); return false; }

        isNumber = int.TryParse(createText[createText.Length - 1].ToString(), out a); //chekker om andet tegn er et tal
        if (isNumber == true) { Console.WriteLine("fejl sidste tegn i password må ikke være et tal"); return false; }

        bool spaces = createText.Contains(" "); //chekker der mellemrum

        if (spaces == true) { Console.WriteLine("fejl password må ikke indeholde mellemrum, baka"); return false; }

        if (createText.ToLower() == Brugernavn) { Console.WriteLine("fejl password må ikke være det samme som brugernavn dumpap"); return false; } //chekker passwordet ikke er det samme som brugernavn

        if (createText == _password) { Console.WriteLine("fejl password må ikke være det samme som tidligere password"); return false; }; //chekker at password ikke er det samme som tidligere password

        File.WriteAllText(path, Brugernavn + Environment.NewLine + createText); //udskriver det nye password og brugernavn til .txt filen

        return true;
    }
}
```

## Kode for Fodbold opgave:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MyLibrary;

namespace Fodbold_Fan_Support_igen
{
    internal class Program
    {
        static void Main(string[] args)
        {
            bool gennemført = false; //her sætter vi en bool til at være falsk og for programmet til at køre indtill den er sat til sandt
            while(gennemført == false) {
                try
```



```
{
    Console.WriteLine("indtast antal afleveringer");//her skriver vi til
    brugeren at de skal indtaste afleveringer

    int afle = Convert.ToInt32(Console.ReadLine());//her gemmer vi
    antallet afleveringer som en int

    Console.WriteLine("indtast mål hvis der var mål");
    string møl = Console.ReadLine(); //her gemmer vi hvad brugeren skriver
    om mål som en string

    Afleveringer Af = new Afleveringer(afle,møl); //her skaber vi så et
    objekt med navn Af sin har værdierne fra tidligere

    Console.WriteLine(Af.Shout()); //her bruger vi så det nye objekt til
    at kalde en metode vi printer resultatet af.
    gennemført = true; //her siger vi så at programmet kan gå videre fra
    while løkken
}
catch (Exception)//catch er til hvis noget går galt, hvis for eksempel
programmet prøver at omdanne bogstaver til int32 format
{ Console.WriteLine("noget gik galt"); }} //her siger bare hvad der skal
ske hvis der går noget galt.

    Console.ReadKey();//denne readkey sikre os bare at programmet ikke lukker
    lige efter det blevet færdigt
}
}
```

## Kode for Danse opgave:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MyLibrary;

namespace Danse_Konkurrence
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int points = -100; //jeg starter med at sætte points til -100 da jeg ikke
            regner med man kan få minus points og jeg skal bruge det senere

            Console.WriteLine("indtast navn");
            string navn = Console.ReadLine(); //her sætter jeg en string til at
            være det som brugern skriver
            Console.WriteLine("indtast points");
            do //her starter jeg så en do while løkke som forsætter indtill points
            ikke er -100, dette er så brugeren kan blive ved med at forsøge
            {
```

```
        try //denne try catch i loppet er til hvis der går noget galt som at
omdanne et bogstav til int format
        {
            points = Convert.ToInt32(Console.ReadLine()); //her bliver points
sat til det brugern skriver
        }
        catch (Exception) //her siger jeg at hvis det tidligere kode slår
fejl så skal det under ske
        { Console.WriteLine("noget gik galt, intast venligst kun tal"); }
//skriver til brugen der gik noget galt
    } while (points == -100); //her er hvor jeg difinere hvor længe looped skal
køre, og det basicly indtill points er sat.

    DancerPoints D1 = new DancerPoints(navn,points); //her laver jeg så et
nyt objekt i dancerpoint classen og bruger de værdier fra tidligere

    Console.Clear(); //ryder bare skærmen så den ikke bliver for rodet
    points = -100; //sætter points til -100 så vi kan bruge den igen
    Console.WriteLine("indtast navn");
    navn = (Console.ReadLine());
    Console.WriteLine("indtast points"); //her gør jeg så det samme som
sidst med points og navn
    do
    {
        try
        {
            points = Convert.ToInt32(Console.ReadLine());
        }
        catch (Exception)
        { Console.WriteLine("noget gik galt, intast venligst kun tal"); }
    } while (points == -100);

    DancerPoints D2 = new DancerPoints( navn,points); //jeg sætter så de
nye værdier for points og navn i et andet objekt
    DancerPoints D3 = (D1 + D2); //til sidst plusser programet de 2
objekter sammen til en ny, og udskriver den
    Console.WriteLine(D3.I_alt());

    Console.ReadKey(); //er her for ikke at lukke programmet når det kørt
igennem.
    }
}

}
```

## Kode for Password opgave

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MyLibrary;
using System.IO;
using System.Text.RegularExpressions;

namespace Oprettelse_af_gyldigt_Password
{
```

```
internal class Oprettelse_af_gyldigt_Password
{
    static void Main(string[] args)
    {
        string path = @"C:\Users\rassør\Desktop\Visuel studio opgaver\teks.txt";
//her sætter jeg en path som programmet skal brug til at finde ud af hvor den finde en
bestemt fil

        string Brugernavn; //her initiliser jeg nogle strings jeg skal bruge
senere
        string Input1;
        string Input2;
        bool Password_oprettet = false; //her sætter jeg en bool til falsk, fordi
programmet skal chekke om den har ændret sig senere i koden.

        bool Bruger_oprettet = true; //samme som linje 21 men denne her skal
starte som true her skaber jeg et nyt objekt som jeg skal bruge senere
        Password_check PC = new Password_check("", "");

        if (!File.Exists(path))//denne if statement checker om der eksistere en fil
på den path jeg skrev tidligere
        {
            //her er hvad der sker hvis filen ikke eksistere
            Bruger_oprettet = false; //her sætter programmet bruger oprettet til
falsk fordi programet skal vide senere at den ikke skal logge ind men oprette en
bruger i stedet.
            Console.WriteLine("ingen bruger fundet opret venligst brugernavn:");
            Brugernavn = Console.ReadLine();
            Password_check ny = new Password_check(Brugernavn, "n"); //skaber et
nyt objekt med værdien fra brugeren.
            ny.Oprt_txt();//kalder en metode der opretter en .txt fil og putter
tekst ind i den
        }
        while (Bruger_oprettet == true)//denne løkke siger at hvis der allerede er
en bruger skal den blive ved med at køre.
        {
            Console.WriteLine("indtast brugernavn");
            Input1 = Console.ReadLine();

            Console.WriteLine("indtast password");
            Input2 = Console.ReadLine();

            if (!PC.Check(Input1 , Input2)) //kalder en metode som laver et
check baseret på input fra brugeren
            {
                Console.Clear();//fjerner tekst fra skærmen så den ikke er så
rodet
                Console.WriteLine("kode eller navn er forkert, prøv igen");//siger
til brugeren de skal prøve igen inden loopet starter igen
            }
            else
            {Bruger_oprettet = false; }//hvis inputtet fra brugeren klaret
check metoden så stopper loopet
        }

        Console.WriteLine("adgang givet");//lad brugeren vide at adgang er
givet
    }
}
```

```
        Console.WriteLine("opret nyt password og brugernavn?  J: ja , N: nej"); //lad brugen vide at de har et valg
        Input1 = Console.ReadLine();
        if (Input1.ToLower() == "j") //chekker om bruges input er j ligemeget
om det stor J eller lille j
        { //hvis det er j så går den videre ned og spørg brugen om at oprette
brugernavn og password.
        Console.WriteLine("opret nyt brugernavn");
        Brugernavn = Console.ReadLine(); //siden der ikke er krav omkring
brugernavn kan den bare være hvad brugeren skriver
        Console.Clear(); //rydder skærmen for tekst
        while (Password_oprettet == false) //dette loop sikre sig at det
næste stykke kode køre indtill password oprettet er true
        {
            Console.WriteLine("opret nyt password");
            string createText = Console.ReadLine();

            if (PC.Create_Pass(createText, Brugernavn) == true) //her
kalder jeg en metode for at se om passwordet brugen har valgt lever op til krævine
            {
                Console.WriteLine("password oprettet"); //hvis passwordet
lever op til krævine sætte programmet Password_oprettet til sandt og bryder loopet
                Password_oprettet = true;
            }
            else { Console.WriteLine("password møder ikke krav prøv
igen" + ""); } //hvis passwordet ikke møder kravne forsætter loopet.
        }

        Console.WriteLine("ses igen"); //en farvel besked :D
        Console.ReadKey(); //stopper programmet for at lukke når det er
kørt igennem
    }

}
```

## Konklusion

Jeg i løbet af dette projekt blevet bedre til at skrive afkoblet kode. Jeg også kommet i bedre bekendt skab med metoder og class. Det samme gælder for objekt orienteret kode. Jeg er har også fundet flere værktøjer til at finde information omkring kode og C Sharp.

## Opgaveforløb

15-11-2021-Mandag:

vi fik opgaven uden noget særligt instruction, så jeg lavet 1 og 2 uden afkoblet kode.

16-11-2021-Tirsdag:

Det var alt sammen forkert så skulle lave det hele om. Jeg fik læst en masse om afkoblet kode og hvordan jeg kan bruge det. Ud over det fik jeg løst opgave 1 og er næsten færdig med opgave 2.

17-11-2021-Onsdag:

Jeg fik læst op på operator overlaod og blev færdig med opgave 2. Opgave 3 er en del svære med at skrive men jeg er nogle lunde færdig med den. Der er dog stadige lidt fejl og bugs der skal rettes

18-11-2021-torsdag:

Jeg har rettet en del på opgave 3 så nu er jeg begyndt på at skrive rapport. Jeg har skrevet en MASSE kommentere da lære var meget efter os omkring det på Frediksberg. Det viser sig dog at jeg overhovedet ikke behøvet så mange kommentere.

19-11-2021-fredag:

Jeg har skrevet rapporten færdig.

Figur 1 Opsætning

4