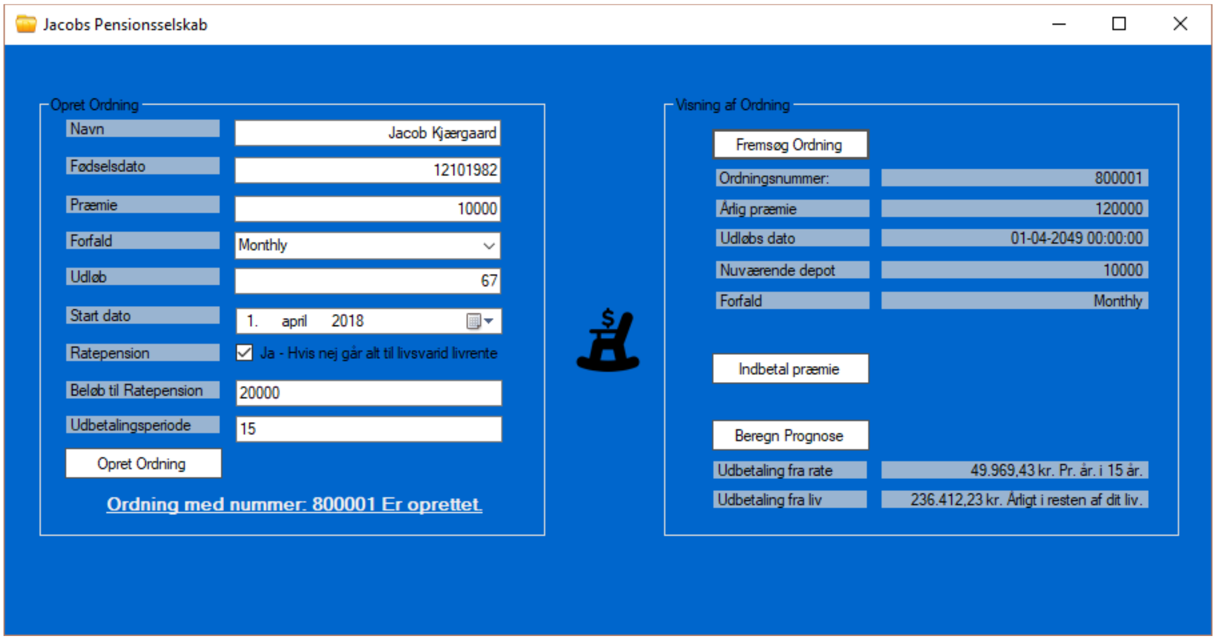
Implementering af en Windows Form for Jacobs Pensionsselskab A/S



Navn	Jacob Kjærgaard
Fødselsdato	12101982
Præmie	10000
Forfald	Monthly
Udløb	67
Start dato	1. april 2018
Ratepension	☑ Ja - Hvis nej går alt til livsvarid livrente
Beløb til Ratepension	20000
Udbetalingsperiode	15

Ordning med nummer: 800001 Er oprettet.

Ordningsnummer:	800001
Årlig præmie	120000
Udløbs dato	01-04-2049 00:00:00
Nuværende depot	10000
Forfald	Monthly
Udbetaling fra rate	49.969,43 kr. Pr. år. i 15 år.
Udbetaling fra liv	236.412,23 kr. Årligt i resten af dit liv.

Forfatter: Jacob Kjærgaard

Dato: 20-04-2018

Version: 1.0

Indhold

Kapitel 1

Jacobs Pensionsselskab er et program til at lave en pensionsordning, se data om ordningen, indbetal et beløb på pensionsordningen og se den forventede udbetaling.

1. Baggrund

Programmet laves som en obligatorisk afleveringsopgave i faget videregående programmering.

2. Problemstilling

For at bruge de tillærte begreber i faget videregående programmering, laves et pensions program, hvor elementer fra faget bliver brugt.

Selve programmet har 4 hoved elementer:

1. Oprettelse af en ordning -> herved instanciering af klasser, der bruges i det følgende.
2. Visning af data -> indhent data fra objekter og vis dem på brugergrænsefladen.
3. Placer en indbetaling på ordning -> brug command og query metoder til at få indformation til at placere indbetalingen på den rigtige opsparingsform.
4. Lav prognose -> bygger videre på punkt 3, men viser resultatet til brugeren.

Programmet er lavet, så der kan laves en pensionsordning med én opsparingsform (livsvarig livrente), og der kan vælges at have 2 typer (ratepension og livsvarig livrente). Der er lovgivningsgrænse på, hvornår indbetalingen skal gå fra den en opsparing til den anden og kun kan vælge en anden grænse.

Udover at holde styr på placering af penge på depotet, skal denne data også gemmes, da indbetalingen skal indberettes til skat med en markering af hvad der er gået til ratepension og hvad der er gået til livsvarig livrente.

Kapitel 2

Der har eksisteret pensionsprogrammer mange år før OOP så dagens lys, så det er muligt at lave som procedural, men en object orienteret tilgang virker bedre til at placere logik og information om et enkelt område et sted og samtidig gøre det mere flexibelt når staten igen ændre reglerne på området.

1. Afgrænsning

Pensionsområdet er et stort og kompleks område, hvor der er mange faktorer, som kunne være medtaget. Eksempelvis kunne implementering af en database give stor værdi, så man bedre kunne gemme data. Et lille udvalg af emner, der kunne implementeres: firmaaftaler, forsikringsdækning, investeringsvalg, skattefrie opsparing mv.

For at holde fokus på struktur vedr. klasser og logikken her, er exception handling og unittests blevet reduceret.

Der er også valgt at lave nogle "lette" udveje på enkelte områder for at komme i mål indenfor tidsgrænsen – eksempelvis beregning af prognose sker kun i hele antal år.

2. Dataopsamling

Dataopsamling består stortset af de felter, som der er på skærmen i forhold til at oprette ordningen. Der er dog valgt at lave et singelton pattern til at generer et pensionsnummer, så det bliver unikt og fortløbende.

For at behandle placering af penge på korrekte depoter, har det været nødvendigt at gemme flere data om denne præmie betaling.

3. DataStorage

Da det er valgt at afgrænse fra brugen af database, betyder det at data skal gemmes i variabler, hvilket har den betydning af pt. Overskriver en ny ordning den gamle. Hvis man havde tilknyttet en database, vil man kunne oprette et vilkårligt antal ordninger og fremsøge data om disse efter ønske.

4. Teknologi valg

Det er valgt at bruge C#, windows form application og bruge VS som IDE.

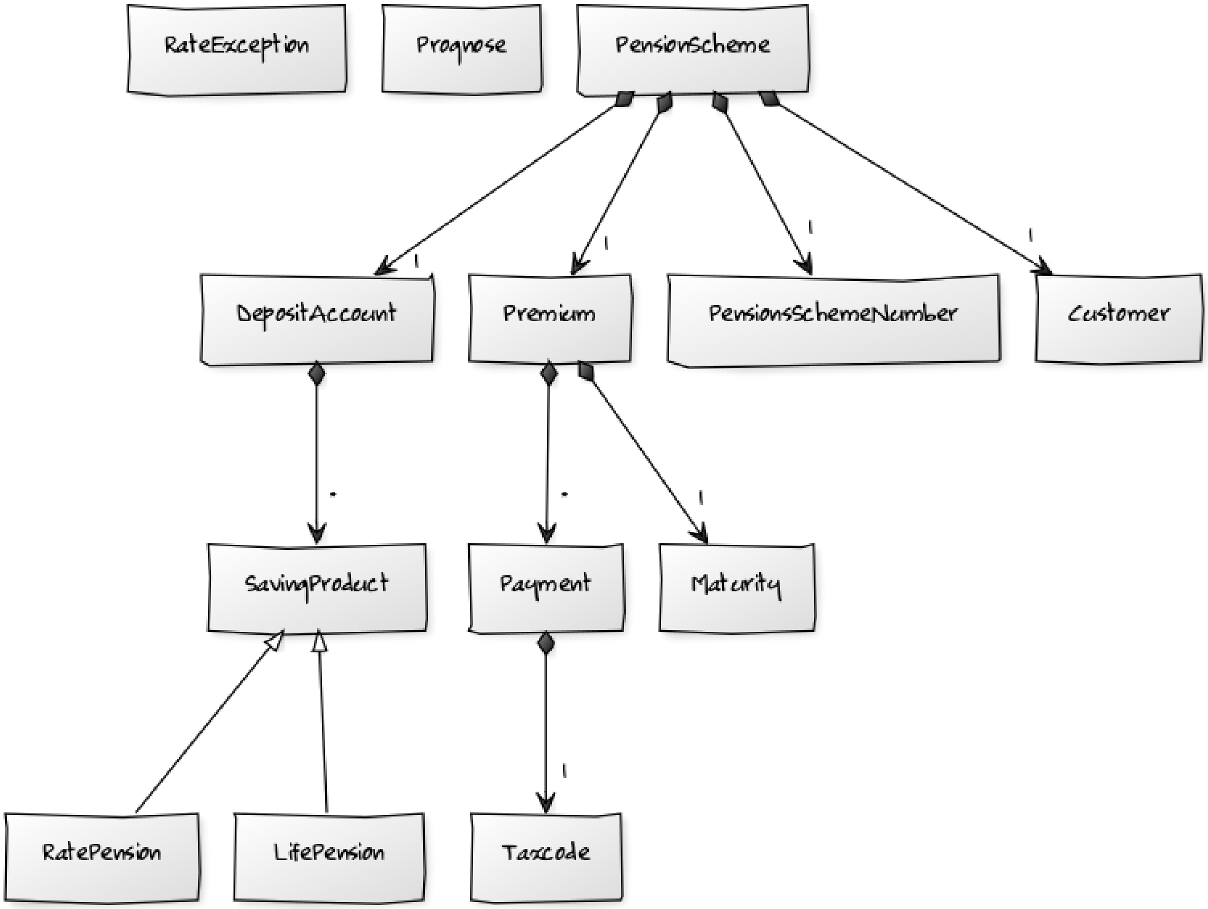
5. Teststrategi

Der er lavet et separat unit test project, hvor der er lavet unit test på enkelte metoder, men der er blevet afgrænset at bruge al for meget tid her, da programmet ikke bliver brugt og det kun skal beskrives i denne rapport.

Der er lavet manuel test for at sandsynliggøre at programmet fungere efter hensigten i alle henseender. Der er fundet flere bugs i processen, som er løst med det samme.

Kapitel 3 Design

Det er forsøgt at lave et UML diagram, for at vise strukturen af programmet.



Koden er vedlagt

1. Valg af klasser

Klasser kan ses ud af UML diagrammet.

2. Valg af lister / andet

Det er valgt at Premium skal holde en liste af payment objecter. Dette er ud fra en større betragtning om. At transaktioner vil der komme rigtig mange af, og ved at lave det som en liste af strukts flytter man nogle data til transaktions strukt payment frem for at mudre premium med for mange data og registreringer. Strukts er gode til at opbevare mange instancieringer, når der er en lille datamængde i hver.

Det er også valgt at man på pensionsordningen har en liste af opsparingsprodukter. Dette er for at øge flexibiliteten, så man kan tilføje aldersopsparing, kapitalpension, index kontrakter mv.

Generelt er det forsøgt at lave en loose coupled struktur, hvor det er muligt. Dog er PensionsSheme klassen tight couplet via komposition til premium, opsparingsprodukter mv. – Men da der har været koblet indbetaling og opsparingsprodukter til pensionsordninger siden de blev skabt og ikke vil give så meget mening uden i hvert fald en abstrakt form for præmie eller pensionsprodukt er det valgt som en farbar vej.

Kapitel 4 Programmering

Der er lavet mange klasser og enkelte udvalgte gennemgås i nedenstående – hele koden er vedlagt til sidst i rapporten.

1. Klasse PensionsSchemeNumber

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

namespace PensionProgram
{
            public sealed class PensionsSchemeNumber
            {
                        private int _nextNumber = 800001;
                        private static readonly PensionsSchemeNumber instance = new
PensionsSchemeNumber();

                        private PensionsSchemeNumber()
                        {
                        }

                        public static PensionsSchemeNumber Instance
                        {
                                    get
                                    {
                                                return instance;
                                    }
                        }

                        public int GetNextNumber()
                        {
                                    int OccupiedNumber = _nextNumber;
                                    _nextNumber++;
                                    return OccupiedNumber;
                        }


            }
}
```

En singelton klasse, hvor der vælges at instanciere variablen med det samme fremfor lazyload metoden, da man ved at der skal laves et objekt af denne klasse, da den skal bruges ved oprettelse af en ordning.

2. Klasse Paymeny

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

namespace PensionProgram
{
            public struct Payment
            {
                        private DateTime _date;
                        private double _amount;
                        private Taxcode _taxcode;

                        public double Amount
                        {
                                    get { return _amount; }
                                    set { _amount = value; }
                        }

                        public Taxcode Taxcode
                        {
                                    get { return _taxcode; }
                                    set { _taxcode = value; }
                        }

                        public Payment(DateTime date, double amount, Taxcode taxcode)
                        {
                                    this._date = date;
                                    this._amount = amount;
                                    this._taxcode = taxcode;
                        }
            }
}
```

Her har jeg valgt at lave en struct, da der i et pensionssystem vil være mange hundrede tusinde indbetalinger hver måned, og jf. MSDN retningslinjer er struct en optimal klasse at bruge når objektet skal indeholder lidt data og der vil være rigtig mange instancieringer af klassen/struct.

Kapitel 5

Som nævnt indledningsvis, har fokus ikke været på en driftsikker application, da den ikke forventes brugt på nuværende tidspunkt og er meget brugt til at afprøve teorien. For også at afprøve unit test principper, er der oprette et unit test projekt, hvor resultatet kan ses af nedenstående.

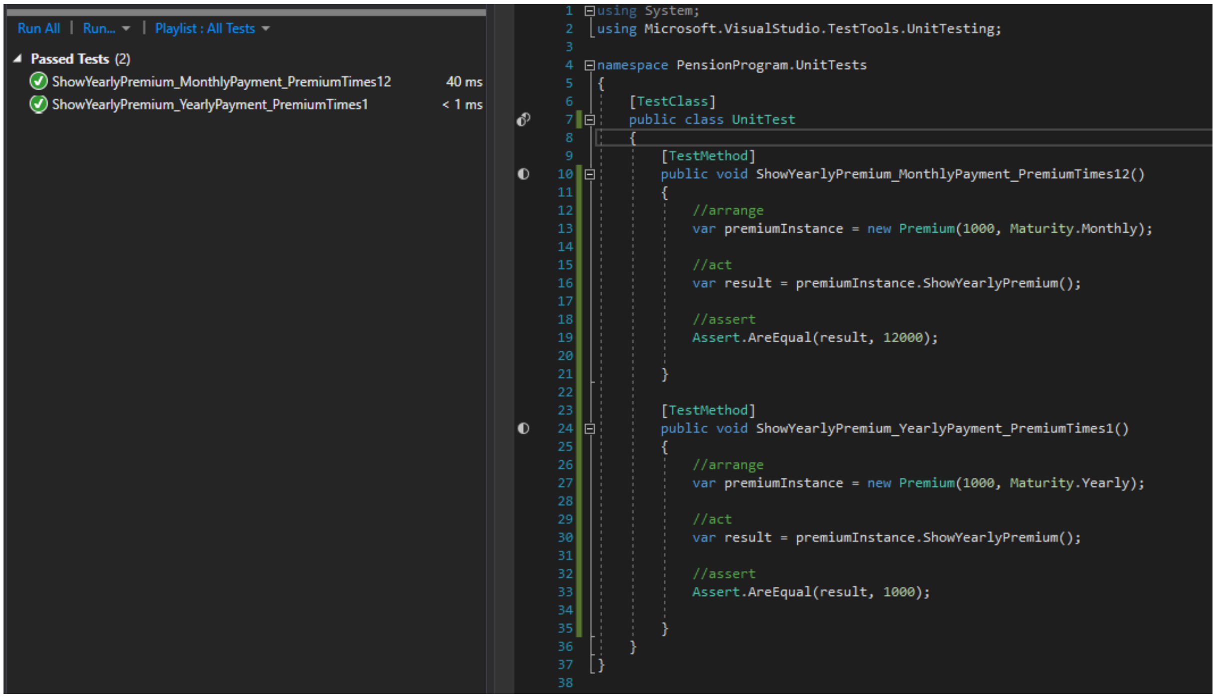
1. Unit test kode

Hele koden er ikke vedlagt, men i det nedenstående vises et eksempel, hvor fremgangsmåden har været, arrange act og assert. Testcase navnet er opbygget efter MetodeNavn_ScenarieDerTestes_ForventetResultat.

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace PensionProgram.UnitTests
{
            [TestClass]
            public class UnitTest
            {
                        [TestMethod]
                        public void ShowYearlyPremium_MonthlyPayment_PremiumTimes12()
                        {
                                    //arrange
                                    var premiumInstance = new Premium(1000, 
Maturity.Monthly);

                                    //act
                                    var result = premiumInstance.ShowYearlyPremium();

                                    //assert
                                    Assert.AreEqual(result, 12000);

                        }

                        [TestMethod]
                        public void ShowYearlyPremium_YearlyPayment_PremiumTimes1()
                        {
                                    //arrange
                                    var premiumInstance = new Premium(1000, 
Maturity.Yearly);

                                    //act
                                    var result = premiumInstance.ShowYearlyPremium();

                                    //assert
                                    Assert.AreEqual(result, 1000);

                        }
            }
}
```

2. Screenshot af gennemførelse samt kommenter resultat

Når test suiten køres ser det således ud:



```
Passed Tests (2)
  ShowYearlyPremium_MonthlyPayment_PremiumTimes12    40 ms
  ShowYearlyPremium_YearlyPayment_PremiumTimes1      < 1 ms
```

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace PensionProgram.UnitTests
{
    [TestClass]
    public class UnitTest
    {
        [TestMethod]
        public void ShowYearlyPremium_MonthlyPayment_PremiumTimes12()
        {
            //arrange
            var premiumInstance = new Premium(1000, Maturity.Monthly);

            //act
            var result = premiumInstance.ShowYearlyPremium();

            //assert
            Assert.AreEqual(result, 12000);

        }

        [TestMethod]
        public void ShowYearlyPremium_YearlyPayment_PremiumTimes1()
        {
            //arrange
            var premiumInstance = new Premium(1000, Maturity.Yearly);

            //act
            var result = premiumInstance.ShowYearlyPremium();

            //assert
            Assert.AreEqual(result, 1000);

        }
    }
}
```

3. Konklusion på test.

Testen har tjent sit formål i det, at applikationen virker til shiny day eksemplet. Da der ikke er brugere der skal udfylde applikationen, har der ikke været fokus på at lave en større validering af input.

Kapitel 6

Opgaven viste sig at være mere omfattende end jeg oprindeligt forventede og tog flere timer at lave, end jeg egentlig havde til rådighed. Det betød også at koden ikke er så ”køn” og så velstruktureret som jeg ønskede.

Jeg havde også en fundamentalt udfordring med at komme på en god måde strukturere at indbetalingen både påvirkede min præmie klasse, for at holde styr på indbetalingen på skattekoder samtidigt med at jeg skulle placere indbetalingen på depotet for at øge værdien af opsparingen.

Jeg endte med en løsning, der virker, men kunne forstiller mig at det var et områder, der kunne forbedres.

Programmet blev også mere tightly koblet end jeg egentlig havde som udgangspunkt. Ville oprindeligt bruge interfaces til at koble opsparingsprodukter til pensionsordningen, men da jeg faktisk skulle bruge data retur fra metoden, og interfaces kun kan bruges med void metoder, måtte jeg bruge nedarvning i stedet.

1. Udvidelser Applikation kan udvides med ....

Jeg vil gerne arbejde videre med dette program – evt. bruge det til afgangsprojekt, hvor det kunne udvides med database adgang (via Linq), der kunne implementeres en aftale struktur, en aftalepart (firma/privat), der kunne tilføjes forsikringsprodukter, investeringsvalg, indberetning til skat, håndtering af AMB mv. – Herudover kunne nærmest alle dele gøres lidt bedre – eksempelvis prognosen, kunne bruge faktisk forsikringsmatematik til at beregne en forventet udbetaling.

--Customer--

```
using System;

namespace PensionProgram
{
    public class Customer
    {
        public string Name { get; }
        public string Fødselsdato { get; }

        public Customer(string name, string fødselsdato)
        {
            Name = name;
            Fødselsdato = fødselsdato;
        }

        public double Age()
        {
            int year = Convert.ToInt32(Fødselsdato.Substring(4));
            int month = Convert.ToInt32(Fødselsdato.Substring(2, 2));

            int currentYear = DateTime.Now.Year;
            int currentMonth = DateTime.Now.Month;

            string now = "" + currentYear + "," + currentMonth;
            string born = "" + year + "," + month;

            double age = Convert.ToDouble(now) - Convert.ToDouble(born);

            return age;
        }
    }
}
```

--DepositAccount--

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


namespace PensionProgram
{
    class DepositAccount
    {
        private readonly List<SavingProduct> _savingProducts;
        private bool _thereIsRatePension = false;

        public DepositAccount()
        {
            _savingProducts = new List<SavingProduct>();
            _savingProducts.Add(new LifePension());
        }

        public void AddRatePension(int maxTax2, int defaultPayoutPeriode = 10)
        {
            _savingProducts.Add(new RatePension(maxTax2,defaultPayoutPeriode));
            _thereIsRatePension = true;
        }

        public bool ThereIsRatePension
        {
            get { return _thereIsRatePension; }
        }

        public int Max2Amount()
        {
            int amount = 0;
            for (var i = 0; i < _savingProducts.Count; i++)
            {
                amount += _savingProducts[i].MaxOnRate();
            }
            return amount;
        }

        public double GetCurrentValue()
        {
            double life = _savingProducts[0].CurrentDeposit();
            if (_savingProducts.Count<2)
            {
                return life;
            }

            else
            {
                return life + _savingProducts[1].CurrentDeposit();
            }
        }

        public void AddPaymentToDepot(double payment, Taxcode taxcode)
        {
            switch (taxcode)
            {
                case Taxcode.Ratepension:
                    for (var i = 0; i < _savingProducts.Count; i++)
                    {
                        if (_savingProducts[i].HaveRate()==true)
                            _savingProducts[i].AddDeposit(payment);
                    }

                    break;
                case Taxcode.Livrente:
                    for (var i = 0; i < _savingProducts.Count; i++)
                    {
                        if (_savingProducts[i].HaveRate()==false)
                            _savingProducts[i].AddDeposit(payment);
                    }

                    break;
            }

        }

        public double GetCurrentLife()
        {
            return _savingProducts[0].CurrentDeposit();
        }

        public double GetCurrentRate()
        {
            double depositRate = 0;
            for (var i = 0; i < _savingProducts.Count; i++)
            {
                if (_savingProducts[i].HaveRate() == true)
                    depositRate += _savingProducts[i].CurrentDeposit();
            }
            return depositRate;
        }
    }
}
```

--Life Pension---

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

namespace PensionProgram
{
    class LifePension : SavingProduct
    {
        private double _deposit;
        public const double Omregningsfaktor = 15.554449;

        public override int MaxOnRate()
        {
            return 0;
        }

        public override bool HaveRate()
        {
            return false;
        }
    }
}
```

--Maturity--

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

namespace PensionProgram
{
    public enum Maturity
    {
        Monthly,
        Yearly
    };
}
```

--Payment--

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

namespace PensionProgram
{
     public struct Payment
     {
          private DateTime _date;
          private double _amount;
          private Taxcode _taxcode;

          public double Amount
          {
               get { return _amount; }
               set { _amount = value; }
          }

          public Taxcode Taxcode
          {
               get { return _taxcode; }
               set { _taxcode = value; }
          }

          public Payment(DateTime date, double amount, Taxcode taxcode)
          {
               this._date = date;
               this._amount = amount;
               this._taxcode = taxcode;
          }
     }
}
```

--PensionScheme--

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

namespace PensionProgram
{
    class PensionScheme
    {
        public int PensionsSchemeNumber { get; }
        public Customer Customer { get; }
        public DepositAccount DepositAccount { get; }
        public Premium Premium { get; }
        public int ExpirationYear { get; set; }
        public DateTime StartDate { get; }

        public PensionScheme(Customer customer, DepositAccount depositAccount, Premium
        premium, int expirationYear, DateTime StartDate, PensionsSchemeNumber
        pensionsSchemeNumber)
        {
            this.PensionsSchemeNumber = pensionsSchemeNumber.GetNextNumber();
            this.Customer = customer;
            this.DepositAccount = depositAccount;
            this.Premium = premium;
            this.ExpirationYear = expirationYear;
            this.StartDate = StartDate;
        }

        public double ShowYearlyPremium()
        {
            return Premium.ShowYearlyPremium();
        }

        public DateTime ShowExpirationDate()
        {
            return StartDate.AddYears((int)ExpirationYear-(int)Customer.Age());
        }

        public double ShowDepositAccount()
        {
            return DepositAccount.GetCurrentValue();
        }
    }
}
```

--PensionsSchemeNumber--

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

namespace PensionProgram
{
    public sealed class PensionsSchemeNumber
    {
        private int _nextNumber = 800001;
        private static readonly PensionsSchemeNumber instance = new PensionsSchemeNumber();

        private PensionsSchemeNumber()
        {
        }

        public static PensionsSchemeNumber Instance
        {
            get
            {
                return instance;
            }
        }

        public int GetNextNumber()
        {
            int OccupiedNumber = _nextNumber;
            _nextNumber++;
            return OccupiedNumber;
        }


    }
}
```

--Premium--

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

namespace PensionProgram
{
     public class Premium
     {
          private Maturity _maturity;
          private double _premium;
          private readonly List<Payment> _payments = new List<Payment>();

          public Premium(double premiumPrMaturity, Maturity maturity)
          {
               this._premium = premiumPrMaturity;
               this.Maturity = maturity;
          }

          public List<Payment> Payments
          {
               get { return _payments; }
          }

          public Maturity Maturity { get => _maturity; set => _maturity = value; }

          public void AddPayment(Payment payment) => _payments.Add(payment);

          public double ShowYearlyPremium()
          {
               if (Maturity == Maturity.Yearly)
               {
                    return _premium;
               }
               else
               {
                    return _premium * 12;
               }
          }

          public double PaymentsOnTaxCode(Taxcode taxcode)
          {
               double sum = 0;
               for (var i = 0; i < _payments.Count; i++)
               {
                    if (_payments[i].Taxcode == taxcode)
                    {
                         sum += _payments[i].Amount;
                    }
               }
               return sum;
          }

     }
}
```

---Prognose---

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

namespace PensionProgram
{
    public abstract class Prognose
    {
        public const double prognoseRate = 0.01112;

    }
}
```

--Program---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PensionProgram
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new PensionCreation());
        }
    }
}
```

--RateException--

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

namespace PensionProgram
{
    public class RateException : Exception
    {

        public RateException()
        {
        }

        public RateException(string message)
            : base(message)
        {
        }

        public RateException(string message, Exception inner)
            : base(message, inner)
        {
        }
    }
}
```

---RatePension---

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

namespace PensionProgram
{
    class RatePension : SavingProduct
    {
        private int _maxTax2ByLaw = 50000;
        private int _maxTax2;
        private int _defaultPayoutPeriode = 10;
        private double _deposit;

        public int MaxTax2ByLaw { get; private set; }

        public RatePension(int maxTax2, int defaultPayoutPeriode = 10)
        {
            MaxTax2 = maxTax2;
            this._defaultPayoutPeriode = defaultPayoutPeriode;
            this._deposit = 0;
        }

        public int DefaultPayoutPeriode
        {
            get { return _defaultPayoutPeriode; }
            set { _defaultPayoutPeriode = value; }
        }

        public int MaxTax2
        {
            get{return _maxTax2;}
            set
            {
                if (value < _maxTax2ByLaw)
                {
                    _maxTax2 = value;
                } else
                {
                   throw new RateException("Du kan ikke sætte en grænse på Ratepension der
                   overstiger: " + _maxTax2ByLaw);
                }
            }
        }

        public override int MaxOnRate()
        {
            return _maxTax2;
        }

        public override bool HaveRate()
        {
            return true;
        }
    }
}
```

--SavingProduct--

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

namespace PensionProgram
{
    public abstract class SavingProduct
    {
        private double _deposit;

        public virtual double CurrentDeposit()
        {
            return _deposit;
        }

        public virtual void AddDeposit(double payment)
        {
            _deposit += payment;
        }

        public abstract int MaxOnRate();

        public abstract bool HaveRate();

    }
}
```

---Taxcode---

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

namespace PensionProgram
{
    public enum Taxcode
    {
        Livrente = 1,
        Ratepension = 2
    };
}
```

```
------------------------------------------------------------Form1-------------------------------------------------------

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PensionProgram
{
    public partial class PensionCreation : Form
    {
        private string customerName;
        private string birthdate;
        private double premiumPrMaturity;
        private Maturity maturity;
        private int expirationYear;
        private int showingPensionScheme;
        private Customer customer;
        private PensionScheme pensionScheme;
        private DateTime startDate;
        private bool isthereRate;
        private int rateYear = 10;
        private int maxRateYear = 0;

        public PensionCreation()
        {
            InitializeComponent();
            maturityComboBox.DataSource = Enum.GetValues(typeof(Maturity));
        }

        private void button1_Click(object sender, EventArgs e) //opret ordning
        {
            try
            {
                customerName = Convert.ToString(customerNameTextBox.Text);
                birthdate = Convert.ToString(FødselsDatoTextBox.Text);
                premiumPrMaturity = Convert.ToInt32(premiumTextBox.Text);
                maturity = (Maturity)maturityComboBox.SelectedItem;
                expirationYear = Convert.ToInt32(expiryTextBox.Text);
                startDate = startDateSelector.Value;
                isthereRate = (isThereRatePensionCheckBox.Checked == true) ? true : false;
                if (isthereRate)
                {
                    rateYear = Convert.ToInt32(rateYearTextBox.Text);
                    maxRateYear = Convert.ToInt32(maxRateTextBox.Text);
                }
            }
            catch (Exception)
            {
                MessageBox.Show("Der er fejl i indtastningen", "Fejl i indtastning",
                MessageBoxButtons.OK);
            }


            //Opret Kunde
            try
            {
                customer = new Customer(customerName, birthdate);
            }
            catch (Exception)
            {
                MessageBox.Show("Der er fejl i navn eller fødselsdag", "Fejl i navn eller
                fødselsdag", MessageBoxButtons.OK);
                throw;
            }

            //Opret ordning
            try
            {
                pensionScheme = new PensionScheme(customer,
                    new DepositAccount(),
                    new Premium(premiumPrMaturity, maturity),
                    expirationYear,
                    startDate,
                    PensionsSchemeNumber.Instance);
                    CreationLabel.Text = "Ordning med nummer: " +
                    pensionScheme.PensionsSchemeNumber + " Er oprettet.";

                if (isthereRate == true)
                {
                    pensionScheme.DepositAccount.AddRatePension(maxRateYear, rateYear);
                }
            }
            catch (Exception)
            {
                MessageBox.Show("Der er fejl i pensionsinfo", "Fejl i pensionsinfo",
                MessageBoxButtons.OK);
                throw;
            }


        }

        private void SeekInfoButton_Click(object sender, EventArgs e)
        {
            //Fremsøg ordning
            ShowSchemeNumberLabel.Text = pensionScheme.PensionsSchemeNumber.ToString();
            ShowYearlyPremiumLabel.Text = pensionScheme.ShowYearlyPremium().ToString();
            ShowExpiryDateLabel.Text = pensionScheme.ShowExpirationDate().ToString();
            ShowCurrentDepositLabel.Text = pensionScheme.ShowDepositAccount().ToString();
            ShowMaturityLabel.Text = pensionScheme.Premium.Maturity.ToString();
        }

        private void PremiumPaymentButton_Click(object sender, EventArgs e)
        {
            double paidOnTax2 =
            pensionScheme.Premium.PaymentsOnTaxCode(Taxcode.Ratepension);
            bool RateOnScheme = pensionScheme.DepositAccount.ThereIsRatePension;
            int max2 = pensionScheme.DepositAccount.Max2Amount();

            if (paidOnTax2 >= max2)
            {
                  pensionScheme.DepositAccount.AddPaymentToDepot(premiumPrMaturity,
                  Taxcode.Livrente);
```

```
           pensionScheme.Premium.AddPayment(new Payment(DateTime.Now,
           premiumPrMaturity, Taxcode.Livrente));
     }
     else if (paidOnTax2<max2)
     {
           if (premiumPrMaturity < (max2 - paidOnTax2))
           {
              pensionScheme.DepositAccount.AddPaymentToDepot(premiumPrMaturity,
              Taxcode.Ratepension);
              pensionScheme.Premium.AddPayment(new Payment(DateTime.Now,
              premiumPrMaturity, Taxcode.Ratepension));
           }
           else
           {
                pensionScheme.DepositAccount.AddPaymentToDepot((max2-paidOnTax2),
                Taxcode.Ratepension);
                pensionScheme.Premium.AddPayment(new Payment(DateTime.Now, (max2 -
                paidOnTax2), Taxcode.Ratepension));
                pensionScheme.DepositAccount.AddPaymentToDepot(premiumPrMaturity -
                (max2 - paidOnTax2), Taxcode.Ratepension);
                pensionScheme.Premium.AddPayment(new Payment(DateTime.Now,
                premiumPrMaturity - (max2 - paidOnTax2), Taxcode.Ratepension));
           }

     }
}

private void isThereRatePensionCheckBox_CheckedChanged(object sender, EventArgs e)
{
     if (isThereRatePensionCheckBox.Checked)
           {
                maxRateTextBox.Enabled = true;
                rateYearTextBox.Enabled = true;
           }
           else
           {
                maxRateTextBox.Enabled = false;
                rateYearTextBox.Enabled = false;
                maxRateTextBox.Text = "";
                rateYearTextBox.Text = "";
           }
}

private void prognoseButton_Click(object sender, EventArgs e)
{
     //Prognose
     double tax1Deposit = pensionScheme.DepositAccount.GetCurrentLife();
     double tax2Deposit = pensionScheme.DepositAccount.GetCurrentRate();

     int yearsToRetirement = expirationYear - (int)customer.Age();

     for (var i = 0; i < yearsToRetirement; i++)
     {
              if (pensionScheme.Premium.ShowYearlyPremium() >
              pensionScheme.DepositAccount.Max2Amount())
           {
           tax1Deposit += pensionScheme.Premium.ShowYearlyPremium()-
           pensionScheme.DepositAccount.Max2Amount() + (tax1Deposit *
           Prognose.prognoseRate);
```

```
                tax2Deposit += pensionScheme.DepositAccount.Max2Amount() + (tax2Deposit *
                Prognose.prognoseRate);
                }
                else
                {
                   tax2Deposit += pensionScheme.Premium.ShowYearlyPremium() + (tax2Deposit
                   * Prognose.prognoseRate);
                }
            }

            if (tax2Deposit > 0)
            {
                string tax2Prognosis = string.Format("{0:N2} kr.", (tax2Deposit /
                rateYear));
                ShowRatePrognosisLabel.Text = "" + tax2Prognosis + " Pr. år. i " +
                rateYear + " år.";
            }
            else
            {
                ShowRatePrognosisLabel.Text = "-";
            }

            if (tax1Deposit > 0)
            {
                string tax1Prognosis = string.Format("{0:N2} kr.", (tax1Deposit /
                LifePension.Omregningsfaktor));
                ShowLifePrognosisLabel.Text = "" + tax1Prognosis + " Årligt i resten af
                dit liv.";
            }
            else
            {
                ShowLifePrognosisLabel.Text = " ";
            }
        }
    }
}
```

```
------------------------------------------------------------Form1.Designer-------------------------------------------------------
namespace PensionProgram
{
    partial class PensionCreation
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(PensionCreation));
            this.label1 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.label11 = new System.Windows.Forms.Label();
            this.label13 = new System.Windows.Forms.Label();
            this.label15 = new System.Windows.Forms.Label();
            this.maturityComboBox = new System.Windows.Forms.ComboBox();
            this.startDateSelector = new System.Windows.Forms.DateTimePicker();
            this.isThereRatePensionCheckBox = new System.Windows.Forms.CheckBox();
            this.createSchemeButton = new System.Windows.Forms.Button();
            this.prognoseButton = new System.Windows.Forms.Button();
            this.label4 = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label10 = new System.Windows.Forms.Label();
            this.SeekInfoButton = new System.Windows.Forms.Button();
            this.label2 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.label12 = new System.Windows.Forms.Label();
            this.label14 = new System.Windows.Forms.Label();
            this.ShowMaturityLabel = new System.Windows.Forms.Label();
            this.ShowCurrentDepositLabel = new System.Windows.Forms.Label();
            this.ShowExpiryDateLabel = new System.Windows.Forms.Label();
            this.ShowSchemeNumberLabel = new System.Windows.Forms.Label();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.rateYearTextBox = new System.Windows.Forms.TextBox();
            this.maxRateTextBox = new System.Windows.Forms.TextBox();
            this.expiryTextBox = new System.Windows.Forms.TextBox();
            this.premiumTextBox = new System.Windows.Forms.TextBox();
            this.FødselsDatoTextBox = new System.Windows.Forms.TextBox();
            this.customerNameTextBox = new System.Windows.Forms.TextBox();
            this.CreationLabel = new System.Windows.Forms.Label();
            this.label17 = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.PremiumPaymentButton = new System.Windows.Forms.Button();
            this.ShowLifePrognosisLabel = new System.Windows.Forms.Label();
            this.ShowRatePrognosisLabel = new System.Windows.Forms.Label();
            this.ShowYearlyPremiumLabel = new System.Windows.Forms.Label();
            this.label21 = new System.Windows.Forms.Label();
            this.groupBox1.SuspendLayout();
            this.groupBox2.SuspendLayout();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.BackColor = System.Drawing.SystemColors.ActiveCaption;
            this.label1.Location = new System.Drawing.Point(46, 56);
            this.label1.MinimumSize = new System.Drawing.Size(115, 10);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(115, 13);
            this.label1.TabIndex = 0;
            this.label1.Text = "Navn";
            //
            // label3
            //
```

```
this.label3.AutoSize = true;
this.label3.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label3.Location = new System.Drawing.Point(46, 113);
this.label3.MinimumSize = new System.Drawing.Size(115, 10);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(115, 13);
this.label3.TabIndex = 2;
this.label3.Text = "Præmie";
//
// label5
//
this.label5.AutoSize = true;
this.label5.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label5.Location = new System.Drawing.Point(46, 167);
this.label5.MinimumSize = new System.Drawing.Size(115, 10);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(115, 13);
this.label5.TabIndex = 6;
this.label5.Text = "Udløb";
//
// label7
//
this.label7.AutoSize = true;
this.label7.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label7.Location = new System.Drawing.Point(46, 140);
this.label7.MinimumSize = new System.Drawing.Size(115, 10);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(115, 13);
this.label7.TabIndex = 4;
this.label7.Text = "Forfald";
```

```
//
// label11
//
this.label11.AutoSize = true;
this.label11.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label11.Location = new System.Drawing.Point(46, 252);
this.label11.MinimumSize = new System.Drawing.Size(115, 10);
this.label11.Name = "label11";
this.label11.Size = new System.Drawing.Size(115, 13);
this.label11.TabIndex = 12;
this.label11.Text = "Beløb til Ratepension";
//
// label13
//
this.label13.AutoSize = true;
this.label13.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label13.Location = new System.Drawing.Point(46, 224);
this.label13.MinimumSize = new System.Drawing.Size(115, 10);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(115, 13);
this.label13.TabIndex = 10;
this.label13.Text = "Ratepension";
//
// label15
//
this.label15.AutoSize = true;
this.label15.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label15.Location = new System.Drawing.Point(46, 196);
this.label15.MinimumSize = new System.Drawing.Size(115, 10);
this.label15.Name = "label15";
this.label15.Size = new System.Drawing.Size(115, 13);
this.label15.TabIndex = 8;
this.label15.Text = "Start dato";
//
// maturityComboBox
//
this.maturityComboBox.FormattingEnabled = true;
this.maturityComboBox.ItemHeight = 13;
this.maturityComboBox.Location = new System.Drawing.Point(146, 102);
this.maturityComboBox.MinimumSize = new System.Drawing.Size(200, 0);
this.maturityComboBox.Name = "maturityComboBox";
this.maturityComboBox.Size = new System.Drawing.Size(200, 21);
this.maturityComboBox.TabIndex = 16;
//
// startDateSelector
//
this.startDateSelector.Location = new System.Drawing.Point(147, 159);
this.startDateSelector.MaxDate = new System.DateTime(2018, 12, 31, 0, 0, 0, 0);
this.startDateSelector.MinDate = new System.DateTime(2018, 1, 1, 0, 0, 0, 0);
this.startDateSelector.Name = "startDateSelector";
this.startDateSelector.Size = new System.Drawing.Size(200, 20);
this.startDateSelector.TabIndex = 17;
this.startDateSelector.Value = new System.DateTime(2018, 4, 1, 0, 0, 0, 0);
//
// isThereRatePensionCheckBox
//
this.isThereRatePensionCheckBox.AutoSize = true;
this.isThereRatePensionCheckBox.Location = new System.Drawing.Point(147, 185);
this.isThereRatePensionCheckBox.Name = "isThereRatePensionCheckBox";
this.isThereRatePensionCheckBox.Size = new System.Drawing.Size(204, 17);
```

```
            this.isThereRatePensionCheckBox.TabIndex = 18;
            this.isThereRatePensionCheckBox.Text = "Ja - Hvis nej går alt til livsvarid livrente";
            this.isThereRatePensionCheckBox.UseVisualStyleBackColor = true;
            this.isThereRatePensionCheckBox.CheckedChanged += new System.EventHandler(this.isThereRatePensionCheckBox_CheckedChanged);
            //
            // createSchemeButton
            //
            this.createSchemeButton.BackColor = System.Drawing.SystemColors.HighlightText;
            this.createSchemeButton.FlatAppearance.BorderColor = System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(64)))), ((int)(((byte)(64)))));
            this.createSchemeButton.FlatAppearance.BorderSize = 2;
            this.createSchemeButton.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Green;
            this.createSchemeButton.FlatAppearance.MouseOverBackColor = System.Drawing.Color.Lime;
            this.createSchemeButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
            this.createSchemeButton.Location = new System.Drawing.Point(45, 302);
            this.createSchemeButton.Name = "createSchemeButton";
            this.createSchemeButton.Size = new System.Drawing.Size(118, 23);
            this.createSchemeButton.TabIndex = 19;
            this.createSchemeButton.Text = "Opret Ordning";
            this.createSchemeButton.UseVisualStyleBackColor = false;
            this.createSchemeButton.Click += new System.EventHandler(this.button1_Click);
            //
            // prognoseButton
            //
            this.prognoseButton.BackColor = System.Drawing.SystemColors.HighlightText;
            this.prognoseButton.FlatAppearance.BorderColor = System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(64)))), ((int)(((byte)(64)))));
            this.prognoseButton.FlatAppearance.BorderSize = 2;
            this.prognoseButton.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Green;
```

```
this.prognoseButton.FlatAppearance.MouseOverBackColor = System.Drawing.Color.Lime;
this.prognoseButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
this.prognoseButton.Location = new System.Drawing.Point(530, 281);
this.prognoseButton.Name = "prognoseButton";
this.prognoseButton.Size = new System.Drawing.Size(118, 23);
this.prognoseButton.TabIndex = 20;
this.prognoseButton.Text = "Beregn Prognose";
this.prognoseButton.UseVisualStyleBackColor = false;
this.prognoseButton.Click += new System.EventHandler(this.prognoseButton_Click);
//
// label4
//
this.label4.AutoSize = true;
this.label4.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label4.Location = new System.Drawing.Point(531, 334);
this.label4.MinimumSize = new System.Drawing.Size(115, 10);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(115, 13);
this.label4.TabIndex = 23;
this.label4.Text = "Udbetaling fra liv";
//
// label8
//
this.label8.AutoSize = true;
this.label8.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label8.Location = new System.Drawing.Point(531, 312);
this.label8.MinimumSize = new System.Drawing.Size(115, 10);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(115, 13);
this.label8.TabIndex = 21;
```

```
        this.label8.Text = "Udbetaling fra rate";
        //
        // label10
        //
        this.label10.AutoSize = true;
        this.label10.BackColor = System.Drawing.SystemColors.ActiveCaption;
        this.label10.Location = new System.Drawing.Point(46, 278);
        this.label10.MinimumSize = new System.Drawing.Size(115, 10);
        this.label10.Name = "label10";
        this.label10.Size = new System.Drawing.Size(115, 13);
        this.label10.TabIndex = 25;
        this.label10.Text = "Udbetalingsperiode";
        //
        // SeekInfoButton
        //
        this.SeekInfoButton.BackColor = System.Drawing.SystemColors.HighlightText;
        this.SeekInfoButton.FlatAppearance.BorderColor =
System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(64)))), ((int)(((byte)(64)))));
        this.SeekInfoButton.FlatAppearance.BorderSize = 2;
        this.SeekInfoButton.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Green;
        this.SeekInfoButton.FlatAppearance.MouseOverBackColor = System.Drawing.Color.Lime;
        this.SeekInfoButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.SeekInfoButton.Location = new System.Drawing.Point(530, 63);
        this.SeekInfoButton.Name = "SeekInfoButton";
        this.SeekInfoButton.Size = new System.Drawing.Size(118, 23);
        this.SeekInfoButton.TabIndex = 29;
        this.SeekInfoButton.Text = "Fremsøg Ordning";
        this.SeekInfoButton.UseVisualStyleBackColor = false;
        this.SeekInfoButton.Click += new System.EventHandler(this.SeekInfoButton_Click);
        //
```

```
// label2
//
this.label2.AutoSize = true;
this.label2.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label2.Location = new System.Drawing.Point(533, 93);
this.label2.MinimumSize = new System.Drawing.Size(115, 10);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(115, 13);
this.label2.TabIndex = 30;
this.label2.Text = "Ordningsnummer:";
//
// label9
//
this.label9.AutoSize = true;
this.label9.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label9.Location = new System.Drawing.Point(533, 138);
this.label9.MinimumSize = new System.Drawing.Size(115, 10);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(115, 13);
this.label9.TabIndex = 31;
this.label9.Text = "Udløbs dato";
//
// label12
//
this.label12.AutoSize = true;
this.label12.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label12.Location = new System.Drawing.Point(533, 185);
this.label12.MinimumSize = new System.Drawing.Size(115, 10);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(115, 13);
this.label12.TabIndex = 33;
this.label12.Text = "Forfald";
//
// label14
//
this.label14.AutoSize = true;
this.label14.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label14.Location = new System.Drawing.Point(533, 162);
this.label14.MinimumSize = new System.Drawing.Size(115, 10);
this.label14.Name = "label14";
this.label14.Size = new System.Drawing.Size(115, 13);
this.label14.TabIndex = 32;
this.label14.Text = "Nuværende depot";
//
// ShowMaturityLabel
//
this.ShowMaturityLabel.AutoSize = true;
this.ShowMaturityLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ShowMaturityLabel.Location = new System.Drawing.Point(657, 185);
this.ShowMaturityLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowMaturityLabel.Name = "ShowMaturityLabel";
this.ShowMaturityLabel.Size = new System.Drawing.Size(200, 13);
this.ShowMaturityLabel.TabIndex = 37;
this.ShowMaturityLabel.Text = "Ordningsnummer:";
this.ShowMaturityLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// ShowCurrentDepositLabel
//
this.ShowCurrentDepositLabel.AutoSize = true;
this.ShowCurrentDepositLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ShowCurrentDepositLabel.Location = new System.Drawing.Point(657, 162);
this.ShowCurrentDepositLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowCurrentDepositLabel.Name = "ShowCurrentDepositLabel";
this.ShowCurrentDepositLabel.Size = new System.Drawing.Size(200, 13);
this.ShowCurrentDepositLabel.TabIndex = 36;
this.ShowCurrentDepositLabel.Text = "Ordningsnummer:";
this.ShowCurrentDepositLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// ShowExpiryDateLabel
//
this.ShowExpiryDateLabel.AutoSize = true;
this.ShowExpiryDateLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ShowExpiryDateLabel.Location = new System.Drawing.Point(657, 138);
this.ShowExpiryDateLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowExpiryDateLabel.Name = "ShowExpiryDateLabel";
this.ShowExpiryDateLabel.Size = new System.Drawing.Size(200, 13);
this.ShowExpiryDateLabel.TabIndex = 35;
this.ShowExpiryDateLabel.Text = "Ordningsnummer:";
this.ShowExpiryDateLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// ShowSchemeNumberLabel
//
this.ShowSchemeNumberLabel.AutoSize = true;
this.ShowSchemeNumberLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ShowSchemeNumberLabel.Location = new System.Drawing.Point(657, 93);
this.ShowSchemeNumberLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowSchemeNumberLabel.Name = "ShowSchemeNumberLabel";
this.ShowSchemeNumberLabel.Size = new System.Drawing.Size(200, 13);
this.ShowSchemeNumberLabel.TabIndex = 34;
this.ShowSchemeNumberLabel.Text = "Ordningsnummer";
this.ShowSchemeNumberLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// groupBox1
//
this.groupBox1.Controls.Add(this.rateYearTextBox);
this.groupBox1.Controls.Add(this.maxRateTextBox);
this.groupBox1.Controls.Add(this.expiryTextBox);
this.groupBox1.Controls.Add(this.premiumTextBox);
this.groupBox1.Controls.Add(this.FødselsDatoTextBox);
this.groupBox1.Controls.Add(this.customerNameTextBox);
this.groupBox1.Controls.Add(this.CreationLabel);
this.groupBox1.Controls.Add(this.label17);
this.groupBox1.Controls.Add(this.isThereRatePensionCheckBox);
this.groupBox1.Controls.Add(this.startDateSelector);
this.groupBox1.Controls.Add(this.maturityComboBox);
this.groupBox1.Location = new System.Drawing.Point(26, 38);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(379, 331);
this.groupBox1.TabIndex = 38;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Opret Ordning";
//
// rateYearTextBox
//
this.rateYearTextBox.Location = new System.Drawing.Point(147, 240);
this.rateYearTextBox.Name = "rateYearTextBox";
this.rateYearTextBox.Size = new System.Drawing.Size(200, 20);
this.rateYearTextBox.TabIndex = 42;
//
// maxRateTextBox
```

```
//
this.maxRateTextBox.Location = new System.Drawing.Point(147, 213);
this.maxRateTextBox.Name = "maxRateTextBox";
this.maxRateTextBox.Size = new System.Drawing.Size(200, 20);
this.maxRateTextBox.TabIndex = 41;
//
// expiryTextBox
//
this.expiryTextBox.Location = new System.Drawing.Point(146, 129);
this.expiryTextBox.Name = "expiryTextBox";
this.expiryTextBox.Size = new System.Drawing.Size(200, 20);
this.expiryTextBox.TabIndex = 41;
this.expiryTextBox.Text = "67";
this.expiryTextBox.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
//
// premiumTextBox
//
this.premiumTextBox.Location = new System.Drawing.Point(146, 75);
this.premiumTextBox.Name = "premiumTextBox";
this.premiumTextBox.Size = new System.Drawing.Size(200, 20);
this.premiumTextBox.TabIndex = 40;
this.premiumTextBox.Text = "10000";
this.premiumTextBox.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
//
// FødselsDatoTextBox
//
this.FødselsDatoTextBox.Location = new System.Drawing.Point(146, 46);
this.FødselsDatoTextBox.Name = "FødselsDatoTextBox";
this.FødselsDatoTextBox.Size = new System.Drawing.Size(200, 20);
this.FødselsDatoTextBox.TabIndex = 7;
```

```
            this.FødselsDatoTextBox.Text = "12101982";
            this.FødselsDatoTextBox.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
            //
            // customerNameTextBox
            //
            this.customerNameTextBox.Location = new System.Drawing.Point(146, 18);
            this.customerNameTextBox.Name = "customerNameTextBox";
            this.customerNameTextBox.Size = new System.Drawing.Size(200, 20);
            this.customerNameTextBox.TabIndex = 40;
            this.customerNameTextBox.Text = "Jacob Kjærgaard";
            this.customerNameTextBox.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
            //
            // CreationLabel
            //
            this.CreationLabel.AutoSize = true;
            this.CreationLabel.BackColor = System.Drawing.SystemColors.HotTrack;
            this.CreationLabel.FlatStyle = System.Windows.Forms.FlatStyle.System;
            this.CreationLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Underline))),
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.CreationLabel.ForeColor = System.Drawing.SystemColors.MenuBar;
            this.CreationLabel.Location = new System.Drawing.Point(41, 298);
            this.CreationLabel.MinimumSize = new System.Drawing.Size(300, 0);
            this.CreationLabel.Name = "CreationLabel";
            this.CreationLabel.Size = new System.Drawing.Size(300, 16);
            this.CreationLabel.TabIndex = 6;
            this.CreationLabel.TextAlign = System.Drawing.ContentAlignment.TopCenter;
            //
            // label17
            //
            this.label17.AutoSize = true;
            this.label17.BackColor = System.Drawing.SystemColors.ActiveCaption;
            this.label17.Location = new System.Drawing.Point(20, 46);
            this.label17.MinimumSize = new System.Drawing.Size(115, 10);
            this.label17.Name = "label17";
            this.label17.Size = new System.Drawing.Size(115, 13);
            this.label17.TabIndex = 4;
            this.label17.Text = "Fødselsdato";
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.PremiumPaymentButton);
            this.groupBox2.Controls.Add(this.ShowLifePrognosisLabel);
            this.groupBox2.Controls.Add(this.ShowRatePrognosisLabel);
            this.groupBox2.Controls.Add(this.ShowYearlyPremiumLabel);
            this.groupBox2.Controls.Add(this.label21);
            this.groupBox2.Location = new System.Drawing.Point(494, 38);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(386, 331);
            this.groupBox2.TabIndex = 39;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Visning af Ordning";
            //
            // PremiumPaymentButton
            //
            this.PremiumPaymentButton.BackColor = System.Drawing.SystemColors.HighlightText;
            this.PremiumPaymentButton.FlatAppearance.BorderColor =
System.Drawing.Color.FromArgb(((int)(((byte)(0)))), ((int)(((byte)(64)))), ((int)(((byte)(64)))));
            this.PremiumPaymentButton.FlatAppearance.BorderSize = 2;
            this.PremiumPaymentButton.FlatAppearance.MouseDownBackColor = System.Drawing.Color.Green;
            this.PremiumPaymentButton.FlatAppearance.MouseOverBackColor = System.Drawing.Color.Lime;
            this.PremiumPaymentButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
            this.PremiumPaymentButton.Location = new System.Drawing.Point(36, 193);
            this.PremiumPaymentButton.Name = "PremiumPaymentButton";
            this.PremiumPaymentButton.Size = new System.Drawing.Size(118, 23);
            this.PremiumPaymentButton.TabIndex = 40;
            this.PremiumPaymentButton.Text = "Indbetal præmie";
            this.PremiumPaymentButton.UseVisualStyleBackColor = false;
            this.PremiumPaymentButton.Click += new System.EventHandler(this.PremiumPaymentButton_Click);
            //
            // ShowLifePrognosisLabel
            //
            this.ShowLifePrognosisLabel.AutoSize = true;
            this.ShowLifePrognosisLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
            this.ShowLifePrognosisLabel.Location = new System.Drawing.Point(163, 296);
            this.ShowLifePrognosisLabel.MinimumSize = new System.Drawing.Size(200, 10);
            this.ShowLifePrognosisLabel.Name = "ShowLifePrognosisLabel";
            this.ShowLifePrognosisLabel.Size = new System.Drawing.Size(200, 13);
            this.ShowLifePrognosisLabel.TabIndex = 39;
            this.ShowLifePrognosisLabel.Text = "Udbetaling fra liv";
            this.ShowLifePrognosisLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
            //
            // ShowRatePrognosisLabel
            //
            this.ShowRatePrognosisLabel.AutoSize = true;
            this.ShowRatePrognosisLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
            this.ShowRatePrognosisLabel.Location = new System.Drawing.Point(163, 274);
```

```
this.ShowRatePrognosisLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowRatePrognosisLabel.Name = "ShowRatePrognosisLabel";
this.ShowRatePrognosisLabel.Size = new System.Drawing.Size(200, 13);
this.ShowRatePrognosisLabel.TabIndex = 38;
this.ShowRatePrognosisLabel.Text = "Udbetaling fra rate";
this.ShowRatePrognosisLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// ShowYearlyPremiumLabel
//
this.ShowYearlyPremiumLabel.AutoSize = true;
this.ShowYearlyPremiumLabel.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.ShowYearlyPremiumLabel.Location = new System.Drawing.Point(163, 78);
this.ShowYearlyPremiumLabel.MinimumSize = new System.Drawing.Size(200, 10);
this.ShowYearlyPremiumLabel.Name = "ShowYearlyPremiumLabel";
this.ShowYearlyPremiumLabel.Size = new System.Drawing.Size(200, 13);
this.ShowYearlyPremiumLabel.TabIndex = 37;
this.ShowYearlyPremiumLabel.Text = "Ordningsnummer:";
this.ShowYearlyPremiumLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// label21
//
this.label21.AutoSize = true;
this.label21.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.label21.Location = new System.Drawing.Point(39, 78);
this.label21.MinimumSize = new System.Drawing.Size(115, 10);
this.label21.Name = "label21";
this.label21.Size = new System.Drawing.Size(115, 13);
this.label21.TabIndex = 36;
this.label21.Text = "Årlig præmie";
//
```

```
            // PensionCreation
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.BackColor = System.Drawing.SystemColors.HotTrack;
            this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
            this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Center;
            this.ClientSize = new System.Drawing.Size(904, 442);
            this.Controls.Add(this.ShowMaturityLabel);
            this.Controls.Add(this.ShowCurrentDepositLabel);
            this.Controls.Add(this.ShowExpiryDateLabel);
            this.Controls.Add(this.ShowSchemeNumberLabel);
            this.Controls.Add(this.label12);
            this.Controls.Add(this.label14);
            this.Controls.Add(this.label9);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.SeekInfoButton);
            this.Controls.Add(this.label10);
            this.Controls.Add(this.label4);
            this.Controls.Add(this.label8);
            this.Controls.Add(this.prognoseButton);
            this.Controls.Add(this.createSchemeButton);
            this.Controls.Add(this.label11);
            this.Controls.Add(this.label13);
            this.Controls.Add(this.label15);
            this.Controls.Add(this.label5);
            this.Controls.Add(this.label7);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.label1);
```

```
    this.Controls.Add(this.groupBox1);
    this.Controls.Add(this.groupBox2);
    this.DoubleBuffered = true;
    this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
    this.Name = "PensionCreation";
    this.Text = "Jacobs Pensionsselskab";
    this.groupBox1.ResumeLayout(false);
    this.groupBox1.PerformLayout();
    this.groupBox2.ResumeLayout(false);
    this.groupBox2.PerformLayout();
    this.ResumeLayout(false);
    this.PerformLayout();

}

#endregion

private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.Label label13;
private System.Windows.Forms.Label label15;
private System.Windows.Forms.ComboBox maturityComboBox;
private System.Windows.Forms.DateTimePicker startDateSelector;
private System.Windows.Forms.CheckBox isThereRatePensionCheckBox;
private System.Windows.Forms.Button createSchemeButton;
private System.Windows.Forms.Button prognoseButton;
private System.Windows.Forms.Label label4;
```

```
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Button SeekInfoButton;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label label12;
        private System.Windows.Forms.Label label14;
        private System.Windows.Forms.Label ShowMaturityLabel;
        private System.Windows.Forms.Label ShowCurrentDepositLabel;
        private System.Windows.Forms.Label ShowExpiryDateLabel;
        private System.Windows.Forms.Label ShowSchemeNumberLabel;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.Label ShowLifePrognosisLabel;
        private System.Windows.Forms.Label ShowRatePrognosisLabel;
        private System.Windows.Forms.Label ShowYearlyPremiumLabel;
        private System.Windows.Forms.Label label21;
        private System.Windows.Forms.Button PremiumPaymentButton;
        private System.Windows.Forms.Label label17;
        private System.Windows.Forms.Label CreationLabel;
        private System.Windows.Forms.TextBox rateYearTextBox;
        private System.Windows.Forms.TextBox maxRateTextBox;
        private System.Windows.Forms.TextBox expiryTextBox;
        private System.Windows.Forms.TextBox premiumTextBox;
        private System.Windows.Forms.TextBox FødselsDatoTextBox;
        private System.Windows.Forms.TextBox customerNameTextBox;
    }
}
```