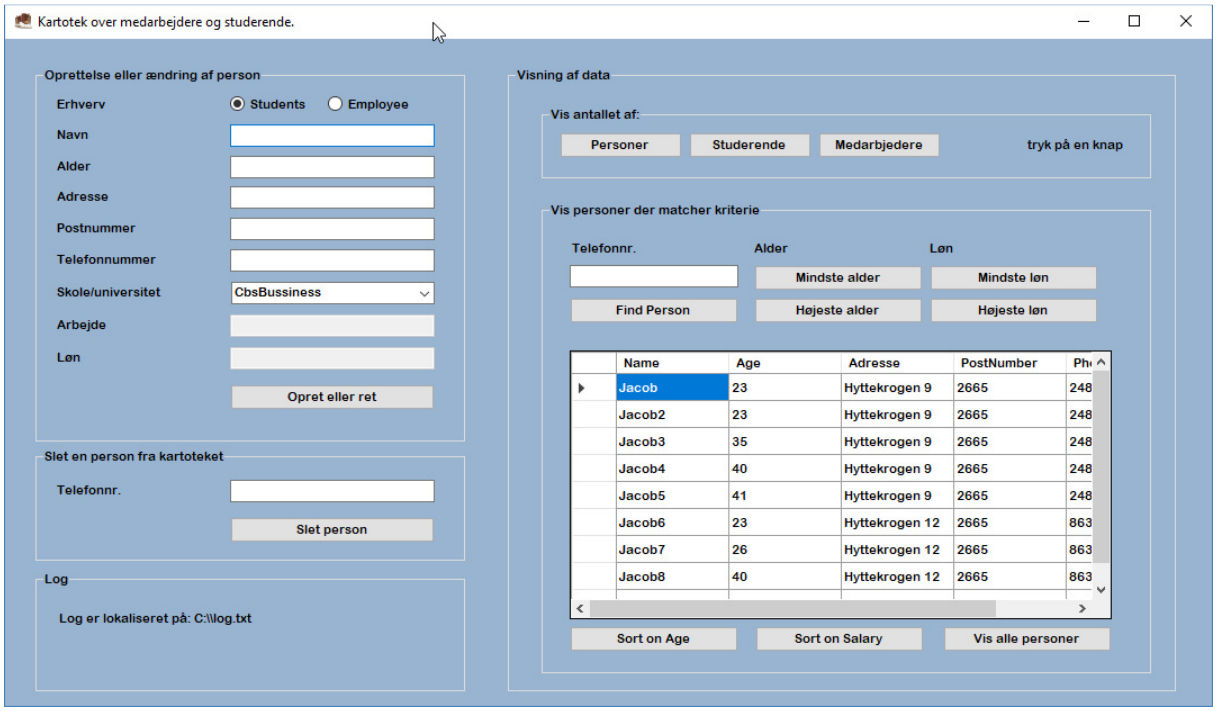
Implementering af en Windows Form for Archive – Kartotek over medarbejdere og studerende



Forfatter: Jacob Kjærgaard

Dato: 20-04-2018

Version: 1.0

Indholdsfortegnelse

Kapitel 1 3

1. Bagrund 3
2. Problemstilling 3

Kapitel 2 3

1. Dataopsætning 3
2. DataStorage 3
3. Teknologi valg 3
4. Teststrategi 3

Kapitel 3 4

Kapitel 4 6

1. Klasse Archive 6
2. Klasse Employees 11
3. Klasse FilePrint 12
4. Klasse SortSalaryDescendingHelper 13

Kapitel 6 14

Kapitel 1

Kartoteket er et CRUD program, hvor det er muligt at oprette, se, rette og slette studerende og medarbejdere. Herudover er der forskellige info om antal og sorteringer.

1. Bagrund

Programmet laves som en obligatorisk afleveringsopgave i faget videregående programmering.

2. Problemstilling

Der skal laves en C# applikation til at oprette en eller flere liste-struktur der kan indeholde objekter af en eller flere klasse datastruktur.
Der skal laves en base klasse, der hedder Person, hvor afledte klasser Student og Employee arver fra Person. De skal også arve fra en interface som gøre man kan lave en Compare af to objekter. Dette bruges til sortering.

Kapitel 2

Det er en rimelig afgrænset problemstilling, som burde kunne løses uden at der skal afgrænses noget. Da det dog er en læringsopgave og det ikke er en applikation, der skal bruges, vil exception handling og unit test blive begrænset, hvor det ikke længere giver læringsmæssigt værdi i forhold til deadlinie.

1. Dataopsætning

Systemet fødes med data for at gøre testen lettere, men ellers indsamles der kun de data, der indtastes i selve applikationen.

2. DataStorage

Pt. Bliver data gemt i lister, hvilket kunne ændres til en database, hvis behovet viste sig at opstå.

3. Teknologi valg

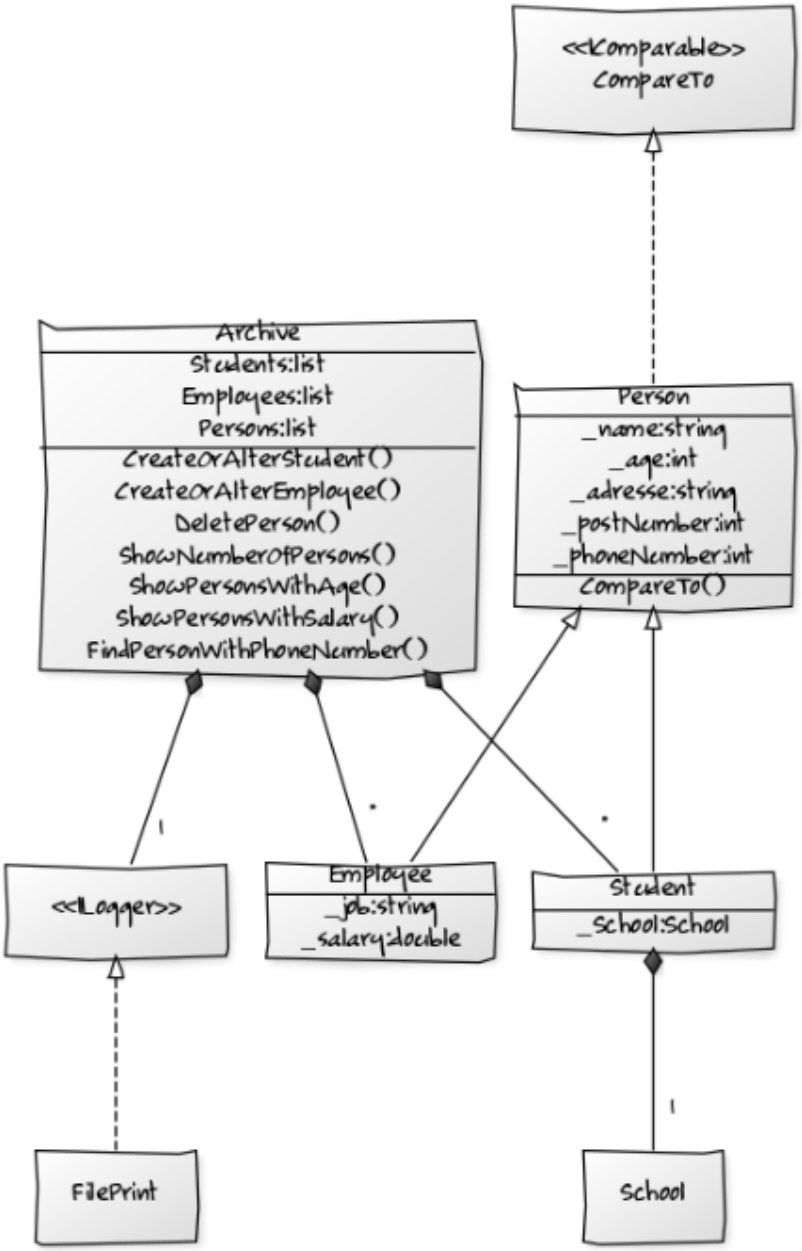
C#, og VS

4. Teststrategi

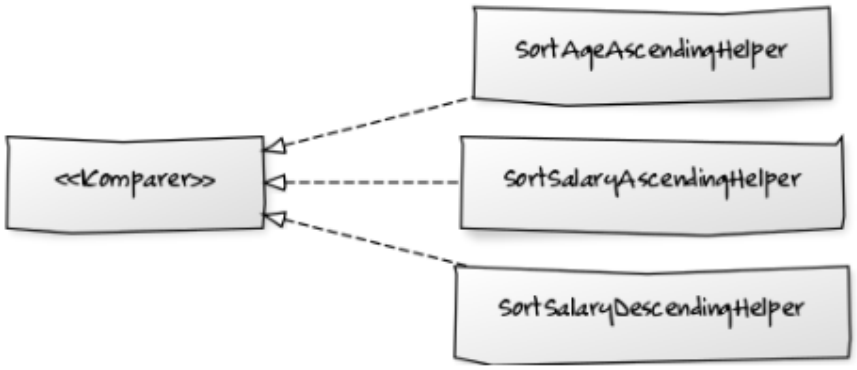
Der er lavet et separat unit test project, hvor der er lavet unit test på enkelte metoder, men der er blevet afgrænset at bruge al for meget tid her.
Der er lavet manuel test for at sandsynliggøre at programmet fungere efter hensigten i alle henseender. Der er fundet flere bugs i processen, som er løst med det samme.
Der udestår dog en enkelt bug, jeg ikke kunne løse. Hvis man sorter employees ved at trykke på ”Sort on Salary” sker der ikke noget når man trykker igen, hvor den burde vende listen om – men hvis man trykker på en anden knap og så sort igen, så vises listen med den korrekte sortering.

Kapitel 3

Der er lavet følgende UML diagram, for at vise strukturen af programmet.



Herudover er der 3 klasser der implementerer icomparer, samt en et unit test project og en enum klasse (MessageType), der bliver brugt i FilePrint.



Person klassen er lavet som en abstrakt klasse, hvor der er to arvede klasser. Person klassen er lavet abstract, da opgaveteksten går meget på, at man skal være enten medarbejder eller studerende og det ikke er meningen, at der skal være en instans af person klassen.

Persons implementere interfacet IComparable, som giver en default sortering (alder descending), som student og employee arver. Der er herudover også blevet lavet 3 classes som implementerer Icomparer<person>/<Employee>, så der kan laves en alternativ sortering på alder i stigende og på faldende og stigende løn.

Sort() har en overload som tager et Icompare interface, det er altså den klasse, der implementer dette interface, der styrer sorteringen. Ved at give sort metoden en classe der sorter Employee efter salary desencending, er det den måde Sort funktionen sorter den pågældende liste af medarbejdere.

Jeg har valgt, at Archive skal holde 2 lister. En med students og en med employees. Jeg tænkte oprindeligt, at have én liste og så lave en for løkke og caste til student/employee (AS operatoren) for så at validere om casting er succesfuldt, så jeg ved om det er en studerende eller ej – Men endte med at konkludere, at det blev for beregningstungt fremfor ”bare” at lave en type-safe liste, der kun holdte det ene eller det andet og så må man lave to løkker hvis man skal gennemgå begge lister.

Det eneste der ellers er lidt interessant i min opbygning er, at jeg har valgt at salary skal være nullable, det er sket dels for at lære denne feature, men også da det ofte sker, at folk ikke ønsker at oplyse netop denne. Ved at lave den nullable, bliver feltet boxed til et objekt og det har en performance straf, men da det er et lille projekt og performace pt. Ikke er et issue accepteres dette.

I min Gui vil der i oprettelse være en radio button som bliver oversat til boolean værdi isStudent, så der bliver kaldt den korrekte CreateOrAlter metode – Jeg valgte 2 metoder, men man kunne have valgt én med optional parameters, hvor der var forskelle, men det virkede forkert.

I mine sorteringsknapper lavede jeg også en ”Toggle”, så hvis man trykker på den samme knap igen, sker sorteringen modsat.

For at udvide brugen af interfaces har jeg valgt at uddelegere beskeder både i form af exceptions og generelle log-beskeder til et interface, hvor jeg har oprettet en klasse, der implementerer interface, der udskriver til en fil. Det sker via dependecy injection i starten af formularen hvor objectet archive dannes (kunne også være lavet i Program.cs), så det vil være let at lave en ny klasse, der eksempelvis udskriver til en printer eller lign. Og så skal archive bare have denne, når den oprettes i stedet.

Kapitel 4

Der er lavet mange klasser og enkelte vises her, mens de andre er vedlagt i rapporten bagerst.

1. Klasse Archive

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Runtime.InteropServices.WindowsRuntime;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Archive
{
    public class Archive
    {
        public ILogger Logger { get; }

        public List<Student> Students { get; } = new List<Student>();
        public List<Employee> Employees { get; } = new List<Employee>();

        //Konstruktor
        public Archive(ILogger logger)
        {
            this.Logger = logger;
        }

        //I nedenstående 2 metoder undersøges det først, om der allerede finde en registrering med samme telefon nummer.
        //Hvis der findes en opdateres personen, med de data der er blevet indsat i formularen og breaker ud af løkken,
        //så løkken ikke bruger tid på at køre listen færdig (der burde ikke kunne forekomme to forekomster med samme telefonnr.)
        public void CreateOrAlterStudent(string name, int age, string adresse, int postNumber, int phoneNumber, School school)
        {
            bool personFound = false;

            foreach (var student in Students)
            {
                if (student.PhoneNumber == phoneNumber)
                {
                    student.Name = name;
                    student.Age = age;
                    student.Adresse = adresse;
                    student.PostNumber = postNumber;
                    student.PhoneNumber = phoneNumber;
                    student.School = school;
```

```
43                  Logger.LogInfo("Studerende ændret kl. " + DateTime.Now);
44                  personFound = true;
45              }
46          }
47
48          if (!personFound)
49          {
50              Students.Add(new Student(name, age, adresse, postNumber, phoneNumber, school));
51              Logger.LogInfo("Studerende oprettet kl. " + DateTime.Now);
52          }
53      }
54
55      public void CreateOrAlterEmployee(string name, int age, string adresse, int postNumber, int phoneNumber, string job,
56          double? salary)
57      {
58          bool personFound = false;
59
60          foreach (var employee in Employees)
61          {
62              if (employee.PhoneNumber == phoneNumber)
63              {
64                  employee.Name = name;
65                  employee.Age = age;
66                  employee.Adresse = adresse;
67                  employee.PostNumber = postNumber;
68                  employee.PhoneNumber = phoneNumber;
69                  employee.Job = job;
70                  employee.Salary = salary;
71                  Logger.LogInfo("Medarbejder ændret kl. " + DateTime.Now);
72                  personFound = true;
73              }
74          }
75          if (!personFound)
76          {
77              Employees.Add(new Employee(name, age, adresse, postNumber, phoneNumber, job, salary));
78              Logger.LogInfo("Medarbejder oprettet kl. " + DateTime.Now);
79          }
80      }
81
82      //I nedenstående metode undersøges det først om der finde en person med et matchende telefonnummer.
83      //Hvis der findes et match slettes denne registrering. Hvis der ikke findes et match bliver smidt
```

```
84          //en exception med info som bliver sendt tilbage til kalderen.
85          public void DeletePerson(int phoneNumber)
86          {
87              bool personFound = false;
88
89              for (int i = 0; i < Students.Count; i++)
90              {
91                  if (Students[i].PhoneNumber == phoneNumber)
92                  {
93                      Logger.LogInfo("Person med telefonnummer " + Students[i].PhoneNumber + " blev slettet kl. " + DateTime.Now);
94                      Students.Remove(Students[i]);
95                      MessageBox.Show("Sletning succesfuld", "Sletning succesfuld", MessageBoxButtons.OK);
96                      personFound = true;
97                      break;
98                  }
99              }
00
01              if (personFound == false)
02              {
03                  for (int i = 0; i < Employees.Count; i++)
04                  {
05                      if (Employees[i].PhoneNumber == phoneNumber)
06                      {
07                          Logger.LogInfo("Person med telefonnummer " + Employees[i].PhoneNumber + " blev slettet kl. " + DateTime.Now);
08                          Employees.Remove(Employees[i]);
09                          MessageBox.Show("Sletning succesfuld", "Sletning succesfuld", MessageBoxButtons.OK);
10                          personFound = true;
11                          break;
12                      }
13                  }
14              }
15
16              if (personFound == false)
17              {
18                  MessageBox.Show("Der findes ikke en person med det telefon nummer i listen", "Ingen person matcher", MessageBoxButtons.OK);
19                  Logger.LogInfo("Person ikke fundet og kan ikk slettes");
20              }
21          }
22
23          //Viser antalet af mennesker
24          public int ShowNumberOfPersons()
25          {
```

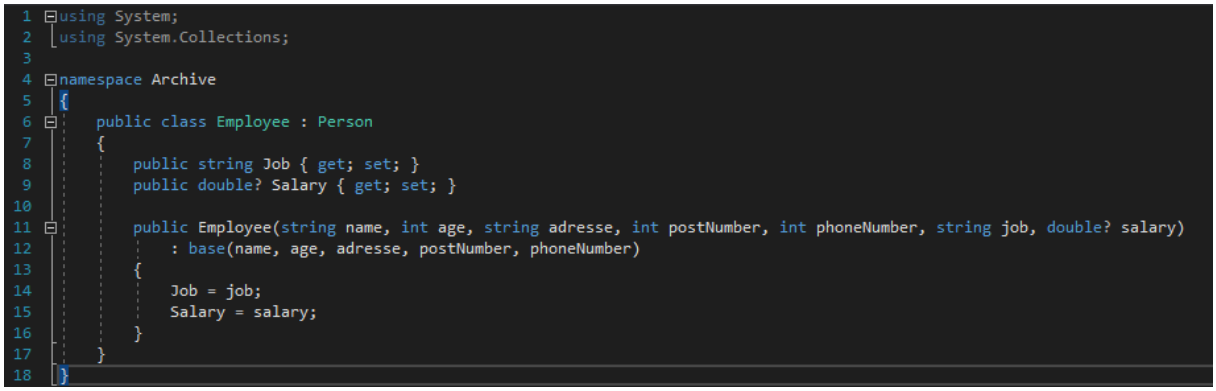
```
126             Logger.LogInfo("Viser antallet af personer kl. " + DateTime.Now);
127             return Employees.Count + Students.Count;
128         }
129
130         //Overload af ShowNumberOfPersons metode
131         //Viser antale af studerende eller employees
132         public int ShowNumberOfPersons(Boolean showStudents)
133         {
134             Logger.LogInfo("Viser antallet af specifikke personer kl. " + DateTime.Now);
135             return (showStudents) ? Students.Count : Employees.Count;
136         }
137
138         //Da der er flere der kan have samme alder eller løn, sker der her flere steps.
139         //1. Sorter listen efter Age/løn for at sikre at første registrering er den mindste og sidste registrering er den højeste.
140         //2. tildel min og max værdier
141         //3. iterere hen over de to løkker (ved age og en ved salary) og finder de personer der matcher min/max løn og skriv dem til en resultat liste.
142         //4. returner den relevante liste.
143         public List<Person> ShowPersonsWithAge(Boolean lowestAge = true)
144         {
145             List<Person> minAgeList = new List<Person>();
146             List<Person> maxAgeList = new List<Person>();
147             int minAge;
148             int maxAge;
149
150             //1. sorterer listen efter CompareTo i persons hvilket er age i descending
151             Students.Sort();
152             Employees.Sort();
153
154             //2. tildel min og max værdier
155             //Da listen er sorteret efter alder, må min og max være de to ydre værdier - men det valideres lige inden det assignes.
156             minAge = (Students[0].Age <= Employees[0].Age) ? Students[0].Age : Employees[0].Age;
157             maxAge = (Students[Students.Count - 1].Age <= Employees[Employees.Count - 1].Age) ? Students[Students.Count - 1].Age : Employees[Employees.Count - 1].Age;
158
159             //3. iterere hen over de to løkker (ved age og en ved salary) og finder de personer der matcher min/max løn og skriv dem til en resultat liste.
160             //hvis en person på listen har samme alder som min eller max alder bliver de tilføjet den respective liste .
161             for (int i = 0; i < Students.Count; i++)
162             {
163                 if (Students[i].Age == minAge)
164                 {
165                     minAgeList.Add(Students[i]);
166                 }
```

```
167
168             if (Students[i].Age == maxAge)
169             {
170                 maxAgeList.Add(Students[i]);
171             }
172         }
173
174         for (int i = 0; i < Employees.Count; i++)
175         {
176             if (Employees[i].Age == minAge)
177             {
178                 minAgeList.Add(Employees[i]);
179             }
180
181             if (Employees[i].Age == maxAge)
182             {
183                 maxAgeList.Add(Employees[i]);
184             }
185         }
186
187         //4. returner den relevante liste.
188         Logger.LogInfo("Retuner person(er) med højeste eller laveste aldre kl. " + DateTime.Now);
189         if (lowestAge)
190         {
191             return minAgeList;
192         }
193         else
194         {
195             return maxAgeList;
196         }
197     }
198
199     //Se kommentar fra ShowPersonsWithAge
200     public List<Employee> ShowEmployeesWithSalary(Boolean lowestSalary)
201     {
202         List<Employee> minSalaryList = new List<Employee>();
203         List<Employee> maxSalaryList = new List<Employee>();
204
205         Employees.Sort(new SortSalaryAscendingHelper()); //efter salary stigende
206
207         //da listen er sorteret efter løn må min og max være de to yderse værdier.
```

```
            //hvis listen er sorteret stigende kunne man bare min = [0], max = count -1
            Double? minSalary = Employees[0].Salary;
            Double? maxSalary = Employees[Employees.Count - 1].Salary;

            //hvis en person på listen har samme løn som min eller max alder bliver de tilføjet den respective liste .
            //hvis jeg ved at listen er sorteret stigende kunne jeg lave while løn = minLøn og spare tid.

            foreach (var employee in Employees)
            {
                if (employee.Salary.Equals(minSalary))
                {
                    minSalaryList.Add(employee);
                }

                if (employee.Salary.Equals(maxSalary))
                {
                    maxSalaryList.Add(employee);
                }
            }

            //returnere den liste der bliver bedt om.
            Logger.LogInfo("Retuner person(er) med højeste eller laveste løn kl. " + DateTime.Now);
            return (lowestSalary) ? minSalaryList : maxSalaryList;

        }

        //I nedenstående metode undersøges det først om der finde en person med et matchende telefonnummer.
        //Hvis der findes et match sendes denne person retur - det sker i en liste da det skal bruges til grid.
        //Hvis der ikke findes et match bliver der smidt en exception med info som bliver sendt tilbage til kalderen.
        public List<Person> FindPersonWithPhoneNumber(int phoneNumber)
        {
            List<Person> allPersons = ShowAllPersons();
            List<Person> resultList = new List<Person>();

            for (int i = 0; i < allPersons.Count; i++)
            {
                if(allPersons[i].PhoneNumber == phoneNumber)
                {
                    resultList.Add(allPersons[i]);
                    return resultList;
                }
            }
```

```
            throw new ArchiveException("Der findes ikke en person med det nummer");
        }

        public List<Person> ShowAllPersons()
        {
            List<Person> allPersons = new List<Person>();

            for (int i = 0; i < Students.Count; i++)
            {
                allPersons.Add(Students[i]);
            }

            for (int i = 0; i < Employees.Count; i++)
            {
                allPersons.Add(Employees[i]);
            }

            return allPersons;
        }
    }
}
```

2. Klasse Employees



```
using System;
using System.Collections;

namespace Archive
{
    public class Employee : Person
    {
        public string Job { get; set; }
        public double? Salary { get; set; }

        public Employee(string name, int age, string adresse, int postNumber, int phoneNumber, string job, double? salary)
            : base(name, age, adresse, postNumber, phoneNumber)
        {
            Job = job;
            Salary = salary;
        }
    }
}
```

3. Klasse FilePrint

```
using System;
using System.IO;
using System.Threading;

namespace Archive
{
    public class FilePrint : ILogger
    {
        private readonly string _path;

        public FilePrint(string path)
        {
            _path = path;
        }

        public void Error(string message, Exception exception)
        {
            Print(message, MessageType.Error);

            using (var streamWriter = new StreamWriter(_path, true))
            {
                streamWriter.WriteLine("stacktrace: " + exception.StackTrace);
            }
        }

        public void LogInfo(string message)
        {
                Print(message, MessageType.Info);
        }

        public void Print(string message, MessageType messageType)
        {
            using (var streamWriter = new StreamWriter(_path, true))
            {
                streamWriter.WriteLine(messageType + ": " + message);
            }
        }
    }
}
```

4. Klasse SortSalaryDescendingHelper

```
using System.Collections.Generic;
using System.Runtime.InteropServices.WindowsRuntime;
using System.Windows.Forms.VisualStyles;

namespace Archive
{
    public class SortSalaryDescendingHelper : IComparer<Employee>
    {
        public int Compare(Employee left, Employee right)
        {
            //da salary er nullable, skal man lige angive dette først.
            if (left.Salary == null) return 1;
            if (right.Salary == null) return -1;

            if (left.Salary > right.Salary)
                return -1;
            if (left.Salary < right.Salary)
                return 1;
            else
                return 0;
        }
    }
```

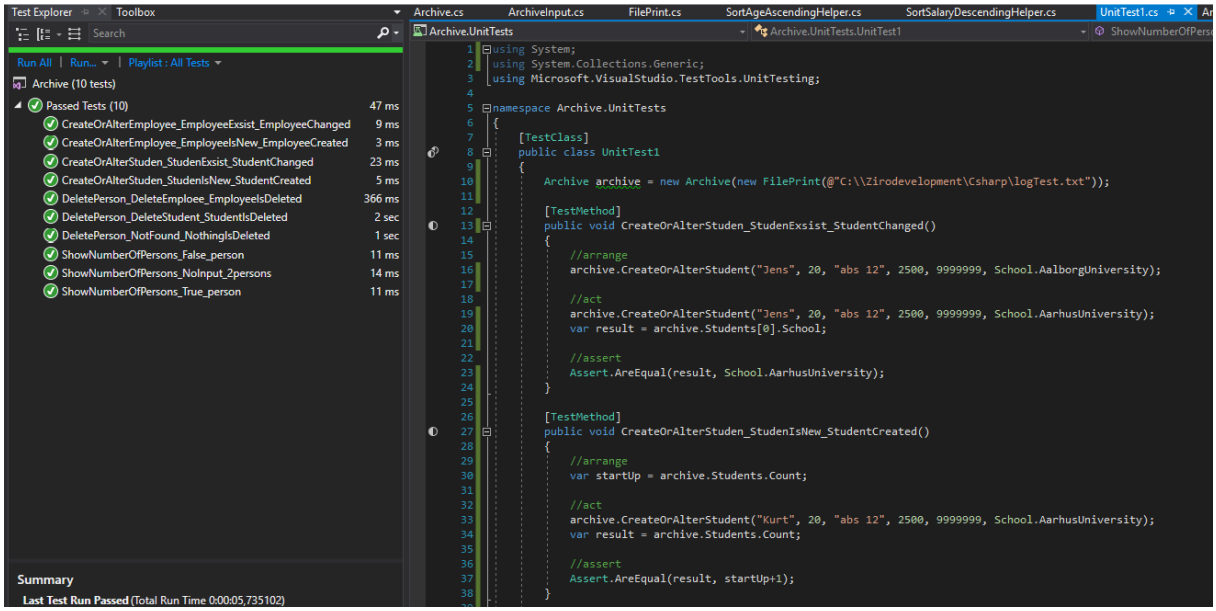
5.1. Fortæl kort om testen, screenshot, samt resultatet og tilrettelse hvis der var fejl

Der er lavet et unitTest project, hvor der er lavet en unit test for enkelte af de hoved metoder, der generer output. Der er lavet en test for hver af udgangene af metoden.
Disse er kørt for at sikre at back-end lever op til kravene.

Der er også lavet manuel test af front-end og for at gøre dette lettere, har jeg preudfyldt nogle studerende og medarbejdere, så man kan se min/max og antal uden at skulle oprette dem via brugergrænsefladen først.

5.2. Konklustion på test.

Det er en velfungerende applikation, dog kunne design og brugervenlighed forbedres.



Kapitel 6

Alt i alt synes jeg at opgaven var god til at lære mere om interfaces og få dem ind under huden.

```
using System;

namespace Archive
{
    public interface ILogger
    {
        void Error(string message, Exception exception);
        void LogInfo(string message);
    }
}
```

```
namespace Archive
{
    public enum MessageType
    {
        Error,
        Info
    }
}
```

```
namespace Archive
{
    public enum School
    {
        CbsBussiness,
        SmartLearning,
        AarhusUniversity,
        AalborgUniversity,
        SyddanskUniversity
    }
}
```

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


namespace Archive
{
    public abstract class Person : IComparable
    {
        public string Name { get; set; }
        public int Age { get; set; }
        public string Adresse { get; set; }
        public int PostNumber { get; set; }
        public int PhoneNumber { get; set; }

        protected Person(string name, int age, string adresse, int postNumber, int phoneNumber)
        {
            Name = name;
            Age = age;
            Adresse = adresse;
            PostNumber = postNumber;
            PhoneNumber = phoneNumber;
        }

        //Implementere IComparable ved at have en default sortering efter alder
        public int CompareTo(object obj)
        {
            var a = (Person)obj;
            return this.Age.CompareTo(a.Age);
        }
    }
}
```

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

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

```
using System.Collections;
using System.Collections.Generic;

namespace Archive
{
    public class SortAgeAscendingHelper : IComparer<Person>
    {
        public int Compare(Person left, Person right)
        {
            return right.Age - left.Age;
        }
    }
}
```

```
    public class SortSalaryAscendingHelper : IComparer<Employee>
    {
        public int Compare(Employee left, Employee right)
        {
            if (left.Salary == null) return -1;
            if (right.Salary == null) return 1;

            if (left.Salary < right.Salary)
                return -1;
            if (left.Salary > right.Salary)
                return 1;
            else
                return 0;
        }
    }
}
```

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


namespace Archive
{
    public class Student : Person
    {
        public School School { get; set; }

        public Student(string name, int age, string adress, int postNumber, int phoneNumber, School school)
            : base(name, age, adress, postNumber, phoneNumber)
        {
            School = school;
        }
    }
}
```

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

namespace Archive
{
    public partial class ArchiveInput : Form
    {
        private bool isStudent;
        private bool _asc = true;
        private bool _ascAge = true;
        private string name;
        private int age;
        private string adress;
        private int postNumber;
        private int phoneNumber;
        private School school;
        private string job;
        private double? salary;
        private int phoneNumberToDelete;
        private int findPersonWithPhoneNumber;
        public Archive archive;

        public ArchiveInput()
        {
            //får istansieret mit arkiv og får tilført start data
            InitializeComponent();
            SchoolComboBox.DataSource = Enum.GetValues(typeof(School));
            archive = new Archive(new FilePrint(@"C:\\Zirodevelopment\Csharp\log.txt"));
            archive.CreateOrAlterStudent("Jacob", 23, "Hyttekrogen 9", 2665, 24862386, School.SmartLearning);
            archive.CreateOrAlterStudent("Jacob2", 23, "Hyttekrogen 9", 2665, 24862387, School.SmartLearning);
            archive.CreateOrAlterStudent("Jacob3", 35, "Hyttekrogen 9", 2665, 24862388, School.SmartLearning);
            archive.CreateOrAlterStudent("Jacob4", 40, "Hyttekrogen 9", 2665, 24862389, School.SmartLearning);
            archive.CreateOrAlterStudent("Jacob5", 41, "Hyttekrogen 9", 2665, 24862390, School.SmartLearning);
```

```
43              archive.CreateOrAlterEmployee("Jacob6", 23, "Hyttekrogen 12", 2665, 86377900, "Tester", null);
44              archive.CreateOrAlterEmployee("Jacob7", 26, "Hyttekrogen 12", 2665, 86377901, "Tester", 400000);
45              archive.CreateOrAlterEmployee("Jacob8", 40, "Hyttekrogen 12", 2665, 86377902, "Tester", 500000);
46              archive.CreateOrAlterEmployee("Jacob9", 41, "Hyttekrogen 12", 2665, 86377903, "Tester", 350000);
47              dataGridView.DataSource = archive.ShowAllPersons();
48          }
49
50          //Tryk på knappen opret - Kunne være mere try catch here - men igen er ikke fokus for nu.
51          private void CreateOrAlterButton_Click(object sender, EventArgs e)
52          {
53              isStudent = StudentsRadioButton.Checked;
54              name = Convert.ToString(NameTextBox.Text);
55              age = Convert.ToInt32(AgeTextBox.Text);
56              adress = Convert.ToString(AdressTextBox.Text);
57              postNumber = Convert.ToInt32(PostNumberTextBox.Text);
58              phoneNumber = Convert.ToInt32(PhoneNumberTextBox.Text);
59
60              //hvis student radio button er selected tager jeg værdi fra dropdown.
61              if (isStudent)
62              {
63                  school = (School) SchoolComboBox.SelectedItem;
64                  archive.CreateOrAlterStudent(name, age, adress,postNumber,phoneNumber,school);
65              }
66              else
67              {
68                  //hvis employee radio button gemmer jeg employee værdier.
69                  job = Convert.ToString(JobTextBox.Text);
70
71                  if (SalaryTextBox.Text == "")
72                  {
73                      salary = null;
74                  }
75                  else
76                  {
77                      try
78                      {
79                          salary = Convert.ToDouble(SalaryTextBox.Text);
80                      }
81                      catch (Exception exception)
82                      {
83                          archive.Logger.Error("Fejl i input vedr. sletningsfelt", exception);
84
```

```
                }
            }
            archive.CreateOrAlterEmployee(name, age, adress,postNumber,phoneNumber,job, salary);
        }

    }

    //På slet har jeg lidt try catch og sender evt. fejl til det der tager sig af ILogger interfacet.
    private void DeleteButton_Click(object sender, EventArgs e)
    {
        try
        {
            phoneNumberToDelete = Convert.ToInt32(DeletePersonTextBoks.Text);
            archive.DeletePerson(phoneNumberToDelete);
        }
        catch (Exception exception)
        {
            archive.Logger.Error("Fejl i sletning af person", exception);
        }
    }

    //Clicker på antallet i de næste tre
    private void ShowNumberOfPersonButton_Click(object sender, EventArgs e)
    {
        ShowNumbersOfLabel.Text = "Antal af personer er: " + archive.ShowNumberOfPersons();
    }

    private void ShowNumberOfStudentsButton_Click(object sender, EventArgs e)
    {
        ShowNumbersOfLabel.Text = "Antal af studerende er: " + archive.ShowNumberOfPersons(true);
    }

    private void ShowNumberOfEmployeesButton_Click(object sender, EventArgs e)
    {
        ShowNumbersOfLabel.Text = "Antal af medarbejdere er: " + archive.ShowNumberOfPersons(false);
    }

    //finder person ud fra nummer med try catch
    private void PersonWithPhoneNumberButton_Click(object sender, EventArgs e)
    {
        try
        {
```

```
127                 findPersonWithPhoneNumber = Convert.ToInt32(PersonWithPhoneNumberTextBoks.Text);
128                 dataGridView.DataSource = archive.FindPersonWithPhoneNumber(findPersonWithPhoneNumber);
129             }
130             catch (Exception exception)
131             {
132                 archive.Logger.Error("Fejl - noget gik galt da kunden skulle findes", exception);
133             }
134         }
135
136         //de næste 5 kalder "bare" medtoder til vise data og viser resultatet i grid.
137         private void MinAgeButton_Click(object sender, EventArgs e)
138         {
139             dataGridView.DataSource = archive.ShowPersonsWithAge();//optional parameter med true som udgangspunkt
140         }
141
142         private void MaxAgeButton_Click(object sender, EventArgs e)
143         {
144             dataGridView.DataSource = archive.ShowPersonsWithAge(false);
145         }
146
147         private void MinSalaryButton_Click(object sender, EventArgs e)
148         {
149             dataGridView.DataSource = archive.ShowEmployeesWithSalary(true);
150         }
151
152         private void MaxSalaryButton_Click(object sender, EventArgs e)
153         {
154             dataGridView.DataSource = archive.ShowEmployeesWithSalary(false);
155         }
156
157         private void ShowAllPersonsButton_Click(object sender, EventArgs e)
158         {
159             dataGridView.DataSource = archive.ShowAllPersons();
160         }
161
162         //Lidt gui opsætning så korrekte felter er enabled når der ændres gruppe
163         private void EmployeeRadioButton_CheckedChanged(object sender, EventArgs e)
164         {
165             if (EmployeeRadioButton.Checked)
166             {
167                 JobTextBox.Enabled = true;
168                 SalaryTextBox.Enabled = true;
```

```
169                 }
170                 else
171                 {
172                     JobTextBox.Enabled = false;
173                     SalaryTextBox.Enabled = false;
174                     JobTextBox.Text = "";
175                     SalaryTextBox.Text = "";
176                     }
177                 }
178 
179             private void StudentsRadioButton_CheckedChanged(object sender, EventArgs e)
180             {
181                 SchoolComboBox.Enabled = StudentsRadioButton.Checked;
182             }
183 
184             //laver sort på salary med togling den ene eller anden vej.
185             private void SortSalary_Click(object sender, EventArgs e)
186             {
187                 var list = archive.Employees;
188                 _asc = !_asc;
189                 if (_asc)
190                 {
191                     list.Sort(new SortSalaryAscendingHelper());
192                 }
193                 else
194                 {
195                     list.Sort(new SortSalaryDescendingHelper());
196                 }
197                 dataGridView.DataSource = list;
198             }
199 
200             //sort på age - ser anderledes ud, da SortAgeDescending er default via IComparable
201             private void SortAge_Click(object sender, EventArgs e)
202             {
203                 var list = archive.ShowAllPersons();
204                 _ascAge = !_ascAge;
205                 if (_ascAge)
206                 {
207                     list.Sort(new SortAgeAscendingHelper());
208                 }
209                 else
210                 {
```

```
211                     list.Sort();
212                 }
213                 dataGridView.DataSource = list;
214             }
215         }
216     }
217 
```

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

namespace Archive.UnitTests
{
    [TestClass]
    public class UnitTest1
    {
        Archive archive = new Archive(new FilePrint(@"C:\\Zirodevelopment\Csharp\logTest.txt"));

        [TestMethod]
        public void CreateOrAlterStuden_StudenExsist_StudentChanged()
        {
            //arrange
            archive.CreateOrAlterStudent("Jens", 20, "abs 12", 2500, 9999999, School.AalborgUniversity);

            //act
            archive.CreateOrAlterStudent("Jens", 20, "abs 12", 2500, 9999999, School.AarhusUniversity);
            var result = archive.Students[0].School;

            //assert
            Assert.AreEqual(result, School.AarhusUniversity);
        }

        [TestMethod]
        public void CreateOrAlterStuden_StudenIsNew_StudentCreated()
        {
            //arrange
            var startUp = archive.Students.Count;

            //act
            archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 9999999, School.AarhusUniversity);
            var result = archive.Students.Count;

            //assert
            Assert.AreEqual(result, startUp+1);
        }

        [TestMethod]
        public void CreateOrAlterEmployee_EmployeeExsist_EmployeeChanged()
        {
```

```
43              //arrange
44              archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 9999999, "tester",null);
45
46              //act
47              archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 9999999, "tester", 300000);
48              var result = archive.Employees[0].Salary;
49
50              //assert
51              Assert.AreEqual(result, 300000);
52          }
53
54          [TestMethod]
55          public void CreateOrAlterEmployee_EmployeeIsNew_EmployeeCreated()
56          {
57              //arrange
58              var startUp = archive.Employees.Count;
59
60              //act
61              archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 9999999, "tester", 300000);
62              var result = archive.Employees.Count;
63
64              //assert
65              Assert.AreEqual(result, startUp + 1);
66          }
67
68          [TestMethod]
69          public void DeletePerson_DeleteStudent_StudentIsDeleted()
70          {
71              //arrange
72              archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 99999999, School.AarhusUniversity);
73              var startUp = archive.Students.Count;
74
75              //act
76              archive.DeletePerson(99999999);
77              var result = archive.Students.Count;
78
79              //assert
80              Assert.AreEqual(result, startUp - 1);
81          }
82
83          [TestMethod]
84          public void DeletePerson_DeleteEmploee_EmployeeIsDeleted()
```

```
85          {
86              //arrange
87              archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 99999999, "tester", 300000);
88              var startUp = archive.Employees.Count;
89
90              //act
91              archive.DeletePerson(99999999);
92              var result = archive.Employees.Count;
93
94              //assert
95              Assert.AreEqual(result, startUp - 1);
96          }
97
98          [TestMethod]
99          public void DeletePerson_NotFound_NothingIsDeleted()
100         {
101             //arrange
102             archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 99999999, School.AarhusUniversity);
103             var startUp = archive.ShowNumberOfPersons();
104
105             //act
106             archive.DeletePerson(99991239);
107             var result = archive.ShowNumberOfPersons();
108
109             //assert
110             Assert.AreEqual(result, startUp);
111         }
112
113         [TestMethod]
114         public void ShowNumberOfPersons_NoInput_2persons()
115         {
116             //arrange
117             archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 99999999, School.AarhusUniversity);
118             archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 99999999, "tester", 300000);
119
120             //act
121             var result = archive.ShowNumberOfPersons();
122
123             //assert
124             Assert.AreEqual(result, 2);
125         }
126
```

```
127         [TestMethod]
128         public void ShowNumberOfPersons_True_person()
129         {
130             //arrange
131             archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 99999999, School.AarhusUniversity);
132             archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 99999999, "tester", 300000);
133 
134             //act
135             var result = archive.ShowNumberOfPersons(true);
136 
137             //assert
138             Assert.AreEqual(result, 1);
139         }
140 
141         [TestMethod]
142         public void ShowNumberOfPersons_False_person()
143         {
144             //arrange
145             archive.CreateOrAlterStudent("Kurt", 20, "abs 12", 2500, 99999999, School.AarhusUniversity);
146             archive.CreateOrAlterEmployee("Jensen", 20, "abs 12", 2500, 99999999, "tester", 300000);
147 
148             //act
149             var result = archive.ShowNumberOfPersons(false);
150 
151             //assert
152             Assert.AreEqual(result, 1);
153         }
154 
155     }
156 
157 
158 }
159 
```

------ArchiveInput--------

```
namespace Archive
{
	partial class ArchiveInput
	{
		/// <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(ArchiveInput));
			this.groupBox1 = new System.Windows.Forms.GroupBox();
			this.SchoolComboBox = new System.Windows.Forms.ComboBox();
			this.CreateOrAlterButton = new System.Windows.Forms.Button();
			this.label9 = new System.Windows.Forms.Label();
			this.SalaryTextBox = new System.Windows.Forms.TextBox();
			this.label8 = new System.Windows.Forms.Label();
			this.JobTextBox = new System.Windows.Forms.TextBox();
			this.label7 = new System.Windows.Forms.Label();
			this.label6 = new System.Windows.Forms.Label();
			this.PhoneNumberTextBox = new System.Windows.Forms.TextBox();
			this.label5 = new System.Windows.Forms.Label();
			this.PostNumberTextBox = new System.Windows.Forms.TextBox();
			this.label4 = new System.Windows.Forms.Label();
			this.AdressTextBox = new System.Windows.Forms.TextBox();
			this.label1 = new System.Windows.Forms.Label();
			this.AgeTextBox = new System.Windows.Forms.TextBox();
			this.label3 = new System.Windows.Forms.Label();
			this.EmployeeRadioButton = new System.Windows.Forms.RadioButton();
			this.StudentsRadioButton = new System.Windows.Forms.RadioButton();
			this.label2 = new System.Windows.Forms.Label();
			this.NameTextBox = new System.Windows.Forms.TextBox();
			this.groupBox2 = new System.Windows.Forms.GroupBox();
			this.groupBox5 = new System.Windows.Forms.GroupBox();
			this.SortSalary = new System.Windows.Forms.Button();
			this.SortAge = new System.Windows.Forms.Button();
			this.ShowAllPersonsButton = new System.Windows.Forms.Button();
			this.label13 = new System.Windows.Forms.Label();
			this.label12 = new System.Windows.Forms.Label();
			this.MaxSalaryButton = new System.Windows.Forms.Button();
			this.MinSalaryButton = new System.Windows.Forms.Button();
			this.MinAgeButton = new System.Windows.Forms.Button();
			this.MaxAgeButton = new System.Windows.Forms.Button();
			this.label11 = new System.Windows.Forms.Label();
			this.dataGridView = new System.Windows.Forms.DataGridView();
```

```
this.PersonWithPhoneNumberTextBoks = new System.Windows.Forms.TextBox();
this.PersonWithPhoneNumberButton = new System.Windows.Forms.Button();
this.groupBox4 = new System.Windows.Forms.GroupBox();
this.ShowNumbersOfLabel = new System.Windows.Forms.Label();
this.ShowNumberOfPersonButton = new System.Windows.Forms.Button();
this.ShowNumberOfEmployeesButton = new System.Windows.Forms.Button();
this.ShowNumberOfStudentsButton = new System.Windows.Forms.Button();
this.DeletePersonTextBoks = new System.Windows.Forms.TextBox();
this.label10 = new System.Windows.Forms.Label();
this.DeleteButton = new System.Windows.Forms.Button();
this.groupBox3 = new System.Windows.Forms.GroupBox();
this.groupBox6 = new System.Windows.Forms.GroupBox();
this.label14 = new System.Windows.Forms.Label();
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.groupBox5.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.dataGridView)).BeginInit();
this.groupBox4.SuspendLayout();
this.groupBox3.SuspendLayout();
this.groupBox6.SuspendLayout();
this.SuspendLayout();
//
// groupBox1
//
this.groupBox1.Controls.Add(this.SchoolComboBox);
this.groupBox1.Controls.Add(this.CreateOrAlterButton);
this.groupBox1.Controls.Add(this.label9);
this.groupBox1.Controls.Add(this.SalaryTextBox);
this.groupBox1.Controls.Add(this.label8);
this.groupBox1.Controls.Add(this.JobTextBox);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Controls.Add(this.label6);
this.groupBox1.Controls.Add(this.PhoneNumberTextBox);
this.groupBox1.Controls.Add(this.label5);
this.groupBox1.Controls.Add(this.PostNumberTextBox);
this.groupBox1.Controls.Add(this.label4);
this.groupBox1.Controls.Add(this.AdressTextBox);
this.groupBox1.Controls.Add(this.label1);
this.groupBox1.Controls.Add(this.AgeTextBox);
this.groupBox1.Controls.Add(this.label3);
this.groupBox1.Controls.Add(this.EmployeeRadioButton);
this.groupBox1.Controls.Add(this.StudentsRadioButton);
this.groupBox1.Controls.Add(this.label2);
this.groupBox1.Controls.Add(this.NameTextBox);
this.groupBox1.Location = new System.Drawing.Point(27, 26);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(382, 333);
this.groupBox1.TabIndex = 0;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Oprettelse eller ændring af person";
//
// SchoolComboBox
//
this.SchoolComboBox.FormattingEnabled = true;
this.SchoolComboBox.Location = new System.Drawing.Point(174, 190);
this.SchoolComboBox.Name = "SchoolComboBox";
this.SchoolComboBox.Size = new System.Drawing.Size(181, 20);
this.SchoolComboBox.TabIndex = 22;
//
// CreateOrAlterButton
//
this.CreateOrAlterButton.Location = new System.Drawing.Point(173, 281);
this.CreateOrAlterButton.Name = "CreateOrAlterButton";
this.CreateOrAlterButton.Size = new System.Drawing.Size(182, 23);
this.CreateOrAlterButton.TabIndex = 21;
this.CreateOrAlterButton.Text = "Opret eller ret";
this.CreateOrAlterButton.UseVisualStyleBackColor = true;
this.CreateOrAlterButton.Click += new System.EventHandler(this.CreateOrAlterButton_Click);
```

```
//
// label9
//
this.label9.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(17, 251);
this.label9.MinimumSize = new System.Drawing.Size(150, 18);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(150, 18);
this.label9.TabIndex = 20;
this.label9.Text = "Løn";
//
// SalaryTextBox
//
this.SalaryTextBox.Enabled = false;
this.SalaryTextBox.Location = new System.Drawing.Point(173, 248);
this.SalaryTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.SalaryTextBox.Name = "SalaryTextBox";
this.SalaryTextBox.Size = new System.Drawing.Size(182, 20);
this.SalaryTextBox.TabIndex = 19;
//
// label8
//
this.label8.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(17, 222);
this.label8.MinimumSize = new System.Drawing.Size(150, 18);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(150, 18);
this.label8.TabIndex = 18;
this.label8.Text = "Arbejde";
//
// JobTextBox
//
this.JobTextBox.Enabled = false;
this.JobTextBox.Location = new System.Drawing.Point(173, 219);
this.JobTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.JobTextBox.Name = "JobTextBox";
this.JobTextBox.Size = new System.Drawing.Size(182, 20);
this.JobTextBox.TabIndex = 17;
//
// label7
//
this.label7.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(17, 193);
this.label7.MinimumSize = new System.Drawing.Size(150, 18);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(150, 18);
this.label7.TabIndex = 16;
this.label7.Text = "Skole/universitet";
//
// label6
//
this.label6.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(17, 164);
this.label6.MinimumSize = new System.Drawing.Size(150, 18);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(150, 18);
this.label6.TabIndex = 14;
this.label6.Text = "Telefonnummer";
//
// PhoneNumberTextBox
//
this.PhoneNumberTextBox.Location = new System.Drawing.Point(173, 161);
this.PhoneNumberTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.PhoneNumberTextBox.Name = "PhoneNumberTextBox";
```

```
this.PhoneNumberTextBox.Size = new System.Drawing.Size(182, 20);
this.PhoneNumberTextBox.TabIndex = 13;
//
// label5
//
this.label5.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(17, 136);
this.label5.MinimumSize = new System.Drawing.Size(150, 18);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(150, 18);
this.label5.TabIndex = 12;
this.label5.Text = "Postnummer";
//
// PostNumberTextBox
//
this.PostNumberTextBox.Location = new System.Drawing.Point(173, 133);
this.PostNumberTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.PostNumberTextBox.Name = "PostNumberTextBox";
this.PostNumberTextBox.Size = new System.Drawing.Size(182, 20);
this.PostNumberTextBox.TabIndex = 11;
//
// label4
//
this.label4.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(17, 108);
this.label4.MinimumSize = new System.Drawing.Size(150, 18);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(150, 18);
this.label4.TabIndex = 10;
this.label4.Text = "Adresse";
//
// AdressTextBox
//
this.AdressTextBox.Location = new System.Drawing.Point(173, 105);
this.AdressTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.AdressTextBox.Name = "AdressTextBox";
this.AdressTextBox.Size = new System.Drawing.Size(182, 20);
this.AdressTextBox.TabIndex = 9;
//
// label1
//
this.label1.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(17, 81);
this.label1.MinimumSize = new System.Drawing.Size(150, 18);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(150, 18);
this.label1.TabIndex = 8;
this.label1.Text = "Alder";
//
// AgeTextBox
//
this.AgeTextBox.Location = new System.Drawing.Point(173, 78);
this.AgeTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.AgeTextBox.Name = "AgeTextBox";
this.AgeTextBox.Size = new System.Drawing.Size(182, 20);
this.AgeTextBox.TabIndex = 7;
//
// label3
//
this.label3.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(17, 53);
this.label3.MinimumSize = new System.Drawing.Size(150, 18);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(150, 18);
```

```
this.label3.TabIndex = 6;
this.label3.Text = "Navn";
//
// EmployeeRadioButton
//
this.EmployeeRadioButton.AutoSize = true;
this.EmployeeRadioButton.Location = new System.Drawing.Point(260, 24);
this.EmployeeRadioButton.Name = "EmployeeRadioButton";
this.EmployeeRadioButton.Size = new System.Drawing.Size(77, 16);
this.EmployeeRadioButton.TabIndex = 5;
this.EmployeeRadioButton.Text = "Employee";
this.EmployeeRadioButton.UseVisualStyleBackColor = true;
this.EmployeeRadioButton.CheckedChanged += new System.EventHandler(this.EmployeeRadioButton_CheckedChanged);
//
// StudentsRadioButton
//
this.StudentsRadioButton.AutoSize = true;
this.StudentsRadioButton.Checked = true;
this.StudentsRadioButton.Location = new System.Drawing.Point(173, 24);
this.StudentsRadioButton.Name = "StudentsRadioButton";
this.StudentsRadioButton.Size = new System.Drawing.Size(72, 16);
this.StudentsRadioButton.TabIndex = 4;
this.StudentsRadioButton.TabStop = true;
this.StudentsRadioButton.Text = "Students";
this.StudentsRadioButton.UseVisualStyleBackColor = true;
this.StudentsRadioButton.CheckedChanged += new System.EventHandler(this.StudentsRadioButton_CheckedChanged);
//
// label2
//
this.label2.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(17, 26);
this.label2.MinimumSize = new System.Drawing.Size(150, 18);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(150, 18);
this.label2.TabIndex = 3;
this.label2.Text = "Erhverv";
//
// NameTextBox
//
this.NameTextBox.Location = new System.Drawing.Point(173, 50);
this.NameTextBox.MinimumSize = new System.Drawing.Size(150, 18);
this.NameTextBox.Name = "NameTextBox";
this.NameTextBox.Size = new System.Drawing.Size(182, 20);
this.NameTextBox.TabIndex = 1;
//
// groupBox2
//
this.groupBox2.Controls.Add(this.groupBox5);
this.groupBox2.Controls.Add(this.groupBox4);
this.groupBox2.Location = new System.Drawing.Point(447, 26);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(594, 555);
this.groupBox2.TabIndex = 1;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Visning af data";
//
// groupBox5
//
this.groupBox5.Controls.Add(this.SortSalary);
this.groupBox5.Controls.Add(this.SortAge);
this.groupBox5.Controls.Add(this.ShowAllPersonsButton);
this.groupBox5.Controls.Add(this.label13);
this.groupBox5.Controls.Add(this.label12);
this.groupBox5.Controls.Add(this.MaxSalaryButton);
this.groupBox5.Controls.Add(this.MinSalaryButton);
this.groupBox5.Controls.Add(this.MinAgeButton);
this.groupBox5.Controls.Add(this.MaxAgeButton);
```

```
this.groupBox5.Controls.Add(this.label11);
this.groupBox5.Controls.Add(this.dataGridView);
this.groupBox5.Controls.Add(this.PersonWithPhoneNumberTextBoks);
this.groupBox5.Controls.Add(this.PersonWithPhoneNumberButton);
this.groupBox5.Location = new System.Drawing.Point(30, 120);
this.groupBox5.Name = "groupBox5";
this.groupBox5.Size = new System.Drawing.Size(542, 419);
this.groupBox5.TabIndex = 26;
this.groupBox5.TabStop = false;
this.groupBox5.Text = "Vis personer der matcher kriterie";
//
// SortSalary
//
this.SortSalary.Location = new System.Drawing.Point(190, 376);
this.SortSalary.Name = "SortSalary";
this.SortSalary.Size = new System.Drawing.Size(150, 23);
this.SortSalary.TabIndex = 34;
this.SortSalary.Text = "Sort on Salary";
this.SortSalary.UseVisualStyleBackColor = true;
this.SortSalary.Click += new System.EventHandler(this.SortSalary_Click);
//
// SortAge
//
this.SortAge.Location = new System.Drawing.Point(25, 376);
this.SortAge.Name = "SortAge";
this.SortAge.Size = new System.Drawing.Size(150, 23);
this.SortAge.TabIndex = 33;
this.SortAge.Text = "Sort on Age";
this.SortAge.UseVisualStyleBackColor = true;
this.SortAge.Click += new System.EventHandler(this.SortAge_Click);
//
// ShowAllPersonsButton
//
this.ShowAllPersonsButton.Location = new System.Drawing.Point(357, 376);
this.ShowAllPersonsButton.Name = "ShowAllPersonsButton";
this.ShowAllPersonsButton.Size = new System.Drawing.Size(150, 23);
this.ShowAllPersonsButton.TabIndex = 32;
this.ShowAllPersonsButton.Text = "Vis alle personer";
this.ShowAllPersonsButton.UseVisualStyleBackColor = true;
this.ShowAllPersonsButton.Click += new System.EventHandler(this.ShowAllPersonsButton_Click);
//
// label13
//
this.label13.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label13.AutoSize = true;
this.label13.Location = new System.Drawing.Point(343, 34);
this.label13.MinimumSize = new System.Drawing.Size(150, 18);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(150, 18);
this.label13.TabIndex = 31;
this.label13.Text = "Løn";
//
// label12
//
this.label12.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label12.AutoSize = true;
this.label12.Location = new System.Drawing.Point(187, 34);
this.label12.MinimumSize = new System.Drawing.Size(150, 18);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(150, 18);
this.label12.TabIndex = 27;
this.label12.Text = "Alder";
//
// MaxSalaryButton
//
this.MaxSalaryButton.Location = new System.Drawing.Point(345, 84);
this.MaxSalaryButton.Name = "MaxSalaryButton";
this.MaxSalaryButton.Size = new System.Drawing.Size(150, 23);
```

```
                this.MaxSalaryButton.TabIndex = 30;
                this.MaxSalaryButton.Text = "Højeste løn";
                this.MaxSalaryButton.UseVisualStyleBackColor = true;
                this.MaxSalaryButton.Click += new System.EventHandler(this.MaxSalaryButton_Click);
                //
                // MinSalaryButton
                //
                this.MinSalaryButton.Location = new System.Drawing.Point(345, 55);
                this.MinSalaryButton.Name = "MinSalaryButton";
                this.MinSalaryButton.Size = new System.Drawing.Size(150, 23);
                this.MinSalaryButton.TabIndex = 29;
                this.MinSalaryButton.Text = "Mindste løn";
                this.MinSalaryButton.UseVisualStyleBackColor = true;
                this.MinSalaryButton.Click += new System.EventHandler(this.MinSalaryButton_Click);
                //
                // MinAgeButton
                //
                this.MinAgeButton.Location = new System.Drawing.Point(189, 55);
                this.MinAgeButton.Name = "MinAgeButton";
                this.MinAgeButton.Size = new System.Drawing.Size(150, 23);
                this.MinAgeButton.TabIndex = 28;
                this.MinAgeButton.Text = "Mindste alder";
                this.MinAgeButton.UseVisualStyleBackColor = true;
                this.MinAgeButton.Click += new System.EventHandler(this.MinAgeButton_Click);
                //
                // MaxAgeButton
                //
                this.MaxAgeButton.Location = new System.Drawing.Point(189, 84);
                this.MaxAgeButton.Name = "MaxAgeButton";
                this.MaxAgeButton.Size = new System.Drawing.Size(150, 23);
                this.MaxAgeButton.TabIndex = 27;
                this.MaxAgeButton.Text = "Højeste alder";
                this.MaxAgeButton.UseVisualStyleBackColor = true;
                this.MaxAgeButton.Click += new System.EventHandler(this.MaxAgeButton_Click);
                //
                // label11
                //
                this.label11.Anchor = System.Windows.Forms.AnchorStyles.Left;
                this.label11.AutoSize = true;
                this.label11.Location = new System.Drawing.Point(25, 34);
                this.label11.MinimumSize = new System.Drawing.Size(150, 18);
                this.label11.Name = "label11";
                this.label11.Size = new System.Drawing.Size(150, 18);
                this.label11.TabIndex = 21;
                this.label11.Text = "Telefonnr.";
                //
                // dataGridView
                //
                this.dataGridView.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
                this.dataGridView.Location = new System.Drawing.Point(25, 131);
                this.dataGridView.Name = "dataGridView";
                this.dataGridView.RowTemplate.Height = 24;
                this.dataGridView.Size = new System.Drawing.Size(482, 239);
                this.dataGridView.TabIndex = 26;
                //
                // PersonWithPhoneNumberTextBoks
                //
                this.PersonWithPhoneNumberTextBoks.Location = new System.Drawing.Point(25, 55);
                this.PersonWithPhoneNumberTextBoks.Name = "PersonWithPhoneNumberTextBoks";
                this.PersonWithPhoneNumberTextBoks.Size = new System.Drawing.Size(150, 20);
                this.PersonWithPhoneNumberTextBoks.TabIndex = 25;
                //
                // PersonWithPhoneNumberButton
                //
                this.PersonWithPhoneNumberButton.Location = new System.Drawing.Point(25, 84);
                this.PersonWithPhoneNumberButton.Name = "PersonWithPhoneNumberButton";
                this.PersonWithPhoneNumberButton.Size = new System.Drawing.Size(150, 23);
```

```
this.PersonWithPhoneNumberButton.TabIndex = 23;
this.PersonWithPhoneNumberButton.Text = "Find Person";
this.PersonWithPhoneNumberButton.UseVisualStyleBackColor = true;
this.PersonWithPhoneNumberButton.Click += new System.EventHandler(this.PersonWithPhoneNumberButton_Click);
//
// groupBox4
//
this.groupBox4.Controls.Add(this.ShowNumbersOfLabel);
this.groupBox4.Controls.Add(this.ShowNumberOfPersonButton);
this.groupBox4.Controls.Add(this.ShowNumberOfEmployeesButton);
this.groupBox4.Controls.Add(this.ShowNumberOfStudentsButton);
this.groupBox4.Location = new System.Drawing.Point(30, 35);
this.groupBox4.Name = "groupBox4";
this.groupBox4.Size = new System.Drawing.Size(542, 64);
this.groupBox4.TabIndex = 25;
this.groupBox4.TabStop = false;
this.groupBox4.Text = "Vis antallet af:";
//
// ShowNumbersOfLabel
//
this.ShowNumbersOfLabel.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.ShowNumbersOfLabel.AutoSize = true;
this.ShowNumbersOfLabel.Location = new System.Drawing.Point(370, 27);
this.ShowNumbersOfLabel.MinimumSize = new System.Drawing.Size(150, 18);
this.ShowNumbersOfLabel.Name = "ShowNumbersOfLabel";
this.ShowNumbersOfLabel.Size = new System.Drawing.Size(150, 18);
this.ShowNumbersOfLabel.TabIndex = 11;
this.ShowNumbersOfLabel.Text = "tryk på en knap";
this.ShowNumbersOfLabel.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// ShowNumberOfPersonButton
//
this.ShowNumberOfPersonButton.Location = new System.Drawing.Point(16, 22);
this.ShowNumberOfPersonButton.Name = "ShowNumberOfPersonButton";
this.ShowNumberOfPersonButton.Size = new System.Drawing.Size(109, 23);
this.ShowNumberOfPersonButton.TabIndex = 22;
this.ShowNumberOfPersonButton.Text = "Personer";
this.ShowNumberOfPersonButton.UseVisualStyleBackColor = true;
this.ShowNumberOfPersonButton.Click += new System.EventHandler(this.ShowNumberOfPersonButton_Click);
//
// ShowNumberOfEmployeesButton
//
this.ShowNumberOfEmployeesButton.Location = new System.Drawing.Point(246, 22);
this.ShowNumberOfEmployeesButton.Name = "ShowNumberOfEmployeesButton";
this.ShowNumberOfEmployeesButton.Size = new System.Drawing.Size(109, 23);
this.ShowNumberOfEmployeesButton.TabIndex = 24;
this.ShowNumberOfEmployeesButton.Text = "Medarbjedere";
this.ShowNumberOfEmployeesButton.UseVisualStyleBackColor = true;
this.ShowNumberOfEmployeesButton.Click += new System.EventHandler(this.ShowNumberOfEmployeesButton_Click);
//
// ShowNumberOfStudentsButton
//
this.ShowNumberOfStudentsButton.Location = new System.Drawing.Point(131, 22);
this.ShowNumberOfStudentsButton.Name = "ShowNumberOfStudentsButton";
this.ShowNumberOfStudentsButton.Size = new System.Drawing.Size(109, 23);
this.ShowNumberOfStudentsButton.TabIndex = 23;
this.ShowNumberOfStudentsButton.Text = "Studerende";
this.ShowNumberOfStudentsButton.UseVisualStyleBackColor = true;
this.ShowNumberOfStudentsButton.Click += new System.EventHandler(this.ShowNumberOfStudentsButton_Click);
//
// DeletePersonTextBoks
//
this.DeletePersonTextBoks.Location = new System.Drawing.Point(173, 27);
this.DeletePersonTextBoks.MinimumSize = new System.Drawing.Size(150, 18);
this.DeletePersonTextBoks.Name = "DeletePersonTextBoks";
this.DeletePersonTextBoks.Size = new System.Drawing.Size(182, 20);
this.DeletePersonTextBoks.TabIndex = 22;
//
```

```
                        // label10
                        //
                        this.label10.Anchor = System.Windows.Forms.AnchorStyles.Left;
                        this.label10.AutoSize = true;
                        this.label10.Location = new System.Drawing.Point(17, 30);
                        this.label10.MinimumSize = new System.Drawing.Size(150, 18);
                        this.label10.Name = "label10";
                        this.label10.Size = new System.Drawing.Size(150, 18);
                        this.label10.TabIndex = 23;
                        this.label10.Text = "Telefonnr.";
                        //
                        // DeleteButton
                        //
                        this.DeleteButton.Location = new System.Drawing.Point(173, 60);
                        this.DeleteButton.Name = "DeleteButton";
                        this.DeleteButton.Size = new System.Drawing.Size(182, 23);
                        this.DeleteButton.TabIndex = 24;
                        this.DeleteButton.Text = "Slet person";
                        this.DeleteButton.UseVisualStyleBackColor = true;
                        this.DeleteButton.Click += new System.EventHandler(this.DeleteButton_Click);
                        //
                        // groupBox3
                        //
                        this.groupBox3.Controls.Add(this.DeleteButton);
                        this.groupBox3.Controls.Add(this.label10);
                        this.groupBox3.Controls.Add(this.DeletePersonTextBoks);
                        this.groupBox3.Location = new System.Drawing.Point(27, 365);
                        this.groupBox3.Name = "groupBox3";
                        this.groupBox3.Size = new System.Drawing.Size(382, 100);
                        this.groupBox3.TabIndex = 2;
                        this.groupBox3.TabStop = false;
                        this.groupBox3.Text = "Slet en person fra kartoteket";
                        //
                        // groupBox6
                        //
                        this.groupBox6.Controls.Add(this.label14);
                        this.groupBox6.Location = new System.Drawing.Point(27, 474);
                        this.groupBox6.Name = "groupBox6";
                        this.groupBox6.Size = new System.Drawing.Size(382, 107);
                        this.groupBox6.TabIndex = 3;
                        this.groupBox6.TabStop = false;
                        this.groupBox6.Text = "Log";
                        //
                        // label14
                        //
                        this.label14.AutoSize = true;
                        this.label14.Location = new System.Drawing.Point(19, 35);
                        this.label14.Name = "label14";
                        this.label14.Size = new System.Drawing.Size(176, 12);
                        this.label14.TabIndex = 0;
                        this.label14.Text = "Log er lokaliseret på: C:\\\\log.txt";
                        //
                        // ArchiveInput
                        //
                        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
                        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
                        this.BackColor = System.Drawing.SystemColors.ActiveCaption;
                        this.ClientSize = new System.Drawing.Size(1073, 593);
                        this.Controls.Add(this.groupBox6);
                        this.Controls.Add(this.groupBox3);
                        this.Controls.Add(this.groupBox2);
                        this.Controls.Add(this.groupBox1);
                        this.Font = new System.Drawing.Font("Arial Rounded MT Bold", 8.25F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
                        this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
                        this.Name = "ArchiveInput";
                        this.Text = "Kartotek over medarbejdere og studerende.";
                        this.groupBox1.ResumeLayout(false);
```

```
            this.groupBox1.PerformLayout();
            this.groupBox2.ResumeLayout(false);
            this.groupBox5.ResumeLayout(false);
            this.groupBox5.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.dataGridView)).EndInit();
            this.groupBox4.ResumeLayout(false);
            this.groupBox4.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            this.groupBox3.PerformLayout();
            this.groupBox6.ResumeLayout(false);
            this.groupBox6.PerformLayout();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.TextBox NameTextBox;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.RadioButton EmployeeRadioButton;
        private System.Windows.Forms.RadioButton StudentsRadioButton;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Button CreateOrAlterButton;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.TextBox SalaryTextBox;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.TextBox JobTextBox;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.TextBox PhoneNumberTextBox;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox PostNumberTextBox;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.TextBox AdressTextBox;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox AgeTextBox;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.GroupBox groupBox5;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.Label label12;
        private System.Windows.Forms.Button MaxSalaryButton;
        private System.Windows.Forms.Button MinSalaryButton;
        private System.Windows.Forms.Button MinAgeButton;
        private System.Windows.Forms.Button MaxAgeButton;
        private System.Windows.Forms.Label label11;
        private System.Windows.Forms.DataGridView dataGridView;
        private System.Windows.Forms.TextBox PersonWithPhoneNumberTextBoks;
        private System.Windows.Forms.Button PersonWithPhoneNumberButton;
        private System.Windows.Forms.GroupBox groupBox4;
        private System.Windows.Forms.Button ShowNumberOfPersonButton;
        private System.Windows.Forms.Button ShowNumberOfEmployeesButton;
        private System.Windows.Forms.Button ShowNumberOfStudentsButton;
        private System.Windows.Forms.TextBox DeletePersonTextBoks;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Button DeleteButton;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.GroupBox groupBox6;
        private System.Windows.Forms.Button ShowAllPersonsButton;
        private System.Windows.Forms.ComboBox SchoolComboBox;
        private System.Windows.Forms.Label ShowNumbersOfLabel;
        private System.Windows.Forms.Button SortSalary;
        private System.Windows.Forms.Button SortAge;
        private System.Windows.Forms.Label label14;
    }
}
```