

HÖHERE BERUFSFACHSCHULE FÜR WIRTSCHAFTSINFORMATIK

KURT-SCHUMACHER-STR. 20

66130 SAARBRÜCKEN

HAUSARBEIT

IM BILDUNGSGANG

“STAATLICH GEPRÜFTE/R WIRTSCHAFTSINFORMATIKER/IN”

<i>Vorgelegt von:</i>	<i>Vanessa Schmelzer</i>
<i>Klasse:</i>	<i>WI21ZAS</i>
<i>Adresse:</i>	<i>Luisenstraße 10</i>
<i>Ort:</i>	<i>66538 Neunkirchen</i>
<i>E-Mail:</i>	<i>Vanessa.schmelzer00@web.de</i>
<i>Abgabetermin:</i>	<i>27.05.2024</i>
<i>Betreuer:</i>	<i>Herr Friedrich</i>

Einleitung.....	3
Nicht umgesetzte Kriterien (siehe Pflichtenheft) .....	4
Zusätzliche Änderungen .....	4
Erstellung des Programms .....	4
Framework .....	4
Datenbank .....	5
Back-End-Code .....	6
Sprach Einstellung .....	6
SubWindow – Aufgaben Fenster.....	9
Front-End-Code .....	11
Window1 – Sprach Selektion .....	12
Nutzerhandbuch.....	16
Quellenangabe .....	18
Eidesstattliche Erklärung.....	19

## Einleitung

In Anbetracht des Abschlusses an der Höheren Berufsfachschule für Wirtschaftsinformatik war es mir erlaubt mich für ein Abschlussprojekt zu entscheiden. Ich habe mich für eine Lernsoftware für Basis-Rechnungswesen entschieden, da es immer mehr Menschen gibt, die Schwierigkeiten haben, sich zurechtzufinden.

Mein Persönliches befinden ist es das, dass heutige Schulsystem veraltet, sei welches vielen Menschen die Möglichkeit nimmt etwas zu lernen durch Faktoren wie Demotivation, starke Auslegung auf Extraversion oder auch strenges bzw. Unverständliches Benotungssystem und vieles Mehr.

Zur heutigen Zeit ist es leichter der Mensch von davon abzubringen eine Leistung zu erbringen, wenn es Mangel nach Hilfe, Motivation und oder Verständnis mit sich bringt.

Mein Ziel ist es ein Kleines und übersichtliches Programm zu liefern in dem Schlüsselpunkte behandelt werden, um ein Grundlegendes Verständnis wiederherzustellen.

Mithilfe dieses Programmes soll es den Mitmenschen leichter fallen sich in Rechnungswesen hineinzufinden und ihr Wissen etwas aufzufrischen.

Kunden können mithilfe von leicht verständlicher Theorie und Aufgaben ihr Wissen Trainieren und die Sprache lässt sich von Deutsch auf Englisch ändern.

Zu jeder Frage gibt es eine Ausführliche Lösung die Lücken zu füllen versucht.

## Nicht umgesetzte Kriterien (siehe Pflichtenheft)

- Die Vermittlung der Wiedergabe der Inhalte kann Verständnis Probleme aufweisen.

Die aus Zeitlichen Gründen nicht gefertigten Teile werden überdacht.

## Zusätzliche Änderungen

- Die Wiedergabe des Feedbacks erfolgt durch einen Extra Button welches ein Fenster hervorruft.
- Navigation zwischen den Fragestellungen möglich vorwärts als auch rückwärts.
- Das Programm beläuft sich auf Nur Deutsch lässt jedoch Spielraum für das hinzufügen weiterer Sprachen

## Erstellung des Programms

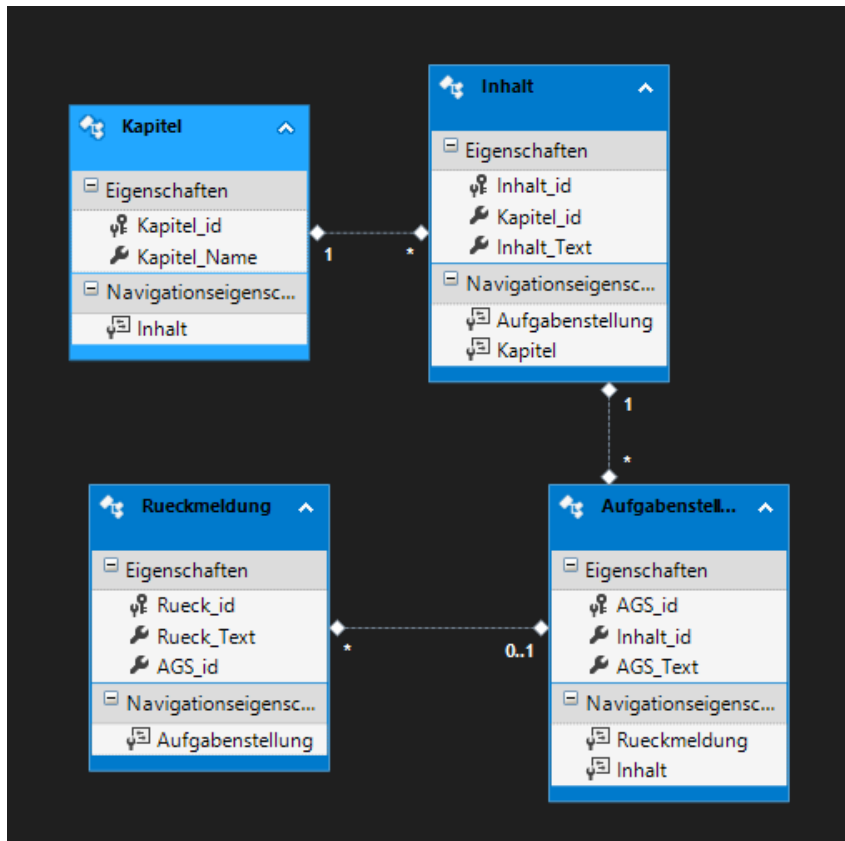
### Framework

Für die Framework wurde verwendet:

- Entity Framework: „Entity Framework Core ist ein moderner Objekt-Datenbank-Mapper für .NET. Er unterstützt LINQ-Abfragen, Änderungsnachverfolgung, Updates und Schemamigrationen. EF Core funktioniert mit vielen Datenbanken, einschließlich SQL-Datenbank (lokal und Azure), SQLite, MySQL, PostgreSQL und Azure Cosmos DB.“ (Microsoft, 2022)

Entity Framework wird in diesem Projekt für den Datenverkehr zwischen der Datenbank und den abfragen des Codes benutzt.

## Datenbank



```

CREATE TABLE Kapitel (
    Kapitel_id INT IDENTITY (1, 1) PRIMARY KEY,
    Kapitel_Name nvarchar(100) not null,
);

CREATE TABLE Inhalt (
    Inhalt_id INT IDENTITY (1, 1) PRIMARY KEY,
    Kapitel_id int not null,
    Inhalt_Text ntext not null,
    constraint fk_KapitelInhalt foreign key (Kapitel_id) references Kapitel(Kapitel_id)
);

CREATE TABLE Aufgabenstellung (
    AGS_id INT IDENTITY (1, 1) PRIMARY KEY,
    --Kapitel_id int not null,
    Inhalt_id int not null,
    AGS_Text ntext not null,
    -- CONSTRAINT fk_KapitelAufgabenstellung FOREIGN KEY (Kapitel_id) REFERENCES Kapitel(Kapitel_id)
    CONSTRAINT fk_InhaltAufgabenstellung FOREIGN KEY (Inhalt_id) REFERENCES Inhalt(Inhalt_id)
);

CREATE TABLE Rueckmeldung (
    Rueck_id INT IDENTITY (1, 1) PRIMARY KEY,
    Rueck_Text ntext not null,
    AGS_id int foreign key references Aufgabenstellung(AGS_id)
);

```

Erstellung der Tabellen für die Lernsoftware:

1. **Tabellenerstellung:** Zunächst werden die erforderlichen Tabellen in der Datenbank erstellt, um die verschiedenen Arten von Daten effizient zu speichern.
2. **Datenverknüpfung:** Die erstellten Tabellen werden miteinander verknüpft, um eine Beziehung zwischen ihnen herzustellen. Dies ermöglicht es, eine logische Kette zu bilden, die eine vereinfachte und effektive Abrufung der Daten ermöglicht.
3. **Relationale Datenbank:** Durch die Verknüpfung der Tabellen in einer relationalen Datenbankstruktur können komplexe Beziehungen zwischen den verschiedenen Datensätzen aufgebaut werden.
4. **Optimierung für den Abruf:** Die Verknüpfung der Tabellen erleichtert den Abruf von Daten, da Informationen über verschiedene Tabellen hinweg miteinander verknüpft werden können.
5. **Sprachspezifische Datenbank:** Da die Datenbank in deutscher Sprache existiert, werden alle Tabellen und Datensätze entsprechend in deutscher Sprache erstellt und verwaltet.

Durch die sorgfältige Erstellung und Verknüpfung dieser Tabellen in der Datenbank wird sichergestellt, dass die Lernsoftware effizient auf die benötigten Informationen zugreifen kann und ein reibungsloser Betrieb gewährleistet ist.

## Back-End-Code

### Sprach Einstellung

```
private void LanguageChange(object sender, RoutedEventArgs e)
{
    Button clickedbutton = sender as Button;
    if (clickedbutton.Content.Equals("Deutsch"))
    {
        MyDataContext.switchCulture("de-DE");
    }
    /*else
    {
        MyDataContext.switchCulture("en-US");
    }
    */
    else
    {
        Console.WriteLine("Keine Datenbank vorhanden");
    }
    Window m = new MainWindow();
    m.Show();
    this.Close();
}
```

Hier wird das Abrufen der Programmsprache zu Deutsch, das Ersetzen der Sprache bei mehreren Datenbanken durch Knopfdruck und das Schließen von Window1 nach der Abfrage beschrieben:

6. **Abrufen der Programmsprache:** Um die Sprache des Programms abzurufen, wird eine Abfrage durchgeführt. Diese Abfrage überprüft die Einstellungen des Programms und gibt die aktuell eingestellte Sprache zurück. In diesem Fall wird die Sprache auf Deutsch abgerufen.

7. **Ersetzen der Sprache bei mehreren Datenbanken:** Wenn mehrere Datenbanken vorhanden sind, kann die Sprache durch einen Knopfdruck ersetzt werden. Dies bedeutet, dass der Benutzer die Möglichkeit hat, die Sprache des Programms auf eine andere Sprache umzustellen. Durch das Drücken des entsprechenden Knopfes wird die Sprache in allen verbundenen Datenbanken geändert.
8. **Schließen von Window1 nach der Abfrage:** Nachdem die Abfrage zur Sprache des Programms durchgeführt wurde, wird Window1 geschlossen. Window1 ist wahrscheinlich ein Fenster oder eine Benutzeroberfläche, die für die Anzeige der Abfrageergebnisse verwendet wird. Nachdem die Informationen abgerufen wurden, ist es üblich, das Fenster zu schließen, um den Benutzerfluss fortzusetzen.

```
public class MyDataContext
{
    3 Verweise
    public static string mycultureinfo { get; private set; } = "de-DE";
    0 Verweise
    public MyDataContext()
    {
    }

    1 Verweis
    public static void switchCulture(string Name)
    {
        mycultureinfo = Name;
    }
}
```

Dieses Bild beschreibt den Vorgang für eine bzw. mehrere benutzte Datenbanken.

Es ist zu sehen, dass durch die MyDataContext Klasse der Standard auf die Deutsche Datenbank "Deutsch" gesetzt wird.

```
private void chosenKapitel(object sender, RoutedEventArgs e)
{
    MenuItem m = (MenuItem)sender;
    // MessageBox.Show((string)m.Header);
    Window w = new SubWindow(k.FirstOrDefault(x => x.KapitelName.Equals(m.Header)).KapitelId, this);
    w.Show();
    w.Focus();
    this.Hide();
}
```

Übergang vom MainWindow zum SubWindow in dem beschriebenen Szenario erfolgt:

9. **Benutzerinteraktion auslösen:** Der Übergang vom MainWindow zum SubWindow wird durch eine Benutzerinteraktion ausgelöst, beispielsweise durch das Auswählen eines bestimmten Kapitels oder einer Aktion innerhalb der Anwendung.
10. **Aufrufen des SubWindows:** Nachdem der Benutzer das gewünschte Kapitel ausgewählt hat, wird das SubWindow aufgerufen. Dies kann durch Code in der Anwendungslogik implementiert werden, um das entsprechende SubWindow zu laden und anzuzeigen.
11. **Fokus setzen:** Das aufgerufene SubWindow wird in den Fokus gesetzt, was bedeutet, dass es aktiv wird und Benutzerinteraktionen empfängt. Der Fokus

wird vom MainWindow auf das SubWindow verschoben, um die Benutzerinteraktion im SubWindow zu ermöglichen.

12. **MainWindow verstecken:** Während das SubWindow im Fokus steht, wird das MainWindow versteckt. Dies bedeutet, dass das Hauptfenster der Anwendung vorübergehend ausgeblendet wird, um dem SubWindow den Vorrang zu geben und die Benutzererfahrung auf das ausgewählte Kapitel zu konzentrieren.

Durch diesen Ablauf wird ein reibungsloser Übergang von dem MainWindow zum SubWindow ermöglicht, wodurch die Benutzerführung verbessert wird und die Anwendung eine klare Hierarchie von Fenstern und Inhalten aufweist.



```
private static bool firstload = true;
List<Kapitel> k = new List<Kapitel>();
List<MenuItem> menuItems = new List<MenuItem>();
1 Verweis
public MainWindow()
{
    InitializeComponent();

    //Scaffold-DbContext "Data Source=(localdb)\MSSQLLocalDB;Initial
    int count = 1;
    if (MyDataContext.mycultureinfo == "de-DE")
    {
        hausarbeit_DEContext ctx = new hausarbeit_DEContext();

        k = ctx.Kapitels.ToList();

        foreach (Kapitel kitem in k)
        {
            MenuItem m = new MenuItem();
            m.Header = kitem.KapitelName;
            m.Click += new RoutedEventHandler(chosenKapitel);
            menuItems.Add(m);
        }
        menuItems.ForEach(x => menü.Items.Add(x));
    }
}
```

Zu sehen ist hier wie das Menü erstellt wird und jedem Menüpunkt der Name eines aus der Datenbank gezogenen Kapitels zugeordnet wird:

13. **Datenbankabfrage durchführen:** Zunächst wird eine Datenbankabfrage durchgeführt, um die Kapitelteile zu extrahieren, die für das Menü verwendet werden sollen. Dies kann durch einen geeigneten SQL-Befehl erfolgen, der die relevanten Kapitel aus der Datenbank abrufen.
14. **Menüerstellung:** Basierend auf den aus der Datenbank abgerufenen Kapiteln wird das Menü erstellt. Jeder Menüpunkt wird dynamisch mit dem Namen eines Kapitels aus der Datenbank verknüpft. Dies kann programmatisch erfolgen, indem die Datenbankergebnisse durchlaufen und jedem Menüpunkt der entsprechende Kapitelname zugewiesen wird.
15. **Menüpunkte benennen:** Die Namen der Kapitel werden den Menüpunkten zugeordnet, um eine klare und verständliche Menüstruktur zu schaffen. Jeder Menüpunkt repräsentiert dabei einen bestimmten Kapitelteil, um dem Benutzer eine intuitive Navigation durch die Anwendung zu ermöglichen.
16. **Dynamische Aktualisierung:** Bei Änderungen oder Aktualisierungen in der Datenbank können die Menüpunkte dynamisch angepasst werden, um sicherzustellen, dass sie immer die aktuellen Kapitelnamen widerspiegeln. Dies gewährleistet die Konsistenz zwischen den Datenbankinhalten und dem Menü der Anwendung.

Durch die Verknüpfung des Menüs mit den aus der Datenbank abgerufenen Kapiteln wird eine flexible und anpassbare Menüstruktur geschaffen, die es dem Benutzer ermöglicht, einfach zwischen den verschiedenen Kapiteln zu navigieren und auf die gewünschten Inhalte zuzugreifen.

```
private void QuestionHelp(object sender, RoutedEventArgs e)
{
    List<int> AqsId = new List<int>();
    selectedAufgaben.ForEach(x => AqsId.Add(x.AqsId));
    MessageBox.Show(rueckmeldungen.FirstOrDefault(x =>
        x.AqsId == selectedAufgaben[_currentQuestionPosition].AqsId).RueckText);
}
```

Die Ausgabe der Lösungen durch eine Message Box, die mit einem gezogenen Datensatz gefüllt wird:

17. **Datensatz abrufen:** Zunächst wird ein Datensatz aus der Datenbank gezogen, der die benötigten Lösungen enthält. Dieser Datensatz kann beispielsweise Informationen zu den Lösungen eines bestimmten Problems oder einer Aufgabe enthalten.
18. **Füllen der Message Box:** Die Message Box wird mit den Lösungen aus dem abgerufenen Datensatz gefüllt. Dies kann durch das Auslesen der Lösungsdaten aus dem Datensatz und das entsprechende Einsetzen in die Message Box erfolgen, um dem Benutzer die korrekten Antworten oder Ergebnisse anzuzeigen.
19. **Anzeige der Message Box:** Nachdem die Message Box mit den Lösungen gefüllt wurde, wird sie angezeigt, um dem Benutzer die Informationen zu präsentieren. Die Message Box wird in der Regel zentral auf dem Bildschirm positioniert und enthält die Lösungen, die aus dem Datensatz abgerufen und eingefügt wurden.

Durch die Verwendung einer Message Box zur Anzeige der Lösungen aus einem Datensatz wird dem Benutzer auf klare und übersichtliche Weise die benötigten Informationen präsentiert. Dies ermöglicht eine effektive Kommunikation von Lösungen oder Ergebnissen innerhalb der Anwendung.

```
1 Verweis
private void zuruck(object sender, RoutedEventArgs e)
{
    _mainWindow.Show();
    this.Close();
}
```

Nach erfolgreicher Erledigung eines Aufgabensatzes ermöglicht die Verbindung eines Knopfes dem Benutzer, zur Hauptansicht (MainWindow) zurückzunavigieren, um den Fortschritt zu überprüfen oder weitere Aktionen auszuführen.

```

private void Back(object sender, RoutedEventArgs e)
{
    if (_currentInhalt > 0)
    {
        _currentInhalt--;
    }

    Inhalt.Text = selectedInhalt[_currentInhalt].Inhalt_Text;
}

1 Verweis
private void Forth(object sender, RoutedEventArgs e)
{
    if (_currentInhalt < selectedInhalt.Count()-1)
    {
        _currentInhalt++;
    }

    Inhalt.Text = selectedInhalt[_currentInhalt].Inhalt_Text;
}

```

Die Codierung der Navigation im SubWindow ermöglicht es dem Benutzer, innerhalb des SubWindows vorwärts und rückwärts zu navigieren, bis alle verfügbaren Datensätze für das jeweilige Kapitel durchgegangen wurden. Sobald kein weiterer Datensatz mehr für das aktuelle Kapitel vorhanden ist, wird die Navigation gestoppt, um dem Benutzer eine klare Anzeige darüber zu geben, dass alle verfügbaren Informationen durchgesehen wurden. Dies gewährleistet eine geordnete und effiziente Navigation innerhalb des SubWindows, wodurch der Benutzer die Inhalte strukturiert, erkunden kann.

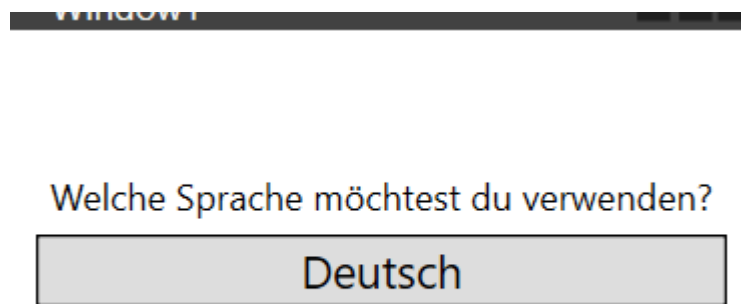
```

1 Verweis
private void NextQuestion(object sender, RoutedEventArgs e)
{
    if (_currentQuestionPosition < selectedAufgaben.Count()-1)
    {
        _currentQuestionPosition++;
    }
    else
    {
        _currentQuestionPosition = 0;
    }

    Frage.Text = selectedAufgaben.ElementAt(_currentQuestionPosition).AGS_Text;
}

```

Die Navigation nach vorne und zurück der Aufgabenstellungen, wie im obigen Bild beschrieben, erfolgt durch wiederholtes Klicken. Dies bedeutet, dass der Benutzer durch Klicken auf eine Schaltfläche oder einen Pfeil nach vorne zur nächsten Aufgabenstellung navigieren kann. Wenn der Benutzer weiterhin auf die Schaltfläche oder den Pfeil klickt, wird die Navigation wiederholt und der Benutzer kann durch die Aufgabenstellungen vorwärts und rückwärts navigieren. Diese Funktion ermöglicht es dem Benutzer, schnell zwischen den Aufgabenstellungen hin und her zu wechseln und den Inhalt effizient zu erkunden.



```
Title="Window1" Height="150" Width="250">
rid>
<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
  <Label Content="Welche Sprache möchtest du verwenden?" />
  <!--<Label Content="Which language do you want to use?" /> -->
  <Button Content="Deutsch" BorderThickness="1" BorderBrush="Black"
    Width="auto" Height="auto" FontSize="15" Click="LanguageChange" />
  <!--<Button Content="English" BorderThickness="1" BorderBrush="Black"
    Width="auto" Height="auto" FontSize="15" Click="LanguageChange" />-->
</StackPanel>
```

Mit Labels wurden Fragen zur Abfrage der Sprache gestellt und als Antwortmöglichkeit wurden Buttons verwendet. Durch einen einfachen Klick auf die Buttons konnte die gewünschte Antwort ausgewählt werden. Wenn du weitere Informationen zu Labels und Buttons benötigst, kann ich dir gerne weiterhelfen.

Hallo und Herzlich Willkommen zu meiner Lernsoftware für Basis-Rechnungswesen.

In diesem Programm findest du eine Anzahl an Theorie mit einer Mischung von Aufgaben und den passenden Lösungen. Du antwortest in einem vorgegebenen Feld und das Programm liefert dir eine ausführliche Antwort. Alles, was du dafür tun musst, ist dir ein Thema auszusuchen und es kann losgehen.

Viel Glück!

Inventur > Inventar > Bilanz    Eigenkapitalvergleich    Bestandskonten    Grundbuch    Hauptbuch    Eröffnungs-/Schlussbilanz konto    Erfolgskonten    Gewinn und Verlust konto    Lineare Abschreibung    Umsatzsteuer

```
<Grid.ColumnDefinitions>
  <ColumnDefinition Width="90"/>
  <ColumnDefinition Width="*/>
  <ColumnDefinition Width="90"/>
</Grid.ColumnDefinitions>

<Grid Grid.Column="1">
  <Grid.RowDefinitions>
    <RowDefinition Height="50"/>
    <RowDefinition Height="350"/>
    <RowDefinition Height="50"/>
    <RowDefinition Height="250"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>
```

Hier wurde das Programm in zwei Teile aufgeteilt, wobei in der mittleren Spalte eine weitere Unterteilung in fünf Abschnitte erfolgte. Schätzungen wurden verwendet, um dem Wireframe so nahe wie möglich zu kommen. Diese Strukturierung hilft bei der Visualisierung und Planung des Programms.

```

<Grid Grid.Row="3">
  <Grid.Background>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />

  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />

  </Grid.RowDefinitions>

  <Menu x:Name="menü"
    Width="auto"
    Height="50"
    VerticalAlignment="Top"
    HorizontalAlignment="Center"
    Background="White"
    BorderBrush="Blue"
    BorderThickness="1"
    FontSize="10"
    Grid.Column="0"
    >

    <Menu.ItemsPanel>
      <ItemsPanelTemplate>
        <VirtualizingStackPanel Orientation="Horizontal" />
      </ItemsPanelTemplate>
    </Menu.ItemsPanel>

```

In der mittleren Reihe wurde die vierte Spalte zunächst in ein Grid aufgeteilt, in dem das Menü realisiert wird, das durch den erforderlichen Backend-Code erstellt wird. Durch den Frontend-Code wird das Menü horizontal ausgerichtet und jeder Spalte wird der benötigte Platz zugewiesen. Diese strukturierte Vorgehensweise ermöglicht eine klare Darstellung und Funktionalität des Menüs.

◀

Der Eigenkapitalvergleich ist eine einfache und effektive Methode, um den Erfolg eines Unternehmens zu bewerten.  
 Es wird ein Vergleich zwischen dem Betrag am Anfang der Periode und dem Betrag am Ende der Periode durchgeführt.  
 Wenn das Eigenkapital steigt, wird dies als Eigenkapitalmehrung bezeichnet, wenn es sinkt Eigenkapitalminderung.  
 Es ist nicht erlaubt, private Einnahmen oder Entnahmen als Erhöhung oder Abnahme zu betrachten.

▶

«

Was ist der Eigenkapitalvergleich und welche Informationen bietet er über die finanzielle Pos

Bitte Text hier eingeben

»

Lösung

>

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="100"/>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="100"/>
  </Grid.ColumnDefinitions>

  <Button Margin="5,700,5,5" Background="Transparent" FontSize="60" Click="zuruck" ><</Button>
  <Button Content="«" Margin="5,100,5,400" Background="Transparent" FontSize="60" Click="Back" />
  <Button Content="»" Margin="5,100,5,400" Background="Transparent" FontSize="60" Grid.Column="2" Click="Forth" />
</Grid>
```

Um ein äußeres Feld zu erstellen, in dem die Knöpfe für die Vorwärts- und Rückwärtsnavigation platziert sind und sich der Inhalt in der Mitte befindet, ist eine Gestaltung erforderlich:

## 20. Layoutplanung:

- Definiere die Größe und Position des äußeren Feldes, in dem die Navigationselemente und der Inhalt platziert werden sollen.
- Berücksichtige die Größenverhältnisse für die Knöpfe und den zentralen Inhalt, um ein ausgewogenes Erscheinungsbild zu gewährleisten.

## 21. Interaktive Navigation:

- Füge C# code hinzu, um sicherzustellen, dass die Navigationsknöpfe funktionieren und den Inhalt entsprechend aktualisieren.
- Implementiere Funktionen für die Vorwärts- und Rückwärtsnavigation, die beim Klicken auf die entsprechenden Knöpfe ausgelöst werden.

```

<Grid Grid.Column="1">
  <Grid.RowDefinitions>
    <RowDefinition Height="50"/>
    <RowDefinition Height="350"/>
    <RowDefinition Height="50"/>
    <RowDefinition Height="250"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>

  <Frame Grid.Column="1" Grid.Row="1" BorderBrush="Purple" Background="PeachPuff" >/Frame>
  <Frame Grid.Column="1" Grid.Row="3" BorderBrush="Purple" Background="#FF225FF5" >/Frame>

  <Button Content="yes/ja" Grid.Row="4" Margin="300,5,500,5"
    Background="Transparent" FontSize="40" Click="QuestionHelp" />

  <TextBox Grid.Row="1" x:Name="Inhalt">/TextBox>

  <Button Content=">" Grid.Row="4" Margin="500,5,300,5"
    Background="Transparent" FontSize="40" Click="NextQuestion" />

```

Die Mittlere Reihe ließ sich wie bei MainWindow in Rahmen belegen. In der untersten Reihe befinden sich die Knöpfe für das Einreichen der Antwort wodurch das Event abgerufen wird und die MessageBox mit den Lösungen heraussticht. Ein weiterer Knopf ist nebenan zum Wechseln der Aufgabenstellungen. Die TextBox bildet sich im oberen Teil der Mittleren Reihe und in dieser werden die gezogenen Datensätze des Inhalts visualisiert.

```

<Grid Grid.Row="3">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*/>
    <ColumnDefinition Width="2*/>
    <ColumnDefinition Width="*/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*/>
    <RowDefinition Height="2*/>
  </Grid.RowDefinitions>

  <TextBox Grid.Column="1" Background="Transparent" Margin="0,5" IsReadOnly="True" x:Name="Frage" >/TextBox>

  <TextBox TextWrapping="Wrap" SpellCheck.IsEnabled="True" Name="myTextBox" Background="Transparent"
    Grid.Column="1" Grid.Row="1" Margin="30,5" >Bitte Text hier eingeben / Please enter your text here</TextBox>

```

In dem Bild gibt es eine Unterteilung in der mittleren Spalte und in der dritten Reihe, in dieser die Textboxen definiert und festgelegt sind.

- **"Frage" Textbox:** Diese Textbox dient zur Anzeige des Aufgabensatzes, der in der Regel statisch ist und sich nicht ändert. Hier wird dem Benutzer die Frage oder Aufgabe präsentiert, auf die er antworten soll.
- **"myTextBox" Textbox:** Die myTextBox ist das Feld, in dem die Benutzerantworten geschrieben werden. Der Benutzer kann in dieses Feld seine Antwort auf die gestellte Frage eingeben.

Durch die klare Unterteilung und Festlegung dieser Textboxen können Benutzer die Fragen lesen und ihre Antworten in einem separaten Feld eingeben. Dieses Design hilft dabei, die Benutzererfahrung zu verbessern und die Interaktion mit der Aufgabe zu erleichtern.

## Nutzerhandbuch



Hinzufügen der Datenbank "hausarbeit\_DE" im Ordner "Datenbank", durch Doppelklick lässt sich diese ausführen. Das Einloggen im vorgegebenen Programm Mssql sollte mit "." und Windows Authentifizierung erfolgen.

Die Lernsoftware ist im Ordner "hausarbeit\Hausarbeit\bin\Debug" zu finden als "Hausarbeit.exe".

Beim Start des Programmes Bitte bevorzugte Sprache auswählen. Dies gilt, wenn Mehrere Datenbanken angefügt wurden.

Nach Sprachauswahl wird man mit einem Freundlichen Benutzer Leitfaden begrüßt. Für erste Benutzung Bitte Lesen.

Über das Menü lässt sich durch Klicken auf ein jeweiliges Kapitel verweisen, was sich in einem Neuen Fenster öffnet.

Mittlerer oberer Kasten stellt Informationsmaterial dar. Mithilfe von den Seiten Pfeilen können Information Sätze gewechselt werden. Seiten gerichteten Pfeile stoppen am Ende und beginn eines Datensatzes Bitte vor oder rückwärts navigieren.

Mittlerer schmaler Kasten stellt das Aufgaben Feld dar. Das sich mit dem ">" Button nach vorne navigieren lässt, Der Aufgaben Satz wiederholt sich durch wiederholendes Klicken des ">" Knopfes.

Eigene Antworten können in das Feld unter dem Aufgabensatz geschrieben werden und durch den "Lösung" Knopf lässt sich ein weiteres Fenster aufrufen mit ausführlicher Lösung zum Abgleich.

Der an dem Link unteren Ecke platzierte Kopf führt zum Haupt Fenster zurück, um ein neues Kapitel auszuwählen.

## Quellenangabe

- Deitermann Rückwart, Rechnungswesen für Berufsfachschulen, Winklers
- Schmolke Deitermann, Industrielles Rechnungswesen IKR , Westermann Gruppe
- <https://www.microtech.de/erp-wiki/bilanz/#:~:text=Die%20Bilanz%20ist%20die%20Gegenüberstellung,die%20Bilanz%20Hauptbestandteil%20eines%20Jahresabschlusses.>
- <https://stackoverflow.com>
- <https://www.steuertipps.de/lexikon/a/abschreibung-linear>
- <https://studyflix.de/wirtschaft/lineare-abschreibung-1192>

## Eidesstattliche Erklärung

“Hiermit erkläre ich an Eides statt, dass ich diese Hausarbeit selbständig und ohne fremde Hilfe angefertigt habe und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.“

Datum

Unterschrift