

Muharemi, Kanalga, Ilic, Brunokowski

I3a

|  |
| --- |
|  |
| Zinsberechnung |
| 29. Februar  LP: Michael Schneider; Brigitte Marti |

Teil 1

|  |  |
| --- | --- |
| Projektname: | Zinsberechnung |
| Gruppenmitglieder mit Klassen und Emailadresse: | Altrim Muharemi; Gabriel Kanalga; Vuk Ilic; Niels Brunokowski  I3a  [altrim.muharemi@students.ksba.ch](mailto:altrim.muharemi@students.ksba.ch) [gabriel.kanalga@students.ksba.ch](mailto:gabriel.kanalga@students.ksba.ch) [vuk.ilic@students.ksba.ch](mailto:vuk.ilic@students.ksba.ch) [niels.brunokowski@students.ksba.ch](mailto:niels.brunokowski@students.ksba.ch) |
| Betreuende Lehrpersonen mit Emailadresse: | Michael Schneider; Brigitte Marti  [michael.schneider@kanti-baden.ch](mailto:michael.schneider@kanti-baden.ch) [brigitte.marti@kanti-baden.ch](mailto:brigitte.marti@kanti-baden.ch) |
| Titel: | Zinsberechnung |
| Beschreibung: | In diesem IDPA-Projekt für die "Goldstein und Klaster Bank" sind wir als Praktikant beauftragt, eine Software zu entwickeln, das den zusätzlichen Zinsbetrag berechnet, den Kunden durch den "Marchzins-Bonus" auf ihre Sparbeträge erhalten können. Das Programm sollte den zu erhaltenden Betrag und die Verrechnungssteuer getrennt ausweisen |
| Startdatum des Vorprojekts: | 15.08.2023 |
| Abgabe mit Zeitpunkt: |  |
| Anforderungen: | Funktionale Anforderungen:   1. Variable Zinssatzeingabe: Das Programm muss es dem Benutzer ermöglichen, den aktuellen Zinssatz manuell einzugeben und die Prozentuale Erhöhung des Bonuszinses. 2. Kundeninformationen: Das Programm muss das Geburtsdatum des Kunden abfragen können, um den speziellen Geburtstagsbonus zu berücksichtigen. 3. Zinsberechnung: Das Programm muss den Bonuszinsbetrag basierend auf dem eingegebenen Zinssatz und dem Geburtsdatum des Kunden berechnen können. 4. Steuerberechnung: Das Programm muss in der Lage sein, die Verrechnungssteuer von 35% auf den Bonuszins getrennt zu berechnen und anzuzeigen. 5. Marchzinsberechnung: Das Programm muss in der Lage sein den normalen Zins und Bonuszins zu berechnen, wenn man einen gewissen Zeitraum angibt.   Qualitative Anforderung:   1. Benutzerfreundlichkeit: Das Programm sollte eine effektive und benutzerfreundliche Oberfläche haben, damit der Benutzer die Informationen schnell und einfach eingeben und die Ergebnisse abrufen kann. |
| Datum und Unterschrift  GruppenmitgliederInnen: |  |
| Datum und Unterschrift betreuende  Lehrpersonen: |  |

Vorkenntnisse

Wirtschaftliche Kenntnisse:   
Wir haben Erfahrung in der Berechnung von Zinsen, durch die Fächer.SWIR und SFRW erhalten. Wir sind auch mit den steuerlichen Aspekten und rechtlichen Rahmenbedingungen vertraut, die bei der Integration der Verrechnungssteuer und Zinsberechnungen relevant sein könnten.   
  
Programmier Kenntnisse:  
Das Projekt wird in der Programmiersprache C# mit WinForms umgesetzt. Jeder in unserer Gruppe verfügt über Programmierkenntnisse in C#. Was Neuland betrifft, könnten die spezifischen Anforderungen für die Berechnung des Marchzinses und die Integration der Verrechnungssteuer in C# WinForms.

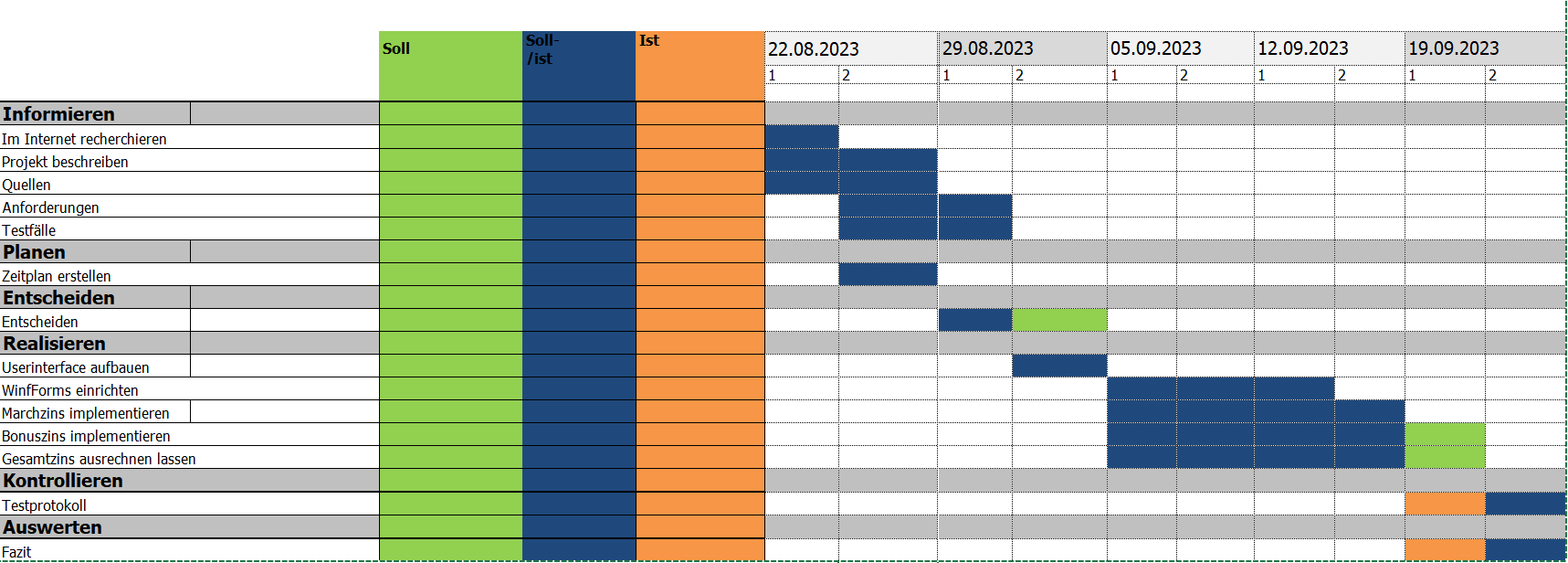
Vorarbeit

Die verschiedenen Zinse und ihre Bedeutung gründlich verstehen: Wir haben am Anfang die Aufgabe analysiert und haben uns damit beschäftigt die verschiedenen Zinssätze zu verstehen und wie man diese berechnet. Wir haben uns über den Marchzins im Internet gut informieren lassen. Dann haben wir uns gegenseitig ausgetauscht, damit jedes Teammitglied die Funktionsweise der Zinse von A bis Z verstanden hat.  
  
Technologieauswahl: Wir haben gründlich darüber diskutiert, welche Technologie wir für die Realisierung des Programms am effektivsten wäre. Zuerst wollten wir das Programm in der C# Konsole realisieren, doch dann haben wir uns aus benutzerfreundlichkeitsgründen für WinForms entschieden.  
  
Anforderungsanalyse: Wir haben die Anforderungen für das Programm gesammelt und wir hatten einige Fragen, worauf Herr Kanalga an Herrn Schneider eine E-Mail geschickt hat

Firmenstandards

IPERKA: Wir haben nach IPERKA gearbeitet und haben unser Projekt und unseren Projektverlauf in einer Word Datei dokumentiert.  
  
Best Practice: Wir haben uns an die normalen Regeln für das Schreiben von C# Code gehalten, da im Auftrag keine Vorgaben bekommen haben. Zu den Richtlinien gehört die verständliche Benennung von Methoden und Klassen. Dadurch wird der Code besser lesbar.

Zeitplan



Arbeitsjournal

29.08.2023

Vergessen

05.09.2023

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | Tätigkeit: Person: | |
| Ausgeführte Tätigkeiten: | Pair Progamming, Angefangen Programm zu realisieren | Muharemi, Brunokowsi |
| Messbare Ziele formuliert und Testfälle erstellt (Am Dokumentation gearbeitet), unterstützende Funktion | Kanalga |
| User Interface designt und erstellt, unterstützende Funktion | Ilic |
| Erreichte Ziele: | Ziel: | Person: |
| Ziele wurden formuliert, Testfälle erstellt, Dokumentation ist bis zum jetzigen Stand vollständig | Kanalga |
| User Interface ist fertig | Ilic |
| Eingaben, Normale Zinsrechnung wurde implementiert | Muharemi, Brunokowski |
|  |  |
| Aufgetretene Probleme: | Die Berechnung vom March-Zinsen und Steuersätze zu implementieren | Muharemi, Brunokowski |
| Keine Probleme | Ilic |
| Keine Probleme | Kanalga |
|  |  |
|  |  |
| Hilfestellung (wer, was): | ChatGPT | |
| Pendenzenliste | Pendenz: | Person: |
| Dokumentation auf dem Laufenden halten | Alle |
| March-Zins Implementierung fertig stellen | Muharemi, Brunokowski |
| Reflexion: | Reflexion: | Person: |
| Das Pair Programming war gut, weil man sich so am Anfang noch direkt helfen konnte. Allerdings wurde es zunehmend anspruchsvoller, als wir uns mit der Berechnung der "March-Zinsen" befassten, da ich anfangs Schwierigkeiten hatte, das Konzept zu erfassen. Nachdem ich es schliesslich verstanden hatte, stand ich vor der Herausforderung, die Formel zur Berechnung des Bonuszinses für Kunden zu entwickeln, basierend auf ihrem Sparbetrag und ihrem Geburtstagsmonat. | Muharemi |
| Das grossartige an Pair Programming war das man die Probleme aus anderem Blickwinkel sah und man somit viel schneller und effizienter voranging. Das Alles am Anfang gut lief, war das wir Pair Programming angewendet haben. Doch wir kamen auf Probleme wie die Umsetzung vom March-Zins. Das kostete uns Zeit und Mühe, aber wir werden weiterarbeiten und dieses Problem lösen | Brunokowski |
| Ich hatte keine grossen Probleme. Durch die Gegebene Ausgangslage konnte ich die Informationen, welche ich für die Ziele und die Tests benötigte extrahieren. Ich bin zufrieden mit meiner Arbeit. | Kanalga |
| Ich finde ich habe das User Interface schön erstellt. Ich habe keine Hürden gehabt. | Ilic |
| Vergleich mit dem Zeitplan: | Soll: | Ist: |
| Programm Realisierung gestartet | Eingabe Konto Guthaben und normale Zinsrechnung auf beliebigen Satz realisiert |
| User interface fertig | User interface fertig |
| Testfälle fertig | Testfälle fertig |

12.09.2023

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | Tätigkeit: Person: | |
| Ausgeführte Tätigkeiten: | Fortsetzung der Programmierung und Fehlerbehebung | Alle |
|  |  |
|  |  |
| Erreichte Ziele: | Ziel: | Person: |
| Erfolgreich an der Implementierung der March-Zinsen gearbeitet | Alle |
| User Interface ist weiterhin im guten Zustand | Ilic |
| Dokumentation auf dem Laufenden gehalten, Testfälle poliert | Kanalga |
|  |  |
| Aufgetretene Probleme: | Keine Probleme | Alle |
|  |  |
|  |  |
|  |  |
|  |  |
| Hilfestellung (wer, was): | ChatGPT | |
| Pendenzenliste | Pendenz: | Person: |
| Dokumentation auf dem Laufenden halten | Alle |
| March-Zins Implementierung fertig stellen | Alle |
| Reflexion: | Reflexion: | Person: |
| Nach ein wenig Überlegung und Besprechung mit dem Team konnten wir unsere Lösungsvorschläge ausfiltern und wie Legosteine zusammensetzen, die Arbeit in der Gruppe kann zu Chaos führen, doch beim Brainstorming bleibt das Team unschlagbar. | Muharemi |
| Für die Berechnung des Marchzinses, bzw. die Berechnungen für den ganzen Auftrag mussten wir uns Schritt-für-Schritt in Einzelteile ausbauen, um konkret sehen zu können, wie man, mit welchen Informationen zur Lösung kommen kann. Nachdem wir diese Teile hatten, konnten wir sie in Code konvertieren. | Brunokowski |
| Der Informationsaustausch innerhalb der Gruppe ist sehr geschmeidig, was die Arbeit an der Dokumentation um vielfaches vereinfacht. | Kanalga |
| Das User Interface blieb stabil, was positiv ist. Ich hatte keine weiteren Probleme und bin zufrieden mit dem Ergebnis. | Ilic |
| Vergleich mit dem Zeitplan: | Soll: | Ist: |
| Programm Realisierung fortgesetzt | March-Zins Implementierung in Arbeit |
|  |  |
|  |  |

19.09.2023

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | Tätigkeit: Person: | |
| Ausgeführte Tätigkeiten: | Vollständige Implementierung der March-Zins-Berechnung durchgeführt | Alle |
| Letzter Debugging-Prozess abgeschlossen | Alle |
| Dokumentation und Testfälle aktualisiert | Alle |
| Erreichte Ziele: | Ziel: | Person: |
| March-Zins-Berechnung ist jetzt fertig implementiert und wurde getestet | Alle |
| User Interface ist optimiert und benutzerfreundlich | Ilic |
| Dokumentation und Testfälle sind vollständig und auf dem neusten Stand | Kanalga |
|  |  |
| Aufgetretene Probleme: | Keine Probleme | Alle |
|  |  |
|  |  |
|  |  |
|  |  |
| Hilfestellung (wer, was): | ChatGPT | |
| Pendenzenliste | Pendenz: | Person: |
| Dokumentation finalisieren | Alle |
| Letzte Tests & Qualitätsichering | Alle |
| Reflexion: | Reflexion: | Person: |
| Die Teamarbeit war effizient und jeder wusste, was zu tun war und die Implementierung der March-Zinsen verlief ohne probleme. | Muharemi |
| Wir konnten alle Code elemente so integrieren, dass sie zusammenarbeiten. Ich habe viel gelernt im Prozess | Brunokowski |
| Wir konnten das Projekt erfolgreich realisieren und wir haben die Benutzeroberfläche sehr gut gemacht. Insgesamt bin ich stolz darauf, wie wir diee Komponen te des Projekts effizient realisiert haben | Kanalga |
| Die March-Zins-Implementierung und die UI-Optimierung haben mich gefordert, aber am Ende habe ich es geschafft | Ilic |
| Vergleich mit dem Zeitplan: | Soll: | Ist: |
| Programm Realisierung fortgesetzt | March-Zins Implementierung und UI abgeschlossen |
|  |  |
|  |  |

26.09.2023

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | Tätigkeit: Person: | |
| Ausgeführte Tätigkeiten: | PowerPoint erstellt | Alle |
| Präsentation üben | Alle |
|  |  |
| Erreichte Ziele: | Ziel: | Person: |
| PowerPoint fertig | Alle |
| Präsentation | Alle |
|  |  |
|  |  |
| Aufgetretene Probleme: | Keine Probleme | Alle |
|  |  |
|  |  |
|  |  |
|  |  |
| Hilfestellung (wer, was): | ChatGPT | |
| Pendenzenliste | Pendenz: | Person: |
|  |  |
|  |  |
| Reflexion: | Reflexion: | Person: |
|  |  |
|  |  |
|  |  |
|  |  |
| Vergleich mit dem Zeitplan: | Soll: | Ist: |
| PowerPoint fertig | PowerPoint fertig |
| Präsentation im Griff | Präsentation im Griff |
|  |  |

Arbeitsergebnisse

Unsere Arbeiten sind im Office SharePoint erstellt und geteilt worden, dies dient der Datensicherung sowie der Zugriffssteuerung.

Installation

1. Dateiextraktion:

Laden Sie die .zip-Datei herunter und speichern Sie sie an einem geeigneten Ort auf Ihrem Computer.

Klicken Sie mit der rechten Maustaste auf die heruntergeladene .zip-Datei und wählen Sie "Alle extrahieren" oder "Hier extrahieren", je nach den verfügbaren Optionen. Dadurch wird der Inhalt der .zip-Datei in einen Ordner entpackt.

2. Öffnen des Projekts in Visual Studio:

Starten Sie Visual Studio auf Ihrem Computer (stellen Sie sicher, dass Visual Studio bereits installiert ist).

Wählen Sie im Visual Studio-Menü "Datei" > "Projekt öffnen" aus.

Navigieren Sie zum zuvor extrahierten Ordner, wählen Sie die Projektdatei (normalerweise mit der Erweiterung .sln) aus und klicken Sie auf "Öffnen".

3. Projekteinstellungen prüfen:

Überprüfen Sie die Projekteinstellungen, um sicherzustellen, dass die Zielplattform und die Framework-Version korrekt konfiguriert sind. Dies kann je nach Projektvariante variieren.

4. Kompilieren und Ausführen:

Drücken Sie die "F5"-Taste oder klicken Sie auf die Schaltfläche "Start" in Visual Studio, um das Projekt zu kompilieren und auszuführen.

5. Bedienung des Winforms-Projekts:

Nachdem das Projekt erfolgreich gestartet wurde, sehen Sie die Benutzeroberfläche des Winforms-Projekts.

Teil 2

Management Summary

Ausgangslage:

Die vorliegende Praxisarbeit wurde im Rahmen der IMS gemacht. Das Projekt behandelt das Thema Marchzinsberechnung und wurde im Zeitraum von 22.08.23 bis 26.09.23 durchgeführt. Die Ausgangslage bestand darin, einen Marchzinskalkulator mit “spielerischen” Elementen, in Form eines Bonuszinses zu erstellen.

Vorgehen:

Zur Erreichung der gesteckten Ziele wurde eine systematische Vorgehensweise gewählt. Die Arbeit wurde in 3 Phasen unterteilt:

1. Planung und Konzeption:

Ein detaillierter Projektplan wurde anhand unseres Vorwissens entwickelt. Dieser enthielt Schritte zur Lösung des Projektes.

2. Durchführung und Umsetzung:

Die geplanten Massnahmen wurden in dieser Phase umgesetzt. Dies umfasste: Design einer passenden UI, die Berechnung des Jahreszins, die Berechnung des Bonuszins und eine sinnvolle Ausgabe der Werte. Während der Durchführung gab es regelmässige Check-ups in der Gruppe, um zu sehen, wie es mit dem Projekt lief.

3. Evaluation und Schlussfolgerungen:

Nach Abschluss der Massnahmen erfolgte eine Evaluation. Dabei wurden die Ergebnisse mit den gesetzten Zielen verglichen, und Schlussfolgerungen gezogen.

Ergebnis:

Die Ergebnisse des Projektes sind vielversprechend. Unsere Ziele wurden erreicht.

Die Arbeit gab uns eine solide Grundlage für zukünftige Projekte im Bereich der Wirtschaft. Detaillierte Ergebnisse kann man im Bericht sehen. Wir sind bereit, weitere Informationen zur Verfügung zu stellen und freuen uns auf eine vertiefende Diskussion dieses Projekts.

Informationsphase

Falls Fragen in der Gruppe auftauchten, wurde zuallererst diese in die Gruppe gestellt. Alle Mitglieder probieren diese Frage zu beantworten und teilten ihr Wissen. Die Informatik dahinter war nicht so komplex, dass wir für jede kleine Frage etwas aus einer Quelle beziehen mussten, oft genügte eine kurze Fragerunde.  
Für die Kalkulation des Marchzinses suchten wir in Google eine Formel, an der wir uns für den Auftrag richten konnten.  
Nach einem ersten Austausch hatten wir offene Fragen zu den Anforderungen, wie z.B. ob der (Bonus-)Zinssatz ein fester Wert ist. Wir schrieben eine Mail zu unseren Auftragsgebern, um diese Fragen zu lösen.

Tätigkeitsliste

|  |  |  |
| --- | --- | --- |
| Tätigkeit | Datum | Person |
| Messbare Ziele formuliert & Testfälle erstellt | 22.08.2023 | Kanalga |
| User Interface designt und erstellt | 29.08.2023 | Ilic |
| Eingaben, Normale Zinsrechnung wurde implementiert | 05.09.2023 | Muharemi |
| Fortsetzung Programmierung (an Implementierung Marchzinsen gearbeitet) | 12.09.2023 | Alle |
| Erfolgreiche und komplette Implementierung der March-Zins-Berechnung durchgeführt | 19.09.2023 | Alle |
| Debugging-Prozess abgeschlossen | 19.09.2023 | Alle |
| Dokumentation und Testfälle aktualisiert | 19.09.2023 | Alle |

Architektur

Überblick

Für das Marchzins-Berechnungsprogramm werden wir eine strukturierte Architektur verwenden, die sicherstellt, dass die Funktionalität klar definiert, erweiterbar und leicht zu warten ist. Die Architektur sollte zudem die Datenverarbeitung und Benutzeroberfläche effizient verknüpfen.

Beschreibung der Architektur

Das Programm wird auf einer Schichtenarchitektur basieren, die die logischen Komponenten klar voneinander trennt. Diese Schichten sind:

a. Präsentationsschicht  
Die Präsentationsschicht enthält die Benutzeroberfläche des Programms, die mithilfe von Windows Forms erstellt wird. Sie ermöglicht die Interaktion mit dem Benutzer und sammelt alle erforderlichen Eingaben.

b. Geschäftsschicht  
Die Geschäftsschicht enthält die Logik für die Marchzins-Berechnung und -Verarbeitung. Hier werden die eingegebenen Daten validiert, die Berechnungen durchgeführt und die Ergebnisse zurückgegeben. Diese Schicht ist unabhängig von der Benutzeroberfläche und kann in Zukunft für andere Anwendungen wiederverwendet werden.

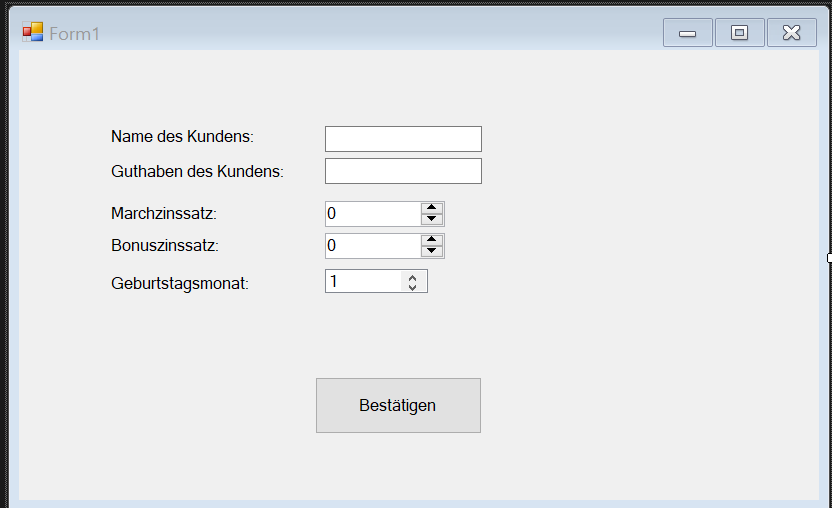
c. Datenschicht  
Die Datenschicht ist für die Speicherung und Verwaltung von Benutzerdaten verantwortlich. In diesem Fall könnten Kundeninformationen, Zinssätze und Geburtsmonate gespeichert werden. Eine einfache Datenstruktur wie eine Datenbank oder eine Datei kann verwendet werden, um die Daten zu speichern.

Begründung

Die Verwendung einer Schichtenarchitektur bietet die Möglichkeit, die Software leichter zu warten und zu erweitern. Da das Programm in Zukunft möglicherweise um zusätzliche Funktionen erweitert wird, können diese in der Geschäftsschicht hinzugefügt werden, ohne die Benutzeroberfläche ändern zu müssen. Dies fördert auch die Wiederverwendbarkeit der Geschäftslogik in anderen Anwendungen.

GUI-Prototypen

Hauptmenü:



Als Erstes überlegten wir uns, wie wir die Eingabeaufforderung gestalten wollten. Dazu haben wir uns überlegt, dass die Benutzereingabefelder klare und verständliche Beschriftungen haben sollten, um den Benutzern deutlich zu zeigen, welche erforderlichen Informationen einzugeben sind. Zum Beispiel sollte das Eingabefeld für den Namen des Kundens, auch so beschriftet sein.

Die Reihenfolge der Eingabe ist nach Logik gegliedert. Zuerst wird der Name des Kunden eingegeben, dann dessen Kontobestand. Ein Zinssatz und ein Bonuszinssatz werden festgelegt (Marchzinssatz wurde aus Versehen geschrieben). Zuletzt abgefragt wird der Geburtsmonat, in dem der Bonuszinssatz aktiviert wird. Und nun wirklich zuletzt steht dort ein grosser, sichtbarer Button mit dem label “Bestätigen”.

Damit der Benutzer auch direkt sehen kann welche Art von Eingaben das Programm verlangt, wurde die Grösse und Typ der Eingabeboxen zur Art der Abfrage angepasst.

Gleiche Typen wurden näher aneinandergerückt, um zu verdeutlichen was für ein Input erwartet wird.

Pop-up für Resultate:



Ein Titel wird gesetzt, um klar zu zeigen was hier jetzt berechnet wird.

Der Gesamtzins wird gezeigt. Die Anzahl Tage, die man in der Berechnung braucht, wird gleich darunter definiert. Die Slider mit den Werten 0 zeigen dem Benutzer sofort, dass man nach rechts verschieben muss, um die gewünschten Daten auszuwählen.

Entscheidungen

Wir haben uns entschieden WinForms als GUI-Framework zu nutzen, um die Anwendung benutzerfreundlich zu gestalten. Die Wahl fiel auf C# als Programmiersprache, da wir alle starken Kenntnisse in dieser Sprache besitzen. Bei der Zinsberechnung implementierten wir eine Funktion, die Marchzinsen genau berechnet und das Ergebnis sowohl auf der Oberfläche anzeigt als auch in einer Variable speichert. Der Marchzins wird ohne und auch mit Verrechnungssteuer gespeichert und auch anschliessend angezeigt.

Beschreibung des Projekt

In diesem IDPA-Projekt für die "Goldstein und Klaster Bank" sind wir als Praktikant beauftragt, eine Software zu entwickeln, die den zusätzlichen Zinsbetrag berechnet, den Kunden durch den "Marchzins-Bonus" auf ihre Sparbeträge erhalten können. Das Programm sollte den zu erhaltenden Betrag und die Verrechnungssteuer getrennt ausweisen. Das Programm kann die Zinsen getrennt in Bonuszins und normaler Zins für das komplette Jahr ausgeben, sowohl für einen gewünschten Zeitraum.

Technologie-Stack:

Entwicklungsumgebung: Visual Studio

Programmiersprache: C#

Framework: WinForms

### Sitzungs-Protokolle und Resultate

#### Sitzung 1: Projekt Start & Anforderungsanalyse

* Datum: 22.08.2023
* Anwesende: Muharemi, Brunokowski, Kanalga, Ilic
* Besprochen: Projektumfang, Messbare Ziele, Testfälle
* Resultat: Messbare Ziele formuliert, Testfälle erstellt

#### Sitzung 2: Fortsetzung der Arbeit

* Datum: 29.08.2023
* Anwesende: Alle
* Besprochen: Statusupdate, Problemlösung, UI-Status, User Interface
* Resultat: User interface in Arbeit, keine neuen Probleme identifiziert

#### Sitzung 3: Fortsetzung der Arbeit

* Datum: 05.09.2023
* Anwesende: Alle
* Besprochen: Statusupdate, Problemlösung, UI-Status
* Resultat: March-Zins Implementierung in Arbeit, keine neuen Probleme identifiziert

#### Sitzung 4: Fortsetzung der Arbeit

* Datum: 12.09.2023
* Anwesende: Alle
* Besprochen: Statusupdate, Problemlösung, UI-Status
* Resultat: March-Zins Implementierung in Arbeit, keine neuen Probleme identifiziert

#### Sitzung 5: Abschlussphase

* Datum: 19.09.2023
* Anwesende: Alle
* Besprochen: Abschluss der Implementierung, Debugging, Dokumentation
* Resultat: March-Zins Implementierung abgeschlossen, UI optimiert, Dokumentation auf dem neuesten Stand

Lösung beschreiben und erklären

#### Umfeld:

Im Projekt haben wir ein kleines Programm entwickelt. Es soll der "Goldstein und Klaster Bank" dazu dienen, den speziellen "Marchzins-Bonus" für Kunden schnell und einfach zu berechnen.

#### Präzisierung der Aufgabenstellung:

Wir mussten eine Software realisieren, das den extra Zinsbetrag für den Geburtstagsmonat eines Kunden ausrechnet. Es sollte sowohl den Bonus als auch die Verrechnungssteuer separat anzeigen.

#### Abklärungen:

* Einige Fragen an Lehrperson, um die Anforderungen klar definieren zu können
* Recherche im Internet, wie Marchzinsen richtig berechnet werden.

#### Auswahlkriterien:

* Die Genauigkeit der Berechnung des Marchzinses (Tage).
* Wie einfach es für den Benutzer ist zu benutzen.

#### Realisierung:

Wir haben uns dazu entschieden das Tool in C# WinForms programmiert. Dabei haben wir eine tägliche Zinsberechnung implementiert, um genaue Ergebnisse zu liefern.

#### Probleme und Lösungen:

Problem 1:

* Wir wussten zuerst nicht, wie Marchzinsen funktionieren.
* Lösung:
* Internetrecherche und Gespräche mit Teammitgliedern.

Problem 1.5:

* Unser erstes Programm hat den Marchzins nur für ganze Monate berechnet.
* Lösung:
* Wir haben den Code geändert, damit er jetzt die Zinsen auf den Tag genau berechnet.

Schlüsselstellen im Programm mit Erklärung

Klasse Marchzinsberechner

public Marchzinsberechner(double guthaben, double marchzinss, …)

{

kunde\_guthaben = guthaben;

…

}

Erklärung:

Setzt die Anfangswerte in die Variablen wie Guthaben und Zinsen, die vom Benutzer eingegeben worden sind.

Methode Kompletezinsberechnung() :

public double Kompletezinsberechnung()

{

bonuszinssatzz = Math.Round(marchzins / 100 \* (bonuserhoehung + 100), 2);

zinssatz = Math.Round(marchzins / 100 + 1, 2);

zins\_ohne\_bonus = Math.Round((kunde\_guthaben \* zinssatz - kunde\_guthaben) / 360 \* (360 - gtag), 2);

bonuszinssatz = Math.Round((zinssatz - 1) / 100 \* (bonuserhoehung + 100) + 1, 2);

bonuszins = Math.Round((kunde\_guthaben \* bonuszinssatz - kunde\_guthaben) / 360 \* gtag, 2);

return Math.Round(zins\_ohne\_bonus + bonuszins, 2);

}

Erklärung:

bonuszinssatzz:

Berechnet den Zinssatz mit Bonus. Der Marchzins wird durch 100 geteilt und dann mit (Bonus + 100) multipliziert. Das Resultat wird auf zwei Dezimalstellen gerundet.

zinssatz:

Findet den Zinssatz heraus. Der Marchzins wird durch 100 geteilt und zu 1 addiert. Das Resultat wird auf zwei Dezimalstellen gerundet.

zins\_ohne\_bonus:

Hier wird der Zins ohne Bonus berechnet. Es wird die Formel (kunde\_guthaben \* zinssatz - kunde\_guthaben) / 360 \* (360 - gtag) verwendet. gtag sind die Tage, an denen der Bonus gilt.

bonuszinssatz:

Ähnlich wie bei zinssatz, wird hier der Zinssatz mit Bonus berechnet.

bonuszins:

Der Zins mit Bonus wird berechnet. Dabei wird die ähnliche Formel wie bei zins\_ohne\_bonus angewendet, nur dass hier gtag (Bonus-Tage) verwendet wird.

return:

Die Methode gibt die Summe von zins\_ohne\_bonus und bonuszins zurück, gerundet auf zwei Dezimalstellen.

Klasse Form1

Bestaetigungs\_btn\_Click

if (string.IsNullOrEmpty(kunde\_Name.Text) || …)

{

MessageBox.Show(…);

return;

}

Erklärung:

Überprüft, ob alle Felder ausgefüllt sind. Wenn nicht, zeigt es eine Fehlermeldung an.

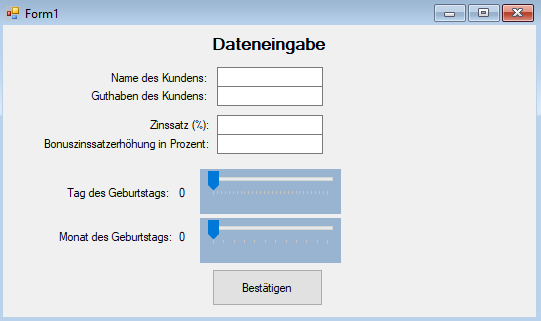
Bestaetigungs\_btn\_Click

Marchzinsberechner mzb = new Marchzinsberechner(…);

Erklärung:

Erstellt ein neues Marchzinsberechner Objekt und übergibt es an das nächste Formular.

Tatsächliches GUI



Im direkten Vergleich zum Prototyp kann man feststellen, dass wir in diesem Formular einen Titel hinzugefügt haben. Die Art von Input wurde abgeändert damit es einen einheitlicheren Look hat. Für die Daten haben wir Slider verwendet, diese sind intuitiver und man versteht direkt, wie diese funktionieren. Tag kommt als erstes und Monat als nächstes da wir, beim Datum ablesen auch als erstes den Tag sagen und dann den Monat.

Das Programm stellt sicher, dass die eingegebenen Daten korrekt und sinnvoll sind. Dies erfolgt durch die Überprüfung von Datenbereichen, in dem Zinssatz (zwischen 0 und 100 Prozent), dem Geburtsmonat (1-12) und dem Geburtstag (1-30), in der. Bei ungültigen Eingaben werden klare Fehlermeldungen angezeigt.  
Einen Defaultwert für jedes Feld zu geben schien überflüssig.



Falls alle Eingaben korrekt kommt man als nächstes zum Marchzinsberechner.   
Mit den jetzigen Daten, die man hat, kann man alle Beträge eines Jahres berechnen, dies kann mit dem ersten Button “Jahreszins Anzeigen” tun. Ein Popup erscheint welches kompakt und klar alle Informationen darstellt.

Will man den Marchzins berechnen so muss man die Laufzeit bestimmen, dafür benutzen wir die gleichen Slider wie bei der vorherigen Seite. Mit dem grossen “Berechnen” Button wird der Marchzins dann berechnet und angezeigt.

Testfälle

# Testfallnummer: 1

Voraussetzung: Programm ist geöffnet

Eingabe: Zinssatz von 2% und Prozentuale Erhöhung ist 100%

Ausgabe: Programm speichert und verwendet den Zinssatz von 2% für die Zinsberechnung und 4 % werden für die Bonuszinsberechnung verwendet.

# Testfallnummer: 2

Voraussetzung: Programm ist geöffnet

Eingabe: Geburtsdatum 15.03

Ausgabe: Programm speichert das Geburtsdatum und verwendet es für die Zinsberechnung mit Geburtstagsbonus

# Testfallnummer: 3

Voraussetzung: Dateneingabe wurde getätigt und Zeitraum für die Zinsberechnung ist eingegeben und der Zeitraum beinhaltet den Geburtstag.

Eingabe: Klick auf "Berechnen"

Ausgabe: Programm zeigt den berechneten Zins insgesamt, zeigt die einzelnen Anteile und alle Verrechnungssteuern. Bonuszins >0

# Testfallnummer: 4

Voraussetzung: Dateneingabe wurde getätigt und Zeitraum für die Zinsberechnung ist eingegeben und der Zeitraum beinhaltet nicht den Geburtstag.

Eingabe: Klick auf "Berechnen"

Ausgabe: Programm zeigt den berechneten Zins insgesamt, zeigt die einzelnen Anteile und alle Verrechnungssteuern. Bonuszins = 0

# Testfallnummer: 5

Voraussetzung: Dateneingabe wurde getätigt

Eingabe: Klick auf "Jahreszins anzeigen"

Ausgabe: Programm zeigt alle Daten (Jahreszins, Verrechnungssteuer total, Nettozins, Anteil normaler Zins, Anteil Bonuszins, Aufteilung normaler Zins, Aufteilung Bonuszins)

# Testfallnummer: 6 (Qualitativ)

Voraussetzung: Programm ist geöffnet

Eingabe: Interaktion mit der Benutzeroberfläche

Ausgabe: Die Benutzeroberfläche ist intuitiv und erlaubt schnelle und einfache Eingabe und Abruf von Informationen

Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Datum | Resultat | Durchgeführt |
| 1 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |
| 2 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |
| 3 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |
| 4 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |
| 5 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |
| 6 | 17.09.2023 | Ok | Brunokowski, Ilic, Kanalga, Muharemi |

Quellenverzeichnis

Marchzins-Formel:

[https://www.gigers.com/matthias/schule/zinsrech.htm#:~:text=Wird%20ein%20Kapital%20nicht%20über,eines%20Jahres%2C%20gilt%20die%20Marchzinsformel.&text=k%20%3D%20Kapital%20in%20Fr.&text=zm%20%3D%20Marchzins%20in%20Fr,%2F100\*m%2F12](https://www.gigers.com/matthias/schule/zinsrech.htm#:~:text=Wird%20ein%20Kapital%20nicht%20über,eines%20Jahres%2C%20gilt%20die%20Marchzinsformel.&text=k%20%3D%20Kapital%20in%20Fr.&text=zm%20%3D%20Marchzins%20in%20Fr,%2F100*m%2F12).

Vuk’s Fazit:

Das Projekt verlief erfolgreich. Die Implementierung der March-Zins-Berechnung war zuerst komplex, aber als Team konnten wir die Herausforderung meistern. Auch die Benutzeroberfläche funktioniert gut und hat ein ansprechendes Design. Ich bin zufrieden mit dem Endergebnis. Durch dieses Projekt habe ich meine Fähigkeiten in WinForms C# vertieft und ich habe gelernt im Team zu arbeiten. Ich habe auch neue Erfahrungen gesammelt, wie man umfangreiche Dokumentationen über ein Projekt schreibt.

Altrim’s Fazit:

Am Anfang vom Projekt haben wir uns entschieden für den Start des Projekts Pair Programming zu machen. Am Anfang war ich noch skeptisch, doch in nachhinein bin ich froh, da wir uns zusammen eingearbeitet haben.

Mit dem "March-Zinsen" gab es Schwierigkeiten, doch das konnten wir mit Teamarbeiten lösen. Wir haben uns Aufgaben zugeteilt und jeder kooperierte.

Niels’ Fazit:

Die Erstellung des Programms an sich, hatte schon noch seine Momente, wo man stecken geblieben ist. Das schöne jedoch an der Gruppenarbeit ist, dass man sich stets austauschen kann bei jeder kleinen Frage oder Unklarheit. Der Informationsaustausch ging auch ohne Probleme. Meine C# Kenntnisse konnte ich endlich wieder Dank diesem Projekt anwenden und entrosten. Neue Fähigkeiten habe ich nicht wirklich dazugelernt, aber allgemeinen Skill schon.  
Die Dokumentation hingegen war sehr mühsam und lief träge im Vergleich zum Programmieren. Lust hatte auch keiner in der Gruppe dazu, aber nicht, weil wir zu faul sind, sondern weil jeder viel lieber am Projekt direkt am Code arbeiten wollte.

Gabriel’s Fazit:

Ich finde im Allgemeinen ist alles sehr gut verlaufen, wir konnten uns gut formatieren, um effizient zu arbeiten. Wir wollten Anfangs, das Coden in Pakete aufteilen, sodass jeder daran einzeln arbeiten kann, aber wir kamen zum Entschluss, dass es ineffizient ist, da es nicht so viel Code ist und wir lieber im Pairprogramming arbeiten. Wir sind froh, dass wir keine großen Probleme hatten, da wir 4 Köpfe hatten, um diese zu lösen, was bei uns immer funktionierte. Etwas, was auch zum größtenteils gut lief, war die Führung der Dokumentation, jeder hat daran gearbeitet, nachdem dieser etwas realisierte. Da aber jeder einzeln daran arbeitete, fehlte uns ein wenig die Übersicht, weswegen wir am Ende viel ergänzen mussten, aber allzu schlimm war dies auch nicht. Das nächste Mal würden wir die Angehens Weise für die Dokumentation anders angehen, denn Rest würden wir beibehalten.