Projektdokumentation

# Informieren

## Ihr Projekt

Herr Muharemi, Brunokowski und Ilic arbeiten an diesem Projekt. Bei diesem Projekt geht es um einen Währungsumrechner, dabei nehmen wir eine API aus dem Internet, die jederzeit die Währungen aktualisiert. Dabei nutzen wir F# als Programmiersprache. Die Idee dahinter ist, dass unser Währungsumrechner es ermöglicht, Geldbeträge von einer Währung in eine andere umzurechnen. Das ermöglicht es den Leuten zu sehen wie viel Wert der Euro in Franken hat. Man kann Währung aus der ganzen Welt eingeben.

## Quellen

https://apilayer.com/marketplace/fixer-api --> API für die Währungen aller Welt

https://entwickler.de/dotnet/entwicklung-leicht-gemacht-mit-f --> C# wird mit F# verglichen, um zu schauen wo die Unterschiede sind

https://www.youtube.com/watch?v=kSfi6kcZrzM --> F# Einführung über Funktionen, Testen usw.

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| **1** | **Muss** | **Funktional** | **Startwährung auswählen können** |
| **2** | **Muss** | **Funktional** | **Zielwährung auswählen können** |
| **3** | **Muss** | **Funktional** | **Betrag auswählen können** |
| **4** | **Muss** | **Funktional** | **Umrechnung, Resultat wird gezeigt** |
| **5** | **Muss** | **Funktional** | **Tabelle mit allen verfügbaren Währungen wird angezeigt** |
| **6** | **Muss** | **Funktional** | **Währungsumrechnung muss mit neuestem Kurs aus API abgelesen werden** |
| **7** | **Muss** | **Rand** | **Fehlermeldung falls API unerreichbar ist** |

## 1.5 Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Vorbereitung** | **Eingabe** | **Erwartete Ausgabe** |
| **1.1** | **Programm gestartet** | 1. **Währung umrechnen [1]** 2. **CHF** 3. **EUR** 4. **Betrag auswählen** | **Umrechnung mit neustem Kurs von Franken in Euro** |
| **2.1** | **Sicherstellen, dass API nicht erreichbar ist (Fehler simulieren)** | 1. **{w}** 2. **{x}** 3. **{y}** 4. **{z}** | **Fehlermeldung erscheint, die besagt, dass die API nicht erreichbar ist** |
| **3.1** | **Programm gestartet** | 1. **Währung umrechnen [1]** 2. **CHF** 3. **CHF** 4. **Betrag auswählen** | **Gleicher Betrag, den man eingegeben hat, wird wieder angezeigt** |
| **4.1** | **Programm gestartet** | 1. **Heutigen Kurs anzeigen [2]** 2. **USD** | **"Der aktuelle Wechselkurs für 1 Euro zu %string (USD) ist: %float (EUR)"** |

1. Planen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** |
| 1.1 | 31.08 | Startwährung eingeben können | 30 |
| 1.2 | 31.08 | Prüfung korrekter Eingabe | 15 |
| 2.1 | 31.08 | Zielwährung eingeben können | 30 |
| 2.2 | 31.08 | Prüfung korrekter Eingabe | 15 |
| 3.1 | 31.08 | Eingabefeld zum Betrag | 90 |
| 3.2 | 31.08 | Prüfung korrekter Eingabe | 15 |
| 4.1 | 07.09 | Logik für die Konversion | 240 |
| 4.2 | 07.09 | Anzeige des Resultates | 60 |
| 5.1 | 07.09 | Darstellung der Währungstabelle im Interface | 30 |
| 6.1 | 14.09 | Verbinden der API | 60 |
| 6.2 | 14.09 | Abrufen der neuesten Kurse | 30 |
| 6.3 | 14.09 | Implementierung neuer Formeln mit API-Kurse | 60 |
| 6.4 | 14.09 | Richtige Anzeige des Resultats | 60 |
| 7.1 | 14.09 | Überwachung der Erreichbarkeit der API | 15 |
| 7.2 | 14.09 | Anzeige einer Fehlermeldung, falls sie nicht erreichbar ist | 15 |
| TOTAL: | | | 765 |

1. Entscheiden
2. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Datum** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 1.1 | 31.08 | Startwährung eingeben können | 30 | 60 |
| 1.2 | 31.08 | Prüfung korrekter Eingabe | 15 | 10 |
| 2.1 | 31.08 | Zielwährung eingeben können | 30 | 30 |
| 2.2 | 31.08 | Prüfung korrekter Eingabe | 15 | 10 |
| 3.1 | 31.08 | Eingabefeld zum Betrag | 90 | 60 |
| 3.2 | 31.08 | Prüfung korrekter Eingabe | 15 | 10 |
| 4.1 | 07.09 | Logik für die Konversion | 240 | 270 |
| 4.2 | 07.09 | Anzeige des Resultates | 60 | 40 |
| 5.1 | 07.09 | Darstellung der Währungstabelle im Interface | 30 | 20 |
| 6.1 | 14.09 | Verbinden der API | 60 | 60 |
| 6.2 | 14.09 | Abrufen der neuesten Kurse | 30 | 30 |
| 6.3 | 14.09 | Implementierung neuer Formeln mit API-Kurse | 60 | 60 |
| 6.4 | 14.09 | Richtige Anzeige des Resultats | 60 | 60 |
| 7.1 | 14.09 | Überwachung der Erreichbarkeit der API | 15 | 15 |
| 7.2 | 14.09 | Anzeige einer Fehlermeldung, falls sie nicht erreichbar ist | 15 | 15 |

Beim Realisieren der geplanten Teilaufgaben haben wir die Pair-Programmieren Methode angewandt. Wir haben meistens zusammen an den verschiedenen Teilaufträgen des Projektes gleichzeitig gearbeitet.

1. Kontrollieren

## **Testprotokoll**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| **1.1** | **20.09** | **OK** | Brunokowski |
| **2.1** | **20.09** | **OK** | Brunokowski |
| **3.1** | **20.09** | **OK** | Brunokowski |
| **4.1** | **20.09** | **OK** | Brunokowski |

1. Auswerten

Fazit:

Wir konnten das Programm effektiv als Team realisieren. Es sind keine Probleme während dem Realisieren aufgetaucht. Was besonders gut gelaufen ist, ist die API-Integration, da wir es in anderen Modulen bereits gelernt haben. Was ein bisschen für uns eine Herausforderung war, war am Anfang bei der Umsetzung von der Logik der Umrechnungen von Währungen. Das Projekt war insgesamt ein Erfolg und wir konnten viele Grundkenntnisse in F# bekommen. Wir haben gelernt, wie man Funktionen in F# anwendet und wie man eine API-Integration in einem F# Programm implementiert.