

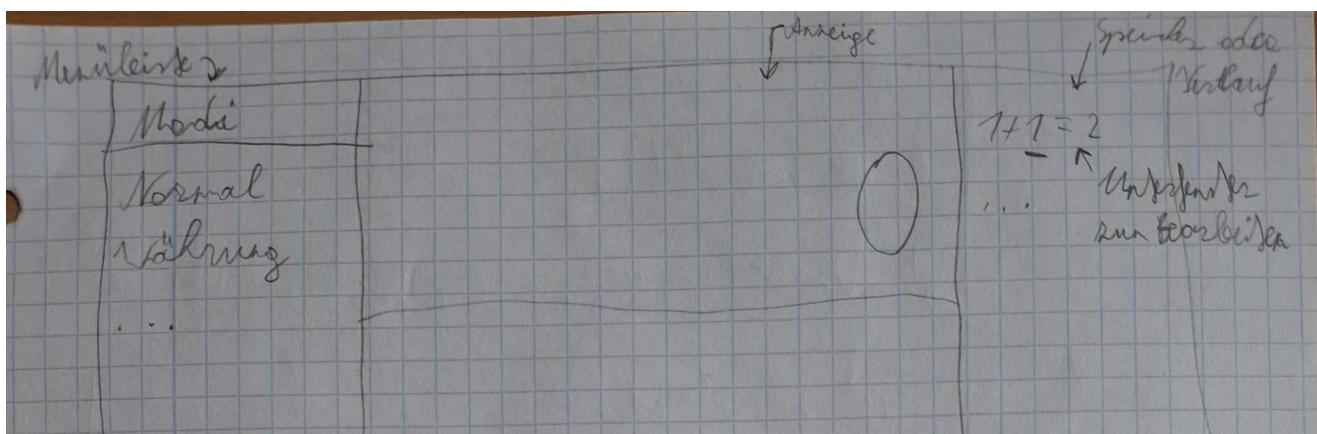
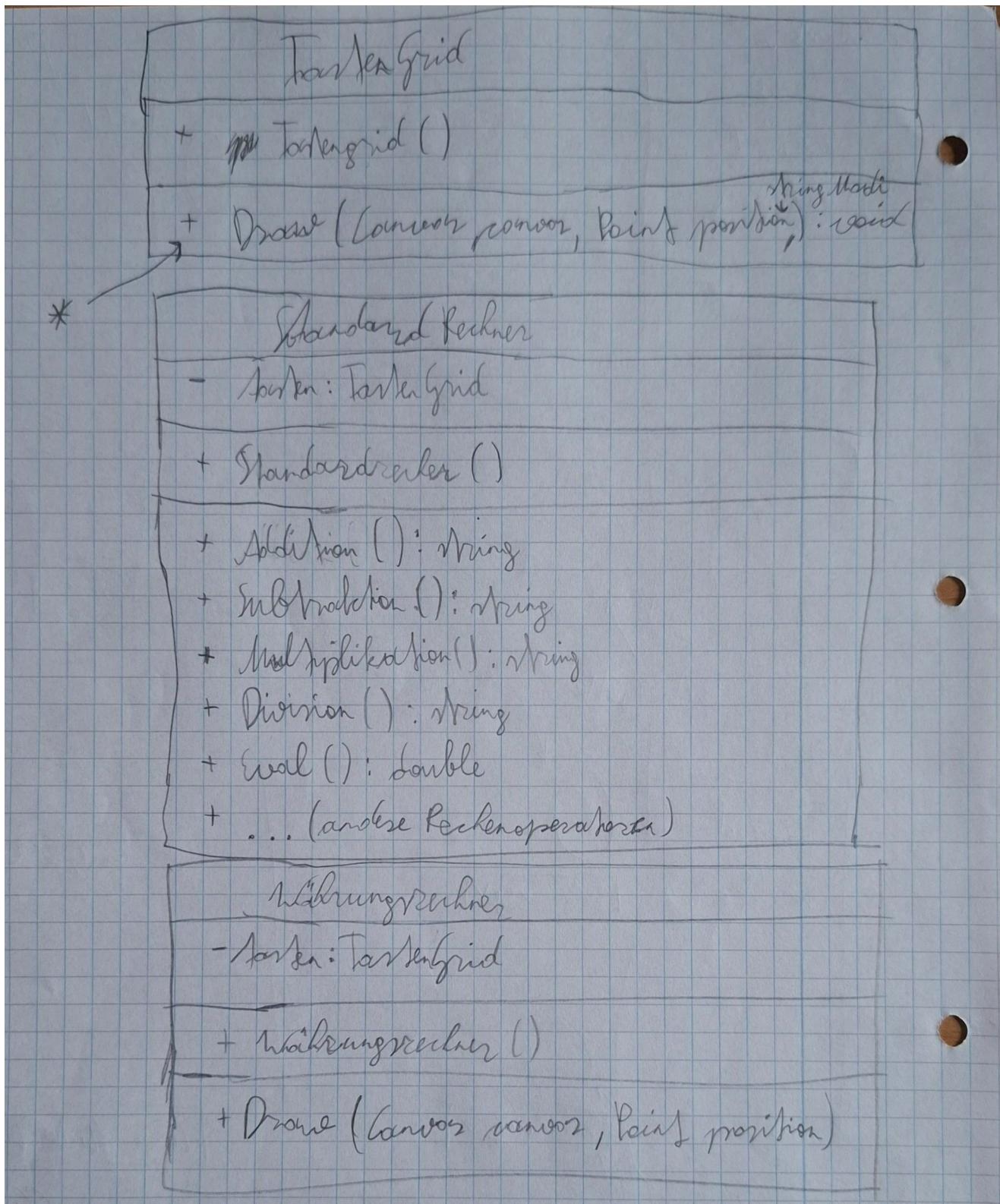
# Dokumentation

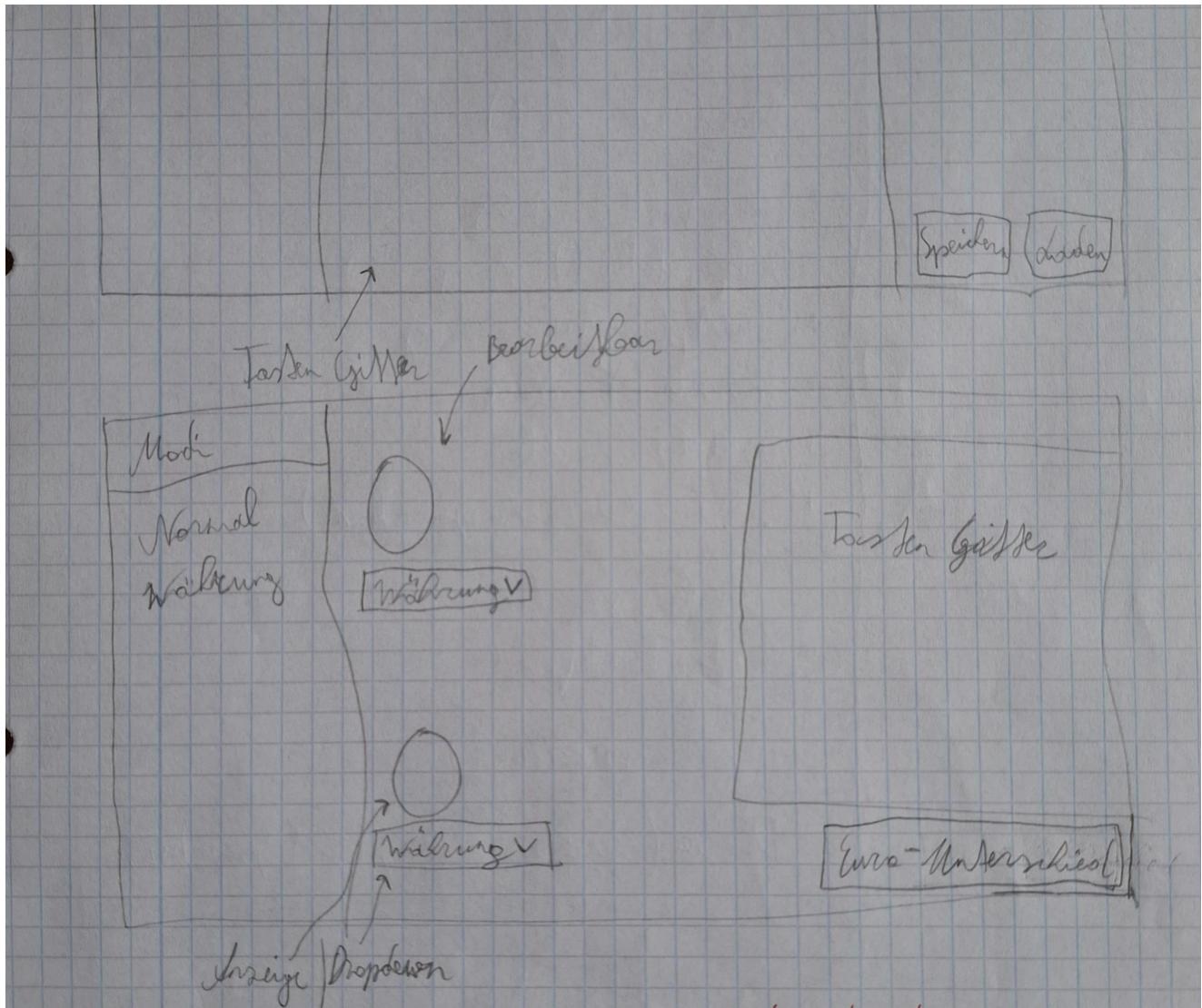
---

## Multi Taschenrechner

---

Erstes Klassendiagramm und Grund Vorstellung





## Must haves

- Normaler Taschenrechner (+, -, /, \*, 2.Wurzel, hoch2, clear)
- Menüleiste
- Rechnung&Ergebniss Speichern/Laden
- Umrechner von Währungen
- Graphen Zeichner
- Logging

## Nice to haves

- Umrechnungen (z.B. Länge, Winkel, Temperaturen)
- Diagramm über den grafischen Unterschied von Währungen
- Graph als png herunterladen
- Web API Anbindung
- Verwendung von externer Bibliothek (NuGet Package Ncalc)

## Wie soll die Umsetzung sein?

Milestones

1. Klassendiagramme, Planungsphase fertig
2. normaler Taschenrechner anfangen; Graphen Zeichner anfangen
3. Graphen Zeichner weiter machen; Serialisierung bei normaler Taschenrechner
4. Graphen Zeichner und normaler Taschenrechner fertig; Währungsrechner und Design anfangen
5. Währungsrechner fertig; Diagramm anfangen
6. Diagramm fertig; Längen-, Winkel-, und Temperaturenrechner anfangen
7. Mit den Rechnern fertig und Design fertigstellen
8. Dokumentation feinschliff

## Wer macht was?

Emil: Fokus: Front End (z.B. Menüleiste) und Geogebra Fake (Back und front end)

Joudi: Fokus: Back End (z.B. der normale Taschenrechner) und auch Front End ein bisschen

## Wie wurde die Software getestet?

Sie wurde durch Visual Studio von Microsoft getestet.

## Welche Softwarevoraussetzungen werden benötigt.

Die neueste Version von Visual Studio mit ".Net6.0".

## Funktionsblöcke bzw. Architektur

**NormalCalc**

Klasse

- Felder
  - dot : bool
  - lastEntry : string
  - lastNums : string
- Eigenschaften
  - Ergebnis { get; set; } : string
  - LastEntry { get; set; } : string
  - LastNums { get; set; } : string
  - Rechnung { get; set; } : string
- Methoden
  - AddString(string entry, Label lblOutput) : void
  - Berechnen(string rechnung) : string
  - Clear(Label lblOutput) : void
  - DeSerializeToCsv(string serialized) : NormalCalc
  - Negate(Label lblOutput) : void
  - NormalCalc()
  - NormalCalc(string rechnung, string ergebnis)
  - OneDividedBy(Label lblOutput) : void
  - Percent(Label lblOutput) : void
  - RemoveString(Label lblOutput) : void
  - SerializeToCsv() : string
  - Squared(Label lblOutput) : void
  - UpdateVariables(string result) : void

**NormalCalcList**

Klasse

- Felder
  - calcs : List<NormalCalc>
- Methoden
  - Add(NormalCalc calc) : void
  - Load(string filename) : void
  - Save(string filename) : void
  - SetLastCalc(string str, Label lbl, NormalCalc calc) : void
  - UpdateListBox(ListBox listBox, Label lbl) : void

**Logging**

Klasse

- Felder
  - logger : Logger

**Currency**

Klasse

- Felder
  - client : HttpClient
  - dot : bool
  - lastTwo : int
- Eigenschaften
  - Name { get; set; } : string
  - Shortcut { get; set; } : string
- Methoden
  - AddString(string entry, Label lblOutput, Label lblOutputTrgt, ComboBox CBBaseCurrency, ComboBox CBTrgtCurrency) : void
  - Clear(Label lblInput, Label lblOutput) : void
  - ConvertCurrency(string baseCurrency, string targetCurrency, double amount) : Task<double>
  - Currency()
  - Currency(string shortcut, string name)
  - RemoveString(Label lblOutput, Label lblOutputTrgt, ComboBox CBBaseCurrency, ComboBox CBTrgtCurrency) : void

**CurrenciesCollection**

Klasse

- Felder
  - client : HttpClient
  - currencies : List<Currency>
- Methoden
  - Add(Currency currency) : void
  - CurrenciesCollection()
  - GetCurrencies(ComboBox CB) : void

**CurrencyComparison**

Klasse

- Felder
  - client : HttpClient
  - CurrencyRates : Dictionary<string, double>
- Methoden
  - CurrencyComparison()
  - DrawRects(Canvas canvas, string currency) : void
  - GetData(string currency) : Task<Dictionary<string, double>>

**TemperatureConverter**

Klasse

- Felder
  - dot : bool
  - fromC : Dictionary<string, Func<double, double>>
  - toC : Dictionary<string, Func<double, double>>
- Methoden
  - AddString(string entry, Label lblOutput, Label lblOutputTrgt, ComboBox CBBaseTemperature, ComboBox CBTrgtTemperature) : void
  - Clear(Label lblInput, Label lblOutput, ComboBox CBBaseTemperature, ComboBox CBTrgtTemperature) : void
  - ConvertTemperature(Label lblOutput, Label lblOutputTrgt, ComboBox CBBaseTemperature, ComboBox CBTrgtTemperature) : void
  - RemoveString(Label lblOutput, Label lblOutputTrgt, ComboBox CBBaseTemperature, ComboBox CBTrgtTemperature) : void

**Graph** Klasse

- Felder
  - .addProperty : Brush
  - function : Func<double, double>
- Eigenschaften
  - Color { get; set; } : Brush
  - Function { get; set; } : Func<double, double>
- Methoden
  - ConvertPowerExpressions(string functionExpression) : string
  - ConvertRootExpressions(string functionExpression) : string
  - ConvertStringToFunc(string functionExpression) : Func<double, double>
  - DrawGraph(Canvas canvas, int scale) : void
  - Edit(string function) : void
  - Graph(string function)

**GraphList** Klasse

- Felder
  - \_brushesArray : Brush[]
  - \_grapheneList : List<Graph>
  - \_rounds : int
  - \_usedBrushes : List<Brush>
- Methoden
  - Add(Graph graph) : void
  - DrawGraphene(Canvas canvas, CoordinateSystem coordinateSystem, int scale) : void
  - GetGraphByLabelName(string labelName) : Graph
  - GetList() : List<Graph>
  - GraphList()

**Length** Klasse

- Felder
  - dot : bool
- Eigenschaften
  - Converting { get; set; } : double
  - LengthName { get; set; } : string
- Methoden
  - AddString(string entry, Label lblOutput, Label lblOutputTrgt, ComboBox CBBASELength, ComboBox CBTrgtLength) : void
  - Clear(Label lblInput, Label lblOutput) : void
  - Length() : double
  - Length(double converting, string lengthName)
  - RemoveString(Label lblOutput, Label lblOutputTrgt, ComboBox CBBASELength, ComboBox CBTrgtLength) : void
  - UpdateOutput(Label lblOutput, Label lblOutputTrgt, ComboBox CBBASELength, ComboBox CBTrgtLength) : void

**LengthDic** Klasse

- Felder
  - \_lengthDic : Dictionary<string, double>
- Methoden
  - Convert(double value, string fromUnit, string toUnit) : double

**CoordinateSystem** Klasse

- Felder
  - \_scale : int
  - \_scaleStep : int
- Methoden
  - CoordinateSystem(int scale, int scaleStep) : void
  - DrawCoordinateSystem(Canvas canvas) : void
  - ZoomIn(Canvas canvas) : void
  - ZoomOut(Canvas canvas) : void

**Angle** Klasse

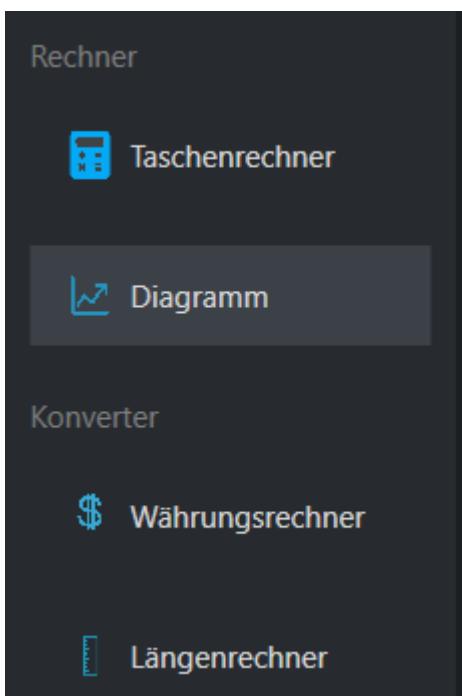
- Felder
  - dot : bool
- Eigenschaften
  - AngleName { get; set; } : string
  - Converting { get; set; } : double
- Methoden
  - AddString(string entry, Label lblOutput, Label lblOutputTrgt, ComboBox CBBASEAngle, ComboBox CBTrgtAngle) : void
  - Angle() : double
  - Angle(double converting, string angleName)
  - Clear(Label lblInput, Label lblOutput) : void
  - RemoveString(Label lblOutput, Label lblOutputTrgt, ComboBox CBBASEAngle, ComboBox CBTrgtAngle) : void
  - UpdateOutput(Label lblOutput, Label lblOutputTrgt, ComboBox CBBASEAngle, ComboBox CBTrgtAngle) : void

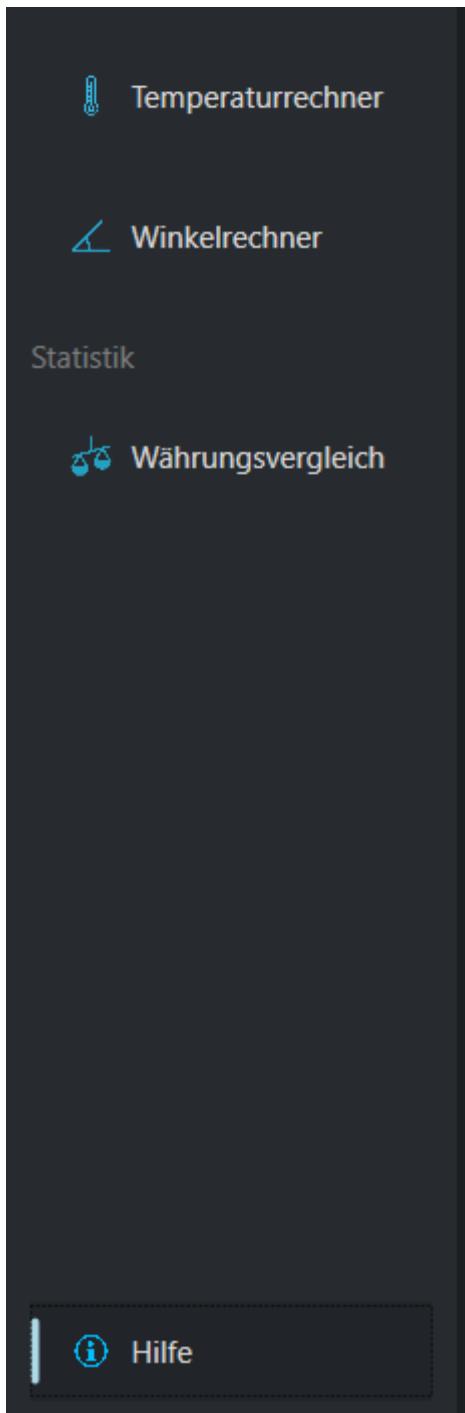
**AngleConverter** Klasse

- Methoden
  - Convert(double value, string fromUnit, string toUnit) : double

## Bedienungsanleitung

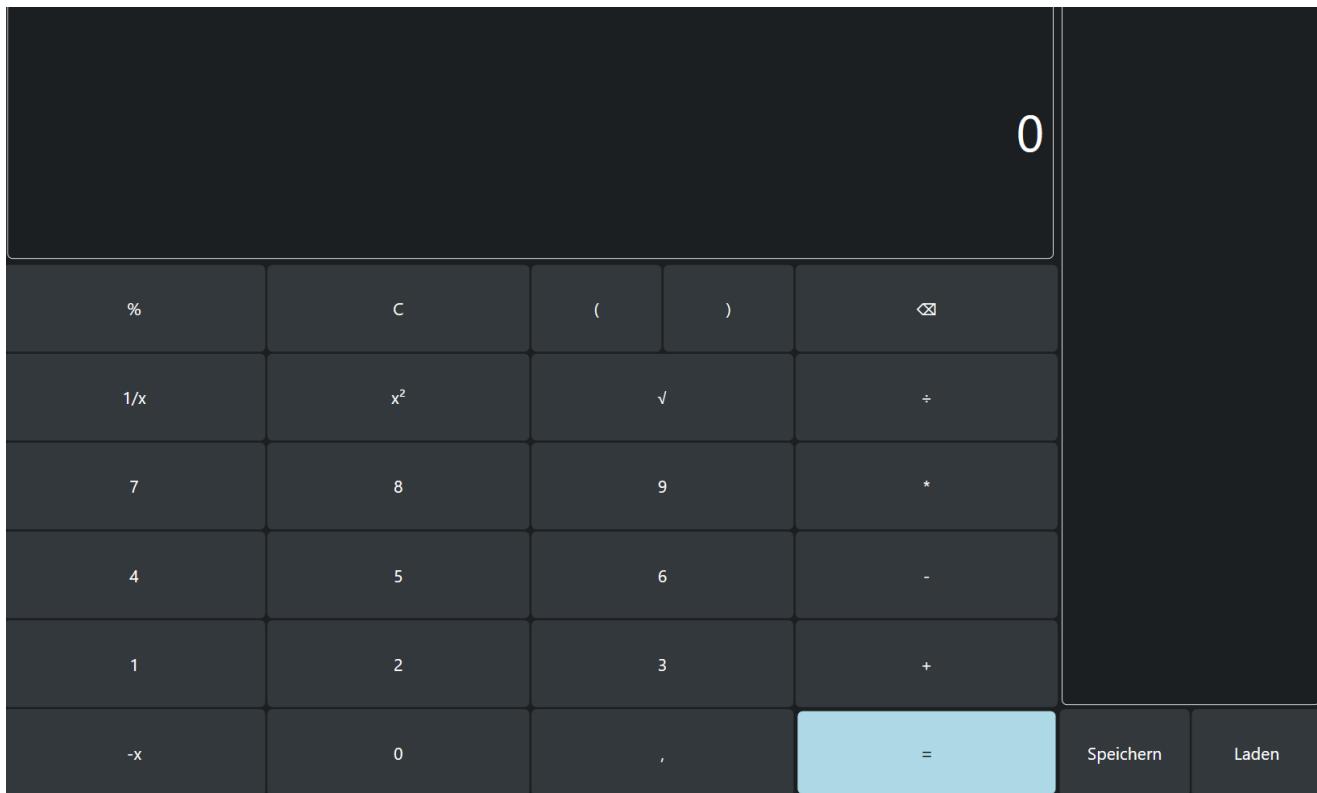
### Menüleiste





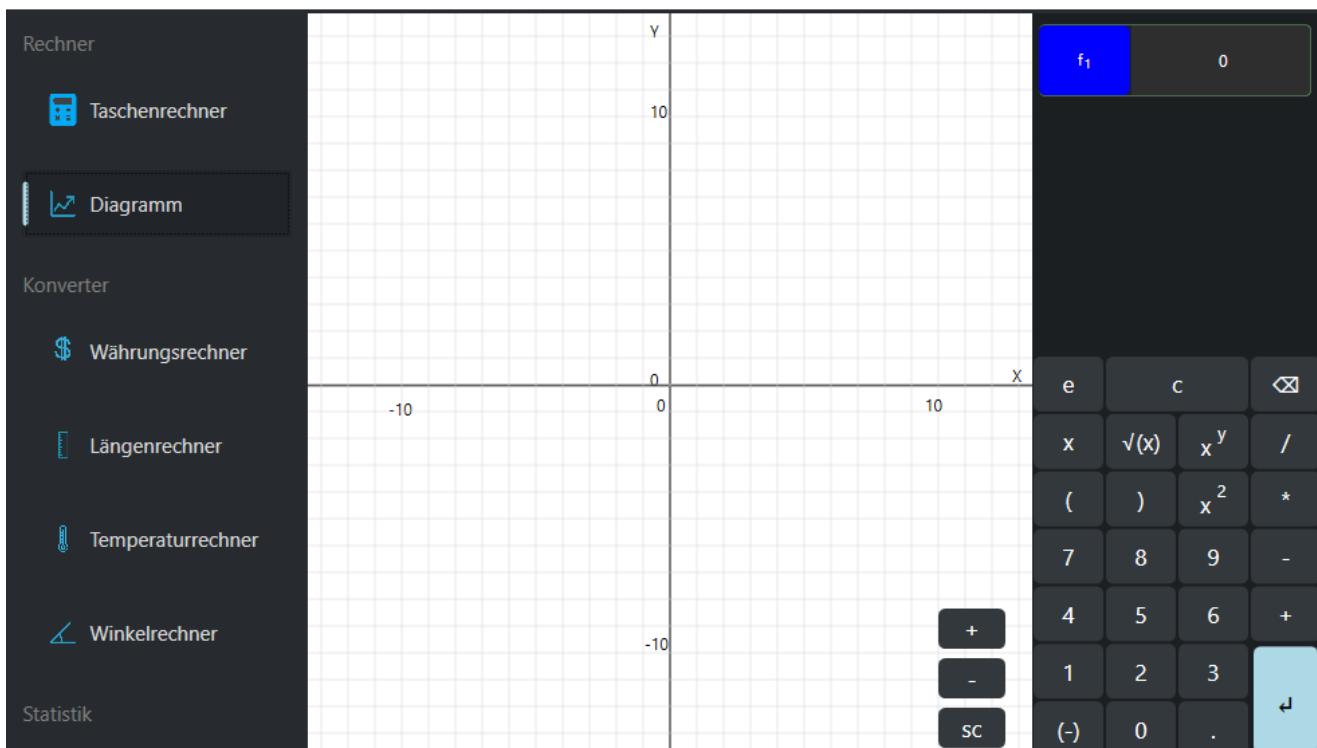
Hier können Sie zwischen den anderen Modi immer wechseln. In den Einstellungen sehen Sie noch ein paar Informationen über die Ersteller des Programms.

Normaler Taschenrechner



Hier können Sie Rechnung berechnen lassen, indem man mithilfe des Zahlengitters unten Zahlen oder Zeichen eingibt oder Rechenverfahren ausführen. Wenn man einen Rechenfehler macht dann kommt als Lösung "undefined" raus. Rechts wird dann ein Verlauf Ihrer Rechnungen angezeigt. Wenn Sie auf eine Rechnung klicken wird es in das Eingabefeld überschrieben. Zuletzt gibt es unter dem Verlauf noch 2 Buttons mit denen man den Verlauf speichern oder von wo anders laden kann.

## Diagramm



Hier kann man auf der rechten Seite eine Funktion eingeben und sie wird dann auf dem Koordinatensystem gezeichnet. Man kann bis zu 10 Graphen zeichnen. Bevor man einen neuen Graphen

machen will muss man zuerst den davor bearbeiten. Wenn man auf den Button z.B. "f1" klickt bei dem Graph kann man diesen dann bearbeiten und die Umrundung wird grün, sodass man sieht welchen Graph Sie gerade bearbeiten. Drücken Sie bitte immer Enter wenn sie etwas geändert haben, um zu bestätigen das sie es wirklich wollen.

## Währungsrechner

The screenshot shows a user interface for a currency converter. On the left, there are two input fields, each containing a '0'. Below each input field is a dropdown menu listing various currencies with their codes: Australian Dollar - AUD, Bulgarian Lev - BGN, Brazilian Real - BRL, Canadian Dollar - CAD, Swiss Franc - CHF, and Chinese Renminbi Yuan - CNY. To the right of the input fields is a numeric keypad grid. The grid consists of a top row with 'C' and a clear/cancel button, followed by three rows of three buttons each, labeled 7, 8, 9 in the top row; 4, 5, 6 in the middle row; and 1, 2, 3 in the bottom row. The number 0 is located at the bottom center of the grid.

Hier können Sie eine Währung auswählen, indem man die Leiste unter der ersten 0 klickt und es erscheint eine Liste mit vielerlei Währungen die man auswählen kann, dies klappt auch mit der unteren Leiste. Jedoch können Sie nur dann die Zahleneingabe mithilfe des Gitters rechts machen, wenn Sie Ihre Basis Währung und die Ziel Währung auswählen.

## Längenrechner

A dark-themed calculator interface. The left side features two input fields, each with a dropdown menu showing unit options. The top input field has '0' and the bottom one also has '0'. To the right is a numeric keypad with a grid layout. The top row contains 'C' and a clear/cancel button. The second row contains '7', '8', and '9'. The third row contains '4', '5', and '6'. The fourth row contains '1', '2', and '3'. The bottom row contains '0', a decimal point '.', and a negative/positive sign '-/+'. A small note at the bottom right of the keypad says 'DEG RAD GRAD'.

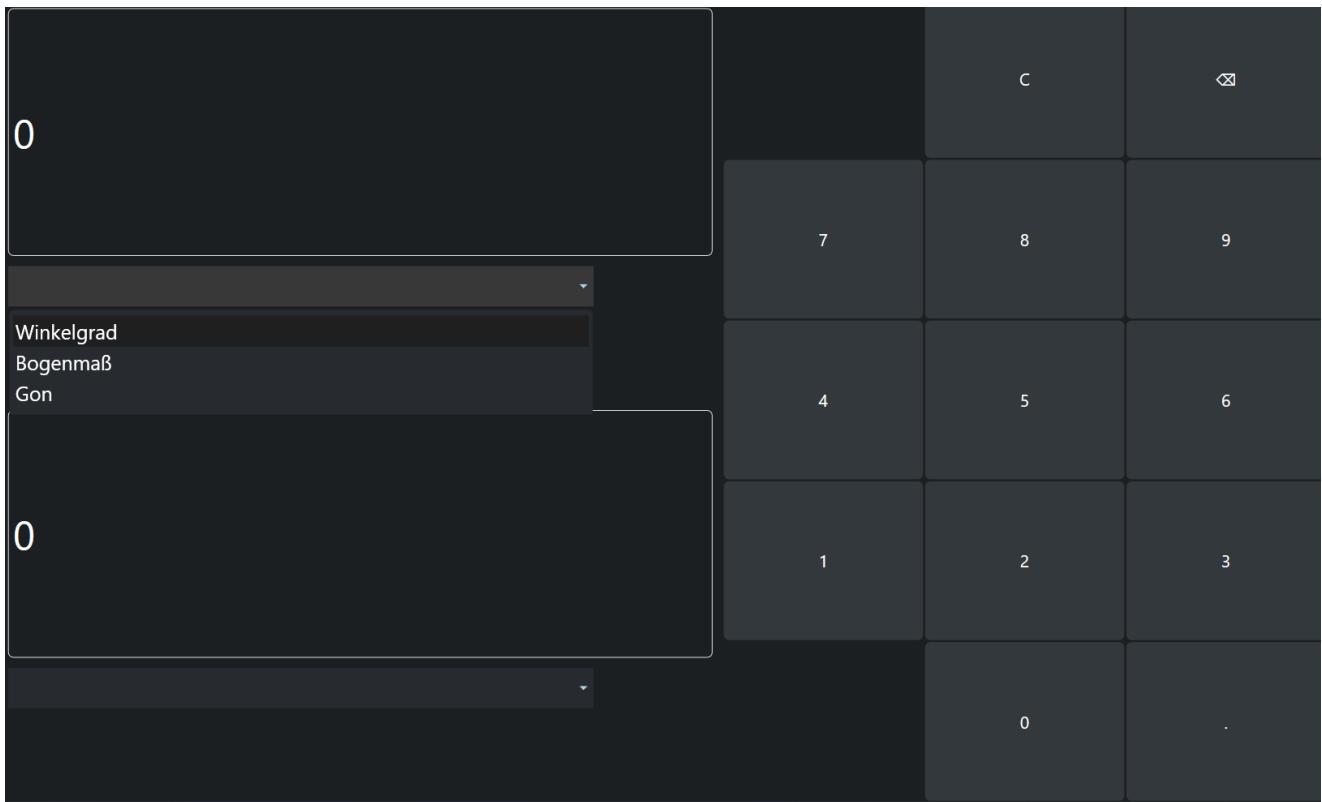
Hier ist es gleich wie beim Währungsrechner, bloß sind es hier Längen anstatt Währungen. Für die Zahlen Eingabe müssen Sie auch wieder zuerst beide Längen auswählen.

### Temperaturrechner

A dark-themed calculator interface. The left side features two input fields, each with a dropdown menu showing unit options. The top input field has '0' and the bottom one also has '0'. To the right is a numeric keypad with a grid layout. The top row contains 'C' and a clear/cancel button. The second row contains '7', '8', and '9'. The third row contains '4', '5', and '6'. The fourth row contains '1', '2', and '3'. The bottom row contains '-/+' (negative/positive), '0', and a decimal point '.'. A small note at the bottom right of the keypad says 'DEG RAD GRAD'.

Hier ist es gleich wie beim Währungsrechner, bloß sind es hier Temperaturen anstatt Währungen und es gibt nun einen Knopf damit Sie Ihre Eingabe negieren können. Für die Zahlen Eingabe müssen Sie auch wieder zuerst beide Temperaturen auswählen.

### Winkelrechner



Hier ist es gleich wie beim Währungsrechner, bloß sind es hier Winkel Einheiten anstatt Währungen. Für die Zahlen Eingabe müssen Sie auch wieder zuerst beide Winkel Einheiten auswählen.

## Währungsvergleich



Hier handelt es sich um eine Statistik, die grafisch angibt wie sehr sich die anderen Währungen von der oben ausgewählten Währung unterscheiden. Zu Ihrer Information: Die grafischen Werte hier sind nicht genau da sie logarithmisch skaliert sind. Und je größer der Balken unten ist, desto mehr Wert hat die ausgewählte Währung zu der Währung mit dem Balken.

## Hilfe



Emil Wagner aka EdgeLordE  
E-mail: emil.wagner@student.htl-rankweil.at  
Snapchat: emil\_wagner19



Joudi Khalil aka Gaul  
E-mail: joudi.khalil@student.htl-rankweil.at  
Snapchat: joudi\_khai9410

Hier sehen sie unser Team und deren Kontaktdaten, die hart an diesem Projekt geschuftet haben.

## Quellen Bilder

<https://www.flaticon.com/de/kostenloses-icon>

## Projekttagebuch

### Joudi

#### Dokumentation

Tag	Was gemacht?
24.04.	Planung
01.05.	Normaler Taschenrechner Basis Design gemacht
08.05.	Normaler Taschenrechner hälften fertig, Paket "ncalc" installiert; an Doku gearbeitet
15.05.	Prozent, negieren, Hoch 2, Wurzel, 1 Durch x fertig; Komma fertig angepasst; Klammern Funktionalität geändert
22.05.	Normaler Taschenrechner fertig; Logging fehlt noch
24.05.	Logging Pakete installiert; Normaler Taschenrechner nötige Logs gemacht; offiziell fertig mit Normaler Taschenrechner

<b>Tag</b>	<b>Was gemacht?</b>
26.05.	Neues UserControl Fenster hinzugefügt für den Währungrechner bzw. Währungsrechner angefangen; Basis Design fertig
29.05.	Währungsrechner fast fertig
04.06.	Währungsrechner fertiggestellt Logging fehlt
05.06.	Währungsvergleich angefangen und 3/4 fertig richtige Balken vom Diagramm fehlen
06.06.	Dezimalstelle Bug behoben bei Währungsrechner und Funktionalitäten verbessert; Währungsvergleich fertig Logging fehlt
08.06.	Logging bei Währungsrechner, Währungsvergleich, und Taschenrechner gemacht; Taschenrechner Bugs behandelt
09.06.	Temperaturrechner fertiggestellt mit Logging

## Probleme

<b>Problem</b>	<b>Lösung</b>
Die Logik hinter der Einabe der einzelnen Zahlen und Zeichen	5h lang dran geblieben um es zu lösen.
Problem mit dem Dezimaltrennzeichen, dass es "." ignoriert	Herausgefunden das es Kulturabhängig ist und man muss es mit einer Zeile Code ändern, dass "." als Dezimaltrennzeichen verwendet wird
API Wahl, weil viele APIs Zahlungsbedingt sind und vielleicht betrügerisch	Herr Diem um Hilfe gebittet und dank Herr Diem eine gefunden
Beim Währungsvergleich: Die Höhe der Balken im Diagramm => waren zu klein	Herausgefunden das man mit logarithmischem Skalieren die kleinen Werte vergrößern kann und die großen Werte komprimieren kann (Habe es in Mathe gelernt, aber ist mir recht Spät aufgefallen)

## Emil

### Dokumentation

<b>Tag</b>	<b>Was gemacht?</b>
24.4	PLanung
29.4	Menü mit einem Tutorial angefangen
30.4	Menü hat nicht geklappt wie es sollte also neu angefangen; Menü fast fertiggestellt (ohne Bilder)

<b>Tag</b>	<b>Was gemacht?</b>
1.5	Menü nochmals überarbeitet und noch den Style des Menüs verschönert; Dem normalen Taschenrechner einen schönen Style verpasst
3.5	Alle Pflichtorder erstellt und die Programme usw. verschoben; Bilder für das Menü eingefügt; Im Menü die Bilder hinzugefügt
8.5	Bei dem Menü noch mehr Bilder hinzugefügt
9.5	Die Tastatur bei dem Fake Geogebra fertig gemacht mit Style
15.5	Der Einstellungsbutton nach unten verschoben; Das Koordinatensystem erstellt wo man rein und raus zoomen kann;
22.5	Man kann jetzt Graphen zeichnen und rechts max 10 Graphen eingeben (noch nicht alle eingaben behandelt)
29.5	Graph kan eingegeben werden und wird dann gezeichnet (man kann noch nicht alles eingeben)
4.6	Man kann bei dem Graph hoch und Wurzel zahlen eingeben und diese werden auch gezeichnet
5.6	Man kann jetzt bei dem Graphen alles eingeben und es funktioniert soweit (noch keine Fehler prüfung), Design von dem Währungsumrechner gemacht (hauptsächlich für die comboBox)
6.6	Menü mit Überschriften versehen sodass alles schön für den Nutzer gegliedert ist, Design für das Diagramm fertig gemacht, Längenumrechner gemacht, Screenshot für das Diagramm gemacht
7.6	Bilder in das Menü hinzugefügt und dann in Gimp so geändert das die bilder Blau anstatt Schwarz sind, Bei dem Diagramm so gestylt, dass das ausgewählte Element grün umrandet wird.
8.6	UserControlSettings fertig gemacht und geloggt, Winkelumrechner gemacht, Alle meine Dateien die ich bis jetzt gecodet habe geloggt
8.6 (Nacht)/9.6	Winkelumrechner gemacht und alles dokumentiert
9.6	Winkelumrechner geloggt und bei dem MainWindow ein ScroolViewer hinzugefügt

## Probleme

<b>Problem</b>	<b>Lösung</b>
Wie wandle ich einen Hochzahl string in eine Rechnung um	Ich ersetze die Hochzahl durch Math.Pow()
Wie rechne ich jetzt einen string "Math.Pow"	Geht nicht ich nehme Ncalc.Pow

<b>Problem</b>	<b>Lösung</b>
Wie rechne ich Wurzel aus wenn es keine Wurzel Funktion bei Ncalc gibt	Ich mache es mit der Hochzahl, weil jede Wurzel kann man auch in eine Hochzahl umwandeln
Wie mache ich ein schwarzes Logo zu einem Blauen	Ich mache in Gimp der Farbvertausch
Wie macht man in Gimp das die Bilder nicht nur Graustufen haben oder nur die Farbe Weiß und Schwarz	Man geht im Menü auf Bild und dann Modus und dann RGB