

---

## Taschenrechner Zahlensysteme mit WPF

---

Objektbasiert programmieren nach Vorgabe

### Inhaltsverzeichnis

---

1	Ziel .....	2
2	Ausgangslage .....	2
3	Aufgabenstellung .....	3
3.1	Generelle Hinweise zur Umsetzung .....	3
3.2	Forderungen an den Zahlensystemtaschenrechner .....	3
4	Geforderte Lösungsbereiche .....	8
5	Hilfsmittel .....	8
6	Zeitbedarf .....	8
7	Abbildungsverzeichnis .....	8

## 1 Ziel

Der Lernende macht weitere Erfahrungen wie eine Benutzerschnittstelle für einen einfachen Taschenrechner für verschiedene Zahlensysteme erstellt wird. Ziel dieser Aufgabe ist es, das bereits durch die Aufgabe: „Einfacher Taschenrechner“ erworbenes Wissen zu vertiefen und neue weiterführende Aspekte des GUI-Designs kennenzulernen. Dieses Dokument hat aber nicht den Anspruch der Vollständigkeit und der perfekten Umsetzung des Problems. In der professionellen Umsetzung würde man das MVC-Prinzip einsetzen. Da dies aber zum jetzigen Zeitpunkt noch nicht optimal erklärt werden kann (Grundwissen fehlt) wird dieser Aspekt zu einem späteren Zeitpunkt nachgeholt und vertieft werden.

## 2 Ausgangslage

In dieser Aufgabe wollen wir einen Taschenrechner für diverse Zahlensysteme umsetzen. Damit die Aufgabe an Übersichtlichkeit nicht verliert, werden wir uns auf die vier meistgebrauchten Zahlensysteme einschränken. Es sind dies:

- Dezimal-System
- Hexadezimal-System
- Oktal-System
- Binär- System (Dual-System)

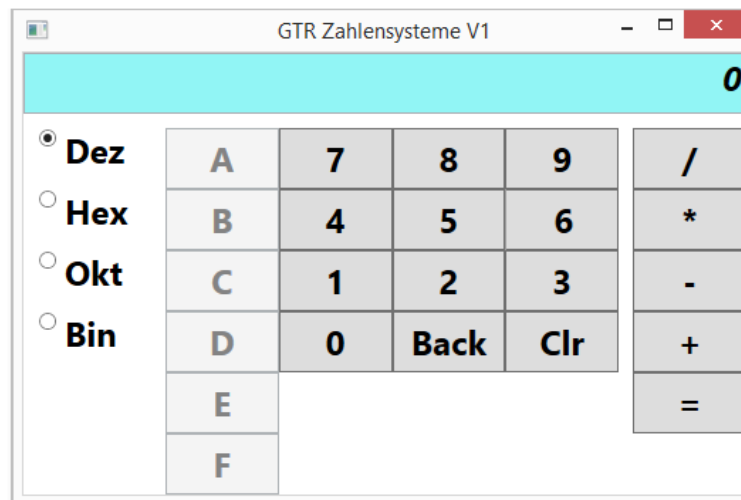


Abbildung 1: GTR Zahlensysteme V1

Wie Sie aus obenstehendem Bild erkennen können, handelt es sich um einen einfachen „Ganzzahl“-Taschenrechner für 4 Zahlensysteme. Das heisst, wir können selber keinen Dezimalpunkt setzen und die Resultatausgabe erfolgt immer als abgerundete Ganzzahl.

Folgende Tasten stehen beim Taschenrechner zur Verfügung:

- |                           |   |
|---------------------------|---|
| Tasten: <b>0 – 9</b>      | Ziffern für Zahleingabe                           |
| Tasten: <b>A – F</b>      | Buchstaben für Zahleingabe beim Hexadezimalsystem |
| Tasten: <b>/, *, -, +</b> | mögliche Rechenoperationen                        |
| Taste: <b>=</b>           | erzeugt die Resultatausgabe                       |
| Taste: <b>Back</b>        | Löscht die letzte eingegebene Ziffer              |
| Taste: <b>Clr</b>         | löscht die Anzeige                                |

Damit eine Division durch 0 nicht durchgeführt werden kann, muss dies überprüft werden. Sollte dieser Fall eintreten, so muss der Benutzer mit einer „MessageBox“ informiert werden.

### 3 Aufgabenstellung

#### 3.1 Generelle Hinweise zur Umsetzung

Da diese Aufgabe eine Fortsetzung der Aufgabe: „Einfacher Taschenrechner“ ist, sollen jetzt nicht mehr alle Teilbereiche der Umsetzung als Tutorial vorgegeben werden. Sie können aber jeder Zeit in die Lösung der letzten Aufgabe hineinschauen um diese Aufgabe selbstständig lösen zu können.

#### 3.2 Forderungen an den Zahlensystemtaschenrechner

##### 3.2.1 Auswahl des Zahlensystems

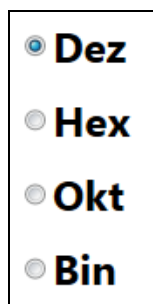


Abbildung 2: RadioButtonGroup

Damit der Benutzer das Zahlensystem frei wählen kann, wollen wir diesmal ein neues Steuerelement einsetzen. Es handelt sich hierbei um das Steuerelement: **RadioButton**.

Ein RadioButton lässt innerhalb einer Gruppe (GroupName) nur immer ein Button aktiv erscheinen. Alle anderen sind deaktiviert. Somit kann der Benutzer nur immer einen Button auswählen um eine gewisse Eigenschaft des Programms einzustellen. In unserem Fall kann also nur immer ein Zahlensystem für die Berechnung ausgewählt werden.

##### 3.2.1.1 Umsetzung im File: MainWindows.xaml

```
<RadioButton GroupName="rbZahlensystem" x:Name="rbDez" Content="Dez" IsChecked="True"
Grid.Column="1" Grid.Row="2" FontSize="24" FontWeight="Bold"
Checked="rbDez_Checked"/>
```

Legende:

GroupName:	Gibt den Namen der Gruppe an zur der alle RadioButtons gehören.
x:Name:	Name des RadioButton. Mit diesem Namen wird der RadioButton im Programmcode angesprochen.
Content:	Sichtbare Beschriftung auf dem Bildschirm.
IsChecked:	Gibt an, dass dieser RadioButton zum Start des Programms selektiert ist. Wenn dieser Eintrag auch bei weiteren RadioButtons vorhanden ist, so wird derjenige Button als selektiert betrachtet, wo der letzte Eintrag vorhanden ist.
Grid.Column:	Angabe in welcher Spalte der RadioButton platziert wird.
Grid.Row:	Angabe in welcher Zeile der RadioButton platziert wird.
FontSize:	Grösse der sichtbaren Buttonbeschriftung.
FontWeight:	Darstellungsart der Buttonbeschriftung.
Checked:	Funktion welche beim Selektieren durch den Benutzer aufgerufen werden soll.

### 3.2.2 Freigabe von Zahlentasten

Damit der Benutzer keine Falscheingaben vornehmen kann, werden nur die für das gewählte Zahlensystem möglichen Ziffern freigeschaltet.

Um dieser Forderung gerecht zu werden, müssen Sie die Tasten sperren oder freigeben. Dies geschieht wie folgt und wird steuerungsmässig im File: MainWindow.xaml.cs eingesetzt:

```
cmd9.IsEnabled = true;
```



Jede Taste kann durch den Befehl:

```
<button>.IsEnabled = true;
```

auf **aktiv** gesetzt werden.

```
cmdA.IsEnabled = false;
```



Jede Taste kann durch den Befehl:

```
<button>.IsEnabled = false;
```

auf **inaktiv** gesetzt werden.

#### 3.2.2.1 Dezimal-Zahlensystem

Bei diesem Zahlensystem arbeiten wir mit der **Basis: 10**. Dies bedeutet, dass die Zahl = 1638 wie folgt geschrieben werden kann:

$$1638 = 1 \cdot 10^3 + 6 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$$



Mögliche Ziffern:

0 .. 9

Abbildung 3: Taschenrechner auf Dezimal-System eingestellt

#### 3.2.2.2 Hexadezimal-Zahlensystem

Bei diesem Zahlensystem arbeiten wir mit der **Basis: 16**. Dies bedeutet, dass die Zahl = 1A3B wie folgt geschrieben werden kann:

$$1A3B = 1 \cdot 16^3 + A \cdot 16^2 + 3 \cdot 16^1 + B \cdot 16^0$$

$$1A3B = 1 \cdot 16^3 + 10 \cdot 16^2 + 3 \cdot 16^1 + 11 \cdot 16^0 = 6715 \text{ (dezimal)}$$

Aus obiger Rechnung haben wir festgestellt, dass den Buchstaben A und B Zahlenwerte zugeordnet werden können. Es gilt beim Hexadezimalsystem die folgende Zuordnung der Zahlen zu den Buchstaben:

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15



Mögliche Ziffern:

0 .. 9

A .. F

Abbildung 4: Taschenrechner auf Hexadezimal-System eingestellt

### 3.2.2.3 Oktal-Zahlensystem

Bei diesem Zahlensystem arbeiten wir mit der **Basis: 8**. Dies bedeutet, dass die Zahl = 1734 wie folgt geschrieben werden kann:

$$1734 = 1 \cdot 8^3 + 7 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 988 \text{ (dezimal)}$$



Mögliche Ziffern:

0 .. 7

Abbildung 5: Taschenrechner auf Oktal-System eingestellt

### 3.2.2.4 Binär-Zahlensystem

Bei diesem Zahlensystem arbeiten wir mit der **Basis: 2**. Dies bedeutet, dass die Zahl = 1011 wie folgt geschrieben werden kann:

$$1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11 \text{ (dezimal)}$$



Mögliche Ziffern:

0 .. 1

Abbildung 6: Taschenrechner auf Binär-System eingestellt

### 3.2.3 Umrechnung von einem Zahlensystem in ein anderes

Entwickeln Sie den Rechner so, dass der Anzeigewert jederzeit in ein anderes Zahlensystem umgewandelt werden kann. Sobald der Benutzer das Zahlensystem ändert, soll der Anzeigewert sofort in das gewählte Zahlensystem gewandelt werden.

Damit die Anzeige gleich den richtigen Umwandlungswert angibt, können Sie die folgende Befehlszeile verwenden:

```
txtAnzeige.Text = Convert.ToString(zwischenwert, basis);
```

Legende:

txtAnzeige.Text	Textfeld, wo der Umwandlungswert dargestellt werden soll.
zwischenwert	Darzustellender Dezimalwert
basis	Basiswert des dazustellenden Zahlensystems

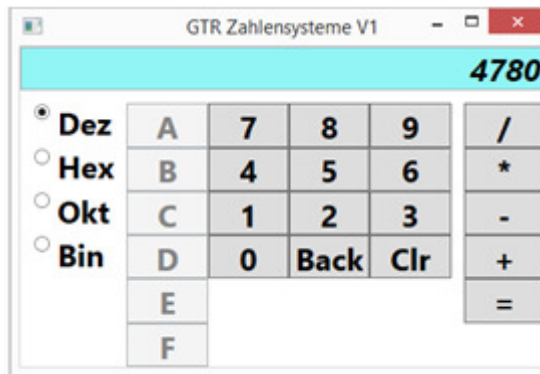


Abbildung 7: Resultatdarstellung im Dezimal-System

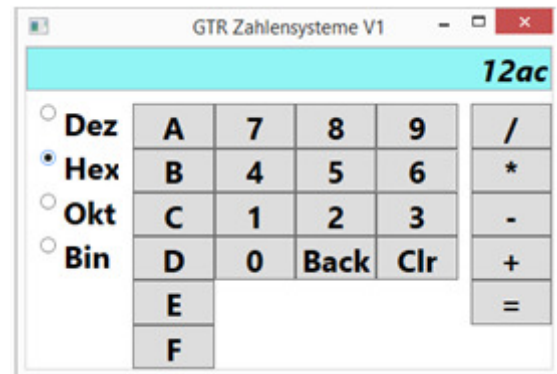


Abbildung 8: Resultatdarstellung im Hexadezimal-System



Abbildung 9: Resultatdarstellung im Oktal-System



Abbildung 10: Resultatdarstellung im Binär-System

### 3.2.4 Generelle Anforderung

Der Taschenrechner selbst soll in der Grösse der Darstellung frei verändert werden können.

---

## 4 Geforderte Lösungsbereiche

---

- Erzeugung des GUI-Oberfläche nach Vorgabe
- Einfügen der Ereignismethoden
- Erklärung der Funktionsweise des Codes
- Auflistung von Verbesserungsmöglichkeiten des GUI

---

## 5 Hilfsmittel

---

Visual Studio

---

## 6 Zeitbedarf

---

ca. 90 Minuten

---

## 7 Abbildungsverzeichnis

---

Abbildung 1: GTR Zahlensysteme V1	2
Abbildung 2: RadioButtonGroup	3
Abbildung 3: Taschenrechner auf Dezimal-System eingestellt	4
Abbildung 4: Taschenrechner auf Hexadezimal-System eingestellt	5
Abbildung 5: Taschenrechner auf Oktal-System eingestellt	5
Abbildung 6: Taschenrechner auf Binär-System eingestellt	6
Abbildung 7: Resultatdarstellung im Dezimal-System	7
Abbildung 8: Resultatdarstellung im Hexadezimal-System	7
Abbildung 9: Resultatdarstellung im Oktal-System	7
Abbildung 10: Resultatdarstellung im Binär-System	7



## Historie

Vers.	Bemerkungen	Verantwortl.	Datum
1.0	- Komplette Aufgabe in C# erstellt	W. Odermatt	23.11.2015

## Referenzunterlagen

LfNr.	Titel / Autor / File / Verlag / ISBN	Dokument-Typ	Ausgabe
1	„Einstieg in Visual C# 2013“ / Thomas Theis / Galileo Computing / 978-3-8362-2814-5	Fachbuch	2014
2	“Visual C# 2012” / Andreas Kühnel / Rheinwerk Computing / 978-3-8362-1997-6	Fachbuch	2013
3	“Objektorientiertes Programmieren in Visual C#” / Peter Loos / Microsoft Press / 3-86645-406-6	Fachbuch	2006