marxer engineering & computing

www.mec.li

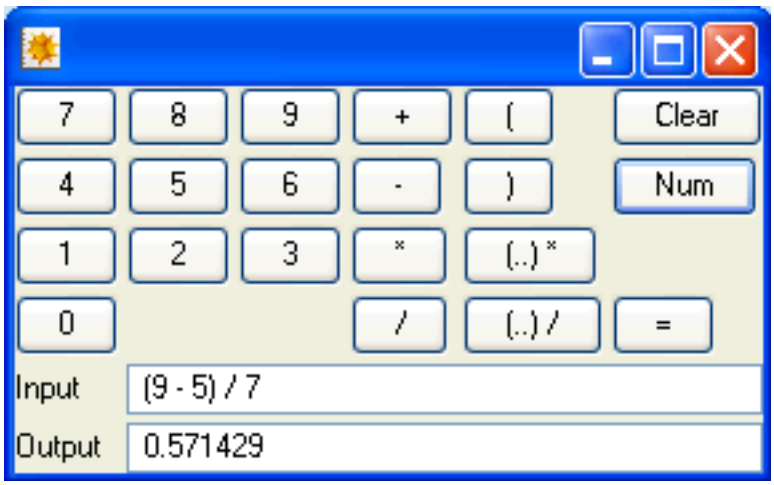
Taschenrechner Graphical User Interface mit GUIKit

Mit dem Package GUIKit ermöglicht es *Mathematica* in speditiver Art und Weise benutzerfreundliche graphische Benutzer Interfaces zu programmieren. Dies wird im Folgenden an Hand eines Taschenrechner demonstriert. Insgesamt sind mehr als 20 Buttons, 2 Label und 2 Textfelder mit den entsprechenden Funktionen zu programmieren.

Ein Screen Shot zeigt das fertige Produkt. Der Code ist - um der besseren Übersichtlichkeit willen - in eine Schritt für Schritt Folge aufgeteilt worden.

ScreenShot

```
Show[GUIScreenShot[gui]];
```



Code

1. Schritt: Laden des Package GUIKit (ausserdem werden einige Messages unterdrückt).

```
Needs["GUIKit`"];
```

2. Schritt: Definition der Funktionen verschiedener Widgets, d.h. der Buttons und des Input Textfeldes.

Es wird einerseits festgelegt, was auf dem Button dargestellt werden soll: z.B. Zahl, Operator,...
Andererseits wird festgelegt, welche Aktion beim Drücken des Buttons durchgeführt werden soll: der Inhalt des Script

```
Clear[wiN, wiZ, wiE, wiK, wiNum, wiClear, wiInfield];
wiN[x_String] :=
  Widget["Button", {"text" → x, BindEvent["action", Script[add[x]]]}];
wiZ[x_String] := Widget["Button",
   {"text" → x, BindEvent["action", Script[add[" " <> x <> " "]]]}];
wiE[x_String] := Widget["Button",
   {"text" → x, BindEvent["action", Script[calc[]]]}];
wiK[x_String] := Widget["Button", {"text" → "(..) " <> x,
    BindEvent["action", Script[addK[x]]]}];
wiNum        := Widget["Button", {"text" → "Num",
    BindEvent["action", Script[toNum[]]]}];
wiClear      := Widget["Button", {"text" → "Clear",
    BindEvent["action", Script[clearFields[]]]}];
wiInfield  := Widget["TextField", {BindEvent["action", Script[calc[]]]},
   Name → "inpField"];
```

3. Schritt: Layouten des Interface

Mit WidgetAlign[] wird Alignment erzielt.

```
wPart = {
   {wiN["7"], WidgetAlign[], wiN["8"], wiN["9"], WidgetAlign[],
    wiZ["+"], WidgetAlign[], wiZ["("], WidgetAlign[], wiClear},
   {wiN["4"], WidgetAlign[], wiN["5"], wiN["6"], WidgetAlign[],
    wiZ["-"], WidgetAlign[], wiZ[")"], WidgetAlign[], wiNum},
   {wiN["1"], WidgetAlign[], wiN["2"], wiN["3"], WidgetAlign[], wiZ["*"],
    WidgetAlign[], wiK["*"]}, {wiN["0"], WidgetAlign[], WidgetAlign[],
    wiZ["/"], WidgetAlign[], wiK["/"], WidgetAlign[], wiE["="]},
   {Widget["Label", {"text" → "Input"}], WidgetAlign[], wiInfield},
   {Widget["Label", {"text" → "Output"}], WidgetAlign[],
    Widget["TextField", Name → "outField"]}};
```

4. Schritt: Definition der Skript Funktionen, die von den Buttons und dem Textfeld aufgerufen werden.

add fügt die neue Zahl bzw. das neue Zeichen hinzu.
addK setzt Klammern über den bisherigen Ausdruck und fügt das neue Zeichen hinzu.
calc führt die Berechnung durch
clear löscht das Input und das Output Feld
toNum wandelt eine rationale Zahl in eine Dezimalzahl um

```
wTotal = Widget["Panel", {
    {wPart}
    , Script[
     add[x_String] := Module[{vOld}, vOld = PropertyValue[{
          "inpField", "text"}]; SetPropertyValue[{"inpField", "text"}, vOld <> x]];
     addK[x_String] := Module[{vOld}, vOld = PropertyValue[{
          "inpField", "text"}];
       SetPropertyValue[{"inpField", "text"}, "(" <> vOld <> ") " <> x <> " "]];
     calc[] := Module[{}, SetPropertyValue[{"outField", "text"}, ToString[
         InputForm[ ToExpression[ PropertyValue[{"inpField", "text"}] ] ]]]];
     clearFields[] := Module[{}, SetPropertyValue[{"inpField", "text"}, ""];
       SetPropertyValue[{"outField", "text"}, ""]];
     toNum[] := Module[{}, SetPropertyValue[{"outField", "text"},
        ToString[ N[ ToExpression[ PropertyValue[{"outField", "text"}] ] ]]]];
    ]
   }];
```

5. Schritt: Starten der Applikation

```
gui = GUIRun[wTotal];
```