IBAN-Projekt mit GUI

G	Graphical
U	User
I	Interface

I	International
B	Bank
A	Account
N	Number

IBAN-Projekt mit GUI

- Alternativer Einstieg in die GUI-Entwicklung mit einem einfachen Programm zum
  - Berechnen

  und
  - Prüfen

  einer IBAN

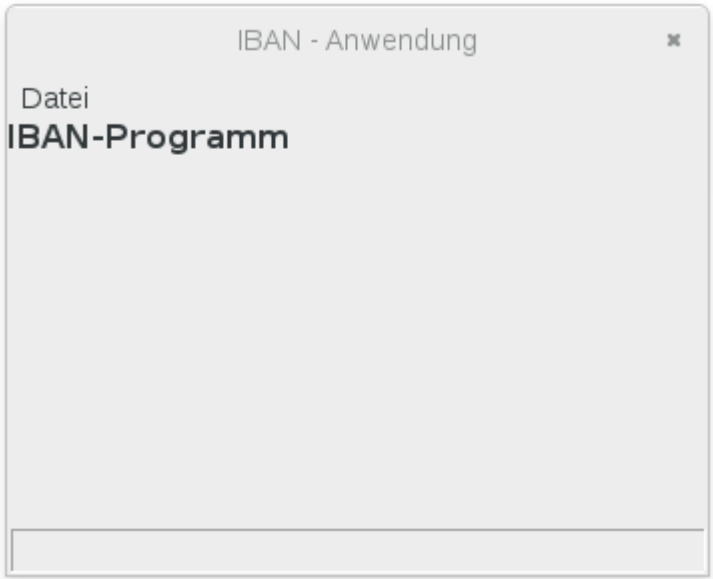
IBAN-Projekt mit GUI

- Nachteil:
  - MVC – Konzept [model-view-controller] ist nur eingeschränkt erkennbar
  - nicht primitive Berechnungen im Modell
- Vorteil:
  - Eine einfache Anwendung mit für Schülerinnen und Schüler erkennbarem Realbezug mit kurzfristiger Anwendbarkeit

IBAN-Projekt mit GUI

View

- Wir beginnen mit einer einfachen Anwendung, die ein Fenster [Frame] mit einer Inhaltsfläche [Panel] erzeugt und anzeigt.
- Quelle ist beispielsweise eines der Demo-Programme.



View

Menue

Binden der …

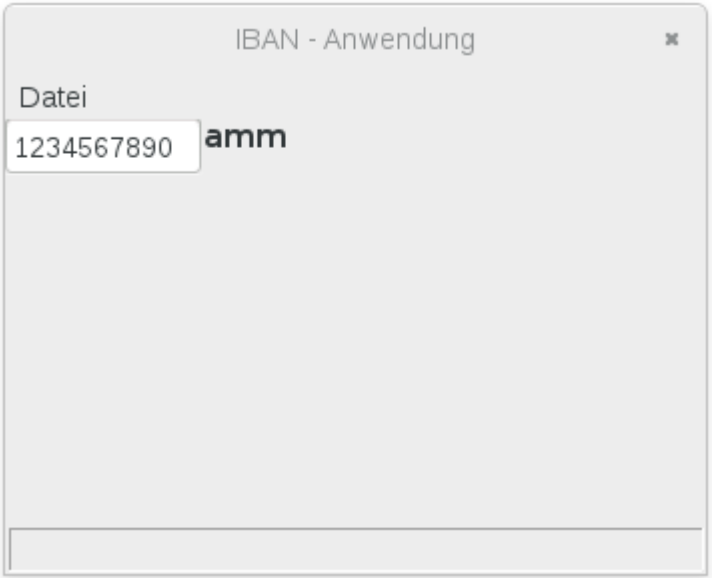
Ereignis-behandlung

Inhalts-Fläche

statischen Text

```
class IBAN_Frame(wx.Frame):
    """
    Frame fuer das IBAN-Programm
    """
    def __init__(self, parent, title):
        wx.Frame.__init__(self, parent, -1, title,
                          pos=(150, 150), size=(350, 250))

        # Create the menubar
        menuBar = wx.MenuBar()

        # and a menu
        menu = wx.Menu()

        # add an item to the menu, using \tKeyName automatically
        # creates an accelerator, the third param is some help text
        # that will show up in the statusbar
        menu.Append(wx.ID_EXIT, "B&eenden\tAlt-X", "Schliessen der Anwendung")

        # bind the menu event to an event handler
        self.Bind(wx.EVT_MENU, self.OnTimeToClose, id=wx.ID_EXIT)

        # and put the menu on the menubar
        menuBar.Append(menu, "&Datei")
        self.SetMenuBar(menuBar)

        self.CreateStatusBar()


        # Now create the Panel to put the other controls on.
        panel = wx.Panel(self)

        # ein Label (statische Beschriftung)
        text = wx.StaticText(panel, -1, "IBAN-Programm")
        text.SetFont(wx.Font(12, wx.SWISS, wx.NORMAL, wx.BOLD))
        text.SetSize(text.GetBestSize())

    def OnTimeToClose(self, event):
        """Event handler fuer Klick auf Beenden."""
        self.Close()
```

IBAN-Projekt mit GUI



View

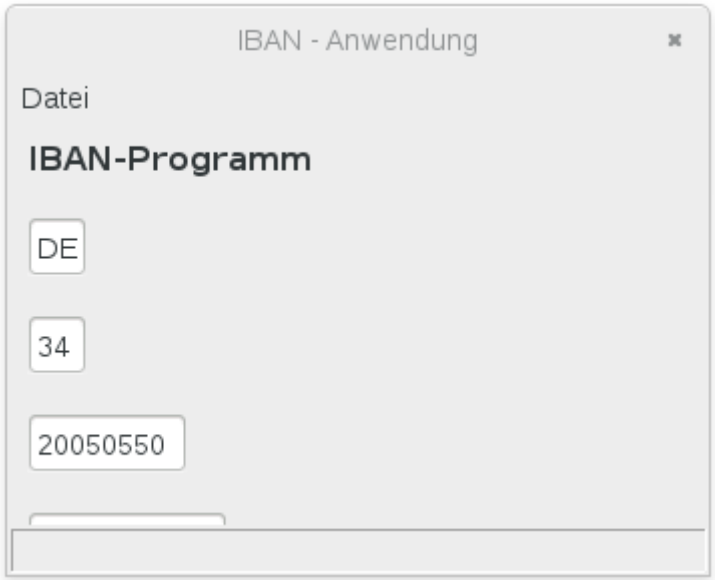
- Textfelder (TextCtrl) einfügen.
- Problem: Überlagerung aller Elemente

```
# ein Label (statische Beschriftung)
text = wx.StaticText(panel, -1, "IBAN-Programm")
text.SetFont(wx.Font(12, wx.SWISS, wx.NORMAL, wx.BOLD))
text.SetSize(text.GetBestSize())

# Text-Ctrl - Abschnitt
self.landTextfeld = wx.TextCtrl(panel, -1, "DE", size=(30, -1))
self.pruefzifferTextfeld = wx.TextCtrl(panel, -1, "34", size=(30, -1))
self.blzTextfeld = wx.TextCtrl(panel, -1, "20050550", size=(80, -1))
self.kontoTextfeld = wx.TextCtrl(panel, -1, "1234567890", size=(100, -1)
```

IBAN-Projekt mit GUI

View

- Layoutmanager einfügen (Sizer)

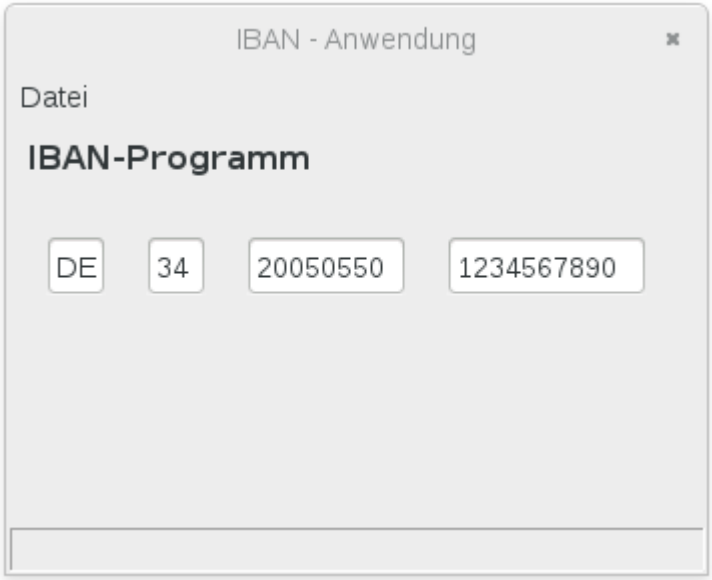



```
# Use a sizer to layout the controls, stacked vertically and with
# a 10 pixel border around each
sizer = wx.BoxSizer(wx.VERTICAL)
sizer.Add(text, 0, wx.ALL, 10)

sizer.Add(self.landTextfeld, 0, wx.ALL, 10)
sizer.Add(self.pruefzifferTextfeld, 0, wx.ALL, 10)
sizer.Add(self.blzTextfeld, 0, wx.ALL, 10)
sizer.Add(self.kontoTextfeld, 0, wx.ALL, 10)

panel.SetSizer(sizer)
panel.Layout()
```

IBAN-Projekt mit GUI

View

- zusätzlich einen "horizontal sizer"




```
sizer = wx.BoxSizer(wx.VERTICAL)
sizer.Add(text, 0, wx.ALL, 10)

horizontalSizer1 = wx.BoxSizer(wx.HORIZONTAL)
horizontalSizer1.Add(self.landTextfeld, 0, wx.ALL, 10)
horizontalSizer1.Add(self.pruefzifferTextfeld, 0, wx.ALL, 10)
horizontalSizer1.Add(self.blzTextfeld, 0, wx.ALL, 10)
horizontalSizer1.Add(self.kontoTextfeld, 0, wx.ALL, 10)
sizer.Add(horizontalSizer1, 0, wx.ALL, 10)

panel.SetSizer(sizer)
panel.Layout()
```

IBAN-Projekt mit GUI

View

- Buttons einfügen,
- auch mit einem "horizontal sizer"


```
# Button - Abschnitt
berechneBtn = wx.Button(panel, -1, "berechne")
berechneBtn.SetToolTipString("berechnet die Prüfziffer zur IBAN")
pruefeBtn = wx.Button(panel, -1, "pruefe")
pruefeBtn.SetToolTipString("prueft die eingegebene IBAN")
beendenButton = wx.Button(panel, -1, "Beenden")
beendenButton.SetToolTipString("Button beendet die Anwendung")
```

IBAN-Projekt mit GUI

View

- Buttons
  an Ereignisse binden


```
# bind the button events to handlers
self.Bind(wx.EVT_BUTTON, self.OnBerechneButton, berechneBtn)
self.Bind(wx.EVT_BUTTON, self.OnPruefeButton, pruefeBtn)
self.Bind(wx.EVT_BUTTON, self.OnTimeToClose, beendenButton)
```

IBAN-Projekt mit GUI

View

- Ereignisbehandlung der Buttons vorbereiten


```
def OnTimeToClose(self, event):
    """Event handler fuer Klick auf Beenden."""
    self.Close()

def OnPruefeButton(self, event):
    """Event handler für pruefe."""
    pass

def OnBerechneButton(self, event):
    """Event handler für berechne."""
    pass
```

IBAN-Projekt mit GUI

View

- Ereignisbehandlung realisieren


```
def OnPruefeButton(self, event):
    """Event handler für pruefe."""
    iban=self.landTextfeld.GetValue()
    if len(iban)!=2:
        self.landTextfeld.SetBackgroundColour(wx.RED)
        return
    iban+=self.pruefzifferTextfeld.GetValue()
    if len(iban)!=4:
        self.pruefzifferTextfeld.SetBackgroundColour(wx.RED)
        return
    iban+=self.blzTextfeld.GetValue()
    if len(iban)!=12:
        self.blzTextfeld.SetBackgroundColour(wx.RED)
        return
    iban+=self.kontoTextfeld.GetValue()
    if len(iban)!=22:
        self.kontoTextfeld.SetBackgroundColour(wx.RED)
        return
    if pruefe(iban):
        self.pruefzifferTextfeld.SetBackgroundColour(wx.GREEN)
    else:
        self.pruefzifferTextfeld.SetBackgroundColour(wx.RED)
```

IBAN-Projekt mit GUI

View

- Ereignisbehandlung realisieren: Berechne-Button


```
def OnBerechneButton(self, event):
    """Event handler für berechne."""
    land=self.landTextfeld.GetValue()
    if len(land)!=2:
        self.landTextfeld.SetBackgroundColour(wx.RED)
        return
    blz=self.blzTextfeld.GetValue()
    if len(blz)!=8:
        self.blzTextfeld.SetBackgroundColour(wx.RED)
        return
    konto=self.kontoTextfeld.GetValue()
    if len(konto)!=10:
        self.kontoTextfeld.SetBackgroundColour(wx.RED)
        return
    self.pruefzifferTextfeld.SetValue(gibPruefzahl(land, blz+konto))
```

IBAN-Projekt mit GUI

View

- Einfärbung zurücksetzen


```
    # Aktualisieren der Farbe nach erneutem Klick in ein Textfeld
    self.landTextfeld.Bind(wx.EVT_LEFT_DOWN, self.OnTFLeftDown)
    self.pruefzifferTextfeld.Bind(wx.EVT_LEFT_DOWN, self.OnTFLeftDown)
    self.blzTextfeld.Bind(wx.EVT_LEFT_DOWN, self.OnTFLeftDown)
    self.kontoTextfeld.Bind(wx.EVT_LEFT_DOWN, self.OnTFLeftDown)

def OnTFLeftDown(self, event):
    """Event handler fuer Klick in ein Textfeld."""
    self.landTextfeld.SetBackgroundColour(wx.WHITE)
    self.pruefzifferTextfeld.SetBackgroundColour(wx.WHITE)
    self.blzTextfeld.SetBackgroundColour(wx.WHITE)
    self.kontoTextfeld.SetBackgroundColour(wx.WHITE)
    event.Skip()
```

IBAN-Projekt mit GUI

View

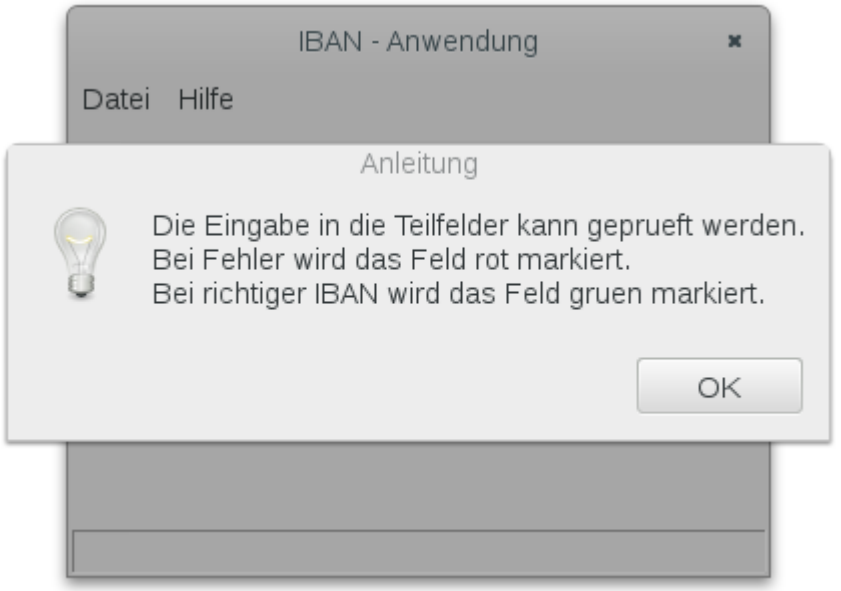
- Hilfetext realisieren


```
# and a menu
menu = wx.Menu()
menu2 =wx.Menu()

# add an item to the menu, using \tKeyName automatically
# creates an accelerator, the third param is some help text
# that will show up in the statusbar
menu.Append(wx.ID_EXIT, "B&eenden\tAlt-X", "Schliessen der Anwendung")
menu2.Append(1000, "&Hilfe", "Hilfetext anzeigen")

self.hilfeText ="Die Eingabe in die Teilfelder kann geprueft werden.\n"
self.hilfeText+="Bei Fehler wird das Feld rot markiert.\n"
self.hilfeText+="Bei richtiger IBAN wird das Feld gruen markiert."

# bind the menu event to an event handler
self.Bind(wx.EVT_MENU, self.OnTimeToClose, id=wx.ID_EXIT)
self.Bind(wx.EVT_MENU, self.OnHilfe, id=1000)
```

IBAN-Projekt mit GUI

View

- Ereignisbehandlung für die Hilfe




```
def OnHilfe(self, event):
    self.MeldungAusgeben(self.hilfeText,'Anleitung')

### ----- Einfache Meldung ausgeben -----
def MeldungAusgeben(self, text, wasIst='Meldung'):
    '''Zeigt ein Dialogfenster mit einer Meldung.'''
    dialog = wx.MessageDialog(self, text, wasIst, wx.OK)
    dialog.ShowModal()
```

IBAN-Projekt mit GUI

Modell

- eine Funktion zum Berechnen der Prüfziffern bei bekannter
    - Landkennung
    - Bankleitzahl
    - Kontonummer
- eine Funktion zum Prüfen einer vorgegebenen vollständigen IBAN
- In der hier vorgestellten Lösung ist dieser Modellteil nicht objektorientiert modelliert.

IBAN-Projekt mit GUI

Modell, Funktion zum Berechnen der Prüfziffern

- Landkennung in Ziffern verwandeln

A->10, B->11, ...
also Alphabet-Position+10 = ASCII-55

  - DE → 1314

- zusammen mit zwei angehängten Nullen für die Prüfziffer

  - also → 131400

- ans Ende von ...

IBAN-Projekt mit GUI

Modell, Funktion zum Berechnen der Prüfziffern

… verbundene

- Bankleitzahl
  - Beispiel 20050550
- Kontonummer
  - Beispiel 1234567890

also

- im Beispiel 200505501234567890131400

IBAN-Projekt mit GUI

Modell, Funktion zum Berechnen der Prüfziffern

Von dieser Zahl wird modulo 97 berechnet [Rest beim Teilen durch 97] und von 98 abgezogen, also

- im Beispiel
  2005055012345678901314 00 % 97 → 64
  98 – 64 → 34

Die IBAN ist dann also
DE 34 2005 0550 1234 5678 90

IBAN-Projekt mit GUI

Modell, Funktion zum Prüfen der IBAN

entsprechender Aufbau, nun mit der bekannten Prüfziffer

- im Beispiel
  200505501234567890131434 % 97 → 1

- Die IBAN ist prinzipiell korrekt, wenn diese Berechnung 1 ergibt.
- Es kann aber nicht festgestellt werden, ob die IBAN wirklich richtig ist, typische Schreibfehler werden jedoch erkannt.

IBAN-Projekt mit GUI

Modell

- Die beiden Funktionen lassen sich natürlich leicht in eine Klasse kapseln, deren Instanzen allein diese beiden Funktionen bereit stellen.
- Vom eigentlichen MVC – Konzept ist zumindest die Trennung zwischen Modell und View realisiert.