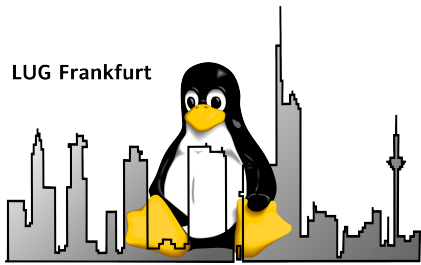


# Programmieren mit Gambas

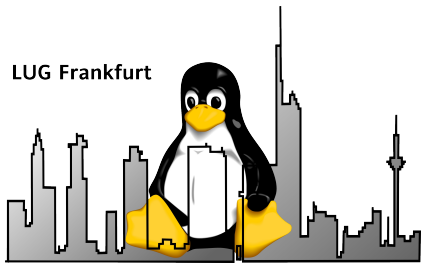
von Rolf Schmidt für  
LUG Frankfurt





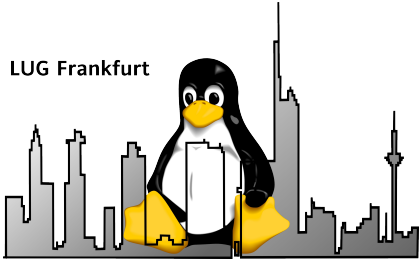
# Ziel des Vortrages

- Kein Programmierkurs
- Kurzvorstellung der Sprache
- ... und Arbeitsumgebung
- Interesse an (anspruchsvoller) Programmierung wecken



# Warum ich Gambas verwende

- Programme sind einfach zu erstellen
- Einfacher Zugriff auf Datenbanken
- Auch komplexe Programmlogik realisierbar
- Schnelleren Erfolg als mit C und QT
- Schneller stabil als C-Programme

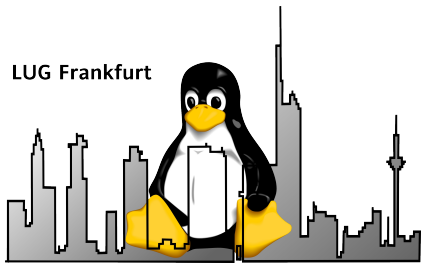


# Was ist Gambas (1)

- Gambas Almost Means BASic
- Entwickler
  - Benoît Minisini
  - [Hall of Fame](#) - die Mitentwickler
- **Gründe für die Entwicklung**

The phenomenal quantity of bugs and inconsistencies that makes Visual Basic so delightful persuaded me to start this project ;-)

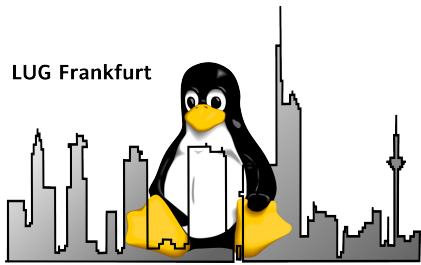
- Mehr als RAD unter Linux



# Was ist Gambas (2)



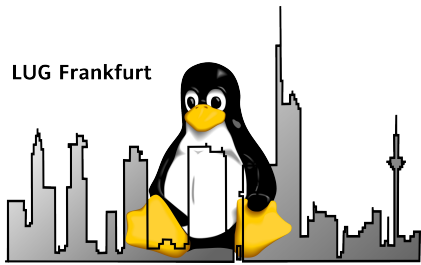
- Immer noch heftig entwickeltes Programmpaket
  - Feature-Requests gelegentlich in wenigen Stunden realisiert
- Anwender
  - Extremadura u.a.
  - z.T. in der Lehre eingesetzt
  - Viele (Hobby-) Programmierer weltweit
- Selbstverständlich OpenSource



# Was ist Gambas (3)

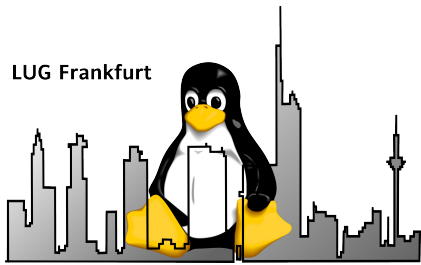


- Wie bekommt man Gambas?
  - Selbst compilieren aus svn
  - Bei Debian in Lenny und Sid vorhanden
  - Div. Webseiten für verschiedene Distributionen
  - <http://gambas.sourceforge.net/>



# Features (1)

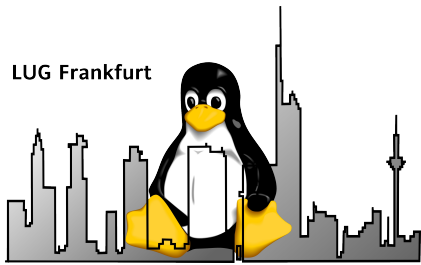
- VB-ähnliche Handhabung (und Syntax)
  - KEIN VB-Clone!
  - Keine Reimplementation der Bugs
  - Konsequenteres OO-Model
- Programmerstellung für die Oberflächen
  - KDE
  - GNOME
- IDE für die Entwicklung



# Features (2)

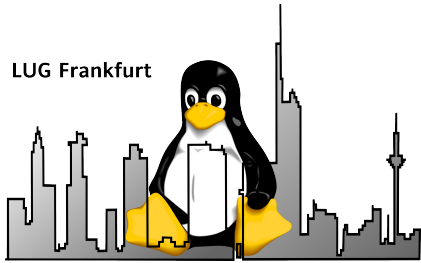
- Einfache Anbindung an
  - MySQL
  - PostgreSQL
  - sqlite (Vers. 2 und 3)
  - firebird
- Internationalisierung
- 32- und 64-Bit Versionen
- Entwicklung eigener Komponenten möglich





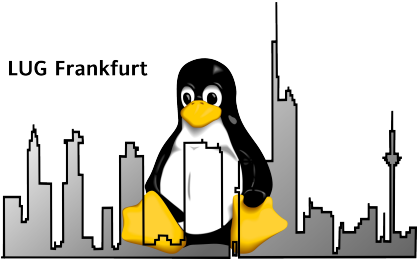
# Features (3)

- Einbetten anderer X-Anwendungen
- Aufrufen anderer Programme mit EXEC
- Einbinden externer Libraries
- Auch Programme ohne GUI möglich
- Auch als Skriptsprache
  - Für Webseiten geeignet  
(siehe Gambas-Doku-Wiki)
- Dokumentation noch nicht vollständig



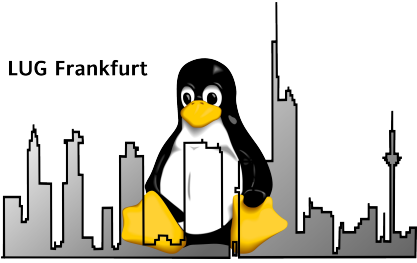
# Features (4)

- Keine Windows-Version - evtl. unter Cygwin
- Programmierkenntnisse notwendig
  - Tutorials vorhanden
  - Mäßiges Hilfesystem



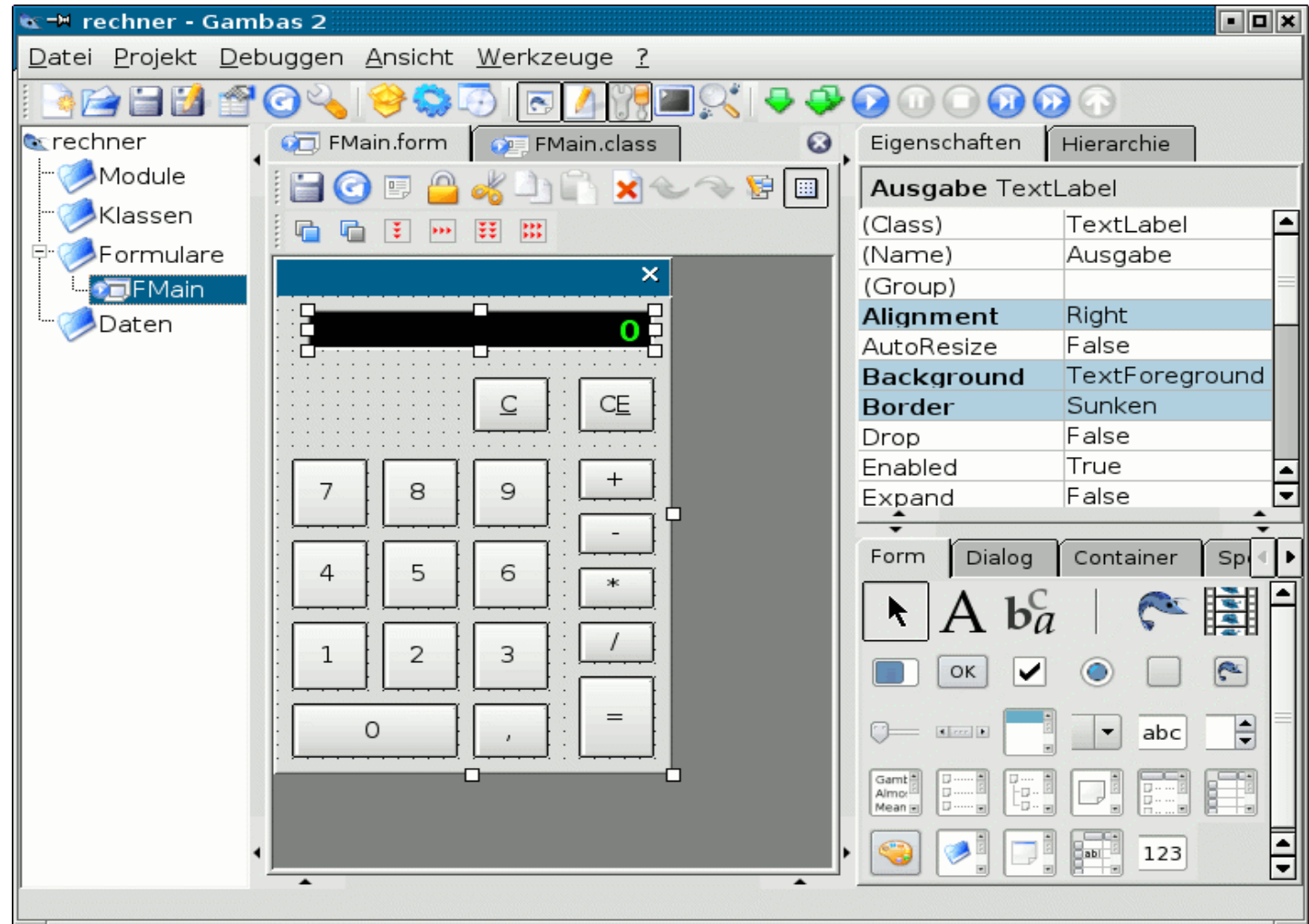
# Die IDE

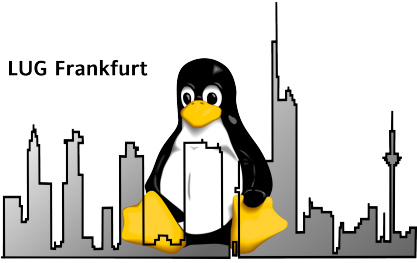
- Bestandteile
  - Projektverwaltung mit Projekteditor
  - Form-Editor
  - Eigenschaften-Editor
  - Menüeditor
  - Komfortablen Programmmeditor mit Eingabeergänzung bzw. -unterstützung
  - Datenbank-Manager
  - Terminalfester (für Programmausgabe)
  - Debugger



# Die IDE – Form-Design

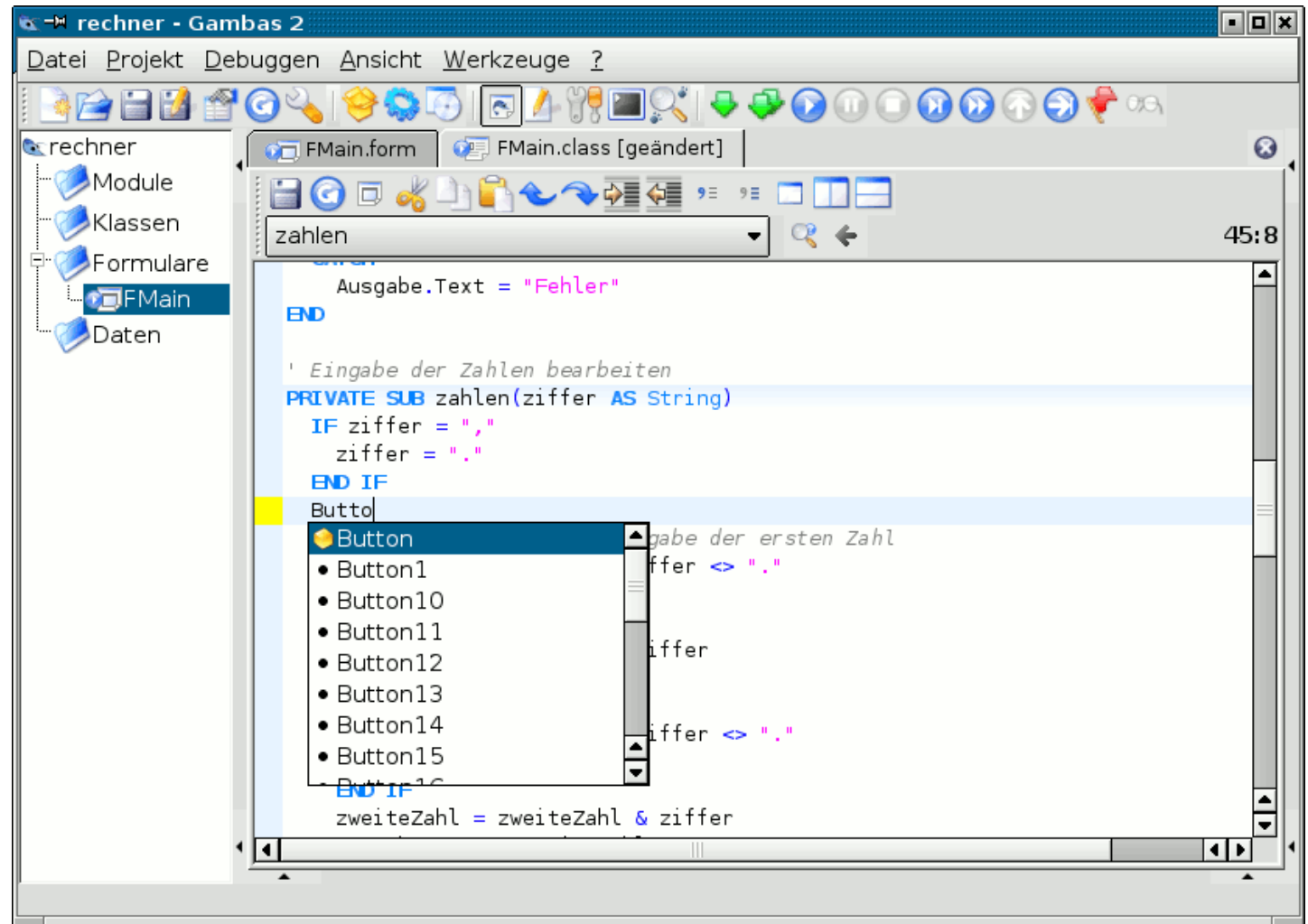
- Menüleiste
- Projektfenster
- Form-Editor
- Eigenschaften-Editor
- Form Elemente

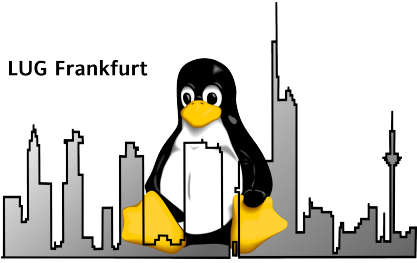




# Die IDE – Programmmeditor (1)

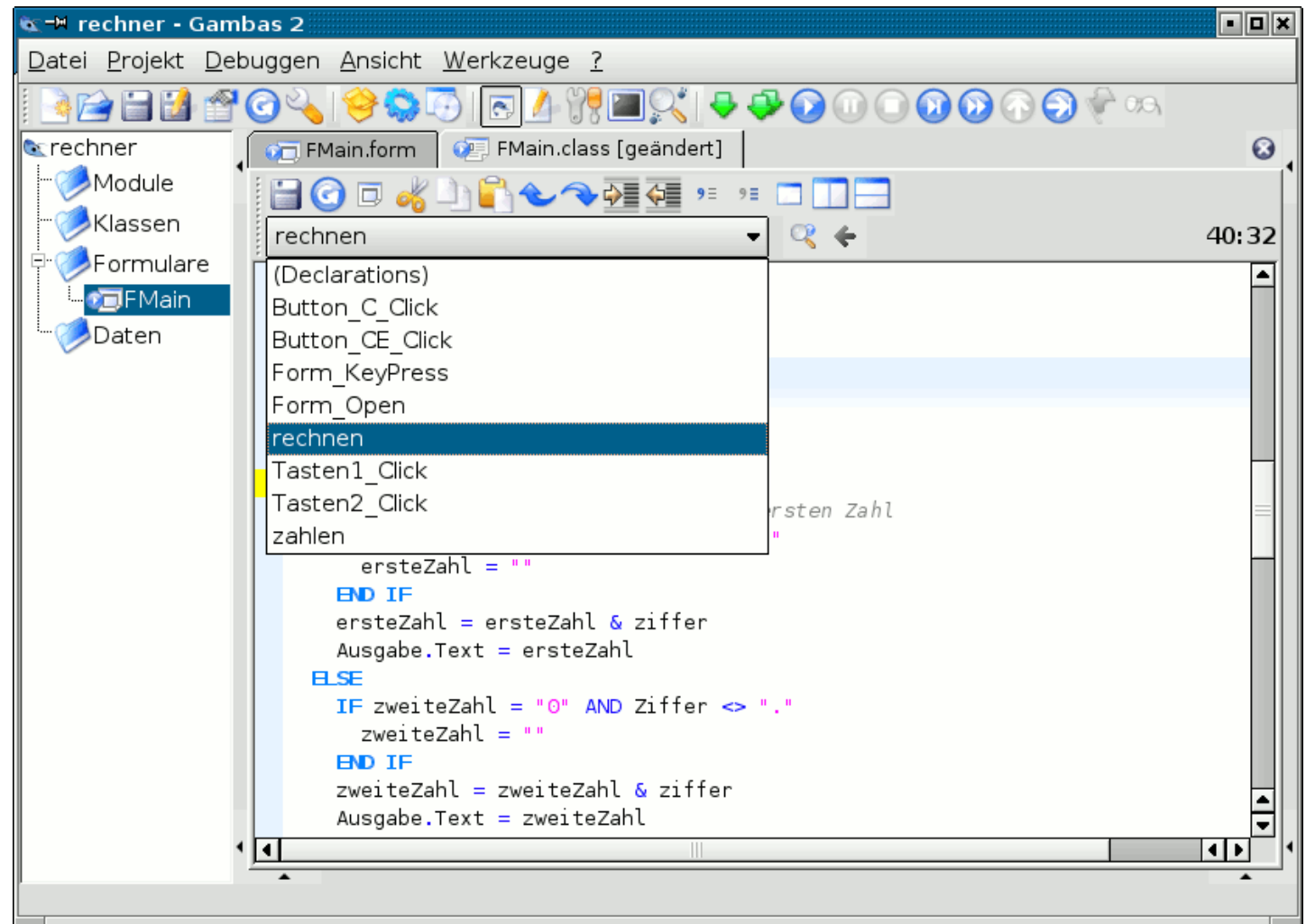
- Element-erkennung
  - Beim Schreiben
- Objekteigenschaften Erkennen
- Highlighting von Sprach-elementen

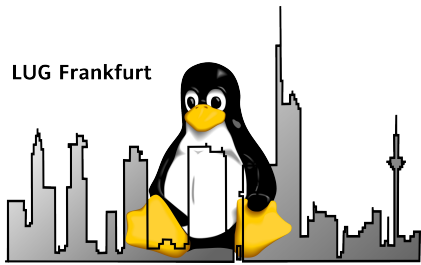




# Die IDE – Programmmeditor (2)

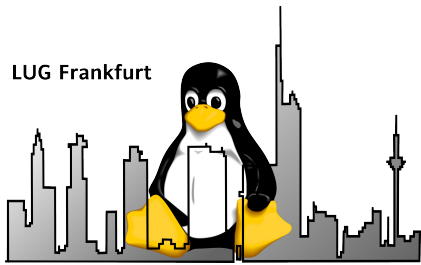
- Direktes Anspringen von Funktionen
- Suchen/Ersetzen
  - In Datei
  - Im Projekt





# Besonderheiten (1)

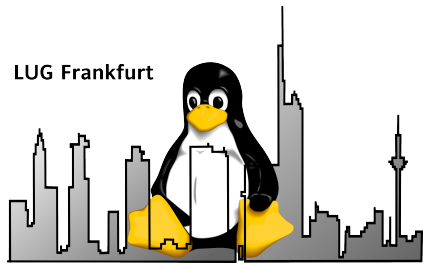
- Alle Variablen müssen deklariert werden
- Nur deklarierte Variablen können verwendet werden
- Variable sind immer typisiert!
- Fehlerverarbeitung mit
  - Try
  - Error
  - Catch
  - Final



# Besonderheiten(2)

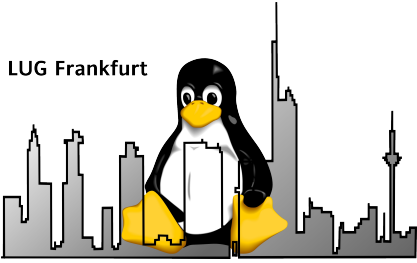
- Konsequente objektorientiert – alle Elemente sind Objekte
- Ein Objekt (z.B. eine Form) kann haben
  - Konstruktor
  - Destruktor
  - Privaten Variablen und Methoden
  - Public Variablen und Methode
  - Ist ableitbar oder eine Ableitung





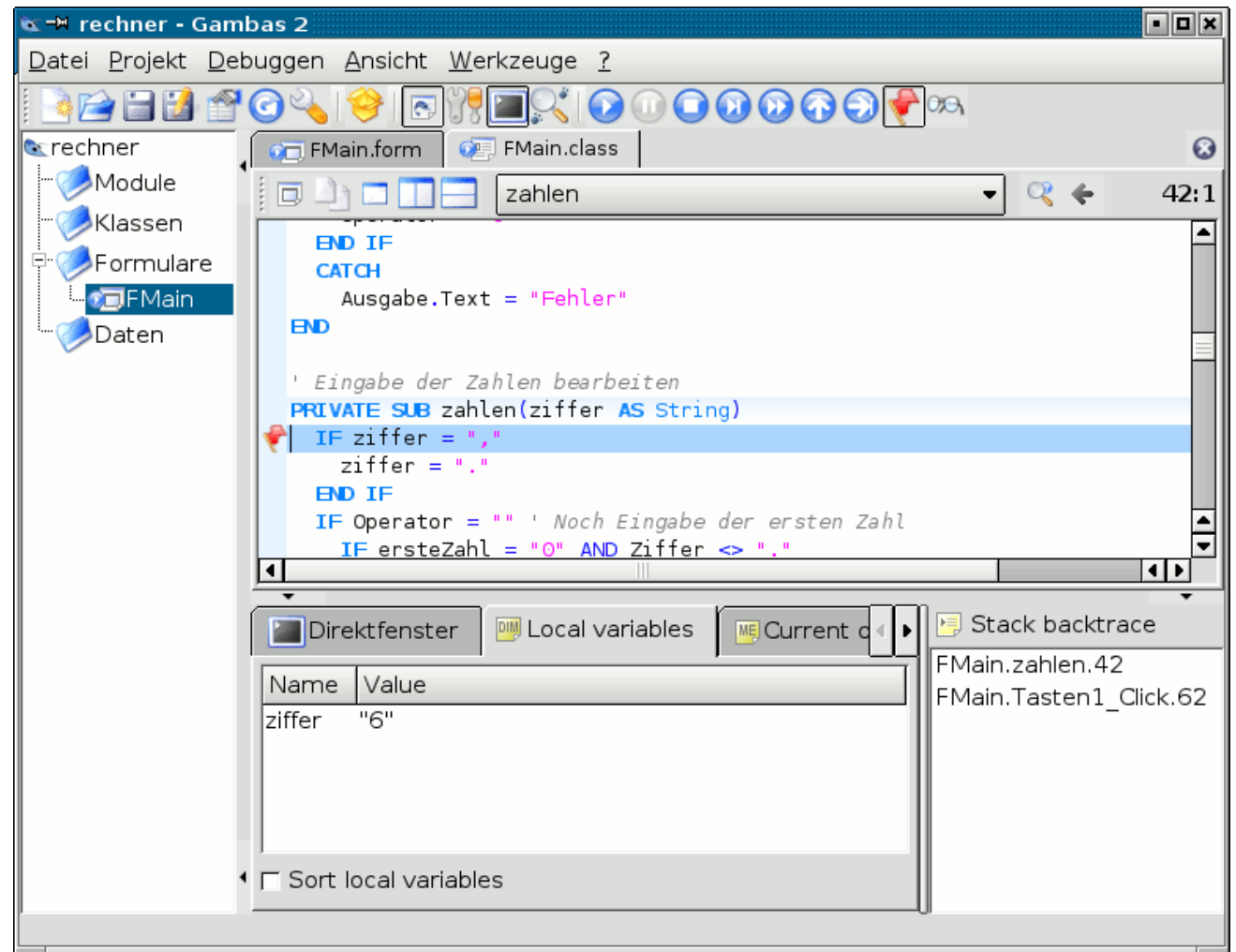
# Besonderheiten(3)

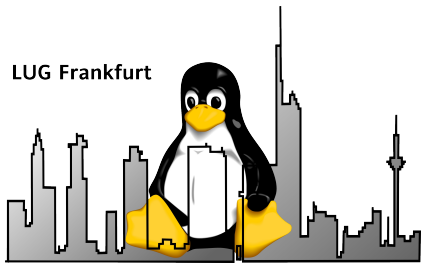
- Statische Klassen (genannt: Module)
  - Für global verfügbare Funktionen
  - Für globalen Daten



# Der Debugger

- Menüleiste
- Projektfenster
- Programm-Editor
- Überwachungsfenster
- Fenster mit Aufrufhierarchie (Stack)

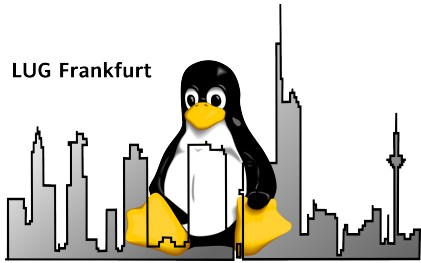




# Programmbeispiele (1)

## Variablendeklaration

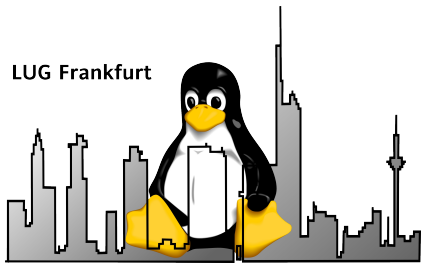
- Deklaration immer mit Datentyp
  - Objekt-Variablen
    - `Public test as string`
    - `Private test1 as string`
  - Methodenvariablen
    - `Dim zahl as integer`



# Programmbeispiele (2)

## Funktionen

- Vordefinierte Methoden
  - Konstruktor
    - `_new()`
  - Destruktor
    - `_free()`
  - Form-Handling
    - `form_open()`



# Programmbeispiele (3)

## Funktionen

- Aus anderen Objekten/Modulen erreichbar

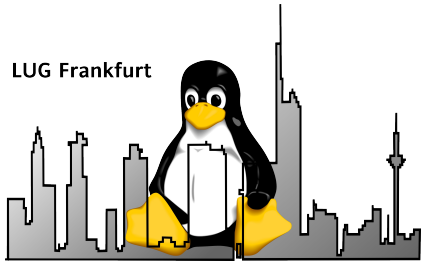
```
Public sub funktion  
end
```

- Private Funktionen eines Objekts

```
Private sub funktion  
end
```

- Aufruf von Funktionen anderer Klassen

```
- klassenname.funktion.eigenschaft
```



# Programmbeispiele (4)

## Fehlerbehandlung

- Try

- Fängt selbständig Fehler ab ohne Programmabbruch.

- Auf Fehler muss „manuell“ geprüft werden

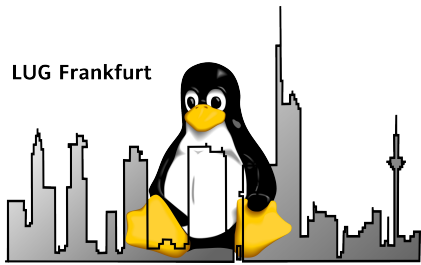
```
TRY MyFile = OPEN "/etc/password" FOR WRITE
```

- Error

```
IF ERROR THEN PRINT "I cannot do what I want!"
```

- Catch

- Fängt alle Fehler ab (wie in C++)

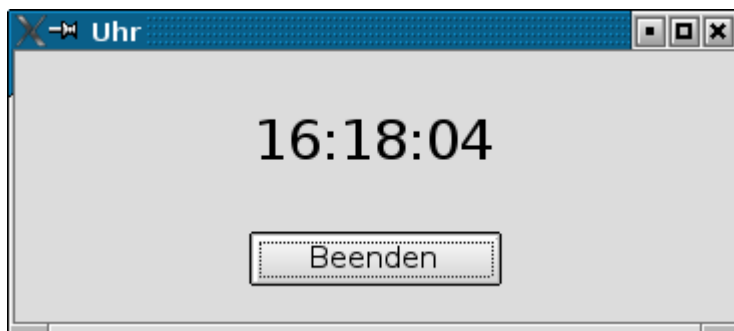


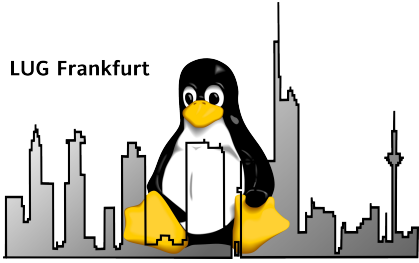
# Beispiele

- Beispiel – entstanden während des Vortrags



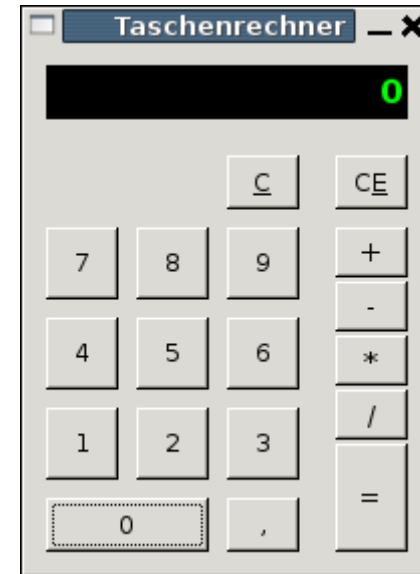
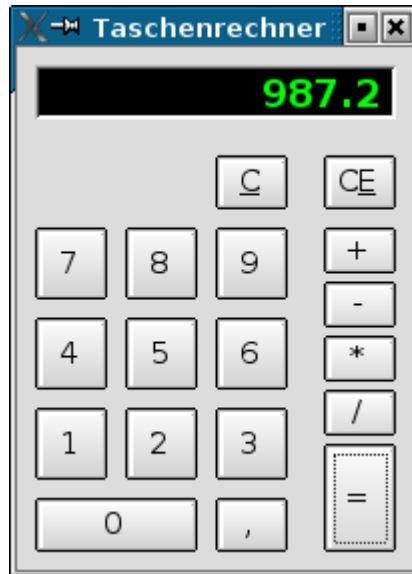
- Einfache Uhr



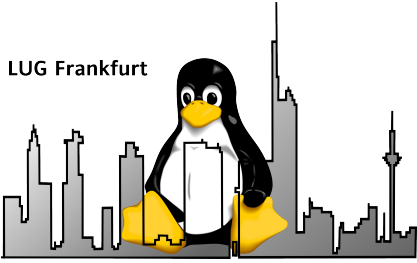


# Beispiele

- Taschenrechner unter KDE
- Taschenrechner unter GNOME



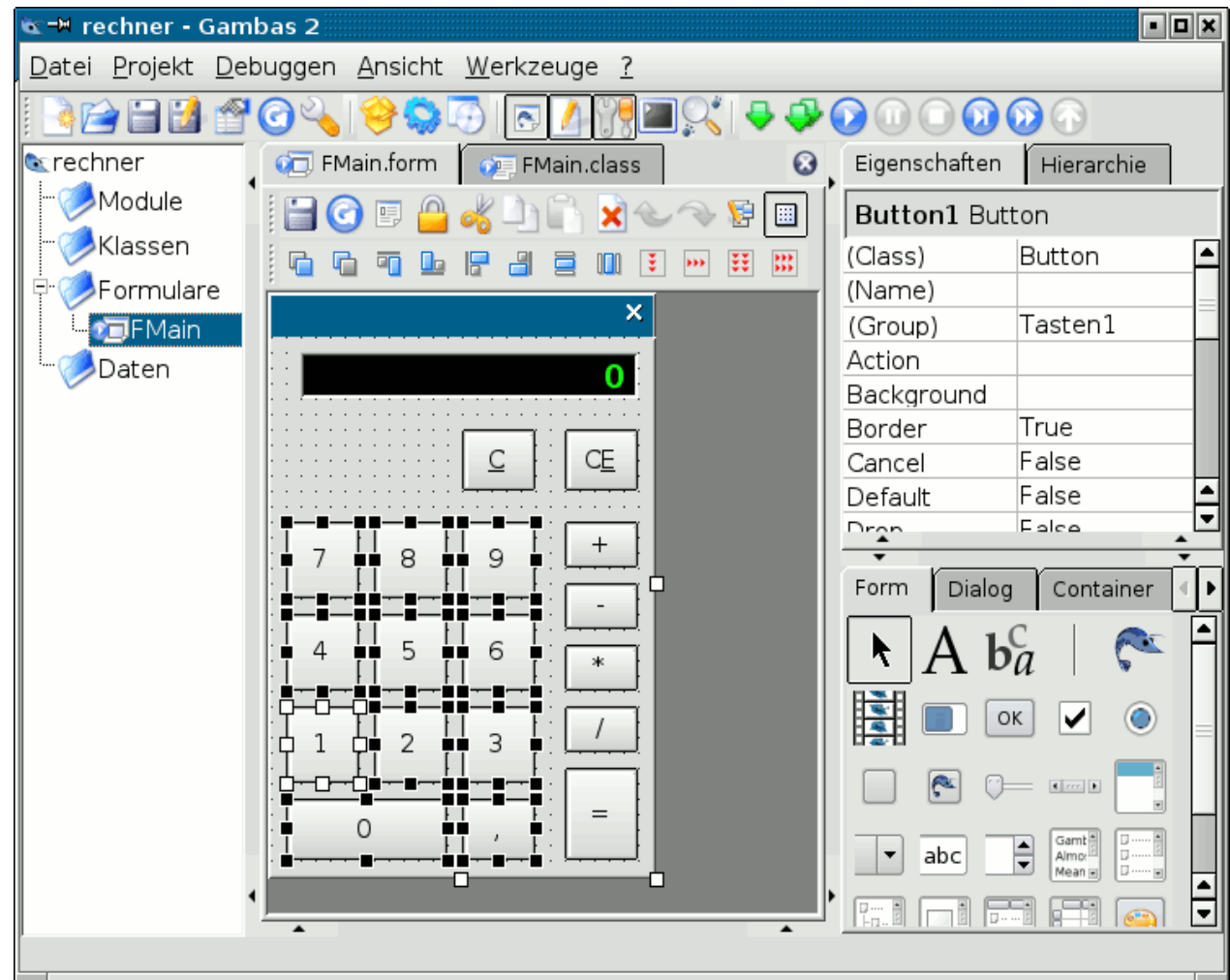


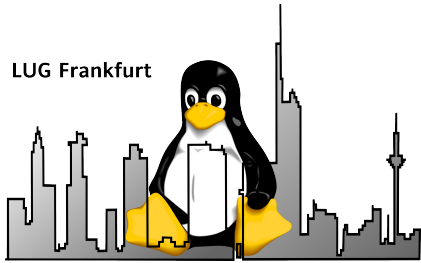


# Gruppieren

```

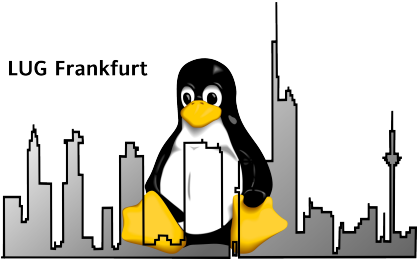
PRIVATE SUB zahlen(ziffer AS String)
  IF ziffer = ","
    ziffer = "."
  END IF
  IF Operator = ""
    IF ersteZahl = "0" AND Ziffer <> "."
      ersteZahl = ""
    END IF
    ersteZahl = ersteZahl & ziffer
    Ausgabe.Text = ersteZahl
  ELSE
    IF zweiteZahl = "0" AND Ziffer <> "."
      zweiteZahl = ""
    END IF
    zweiteZahl = zweiteZahl & ziffer
    Ausgabe.Text = zweiteZahl
  END IF
END
  
```





# Links

- [Gambas - Homepage](#)
- [Gambas FAQs](#)
- [Gambas - Einführung](#)
- [Unterschiede zu VisualBasic](#)
- [Gambas Tutorial](#)



# Alternative Methode

```
Invite ("Rolf Schmidt")  
Order (Paella, Rotwein)  
While not Paella = empty  
    Esst  
    Genießt  
End While
```





Danke  
und  
Tschüß  
fürs  
Mitnehmen

LUG Frankfurt

