



Playing Card Games with Object-oriented APL

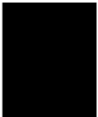
Präsentator: Alexander Krämer


24.04.2023, Bingen






Blackjack

BlackJack

**Karten
Gegner:** 



**Ihre
Karten:**   

The image shows a screenshot of a web-based Blackjack game interface. At the top left, there is a logo for 'DPC' (Deutscher Poker Club) consisting of a white 3D cube with a grid pattern and the letters 'DPC' in red. The top right corner displays the date '28.04.2023' and 'SEITE 2'. The main title 'Blackjack' is centered at the top. Below the title, the game area is titled 'BlackJack' in the window header. On the left, the text 'Karten Gegner:' is followed by a solid black rectangle, indicating that the dealer's cards are hidden. In the center of the game area, a single card is visible: the 6 of Clubs. At the bottom left, the text 'Ihre Karten:' is followed by three cards: the 4 of Hearts, the 6 of Clubs, and the King of Diamonds. At the bottom right, there are three buttons: 'Draw' (highlighted with a blue border), 'Stand', and 'Neues Spiel'.

Rommee

Rommee

Karten Gegner: [Black Card] [Black Card] [Black Card] [Black Card]

Karten Tisch:

[♣ 10] [♦ 10] [♥ 10] [♠ 7] [♠ 8] [♠ 9] [♠ 10] [♠ 6]

Tisch bereit [X]

i Sie können die ausgewählten Karten per Doppelklick an einen der markierten Stapel anlegen

OK

[♥ B] [♣ J] [♣ J] [♥ 8] [♥ 7] [♥ 6]

Ihre Karten: [♥ 5] [♣ 2] [♠ B] [♠ 10] [♥ 4] [♣ 9] [♣ 8] [♥ 3] [♥ 2]

Neuer Stapel Vorh. Stapel

Beende Zug Neues Spiel



Overview

1. Was ist objektorientierte Programmierung?
2. Dialog-Klassen als „Namespaces+Extras“
3. Kapselung
 - Private, Public, ReadOnly
 - Properties – Zugriff auf Variablen steuern
 - Property SubForms
4. Vererbung
 - Spezialisierung von Basisfunktionen
 - Von Base_GUICardGame zu BJ_GUI und Rommee_GUI
 - „Protected“ Member? Mehrfache Vererbung?



Was ist objektorientierte Programmierung?



Programmstruktur

Objektorientiert

Prozedural

Programm

Programm

Klassen

Hauptfunktion(en)

Hauptfunktion(en)

Globale Variablen

Felder

Properties

Methoden

Instanz 1

Instanz 2

Unterfunktionen

Lokale Variablen
(Hauptfunktion)

Felder
(Werte)

Felder
(Werte)

Lokale Variablen
(Unterfunktionen)

Programmstruktur

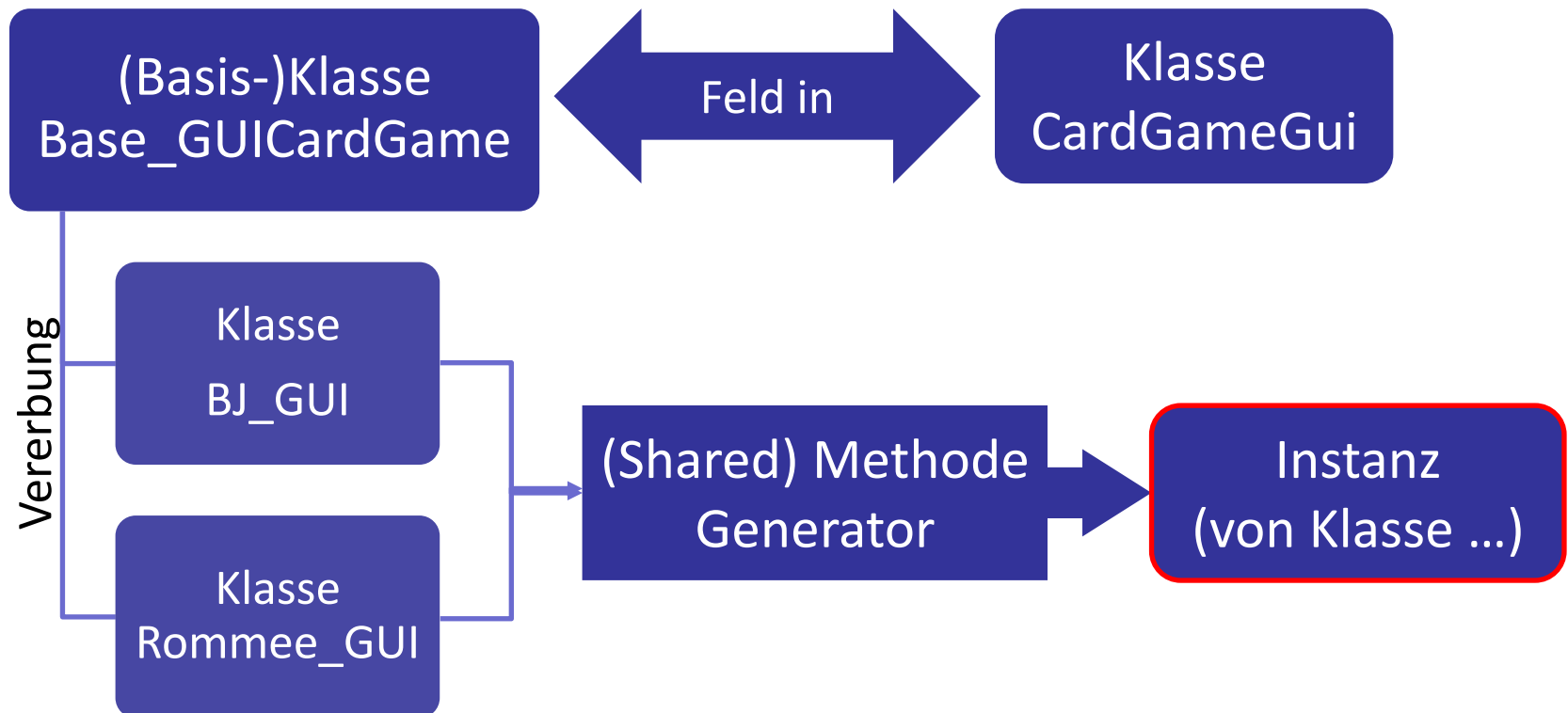
Objektorientiert

- Basiert auf Klassen
- Datenspeicherung in Instanzen der Klassen (Felder)
- Datenverarbeitung durch Methoden der Klassen; Programmfluss gesteuert durch Auswahl von Klassen und Methoden

Prozedural

- Basiert auf Funktionen
- Datenspeicherung in Variablen
- Datenverarbeitung durch Funktionen; Programmfluss gesteuert durch Kontrollstrukturen

Anwendung CardGame



Methode Generator

BJ_GUI

```

▽ Inst←{lArg} Generator Packs;NS;Path
:Access shared public
:if 0=□nc 'lArg'
  NS←#
:elseif 1=p,lArg
  NS←lArg
:else
  (NS Path)←lArg
:endif
:if 0=□nc 'Path'
  Inst←NS.□new □this Packs
:else
  Inst←NS.□new □this (Packs,öcPath)
:endif
▽
  
```

Rommee_GUI

```

▽ Inst←{lArg}Generator rArg;NS;Path;Index
:Access shared public
:If 0=□NC'lArg'
  NS←#
:Elseif 1=p,lArg
  NS←lArg
:Else
  (NS Path)←lArg
:EndIf
:If 0=□NC'Path'
  Inst←NS.□NEW □THIS rArg
:Else
  Inst←NS.□NEW □THIS(rArg,cPath)
:EndIf
▽
  
```



Dyalog-Klassen als „Namespaces+Extras“



Parallelen zu Namespaces

- „Bauplan“ für „Pseudonamespaces“
- Eigene Systemvariablen u. zusätzliche Objekte
- `:Include` \cong `□NS`

```
:Class CardGameGui|  
:Include #.CardGameGui_Tools
```



Kapselung

Public, Private, ReadOnly

- Public: wie in einem Namespace
- Private: „reserviert“ für Member der Klasse
- ReadOnly: Zuweisungen gesperrt
geeignet für public (shared) Felder o. ä.

```
:Class CardGameGui|
:Include #.CardGameGui_Tools

:field public ReadOnly FullCardSet←'J',, 'HCKP'°.., (□D~'01'), (c'10'), 'BDKA'
:field private Main
```

Properties – Zugriff auf Variablen steuern

- Generell: Felder stets private!
- Lese- und v. a. Schreibzugriff durch Properties
- Properties: Set bestimmter Funktionen – Getter und Setter
- Trigger als mögliche Alternative

```
▽ ResetEmptyDeck
:Implements Trigger CardDeck
:if 0=pDeck
    Deck←NewDeck
:endif
▽
```



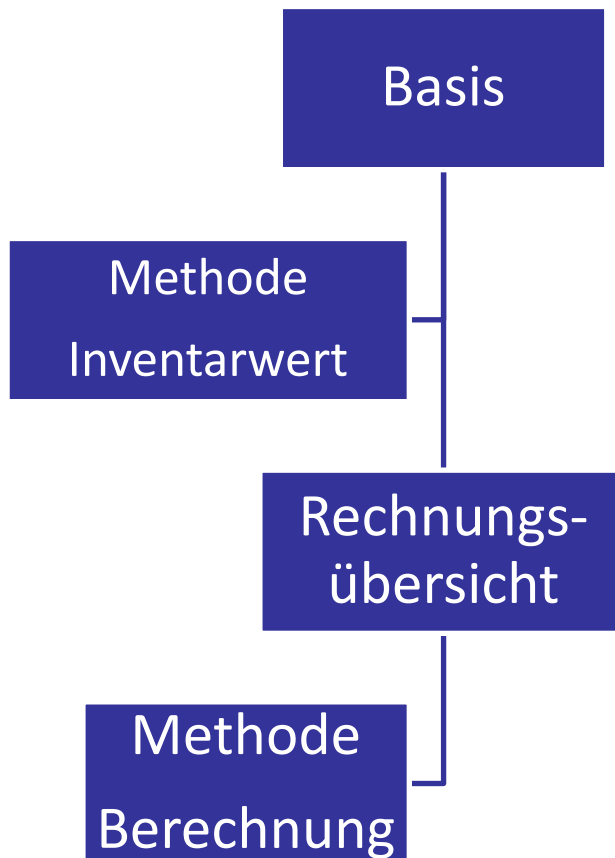
Property SubForms

```
:property Keyed SubForms
:Access public
  ▽ r←Get args;Obj;Prop
  args←targs.Indexers
  :if 0=ppargs
    args←targs
    (Obj Prop)←c"args
  :else
    (Obj Prop)←c[1]>args
  :endif
  r←Obj [WG" Prop
  ▽
  ▽ Set args;Obj;Prop;Val
  args.Indexers←targs.Indexers
  :if 0=ppargs.Indexers
    args.Indexers←targs.Indexers
    (Obj Prop)←c"args.Indexers
  :else
    (Obj Prop)←c[1]>args.Indexers
  :endif
  :if v/~Propc'Event' 'Active' 'Visible' 'Caption' 'BCol' |
    0 OpWmsg 'Some of the specified properties may not be changed!' 'Error CardGameGui - Property SubForms' 16
    →0
  :endif
  Val←args.NewValue
  Obj [WS" Prop(,)ø(c"Val
  ▽
:endproperty
```



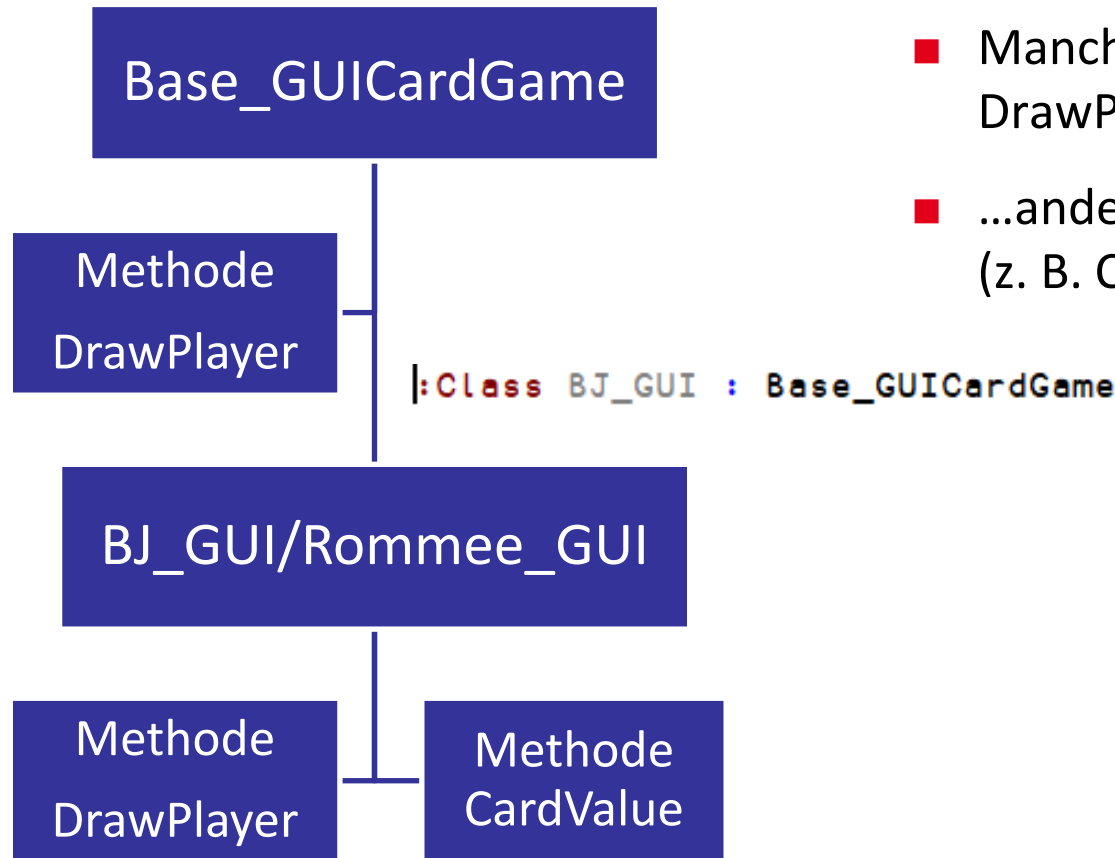
Vererbung

Spezialisierung von Basisfunktionen



- Public Methode Inventarwert erhält Vektor von Preisen und Quantitäten – berechnet Gesamtwert
- Public Methode Berechnung liest eine Textdatei (Lieferschein), berechnet Gesamtbetrag und erstellt eine Übersicht der Werte
- Mit Vererbung: Das Skript von Rechnungsübersicht muss Inventarwert nicht enthalten!

Von Base_GUICardGame zu BJ_GUI und Rommee_GUI



- Manche Methoden (z. B. DrawPlayer) abgeleitet...
- ...andere Methoden originär (z. B. CardValue)



Methode DrawPlayer

Base_GUICardGame

```
▽ {rc}+{Wmsg} DrawPlayer Anz;I;Card
:Access Public
:if 0=nc 'Wmsg'
    Wmsg←'' ''
:endif
:for I :in tAnz
    Card←Deck[1]
    rc←GUI.AddCardToSelf Wmsg,Card
    ±(~rc)/':leave'
    Player←Card,Player
    Deck←1↓Deck
:endif
▽
```

BJ_GUI

```
▽ DrawPlayer Anz
:Access public
'' BASE.DrawPlayer Anz
ValuePlayer++€/CardValue"Anz↑Player
▽
```

„Protected“ Member? Mehrfache Vererbung?

- Private Member werden nicht geerbt
- Protected: Zugriff erlaubt auch in abgeleiteten Klassen
- In Dyalog nur private und public
- Keine mehrfache Vererbung
- Lösung für Problem #2: Benutzung von :Include, Namespaces oder Klasseninstanzen als Felder, geschachtelte Klassen



Danke für Ihre Aufmerksamkeit!

Für weitere Fragen stehen ich und das Dyalog
Documentation Centre zu Ihrer Verfügung