



FACTOR:

Stackprogrammierung für Real-World-Anwendungen

Abnahmepräsentation

Andreas Maier
Marcel Steinle



Heutige Themen

- Anfängliche Schwierigkeiten mit Factor
- Factor – Ein Crashkurs
- ReFACTORing – ein einfaches Beispiel
- Demo Sokoban-Spiel
- Aufbau und Architektur des Spiels
- Zusammenfassung & Ausblick

2

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Anfängliche Schwierigkeiten

In Factor



Anfängliche Schwierigkeiten

- Herangehensweise
- Unlesbarer Code
 - Viele Stackoperationen hintereinander
 - Auftrennung in mehrere Words
- Neue Arbeitsweise mit dem Stack
 - Keine namentliche Referenzierung von Parametern
- Ungewohnter Umgang mit Quotations
- Schnelle Releasezyklen von Factor

4

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Einführung in Factor

Crashkurs



Einführung in Factor – Crashkurs I

- Einführung in die Stackprogrammierung
 - Syntax
 - Verwendung des Stacks
 - Spezielle Stackoperationen (Shuffle Words)
- Definition eigener Words

6

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



- Objektorientierung in Factor
 - Basistyp: Tuple
 - Konstruktoren
 - Accessors
- Anonyme Funktionen (Quotations)

ReFACTORing

Wird Factor seinem Namen gerecht?



Beispiel für das Refactoring

```
: hello-world-ui ( -- )
  [ <pile> "Hello World" <label>
    add-gadget "OK"
    [ close-window ] <border-
button> add-gadget
  "Refactoring" open-window ]
with-ui ;
```

9

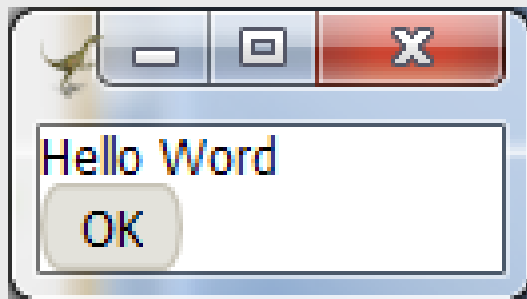
11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Beispiel für das Refactoring

```
: hello-world-ui ( -- )  
  [ <pile> "Hello World" <label>  
    add-gadget "OK"  
    [ close-window ] <border-  
button> add-gadget  
    "Refactoring" open-window ]  
  with-ui ;
```



10

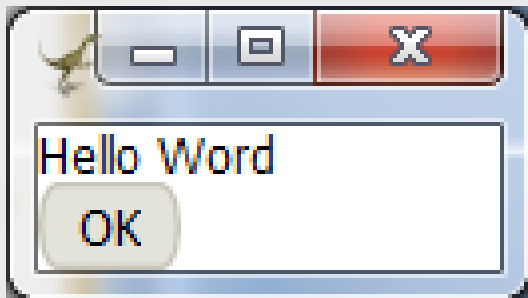
11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Beispiel für das Refactoring

```
: hello-world-ui ( -- )  
  [ <pile> "Hello World" <label>  
    add-gadget "OK"  
    [ close-window ] <border-  
button> add-gadget  
  "Refactoring" open-window ]  
  with-ui ;
```

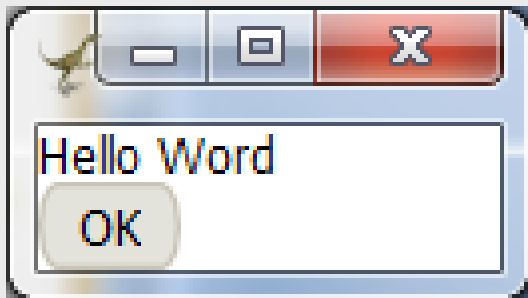


```
: <hello-label> ( -- gadget )  
  "Hello Word" <label> ;  
  
: <ok-button> ( -- gadget )  
  "OK" [ close-window ] <border-  
button> ;  
  
: hello-world-gadget ( -- gadget )  
  <pile>  
    <hello-label> add-gadget  
    <ok-button> add-gadget ;  
  
: hello-world-ui ( -- )  
  [ hello-world-gadget "Refactoring"  
  open-window ] with-ui ;
```

11

Beispiel für das Refactoring

```
: hello-world-ui ( -- )  
[ <pile> "Hello World" <label>  
  add-gadget "OK"  
  [ close-window ] <border-  
button> add-gadget  
  "Refactoring" open-window ]  
with-ui ;
```



```
: <hello-label> ( -- gadget )  
  "Hello Word" <label> ;  
  
: <ok-button> ( -- gadget )  
  "OK" [ close-window ] <border-  
button> ;  
  
: hello-world-gadget ( -- gadget )  
  <pile>  
    <hello-label> add-gadget  
    <ok-button> add-gadget ;  
  
: hello-world-ui ( -- )  
  [ hello-world-gadget "Refactoring"  
  open-window ] with-ui ;
```

12

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Demo Sokoban-Spiel

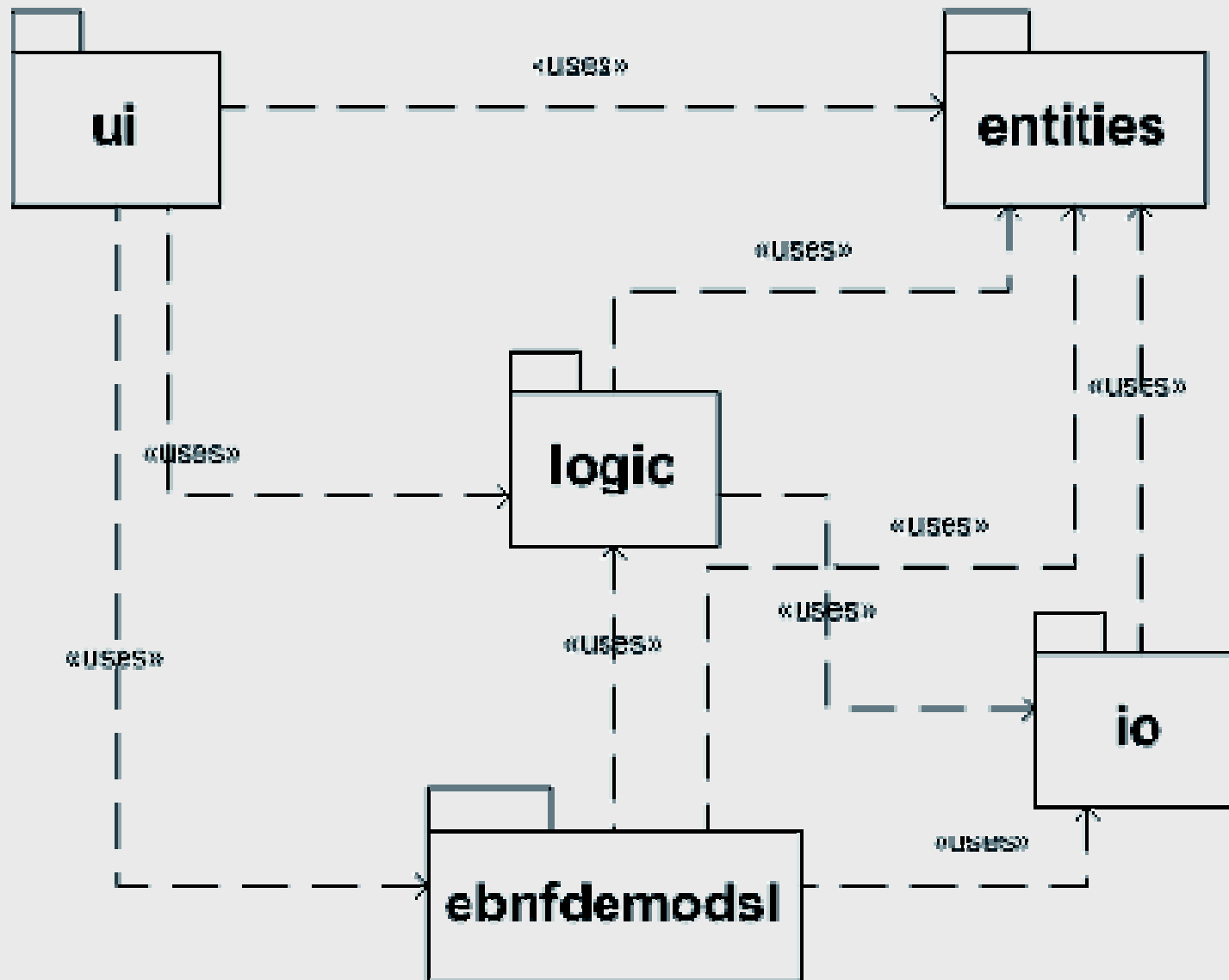
Demonstration

Aufbau und Architektur

Von Sokoban



Grundkomponenten des Spiels



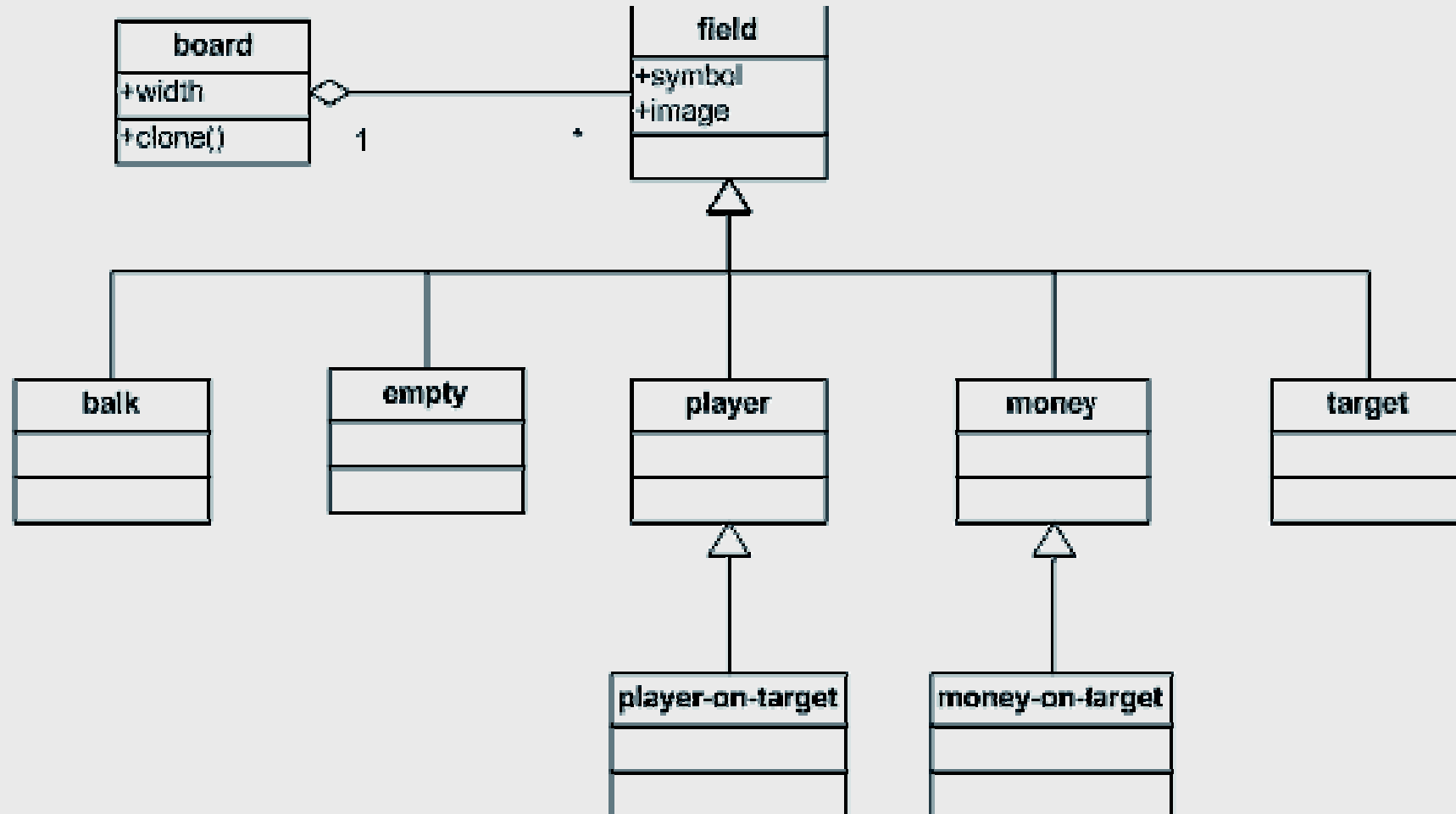
15

11.12.2009

Marcel Steinle, Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Basisklassen des Spiels



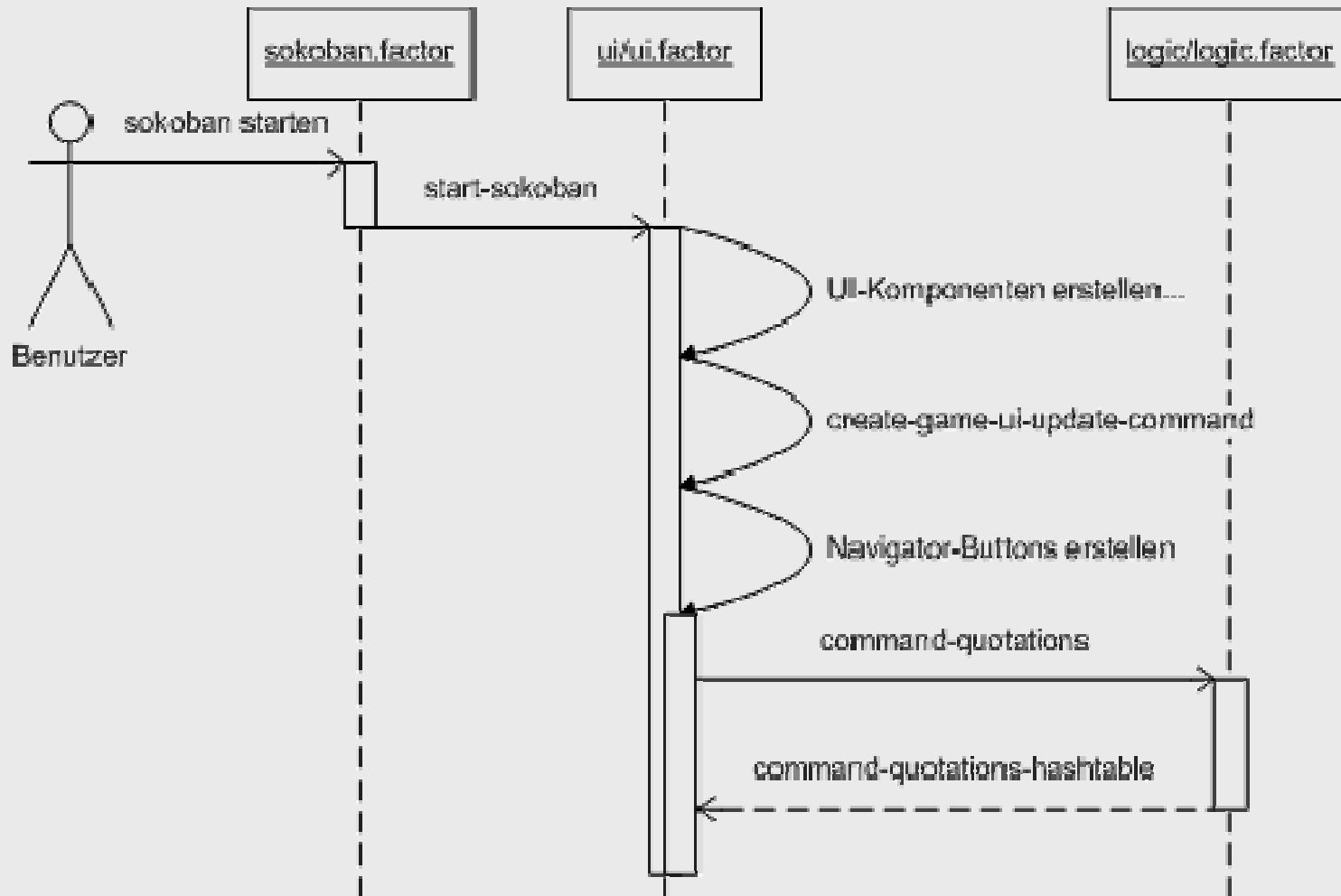
16

11.12.2009

Marcel Steinle, Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Kommunikation zwischen Logik und UI (1)



17

11.12.2009

Marcel Steinle, Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Kommunikation zwischen Logik und UI (2)

- Verwendung einer Variante des Observer-Entwurfsmusters
 - UI-Komponente erstellt Update-Command
 - Logik ruft Update-Command nach Änderungen am Spielfeld (nach einem Spielzug) auf
 - Durch den Aufruf des Update-Commands wird UI aktualisiert
- ➔ Lose Kopplung zwischen UI und Logik

18

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Vergleich Java – Factor (1)

- Word zum Erstellen des Update-Commands:

```
! updateCommand, Stack-Effekt: ( arrayOfQuot -- )  
:: createUpdateCommand ( uiBoard - updateCommand )  
  [ [ call( -- newObject index )  
    uiBoard change-image ] each ] ;
```

- Verwendung von Quotations (anonymen Funktionen)
- Verwendung von Closure durch Locals
- Verwendung einer each-Schleife

19

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Vergleich Java – Factor (2)

```
// UI-----  
// Definieren einer inneren Klasse, um die Umgebung für die Variable  
// uiBoard selber herzustellen, da Java 6 kein Closure unterstützt!  
private class UpdateCommandCreator {  
    private String uiBoard;  
  
    public UpdateCommandCreator(String uiBoard) {  
        this.uiBoard = uiBoard;  
    }  
  
    public UpdateCall createUpdateCommand () {  
        return new UpdateCall()  
        {  
            public void updateCommand(UpdateStep[] arrayOfQuot) {  
                for (UpdateStep step : arrayOfQuot) {  
                    CallUpdateStepResult res = step.callUpdateStep();  
                    changeImage(res.newObject, res.index, uiBoard);  
                }  
            }  
        };  
    }  
}
```

20

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



Parser für die Demo-DSL

- Eine textuelle Beschreibung für den Weg, den die Spielfigur zum Erreichen des Ziels ablaufen muss.
z.B.: „2*up down 3*right left end“
- Definieren der Grammatik durch die erweiterte Backus-Naur-Form (EBNF) direkt in Factor:

```
EBNF: demo-dsl-grammar
number-less-zero = [1-9] ([0-9])* => [[ ... ]]
route = "up" | "down" | "left" | "right"
command = (number-less-zero "*" )? route => [[ ... ]]
commands = command " " commands | "end"
;EBNF
```

21

Zusammenfassung & Ausblick

Jetzt haben wir es gleich geschafft ...



Zusammenfassung

- Factor hat eine einfache Syntax
- Zu Beginn ungewohnte Arbeitsweise
- Real-World-Anwendungen sind definitiv gut umsetzbar
- Durch den Einsatz verschiedener Sprachparadigmen sehr mächtig (stackorientiert, objektorientiert, funktional, dynamische Typisierung)

23

11.12.2009

Marcel Steinle , Andreas Maier - FACTOR: Stackprogrammierung für Real-World-Anwendungen



- Erweiterung von Sokoban z.B. durch Level-Editor, automatische Zielführung, ...
- Unit-Tests in Factor
- Erweiterungsmöglichkeiten in Factor betrachten: Parsing Words (SYNTAX), Metaprogrammierung, ...
- Vergleich mit anderen Stackprogrammiersprachen wie Joy, Forth, ...

Fragen?

- Andreas Maier
anmaier@stud.hs-heilbronn.de
- Marcel Steinle
msteinle@stud.hs-heilbronn.de



25