



## Sopratricks

# Überblick

## 1 GUIs und Kommentare

- GroupLayout unter NetBeans

- Java Coding Conventions

- Javadoc Comments

- Eingabeüberprüfung auf GUI-Ebene

## 2 Arbeiten mit Versionskontrolle

## 3 Zusammenfassung

# Sopratricks

## 1 GUIs und Kommentare

GroupLayout unter NetBeans

Java Coding Conventions

Javadoc Comments

Eingabeüberprüfung auf GUI-Ebene

## 2 Arbeiten mit Versionskontrolle

## 3 Zusammenfassung

## GroupLayout

- ▶ Neues Layoutprinzip
- ▶ Seit Java 6
- ▶ Horizontales Layout getrennt von vertikalem Layout
- ▶ Relativ kompliziert von Hand zu programmieren
- ▶ Gui-Builder-Tutorial:  
<http://www.netbeans.org/kb/55/quickstart-gui.html>
- ▶ Hand-Coding-Tutorial: <http://java.sun.com/docs/books/tutorial/uiswing/layout/group.html>

## Warum Coding Conventions?

- ▶ Bessere Wartbarkeit des Codes
- ▶ 8 Leute im Team
- ▶ Code besser lesbar → besser verständlich
- ▶ Siehe auch: <http://java.sun.com/docs/codeconv/>

## Was nenne ich Wie?

- ▶ Klassen → Substantive (beginnend mit Großbuchstabe)

```
public class Beispiel { ..
```

- ▶ Methoden → Verben (beginnend mit Kleinbuchstabe)

```
private void machWas() { ..
```

- ▶ Variablen

- ▶ Möglichst keine Ein-Buchstaben-Variablen, falls doch:

- ▶ i,j,k,m,n für int

- ▶ c,d,e für char

- ▶ Konstanten in Großbuchstaben, Wörter getrennt durch '\_', z.B.:

```
static final int MAX_PERSONEN = 100;
```

## Leerzeichen und Leerzeilen

- ▶ Leerzeilen

- ▶ Zwischen Methoden
- ▶ Zwischen Variablen und Statements
- ▶ Vor Kommentaren

- ▶ Leerzeichen

- ▶ Vor und nach Operatoren:

```
a + b = c;
```

```
String s = "Irgendwas " + a + " noch etwas";
```

- ▶ Ausnahme:

```
a++;
```

- ▶ Nach Casts:

```
a = (int) b;
```

- ▶ Nach Schlüsselwörtern

```
while (true) ↔ methode()
```

## Verschiedene Arten von Kommentaren

```
/* single line comment */
```

```
a + b = c;      /* trailing comment */
```

```
// end-of-line comment
```

```
/*  
 * block comment  
 */
```

```
/**  
 * javadoc comment  
 */
```

## Aufbau

- ▶ Tags

- @author** Name des Autors

- @param** Name des Parameters Beschreibung

- @return** Beschreibung des Rückgabewerts

- @throws** Mögliche Exceptions

- ▶ Formatierungen

- ▶ Einfache HTML-Elemente können zur Formatierung benutzt werden (`<p>`, `<ul>`, `<li>`, ...)

- ▶ `<code></code>`-Style für z.B. Package- oder Klassennamen

- ▶ Erster Satz tauch in Zusammenfassung auf

## Eingabeüberprüfung auf GUI-Ebene

- ▶ Eingabeüberprüfung sollte wenn möglich in der GUI stattfinden
  - ▶ Erspart komplizierte Abfrageroutinen
  - ▶ Unzulässige Eingabemöglichkeiten sind direkt sichtbar
- ▶ 2 Heransgehenweisen möglich
  1. Überprüfung im Nachhinein
    - ▶ `InputVerifier`  
Nur zulässige Eingaben erlauben den Focus weiterzugeben
  2. Überprüfung während der Eingabe
    - ▶ `JFormattedTextField`  
Mittels `MaskFormatter` wird eine eindeutige Eingabemaske vorgegeben
    - ▶ `DocumentFilter`  
Nur zulässige Eingaben werden übernommen
    - ▶ Verwendung freier Bibliotheken für spezielle Daten (bsp. Datum)

# Sopratricks

- 1 GUIs und Kommentare
  - GroupLayout unter NetBeans
  - Java Coding Conventions
  - Javadoc Comments
  - Eingabeüberprüfung auf GUI-Ebene
- 2 Arbeiten mit Versionskontrolle
- 3 Zusammenfassung

# SVN

- ▶ SVN

- ▶ TortoiseSVN (Shell-Integrierung von SVN-Kommandos)  
<http://tortoisesvn.tigris.org/>
- ▶ Subclipse (SVN Plug-in für Eclipse)  
<http://subclipse.tigris.org/>
- ▶ Subversion for NetBeans  
<http://subversion.netbeans.org/>

## Arbeiten mit SVN

- ▶ Genereller Arbeitszyklus
  1. Update
  2. evtl. Konflikte bereinigen
  3. Bearbeiten
  4. Commit (immer mit sinnvoller LogMessage)
- ▶ Repository sollte folgende Unterverzeichnisse enthalten
  - trunk** Für alle Dateien im Hauptentwicklungszweig
  - branches** Für abzweigende Entwicklungen
- ▶ Abzweigungen können mittels Branch erstellt werden
- ▶ Merge führt verschiedene Zweige wieder zusammen

# Sopratricks

- 1 GUIs und Kommentare
  - GroupLayout unter NetBeans
  - Java Coding Conventions
  - Javadoc Comments
  - Eingabeüberprüfung auf GUI-Ebene
- 2 Arbeiten mit Versionskontrolle
- 3 Zusammenfassung

## Zusammenfassung

- ▶ Beispielcode online verfügbar  
`http://www.informatik.uni-ulm.de/sgi/progwerkstatt`
- ▶ Bei weiteren Fragen: `progwerkstatt.informatik@uni-ulm.de`