

Komponenten und Layout

Konzepte und Umsetzung

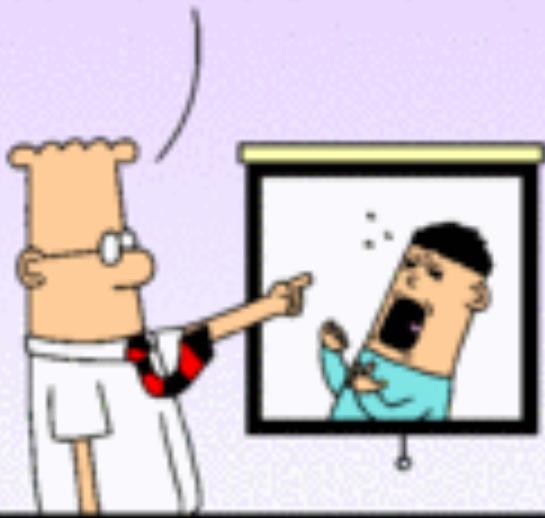


Agenda

1. Entwurf und Layout grafischer Oberflächen (25 Minuten)
2. Überblick JavaFX (25 Minuten)
3. Komponenten und Layout mit JavaFX (25 Minuten)



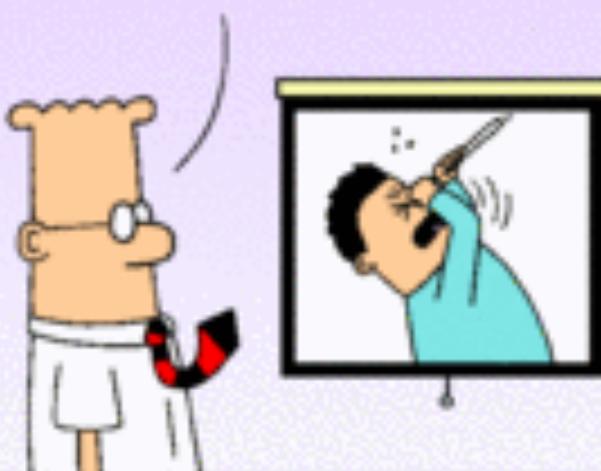
THE RESULTS OF
OUR BETA TESTING
ARE IN.



OUR USER INTERFACE
TRIGGERED WIDE—
SPREAD DESPONDENCY
AND SELF-MUTILATION.

scottadams@aol.com

www.dilbert.com



OBVIOUSLY WE'LL NEED
TO DELAY OUR LAUNCH
FOR THE PUBLIC GOOD.

WHEN DID YOU
BECOME A
COMMUNIST?

7-1-09 © 2001 Scott Adams, Inc./Digit. by UFS, Inc.



wGetGUI v1.0 | You are using GNU Wget 1.9-beta - 1.7 is minimum.

URL:

Hosts

Span All
 Allow List ->

 Reject List ->

Accept/Reject

Accept: Reject:
 htm(l) gif
 jpg txt
 zip exe
 doc All

Custom list:
"thm"
"thumb"
"small"

Retries:

Additional Parameters:

 Act like a browser
 Convert links
 Ignore robots.txt

Configure Proxy

Save settings Load settings

Running Options

Go 2 background
 No info
 All info
 Some info
 Append to logfile
 Overwrite Logfile

Logfile:

About Exit

Start wGetStart.bat Add to wGetStart.bat Empty wGetStart.bat Pro Mode

Retrieval Options

No clobber
 Timestamping
 Continue file download

Quota (kB):

Spider (check for files)

No directories
 Force directories
 Save to custom dir:

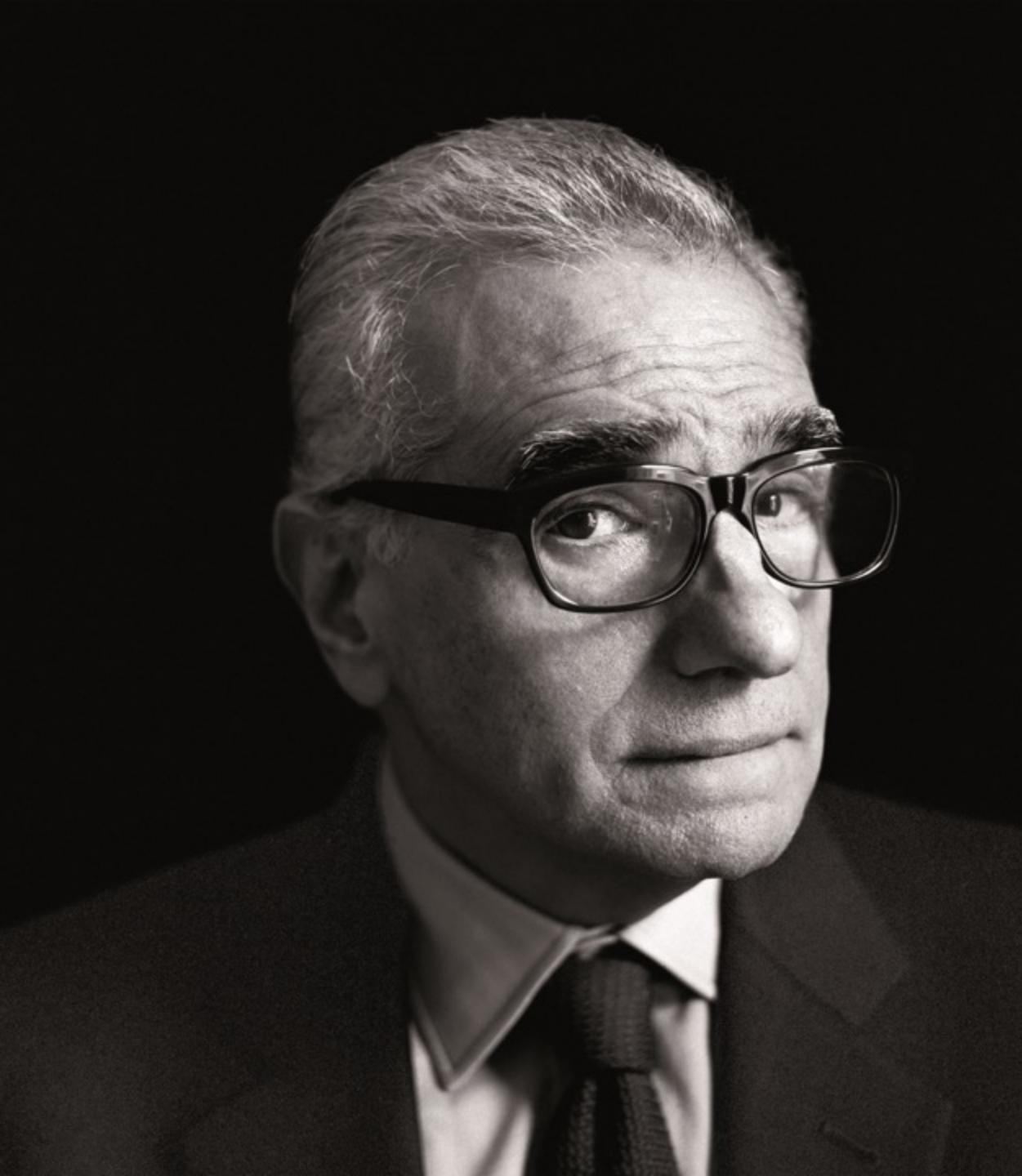
 Clear Server Cache

Recursive Retrieval

Depth:

Download "as-is"
 Mirror site
 add HTML suffix
 Only go deeper





There is no such
thing as simple.
Simple is hard.

Martin Scorsese

Die ISO Norm 9241 regelt Grundsätze der Dialoggestaltung

EN ISO 9241-110 Grundsätze der Dialoggestaltung [[Bearbeiten](#) | [Quelltext bearbeiten](#)]

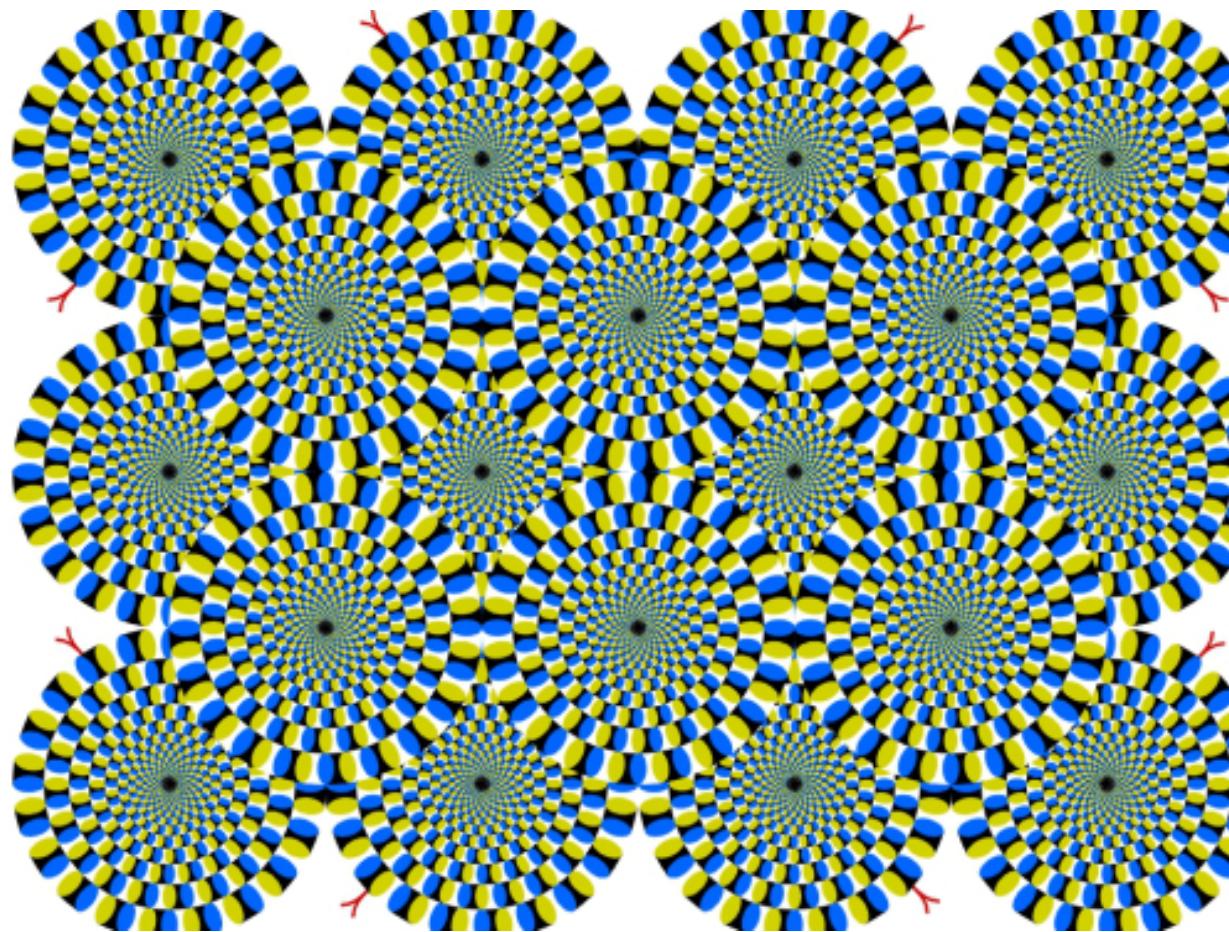
Benutzungsschnittstellen von **interaktiven** Systemen, wie **Webseiten** oder **Software**, sollten vom Benutzer leicht zu bedienen sein. Der Teil 110 der DIN EN ISO 9241 beschreibt folgende Grundsätze für die Gestaltung und Bewertung einer Schnittstelle zwischen Benutzer und System (Dialoggestaltung) und ersetzt den bisherigen Teil 10:

- **Aufgabenangemessenheit** – geeignete Funktionalität, Minimierung unnötiger Interaktionen
- **Selbstbeschreibungsfähigkeit** – Verständlichkeit durch Hilfen/Rückmeldungen
- **Lernförderlichkeit** – Anleitung des Benutzers, Verwendung geeigneter **Metaphern**, Ziel: minimale Erlernbarkeit
- **Steuerbarkeit** – Steuerung des **Dialogs** durch den Benutzer
- **Erwartungskonformität** – Konsistenz, Anpassung an das **Benutzermodell**
- **Individualisierbarkeit** – Anpassbarkeit an Bedürfnisse und Kenntnisse des Benutzers
- **Fehlertoleranz** – Das System reagiert tolerant auf Fehler oder ermöglicht eine leichte Fehlerkorrektur durch den Benutzer

https://de.wikipedia.org/wiki/EN_ISO_9241#EN_ISO_9241-110_Grunds%C3%A4tze_der_Dialoggestaltung



Was bedeutet das konkret ???



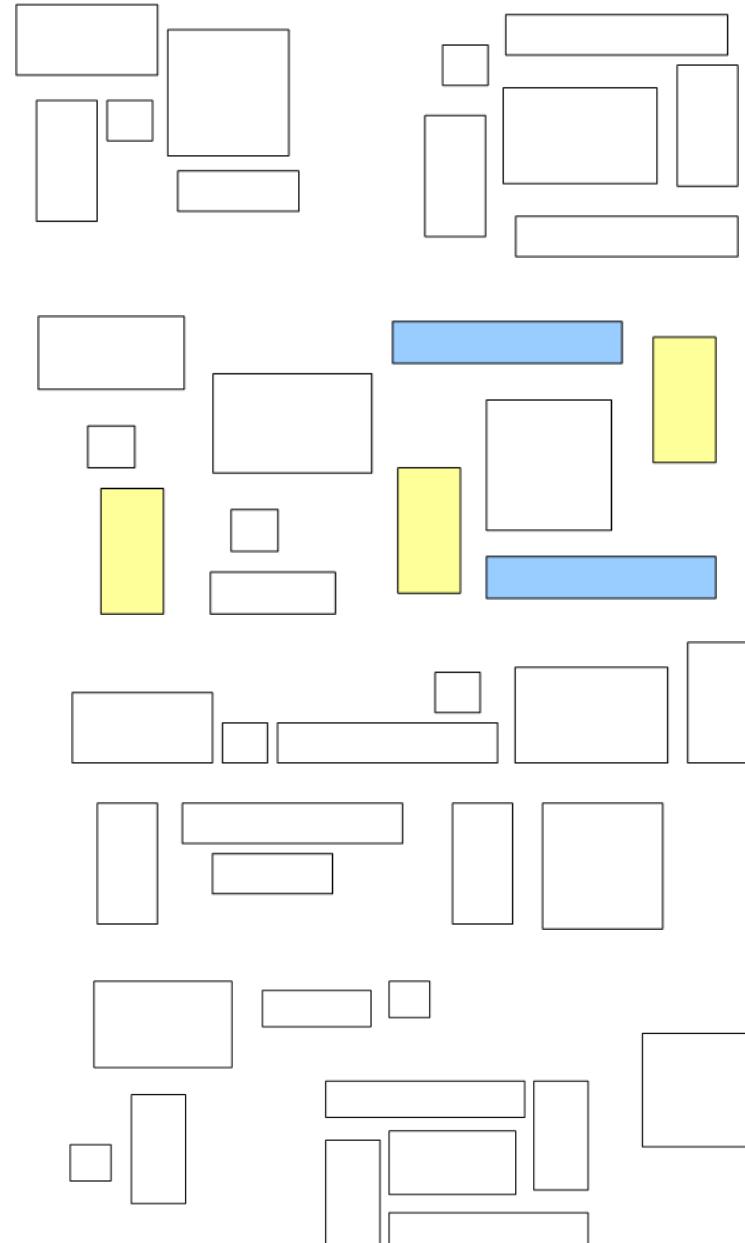
GUI Design - Es gibt nicht nur eine Wahrheit. Es gibt aber viele gute Quellen je Platform.

- Apple
 - <https://developer.apple.com/design/tips/>
 - <https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html>
- Google
 - <http://developer.android.com/design/index.html>
- Microsoft
 - <https://msdn.microsoft.com/windows/uwp/layout/design-and-ui-intro>
- Diverse
 - <http://t3n.de/news/7-goldene-regeln-ui-design-582053/>



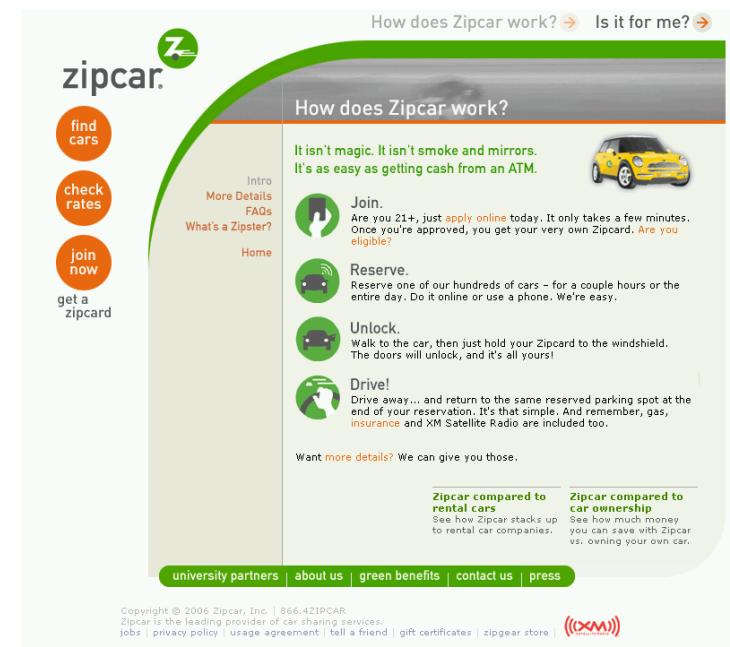
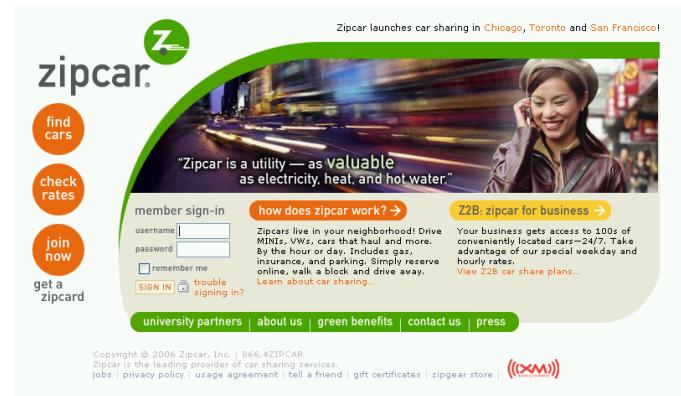
Gestaltungsregeln an der UI

- Räumliche Nähe
 - Zusammengehörende Nutzungselemente / Eingabefelder nahe beieinander
- Ähnlichkeit
 - Elemente mit ähnlicher Farbe, Font, Form, Größe, Orientierung werden als zusammengehörend wahrgenommen
- Spalten und Zeilen Bildung
 - Augen folgen Orientierungslinien
- Abgeschlossenheit
 - Abgeschlossene Formen wie Rechtecke bündeln Zusammenhänge



Einheitlichkeit

- Layout aller Fenster ähnlich wählen
 - Groblayout
 - Farben, Fonts, Formen
 - Position und Art der Buttons
- Damit: Orientierung der Nutzer verbessern
- Hilfen dazu
 - Styleguide
 - Look and Feels, CSS ...
 - Framework mit Basisklassen für die Standard-Dialoge



Orientierung bieten

- **Blocksatz, Gruppierung**
 - gruppierte Edittexte in Länge angelichen
 - Ausnahmen: Kurzformate
(1 Zeichen, PLZ)
 - Separatoren verwenden
- **Spaltenbildung**
 - Mehrere Spalten nutzen den Raum besser aus.
- **Ausnutzung der Fläche**
 - Freie/leere Flächen einfügen

Ebene 1

Stil Layout

Schrift

Helvetica Neue

Normal 28 pt

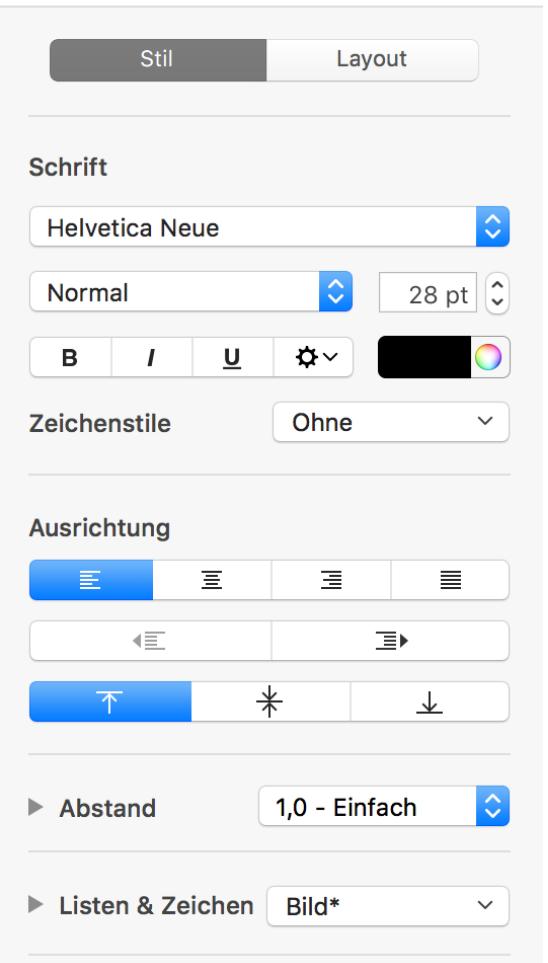
B I U ⚙

Zeichenstile Ohne

Ausrichtung

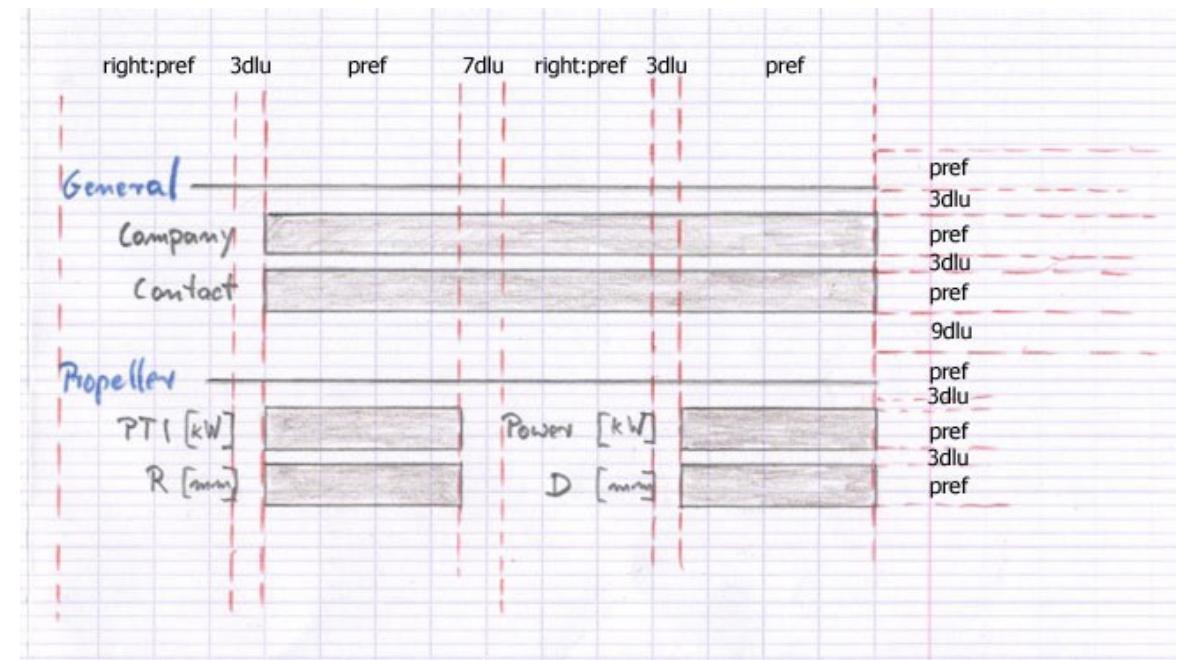
Abstand 1,0 - Einfach

Listen & Zeichen Bild*



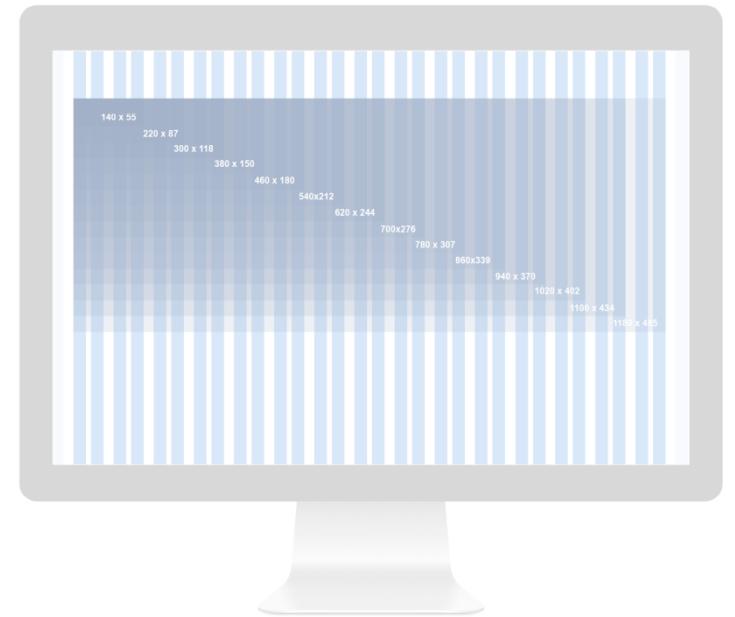
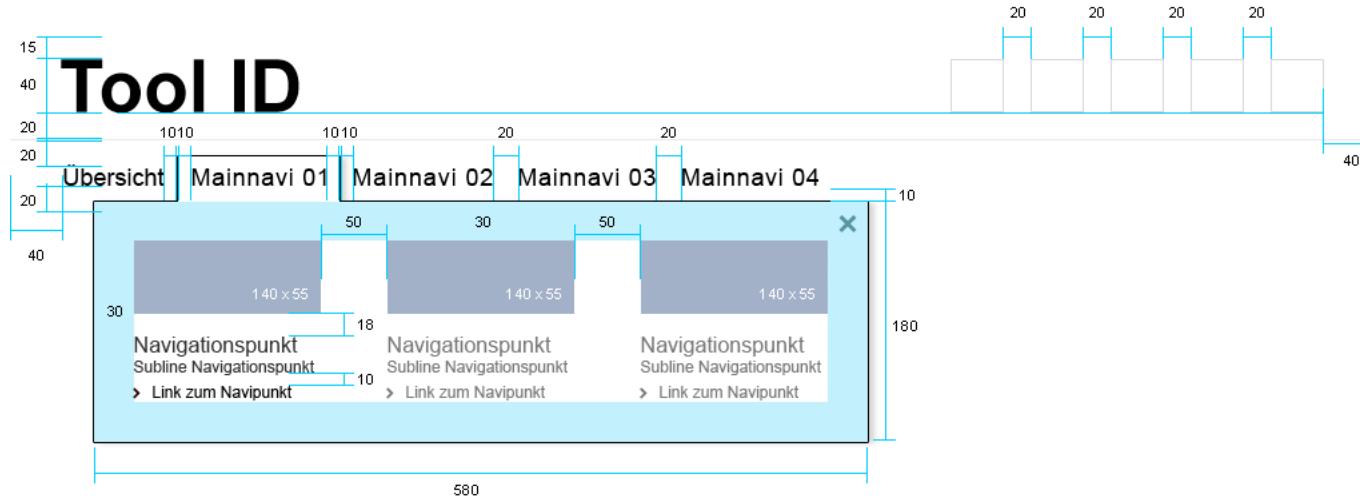
Hilfsmittel: Dynamisches Raster

- Ein Raster hilft,
 - Dialogelemente einzuordnen
 - Spalten und Blockbildung zu fördern
- Ein Raster enthält:
 - Was ist dehnbar?
 - Abstände in Pixeln / dlu
 - Evtl Bündigkeiten
(links oder Rechts?)



Grafik aus www.jgoodies.com

Hilfsmittel: Spaltenorientiertes Raster



Responsive Layout / Design

- **Grundidee:** Baue eine UI oder Webseite so, dass auf verschiedenen Auflösungen unterschiedliche Fonts, Bilder und Layouts verwendet werden.
 - Twitter hat mit der CSS Bibliothek Bootstrap Responsive Design zum Durchbruch verholfen. Seitdem sind viele Webseiten und Anwendungen „Responsive“
 - <http://getbootstrap.com>
- Responsive Design basiert im Browser auf CSS Media Queries
 - Media Queries sind ein W3 Standard
 - <http://www.w3.org/TR/css3-mediaqueries/>
- Responsive Design mit JavaFX funktioniert nur auf Umwegen (n Szenen)
 - <https://youtu.be/nQiFiXJTZzc>



Konsistente Farbgebung

- Wenige verschiedene Farben verwenden
- Vermeiden:
 - Grelle Farben, volle Sättigung
(z.B. Leuchtendes Rot, Gelb, Blau)
 - Große weiße Flächen (das ermüdet)
 - Große dunkle Flächen
- Auf konsistente Verwendung der Farben achten!
 - Farben haben Bedeutung (z.B. Rot = Fehler/Achtung)
 - Styleguides / Look and Feel verwenden

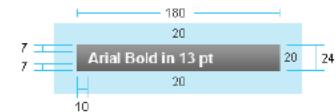


Hilfsmittel: Farbschema

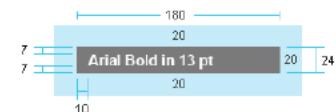
- Ziel: Intuitives Verständnis der GUI, Einheitlichkeit
- Z.B. Aktive/ markierte Teile immer blau, inaktive Teile grau
- Farbraster und Farbschemata



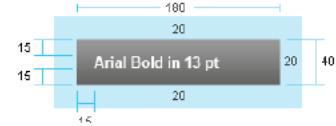
DEFAULT
Schriftart: Arial Bold / 13 pt
Schriftfarbe: #ffffff
Hintergrundfarbe: Linearer Verlauf



AKTIV / HOVER
Schriftart: Arial Bold / 13 pt
Schriftfarbe: #ffffff
Hintergrundfarbe: #666666



MOBILE / TABLET VERSION
Schriftart: Arial Bold / 13 pt



Hilfsmittel: UI Mockups / Wireframes

- Oberflächen werden nur skizziert – ideal für die Spezifikation von Oberflächen
 - <http://www.balsamiq.com/products/mockups>
 - moqups.com

The wireframe shows a user interface for analyzing log files. It is divided into two main sections: 'Logfile' and 'Analysis'.
Logfile: Contains fields for 'File' (set to 'C:\test.log'), 'Format' (set to 'C:\test.log'), and time range controls ('From 8:00:00' and 'To 12:00:00').
Analysis: Contains fields for 'Series' (set to '.*(Exception).*') and 'Type' (set to 'Summary'). It also includes a 'Value' field ('.*(x).*'), a 'Format' field ('12:00:00'), and a 'save' button.



Vorgehen beim Bau grafischer Oberflächen

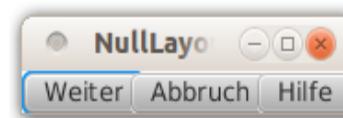
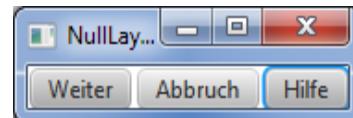
- Phasen bei der Erstellung von Layouts
 - P1: Grobe Skizze „Wireframe“
 - P2: Feine Skizze „Mock-Up“ incl. Größenanpassung
 - P3: Design (Photoshop, Visio o.ä.)
 - P4: UI-Umsetzung (ohne Funktion)
 - P5: UI-Umsetzung (mit Funktion)
- Die Aufwände steigen pro Phase an – Änderungen in einer späten Phase werden teuer
 - Beispiel: P1=1h, P2=4h, P3=1BT, P4=2BT, P5=5BT
 - 1h = 1 Stunde
 - 1 BT = 1 Bearbeitertag



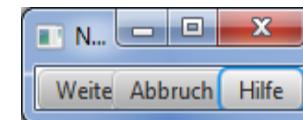
Dynamisches Layout

- Warum sollen Widgets nicht einfach absolut positioniert werden?

- Problem 1: Unterschiedliche Plattformen / Auflösungen (DPI)
 - Windows 7 mit Java 7
 - Ubuntu 12.04 mit Java 7

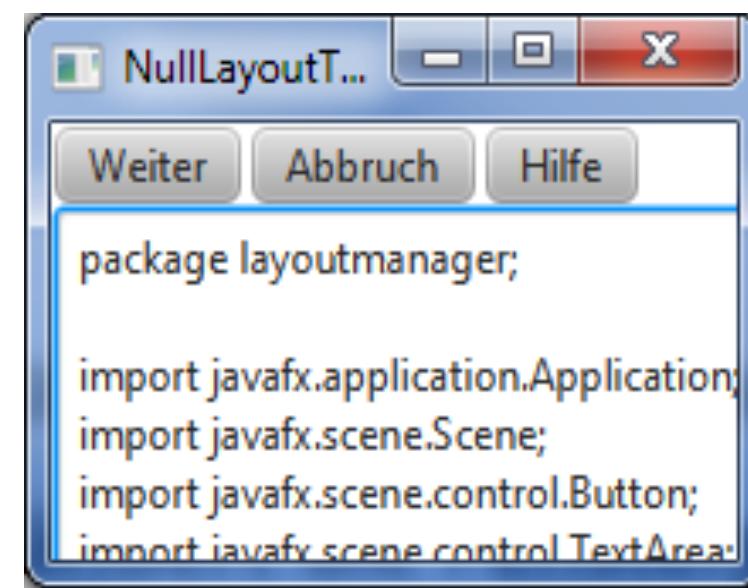
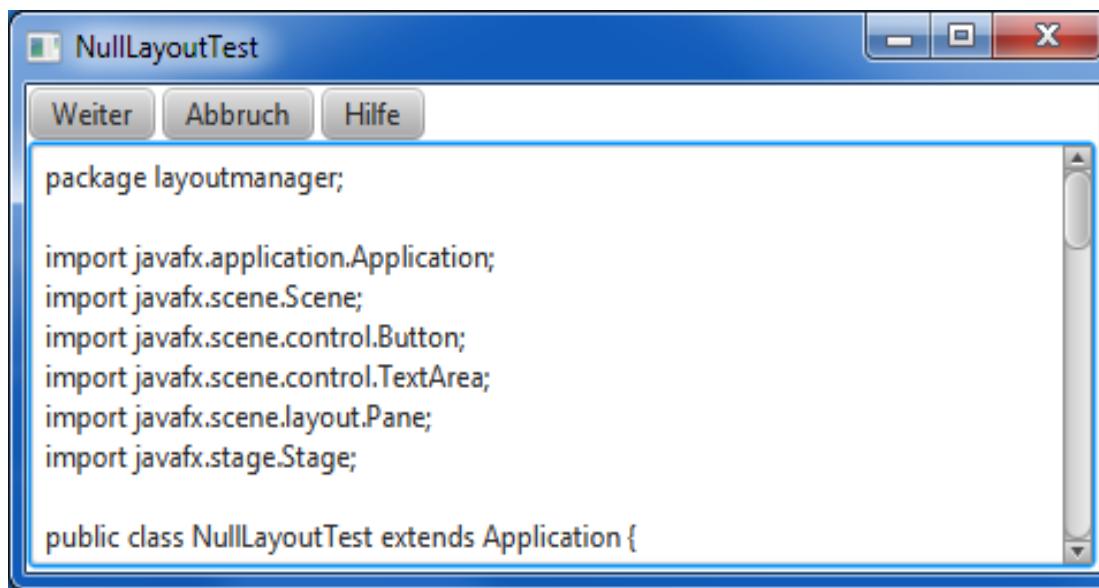


- Problem 2: Internationalisierung
 - Nachträgliche Übersetzung (Internationalisierung) der Texte in andere Sprachen
 - Ausgaben (Sprachvarianten):



Dynamisches Layout

- Problem 3: Interaktive Größenänderungen der Dialoge
 - Der Anwender ändert die Fenstergröße interaktiv. Der Inhalt passt sich nicht automatisch an (Platzverschwendungen oder „Verschwinden“ von Inhalten):



Dynamisches Layout - Grundlagen

- Problem 4: Textorientierung
 - Container unterstützen eine Ausrichtung der Komponenten anhand der Textausrichtung der eingestellten Sprache und des Landes.



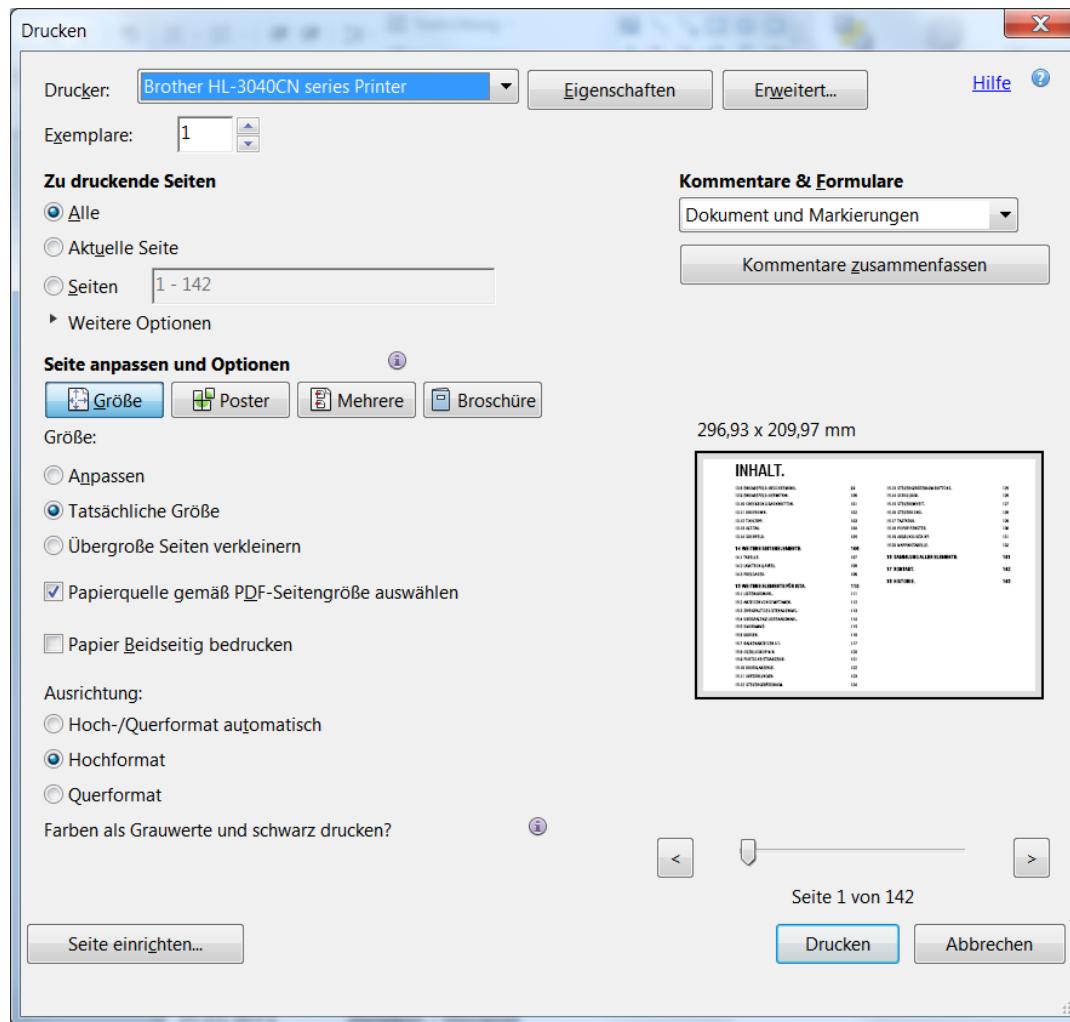
Dynamisches Layout - Grundlagen

■ Konsequenzen:

- Plattformunabhängige Programmierung ist mit absoluten Layouts nicht sinnvoll möglich (unterschiedliche Zeichensätze oder Zeichengrößen).
- Portierung einer Anwendung in mehrere Sprachen erfordert bei absoluten Layouts manuelle Nacharbeiten.



Bewerten Sie den folgenden Dialog!



Agenda

1. Entwurf und Layout grafischer Oberflächen (25 Minuten)
2. Überblick JavaFX (25 Minuten)
3. Komponenten und Layout mit JavaFX (25 Minuten)



Oberflächenprogrammierung mit JAVA

- AWT (Abstract Windows Toolkit)
 - Bestandteil des JDK seit Version 1.0
 - Konzept = Einheitliches API auf GUIs von Windows, Unix, Mac OS (D.h. GUI besteht aus Windows-Buttons oder Mac OS Buttons, etc.)
- Swing
 - Eingeführt: März 1998, Bestandteil des JDK ab Version 1.2 „Java 2“
 - ersetzt/erweitert AWT
 - Konzept = GUI wird komplett in Java „gemalt“ (D.h. GUI besteht aus Grafikelementen, die Swing selbst verwaltet)
 - Trotz verbreiteter Web-GUIs immer noch hohe Praxisrelevanz (IntelliJ + Netbeans sind mit Swing entwickelt)
- SWT (Standard Widget Toolkit)
 - GUI Framework von Eclipse
 - Konzept: Verwendung der Betriebssystem-GUIs - Benötigt nativen Wrapper-Code
 - Relevant für Entwicklung von Eclipse-Plugins oder als Rich-Client-Plattform
- JavaFX

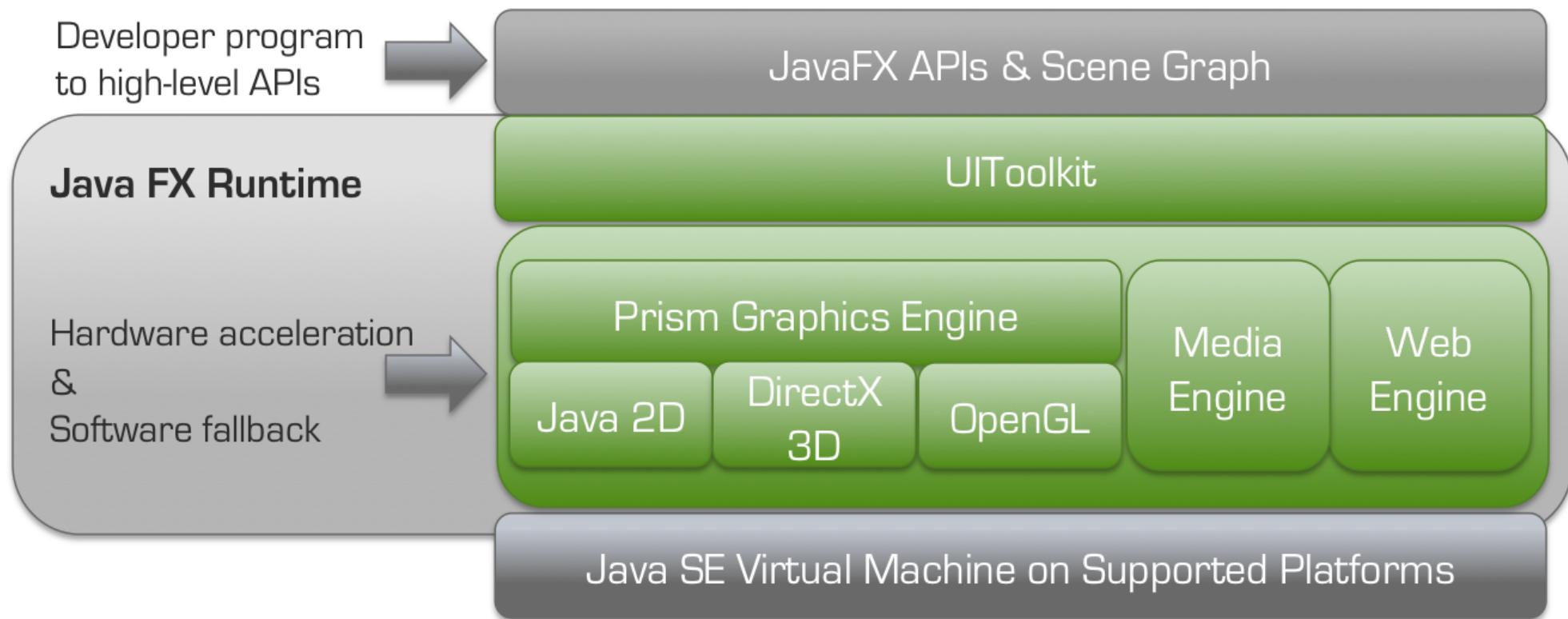


JavaFX

- 2007 als Konkurrenz zu Silverlight / Flash gestartet
 - RIA (Rich-Internet-Applications)
 - Eigene Scriptsprache (JavaFX Script)
 - Die letzte Version 1.3 war nicht erfolgreich
- Relaunch V2 - 2011 auf der JavaOne als **Ersatz für Swing**
 - Java-API / JavaFX Script wird nicht mehr offiziell unterstützt
 - Swing ist damit Vergangenheit
 - Das UI-Framework für Rich-Client-Entwicklung mit JAVA



JavaFX - Architektur



JAVAFX läuft auf embedded Hardware

Raspberry Pi

Raspberry Pi Type B Single Board Computer

The Raspberry Pi is a credit-card sized computer board that plugs into a TV and a keyboard. It's a miniature ARM-based PC which can be used for many of the things that a desktop PC does, like spreadsheets, word-processing and games. It also plays High-Definition video.

Raspberry Pi Model B comes as a single board computer, as shown above, and is not cased. Accessories are not included, however can be ordered below.

£25.92 (£21.60 exc. VAT)



Despatch expected in 8 week(s)



ODROID-C1 Einplatinen-Computer, 1,5 GHz QuadCore, USB, 1 GB



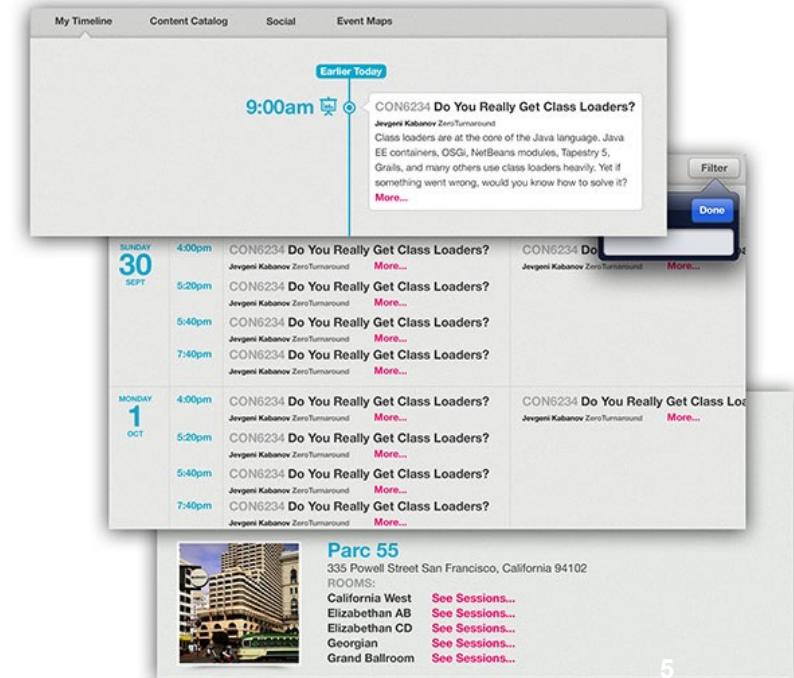
Bestellnummer: 810 378

Preis: 44,00 €

Inkl. MwSt., zzgl. Versandkosten

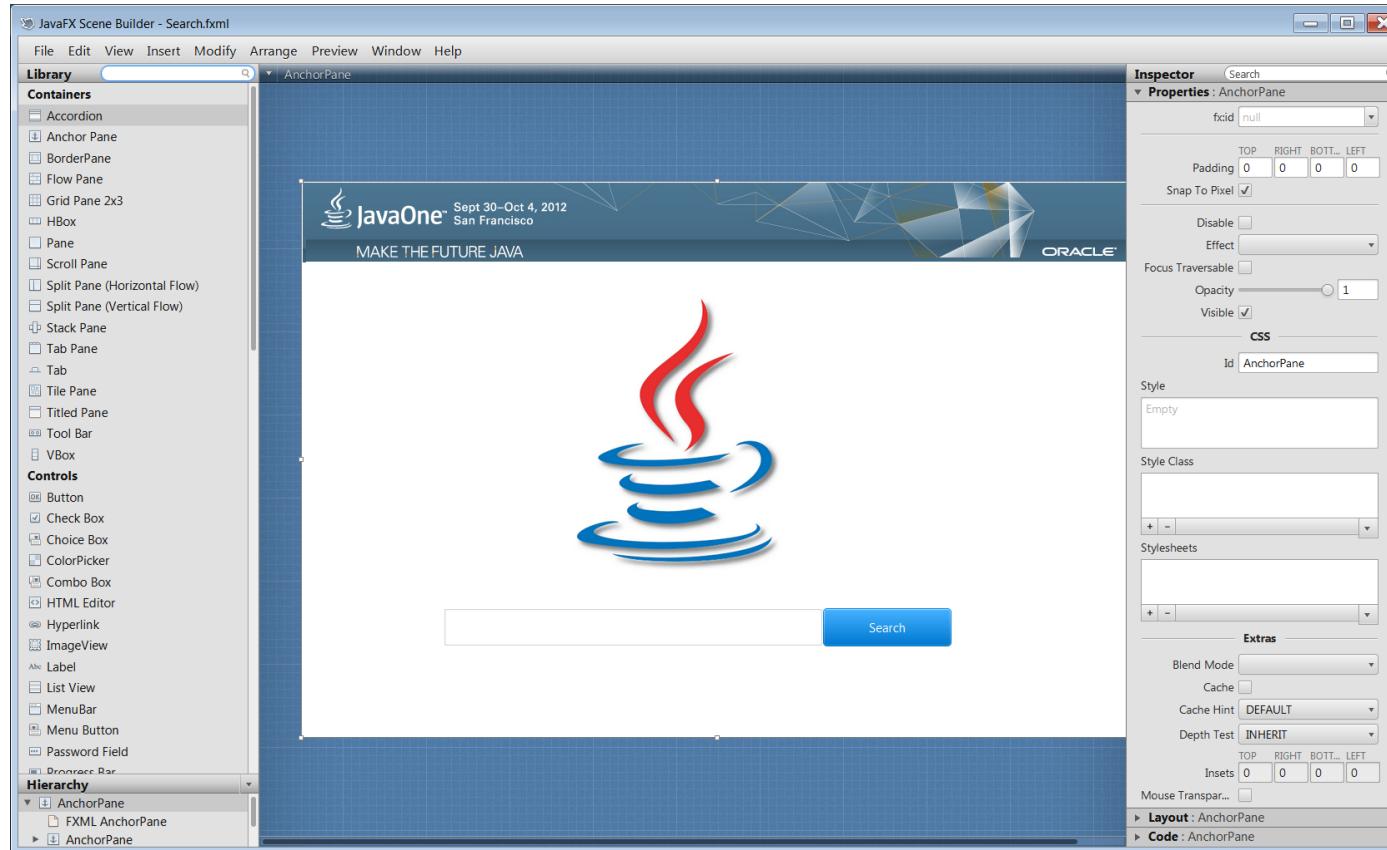
Verfügbarkeit: ✓ Artikel verfügbar.

1 Stück

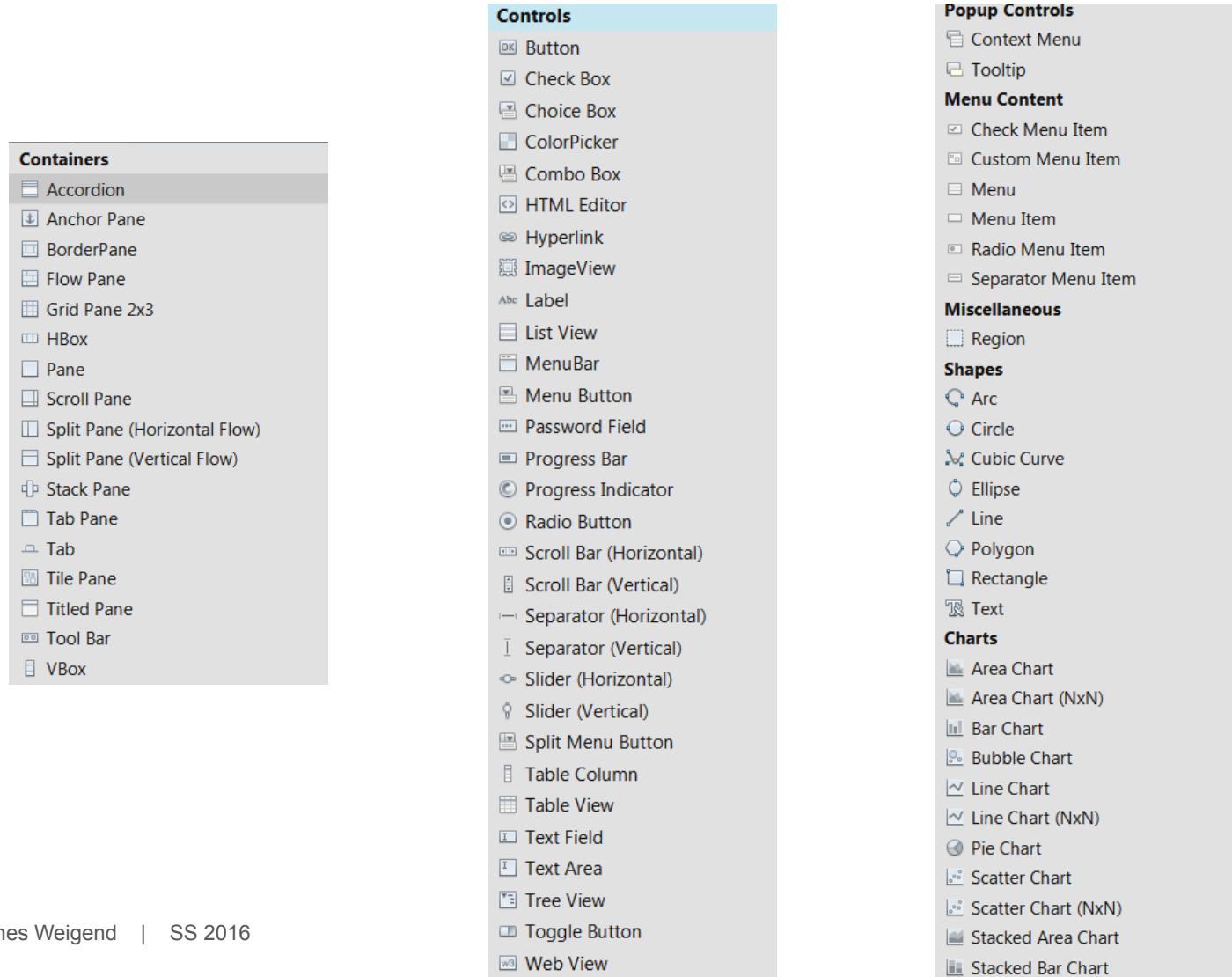


JAVAFX hat mit Scenebuilder einen mächtigen grafischen Editor

- Scenebuilder generiert (F)FXML
- Ist selbst eine FX-Anwendung

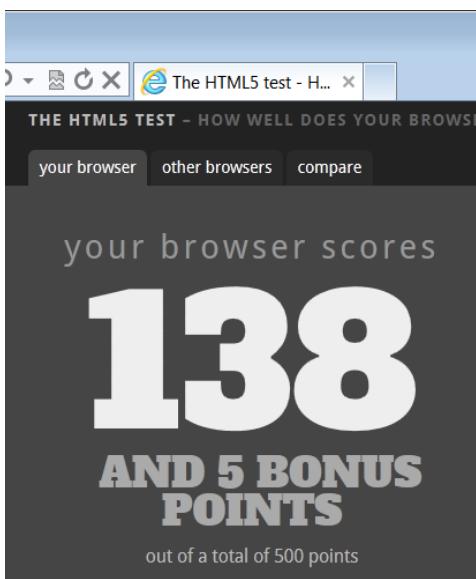


JavaFX hat eine umfangreiche UI-Komponentenbibliothek incl. Webkit-Browser

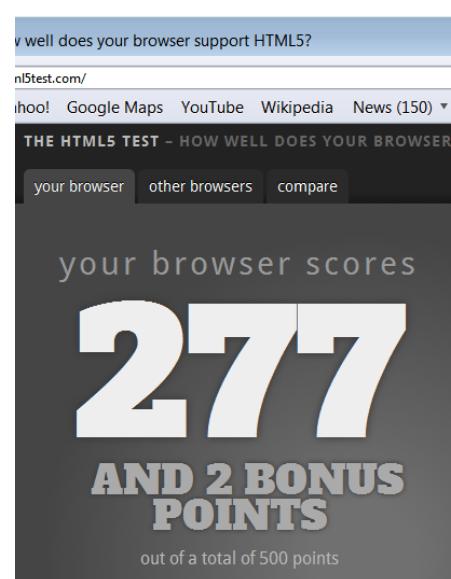


JavaFX Webview unterstützt HTML-5

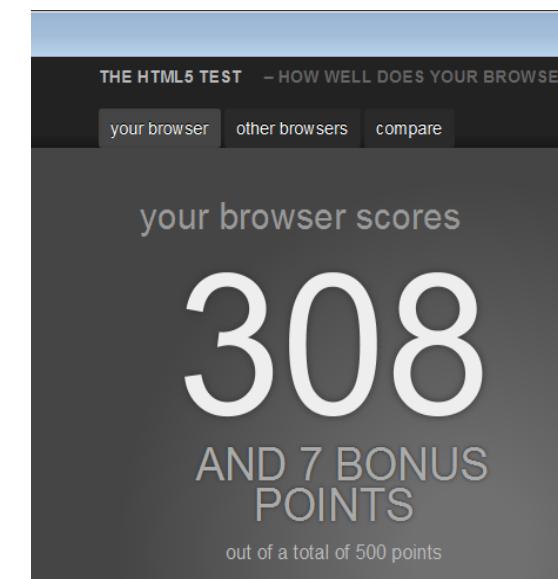
IE9



Safari 5.1



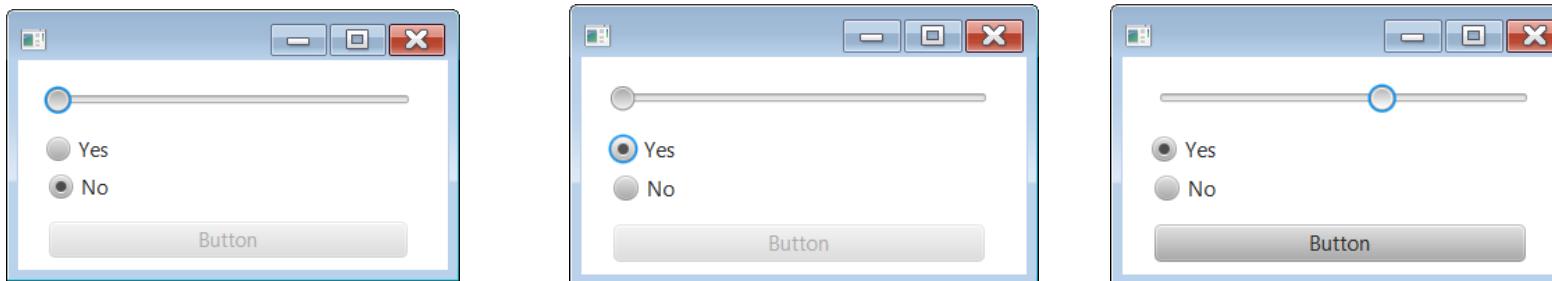
JavaFX 2.2 - Webkit



JavaFX hat ein mächtiges Databinding

- Alle FXControls unterstützen Binding

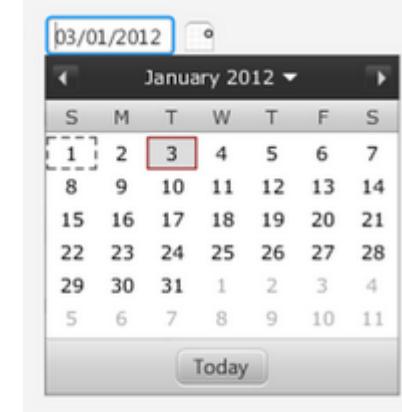
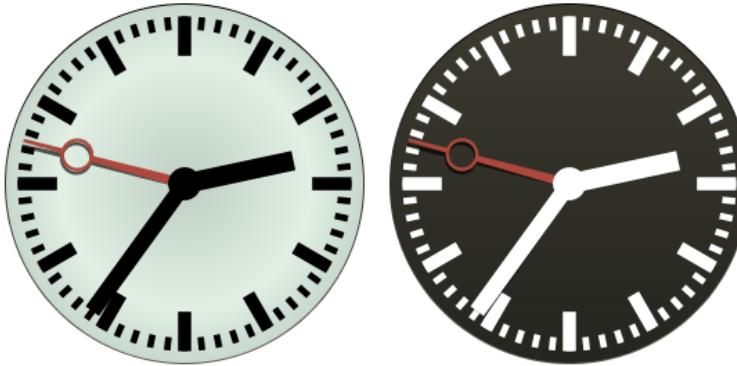
```
IntegerProperty ip = new SimpleIntegerProperty(40);  
slider.valueProperty().bindBidirectional(ip);
```



- Expressions (OR, AND, NOT)

```
button.disableProperty().bind(  
    radioNo.selectedProperty().  
    or(slider.valueProperty().greaterThan(50d).not())  
);
```

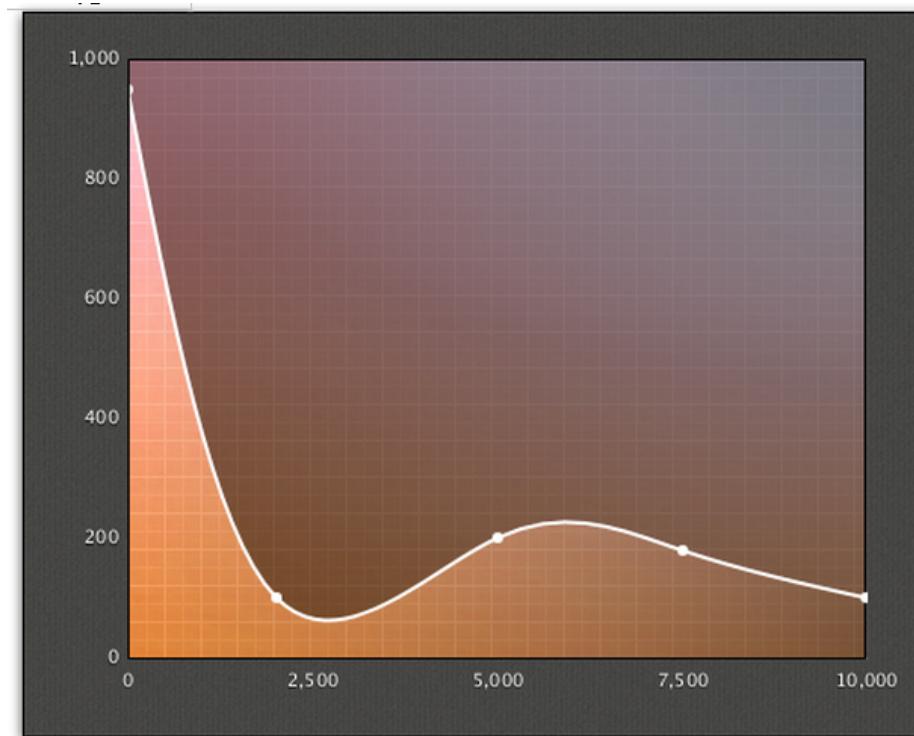
Eigene Controls (Custom-Control) können integriert werden



- a) Skinning von bestehenden Controls
- b) Kombination von bestehenden Controls
- c) Neues Control (Skin + Control + Model)

JavaFX hat eine Chart-Komponente.

- http://www.youtube.com/watch?v=kqWGD_EXUHY



JavaFX kennt Animationen

■ Transition

- Parallel
- Sequenziell
- Geschachtelt

■ Timeline

```
private void fadeOutIn(Node out, Node in) {  
    FadeTransition fadeOutTransition = new FadeTransition(Duration.millis(500), out);  
    fadeOutTransition.setFromValue(1.0);  
    fadeOutTransition.setToValue(0.0);  
    fadeOutTransition.play();  
    out.setVisible(false);  
    in.setVisible(true);  
    FadeTransition fadeInTransition = new FadeTransition(Duration.millis(500), in);  
    fadeInTransition.setFromValue(0.0);  
    fadeInTransition.setToValue(1.0);  
    fadeInTransition.play();  
}  
}
```

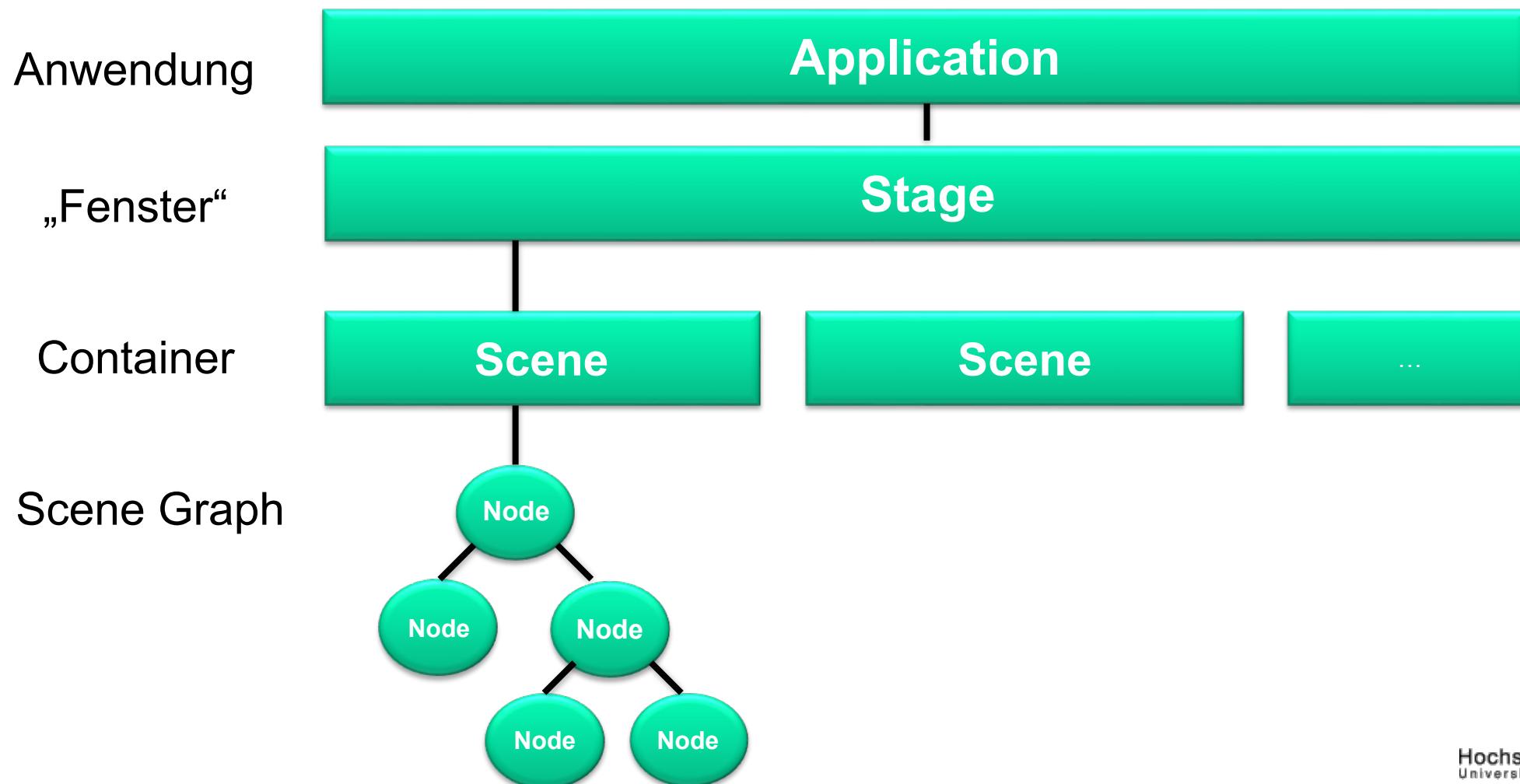


Agenda

1. Entwurf und Layout grafischer Oberflächen (25 Minuten)
2. Überblick JavaFX (25 Minuten)
3. Komponenten und Layout mit JavaFX (25 Minuten)



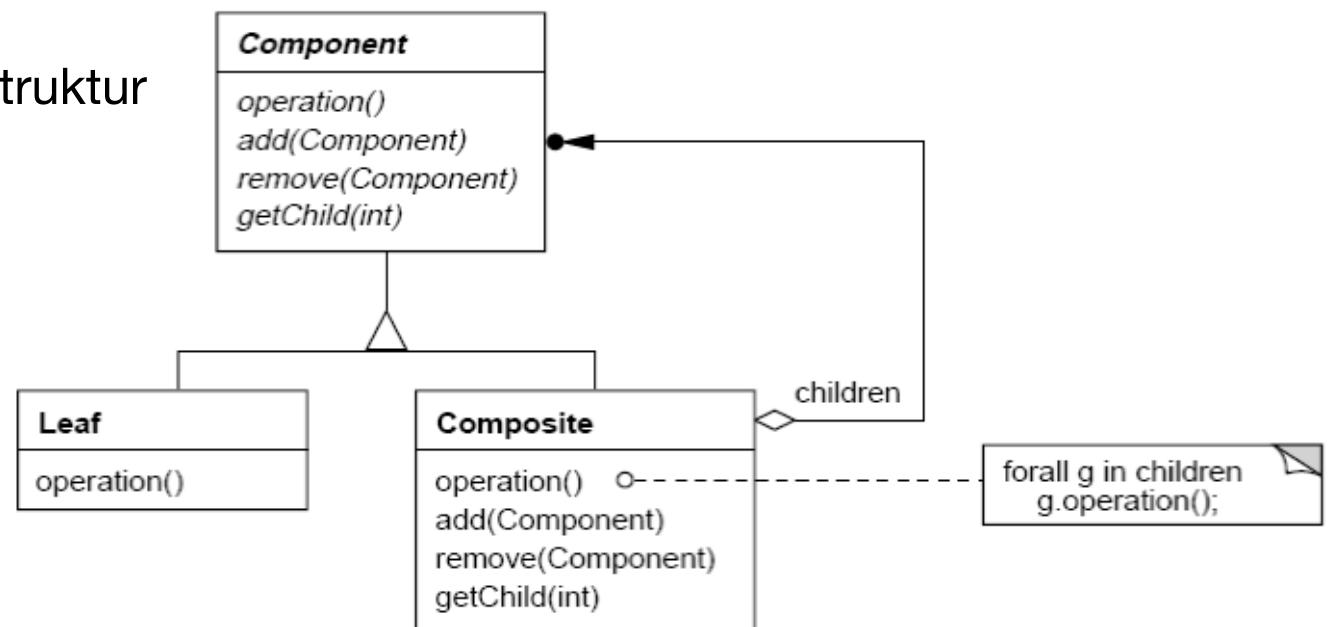
Aufbau einer JavaFX Anwendung



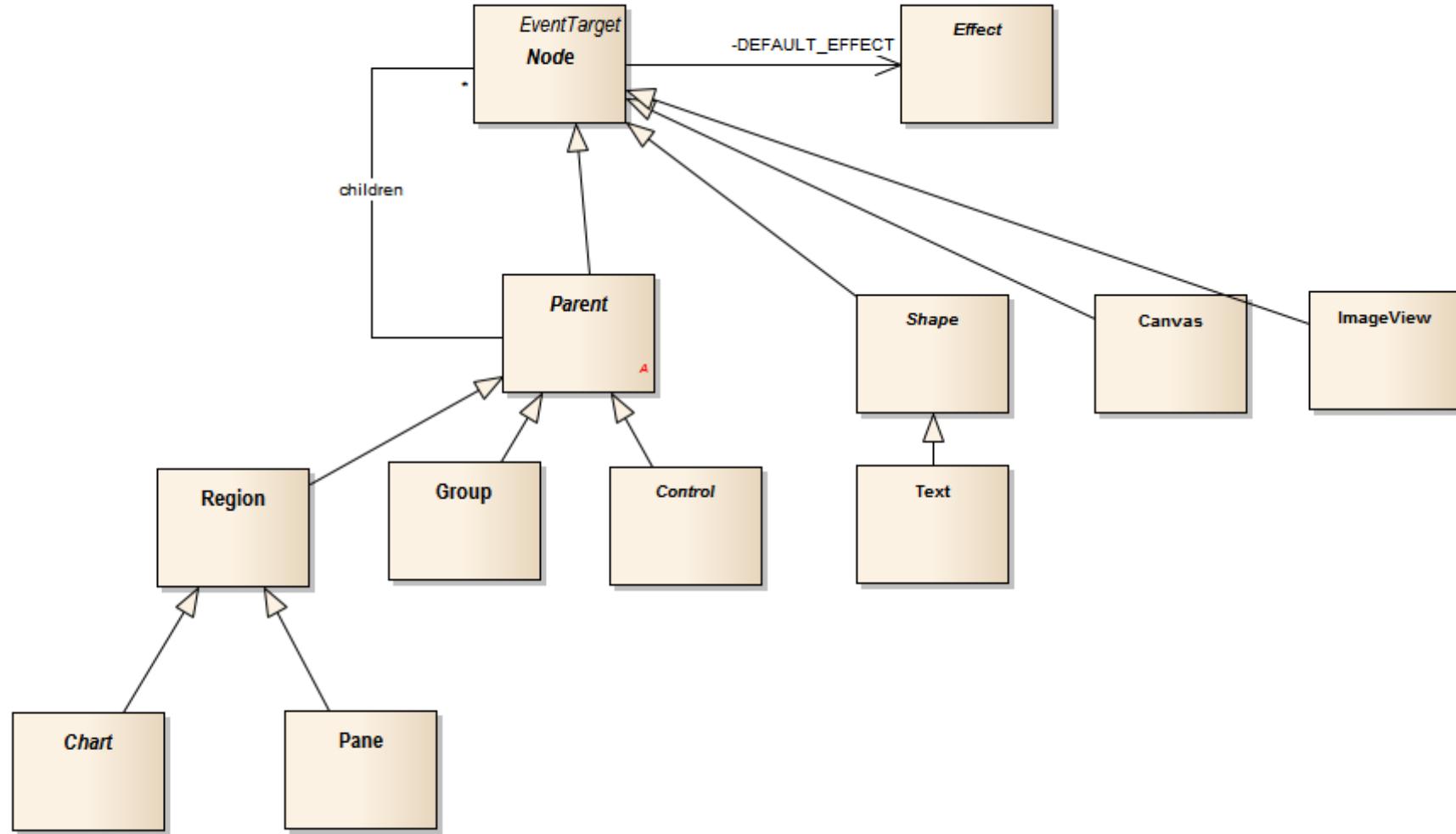


Exkurs: Composite Pattern

- Name: Kompositum (*Composite*)
- Ziel: “*Treat individual objects and multiple, recursively-composed objects uniformly.*”
- Anwendbarkeit:
 - rekursive Objekt-Kompositions-Struktur
 - einheitliche Verarbeitung von einzelnen Objekten und **Objekt-Hierarchien**



Die JavaFX Klassenhierarchie. Wo ist das Composite Pattern?



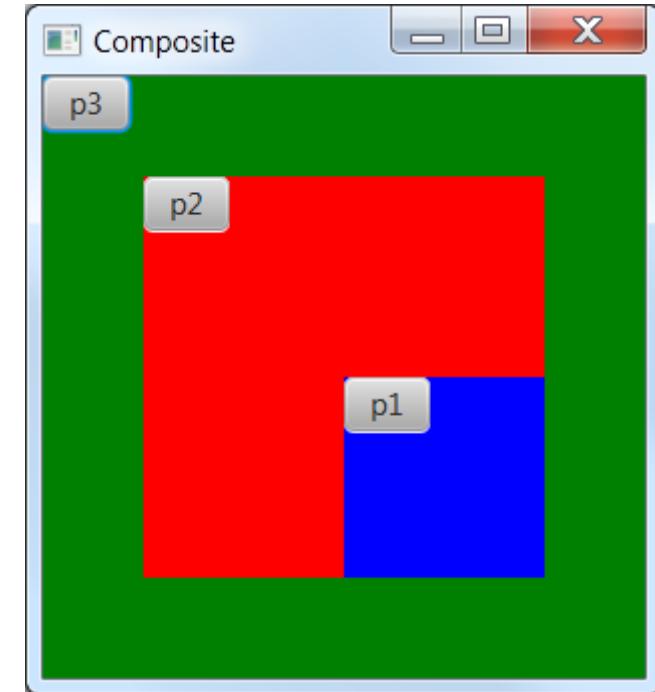
Beispiel für Komposition

```
Pane p1 = new Pane();
p1.setStyle("-fx-background-color: blue;");
p1.setPrefSize(100, 100);
p1.getChildren().add(new Button("p1"));
p1.setLayoutX(100);
p1.setLayoutY(100);

Pane p2 = new Pane();
p2.setStyle("-fx-background-color: red;");
p2.setPrefSize(200, 200);
p2.getChildren().add(p1);
p2.getChildren().add(new Button("p2"));
p2.setLayoutX(50);
p2.setLayoutY(50);

Pane p3 = new Pane();
p3.setStyle("-fx-background-color: green;");
p3.setPrefSize(300, 300);
p3.getChildren().add(p2);
p3.getChildren().add(new Button("p3"));

primaryStage.setScene(new Scene(p3, 300, 300));
primaryStage.show();
```



Layout Management

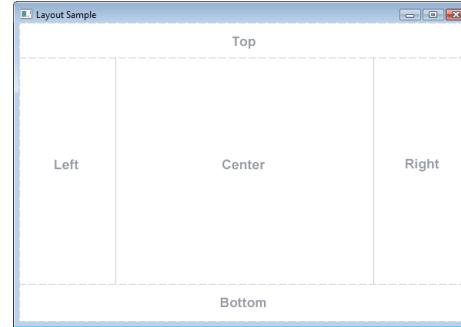
Layout Management erfolgt mit Panes

Panes sind Container die Ihre Kinder dynamisch anordnen

Die wichtigsten Panes sind:

- BorderPane
- HBox
- VBox
- StackPane
- GridPane
- FlowPane
- TilePane
- AnchorPane

Anordnung am Rand
Horizontale Anordnung
Vertikale Anordnung
Überlappende Anordnung (mehrere Panes)
erlaubt beliebig komplexe Anordnungen
Vertikale oder Horizontale Anordnung (wrap)
Ähnliche zu FlowPane für gleich große Elemente
Verankerung an der Parent-Pane



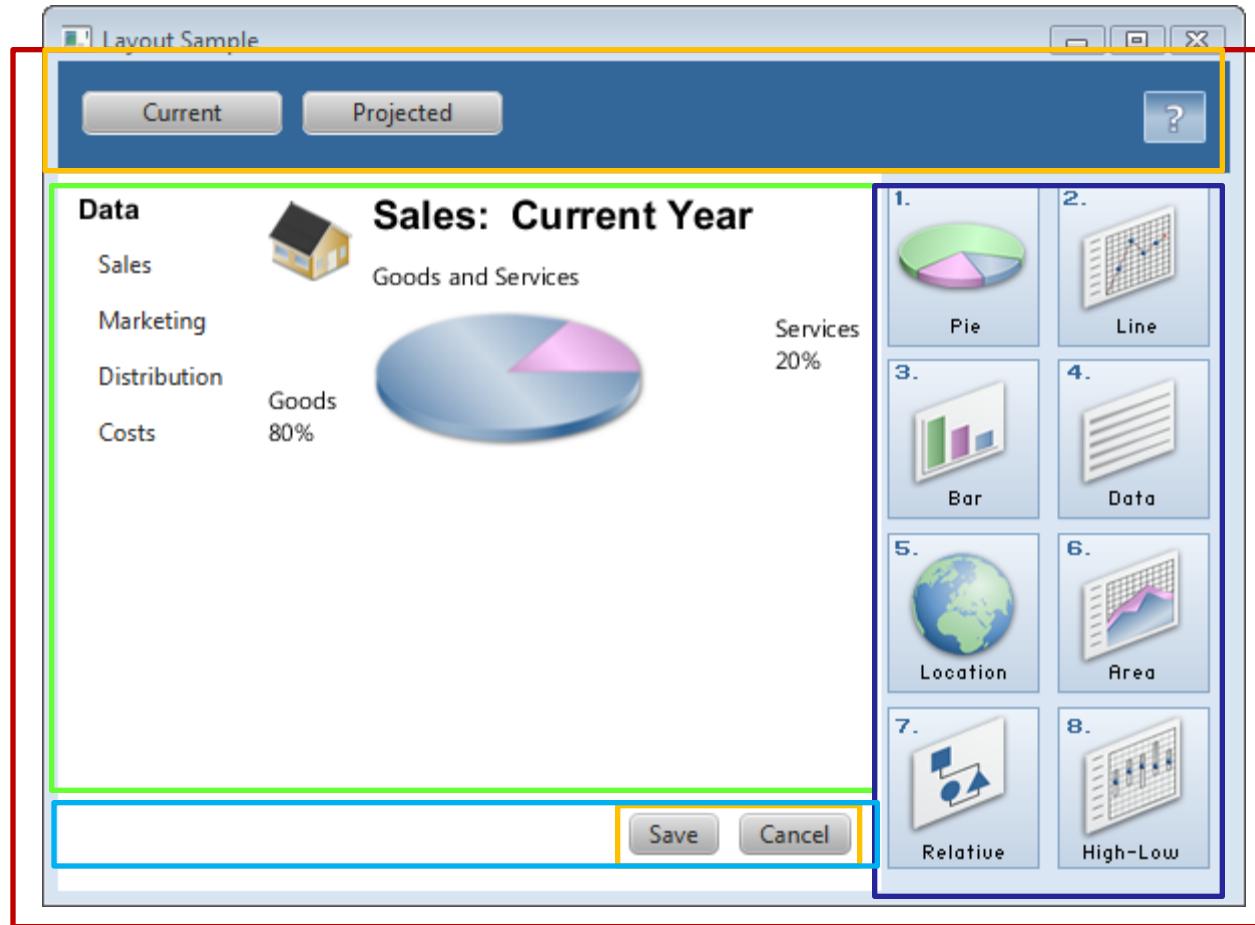
Beispiel:

```
BorderPane border = new BorderPane();  
HBox hbox = ... // create HBox  
border.setTop(hbox);
```

Hochschule Rosenheim
University of Applied Sciences



Der Schlüssel zum dynamischen Layout liegt in der Komposition von Layoutcontainern



BorderPane

FlowPane

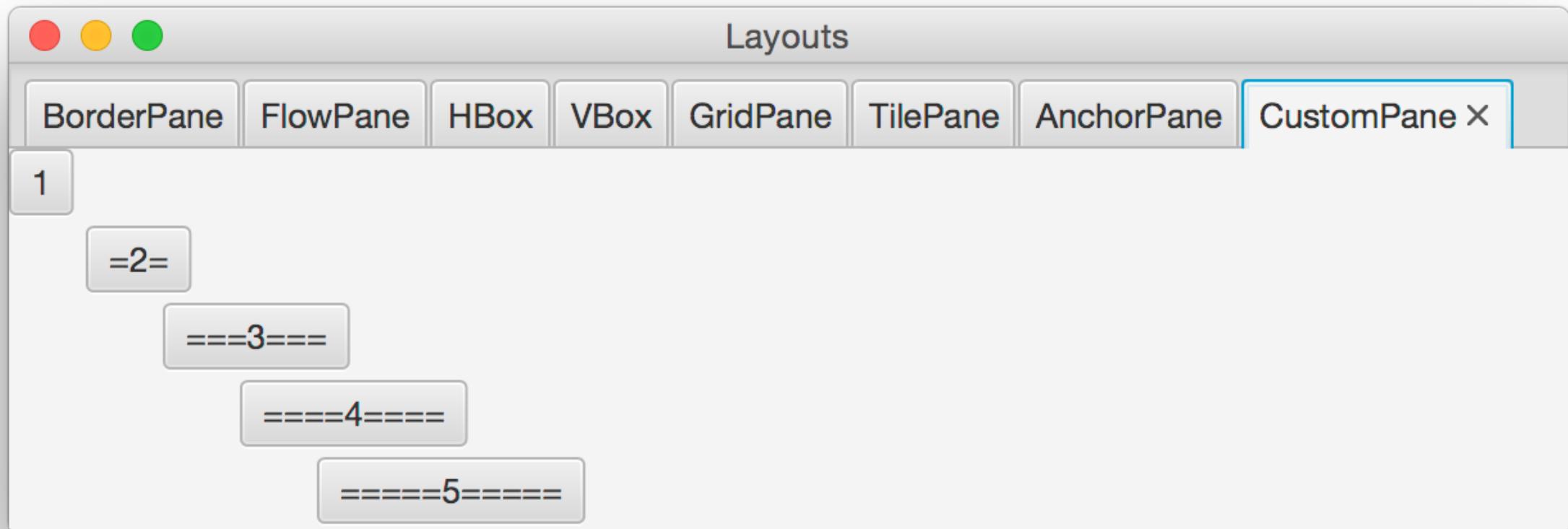
GridPane

HBox

AnchorPane



Demo



<http://teamserver.fh-rosenheim.de/svn/gui/trunk/codebase/code/v2-layout>



Designer stellen Entwickler vor Herausforderungen. Kann man das mit JavaFX bauen?

The screenshot shows a JavaFX application window with a dark theme. At the top, there is a header bar with the number '123' on the left, followed by language and help buttons ('Hilfe aus', 'Deutsch', 'Berater'). Below the header is a search bar with a placeholder 'Recherche starten' and a magnifying glass icon. To the right of the search bar are navigation icons for back, forward, and information, along with a document icon with a '7' notification.

The main content area displays a vehicle detail page. On the left, a sidebar menu lists categories: Recherche, Arbeitswerte, Befunde, Instandsetzungen, Pakete, PuMA-Maßnahmen, Reparaturmaßnahmen, and Teile. Under 'Fahrzeugdetails', the 'Fahrzeugdetails' option is highlighted with a mouse cursor icon. Other options include Technische Aktionen, Service Vorträge, Fehlinformationen, and Werkstatt/Service Beratung.

The central part of the screen shows a red BMW 320i Tour (VIN: A454367) with the text 'Fahrzeug mit Fahrgestellnummer (VIN): **BMW 320i, 160 KW**'. To the right of the car image are technical specifications:

Entwicklungsbezeichnung	E91
Farbe	Estoril Rot metallic
I-Stufe ist	E89X-05-09-521
Erstzulassung	31. Oktober 2012

Below this is a section titled 'Fahrzeugdetails' with the following table:

Fahrzeugart PKW Marke	BMW
Modell	320i
Karosserie	Tour
Fahrgestellnummer	A454367
Fahrgestellnummer (17-Stellen)	WBAVB13566KX62000

