Karsten Lentzsch – JGoodies

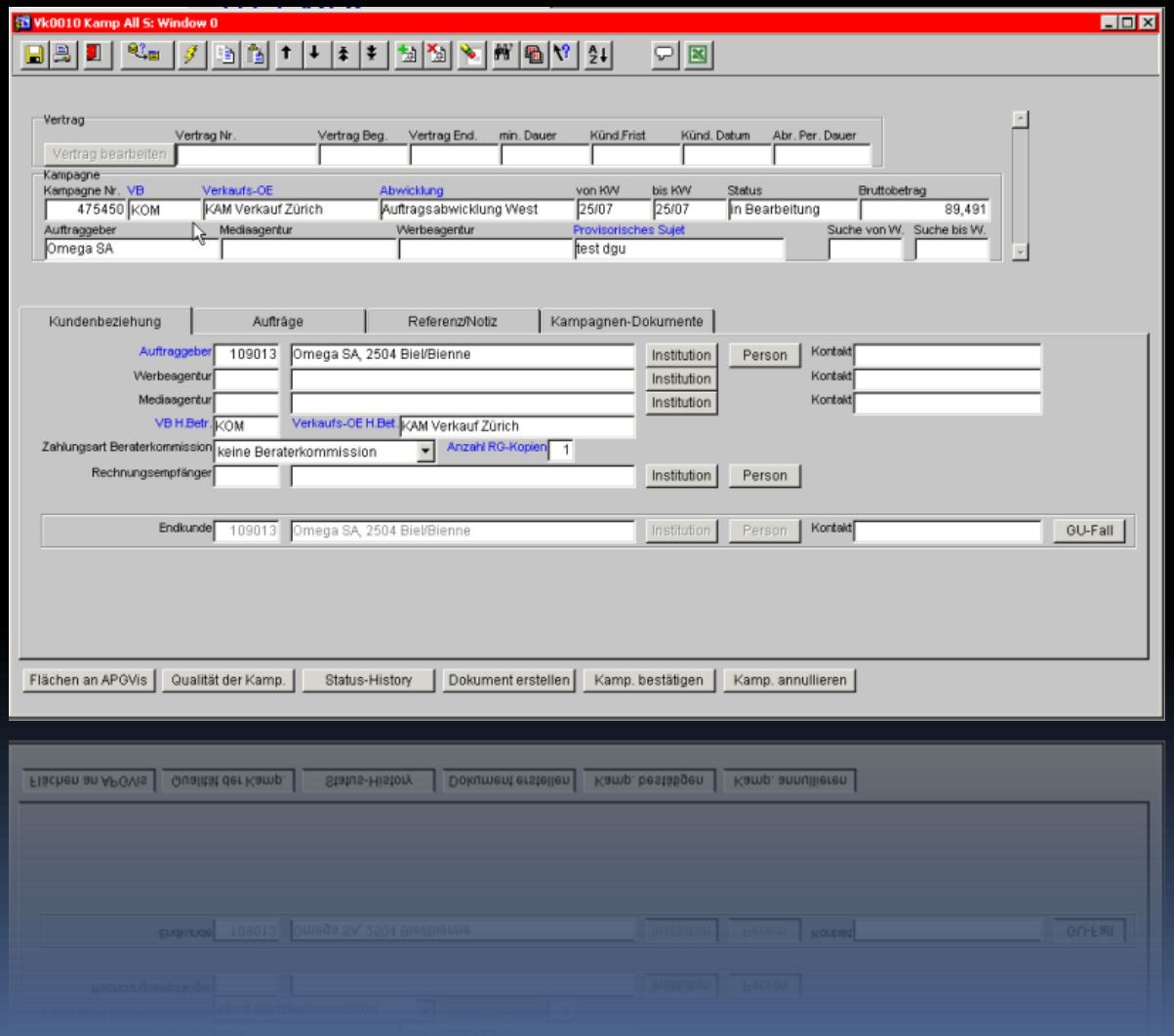
JAVA UI DESIGN WITH STYLE

JGoodies: Karsten Lentzsch

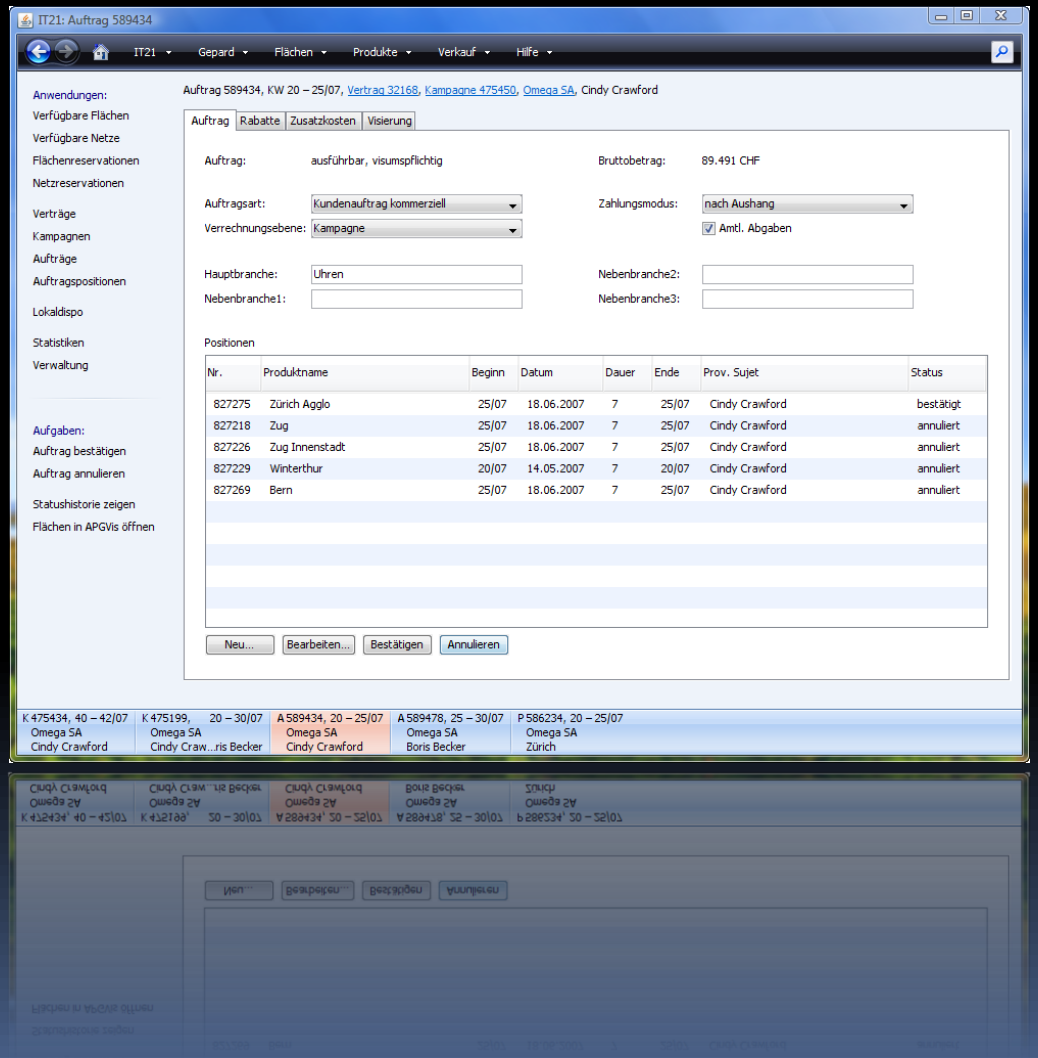
- Quelloffene Swing-Bibliotheken
- Beispielanwendungen
- Berate zu Java-Desktop
- Helfe beim Oberflächen-Bau
- Betreue und schule Teams

- Studiere GUI-Produktionsprozess

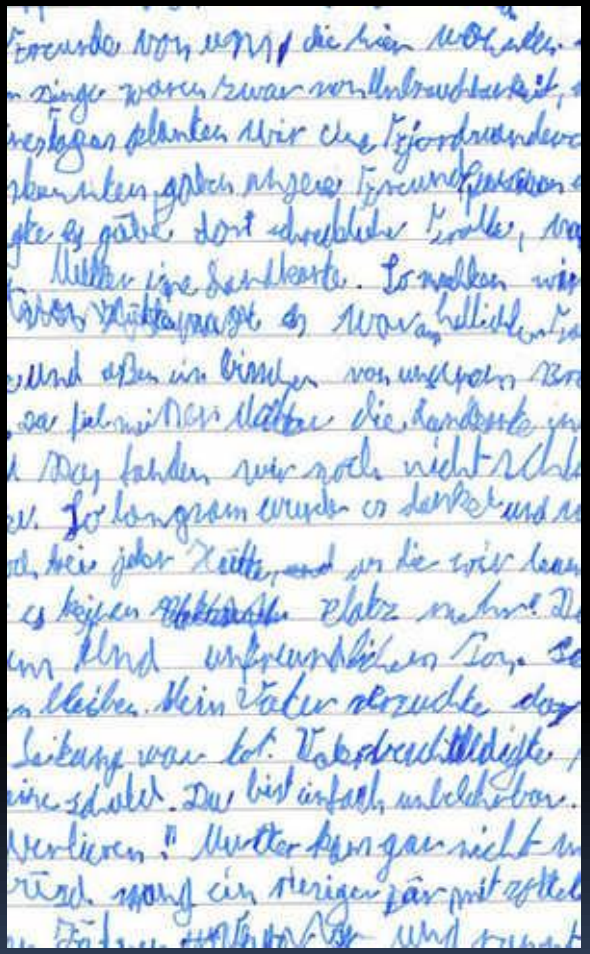
Vorher


Vk0010 Kamp All S: Window 0
Vertrag
Vertrag Nr.
Vertrag Beg.
Vertrag End.
min. Dauer
Künd.Frist
Künd. Datum
Abr. Per. Dauer
Vertrag bearbeiten
Kampagne
Kampagne Nr.
VB
Verkaufs-OE
Abwicklung
von KW
bis KW
Status
Bruttobetrag
475450
KOM
KAM Verkauf Zürich
Auftragsabwicklung West
25/07
25/07
in Bearbeitung
89,491
Auftraggeber
Mediaagentur
Werbeagentur
Provisorisches Sujet
Suche von W.
Suche bis W.
Omega SA
test dgu
Kundenbeziehung
Aufträge
Referenz/Notiz
Kampagnen-Dokumente
Auftraggeber
109013
Omega SA, 2504 Biel/Bienne
Institution
Person
Kontakt
Werbeagentur
Mediaagentur
VB H.Betr.
KOM
Verkaufs-OE H.Bet.
KAM Verkauf Zürich
Zahlungsart Beraterkommission
keine Beraterkommission
Anzahl RG-Kopien
1
Rechnungsempfänger
Endkunde
GU-Fall
Flächen an APGVis
Qualität der Kamp.
Status-History
Dokument erstellen
Kamp. bestätigen
Kamp. annullieren

Nachher


IT21: Auftrag 589434
IT21
Gepard
Flächen
Produkte
Verkauf
Hilfe
Anwendungen:
Verfügbare Flächen
Verfügbare Netze
Flächenreservationen
Netzreservationen
Verträge
Kampagnen
Aufträge
Auftragspositionen
Lokaldispo
Statistiken
Verwaltung
Aufgaben:
Auftrag bestätigen
Auftrag annulieren
Statushistorie zeigen
Flächen in APGVis öffnen
Auftrag 589434, KW 20 – 25/07, Vertrag 32168, Kampagne 475450, Omega SA, Cindy Crawford
Auftrag
Rabatte
Zusatzkosten
Visierung
Auftrag:
ausführbar, visumspflichtig
Bruttobetrag:
89.491 CHF
Auftragsart:
Kundenauftrag kommerziell
Zahlungsmodus:
nach Aushang
Verrechnungsebene:
Kampagne
Amtl. Abgaben
Hauptbranche:
Uhren
Nebenbranche2:
Nebenbranche1:
Nebenbranche3:
Positionen
Nr.
Produktname
Beginn
Datum
Dauer
Ende
Prov. Sujet
Status
827275 Zürich Agglo 25/07 18.06.2007 7 25/07 Cindy Crawford bestätigt
827218 Zug 25/07 18.06.2007 7 25/07 Cindy Crawford annuliert
827226 Zug Innenstadt 25/07 18.06.2007 7 25/07 Cindy Crawford annuliert
827229 Winterthur 20/07 14.05.2007 7 20/07 Cindy Crawford annuliert
827269 Bern 25/07 18.06.2007 7 25/07 Cindy Crawford annuliert
Neu...
Bearbeiten...
Bestätigen
Annulieren
K 475434, 40 – 42/07 Omega SA Cindy Crawford
K 475199, 20 – 30/07 Omega SA Cindy Craw...ris Becker
A 589434, 20 – 25/07 Omega SA Cindy Crawford
A 589478, 25 – 30/07 Omega SA Boris Becker
P 586234, 20 – 25/07 Omega SA Zürich

Vorher



Nachher

MADAM, After her six years' residence at the Mall, I have the honour and happiness of presenting Miss Amelia Sedley to her parents, as a young lady not unworthy to occupy a fitting position in their polished and refined circle. Those virtues which characterize the young English gentlewoman, those accomplishments which become her birth and station, will not be found wanting in the amiable Miss Sedley, whose INDUSTRY and OBEDIENCE have endeared her to her instructors, and whose delightful sweetness of temper has charmed her aged and her youthful companions.

Ziele

Erfolgsfaktoren für die Client-Seite sehen

Wie soll ich Client-Code schreiben?

Kriterien für guten Schreibstil lernen

Gliederung

Visuelle Gebote und Verbote

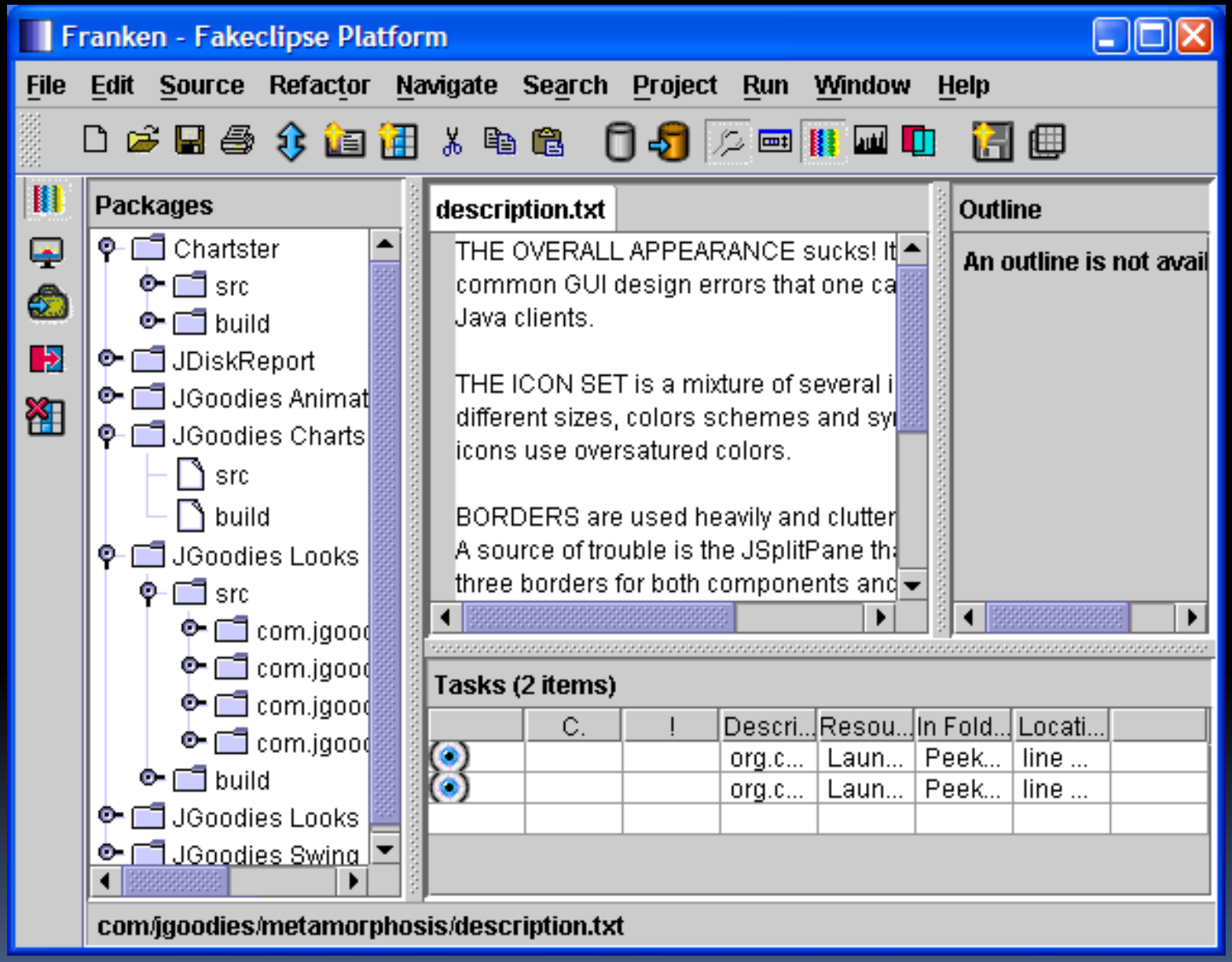
Implementierungs-Muster

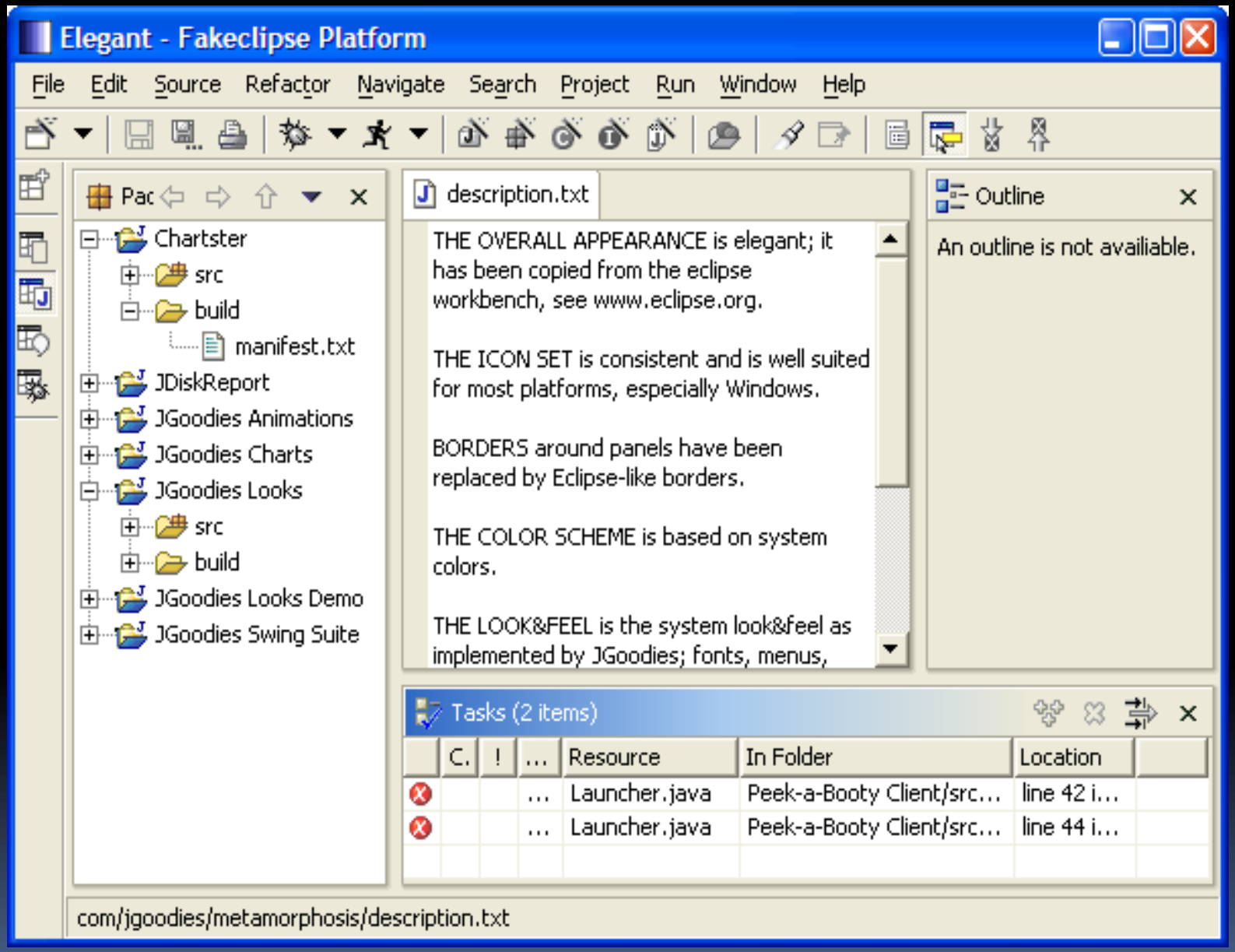
Schreibstil

Visuelle Muster

Visuelle Verbote

- Pfusche nicht mit Farben!
- Pfusche nicht mit Schriften!
- Sei vorsichtig mit Icons!


Franken - Fakeclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Packages
Chartster
src
build
JDiskReport
JGoodies Animat
JGoodies Charts
src
build
JGoodies Looks
src
com.jgooc
com.jgooc
com.jgooc
com.jgooc
build
JGoodies Looks
JGoodies Swing
description.txt
THE OVERALL APPEARANCE sucks! It
common GUI design errors that one ca
Java clients.
THE ICON SET is a mixture of several i
different sizes, colors schemes and sy
icons use oversatured colors.
BORDERS are used heavily and clutter
A source of trouble is the JSplitPane th
three borders for both components and
Outline
An outline is not avail
Tasks (2 items)
C.
!
Descri...
Resou...
In Fold...
Locati...
org.c...
Laun...
Peek...
line ...
org.c...
Laun...
Peek...
line ...
com/jgoodies/metamorphosis/description.txt


Elegant - Fakeclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Pac
Chartster
src
build
manifest.txt
JDiskReport
JGoodies Animations
JGoodies Charts
JGoodies Looks
src
build
JGoodies Looks Demo
JGoodies Swing Suite
description.txt
THE OVERALL APPEARANCE is elegant; it has been copied from the eclipse workbench, see www.eclipse.org.
THE ICON SET is consistent and is well suited for most platforms, especially Windows.
BORDERS around panels have been replaced by Eclipse-like borders.
THE COLOR SCHEME is based on system colors.
THE LOOK&FEEL is the system look&feel as implemented by JGoodies; fonts, menus,
Outline
An outline is not availiable.
Tasks (2 items)
C. ! ... Resource In Folder Location
... Launcher.java Peek-a-Booty Client/src... line 42 i...
... Launcher.java Peek-a-Booty Client/src... line 44 i...
com/jgoodies/metamorphosis/description.txt

Visuelle Gebote

- Richte aus
- Entferne unnötige Rahmen
- Entferne alle unnötigen Elemente

- Verwende Leerraum
- Lerne den Umgang mit Kontrast und Balance

Oracle Forms Runtime - [Institutionen verwalten]

Datei Bearbeiten Gepard Basisdaten Flaechenmanagement Produktmanagement Verkauf Traffic Logistik System Fenster Hilfe

Institution

Gepard-Nr.

Favorit für Sie
Favorit für andere MA
Hierarchie
Inst. beenden

Bezeichnung: Hans Muster AG
Interne Bezeichnung:
Bezeichnungs-Zusatz:
Sprache: Deutsch
MWST-pflichtig: Ja

Verkaufspartner: Nein
Hauptbetreuer APG:
Hauptbetreuer Montagne:

9-stellige ESR-Nr:
ESR-Nr.:
Kopien Faktura: 1
Gültig von:
Gültig bis:

Unternehmung/Enterprise

Rechtsform: Aktiengesellschaft
MWST-Nr.:
MWST 90.2.a Internat. Org.

Auftraggeber/Donneur d'ordre

Bonität: gut
Begründg.:
ABC-Kunden: A

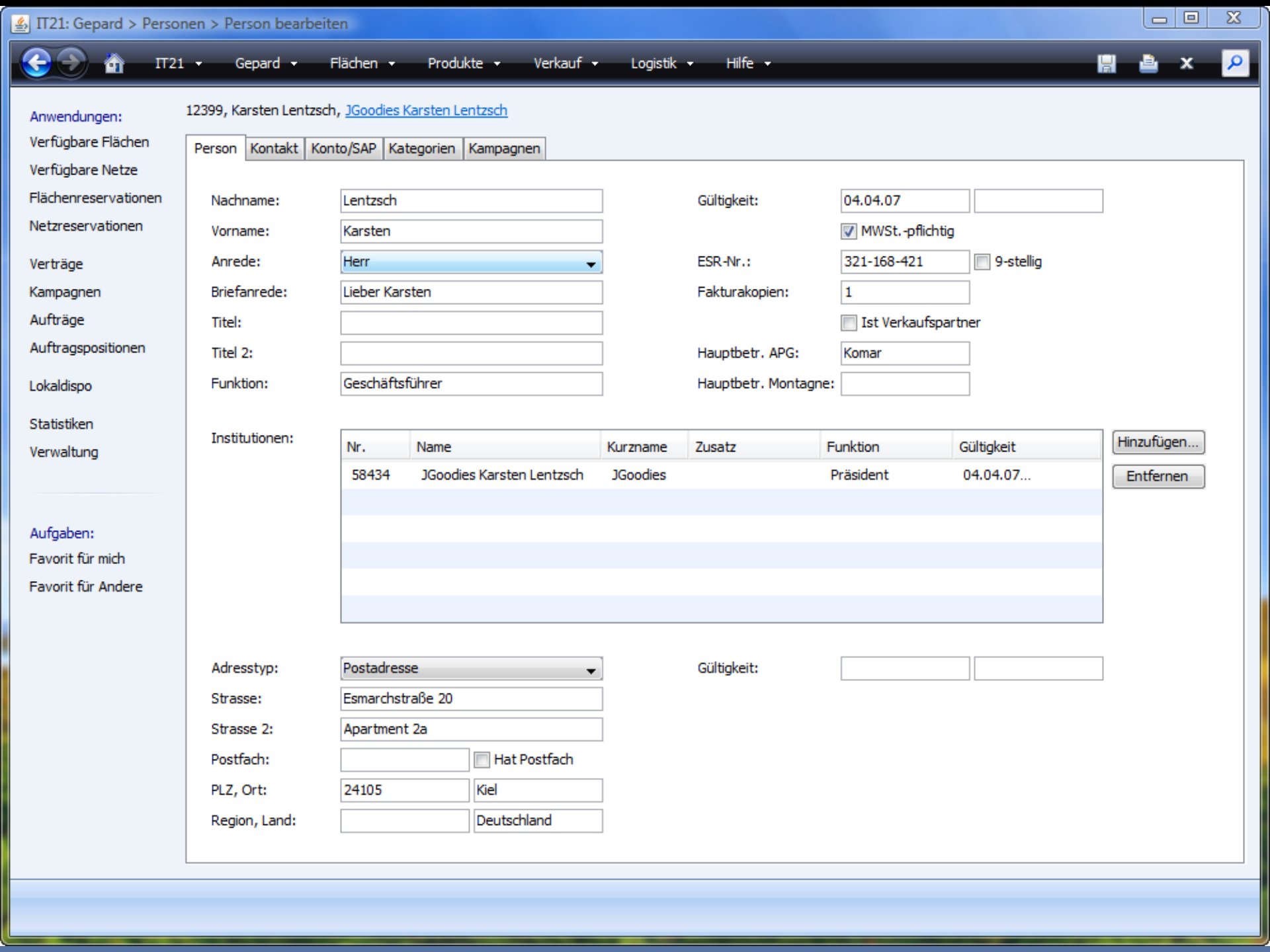
Kontakt/Personne de contact

HauptGepard-Nr.	Nachname	Vorname	Titel	Email	Direktwahl	Funktion	Gültig von	Gültig bis		
111	Mihu	Lucian							Zuordn. erstellen	Zuordn. löschen
									Zuordn. erstellen	Zuordn. löschen

Adresse | Fon/Fax | Konto | Druckerei | SAP | Ext. Logistik | Logging-Einträge | Kategorien | Ka

Adress Typ: Post-Adresse
Strasse: Blumenweg 5
Strasse 2:
Postfach: Nein
Postfach-Nr.:
PLZ: 8000
Ort: Zürich
Region:
Land: Schweiz
Gültig von:
Gültig bis:

Record: 1/1


IT21: Gepard > Personen > Person bearbeiten
IT21
Gepard
Flächen
Produkte
Verkauf
Logistik
Hilfe
Anwendungen:
Verfügbare Flächen
Verfügbare Netze
Flächenreservationen
Netzreservationen
Verträge
Kampagnen
Aufträge
Auftragspositionen
Lokaldispo
Statistiken
Verwaltung
Aufgaben:
Favorit für mich
Favorit für Andere
12399, Karsten Lentzsch, JGoodies Karsten Lentzsch
Person
Kontakt
Konto/SAP
Kategorien
Kampagnen
Nachname: Lentzsch
Vorname: Karsten
Anrede: Herr
Briefanrede: Lieber Karsten
Titel:
Titel 2:
Funktion: Geschäftsführer
Gültigkeit: 04.04.07
MWSt.-pflichtig
ESR-Nr.: 321-168-421
9-stellig
Fakturakopien: 1
Ist Verkaufspartner
Hauptbetr. APG: Komar
Hauptbetr. Montagne:
Institutionen:
Nr. Name Kurzname Zusatz Funktion Gültigkeit
58434 JGoodies Karsten Lentzsch JGoodies Präsident 04.04.07...
Hinzufügen...
Entfernen
Adresstyp: Postadresse
Gültigkeit:
Strasse: Esmarchstraße 20
Strasse 2: Apartment 2a
Postfach:
Hat Postfach
PLZ, Ort: 24105 Kiel
Region, Land: Deutschland

Visuelle Gebote und Verbote

Mehr Informationen unter:

JGoodies.com -> Downloads -> Presentations

„Effektiv gestalten mit Swing“

Gliederung

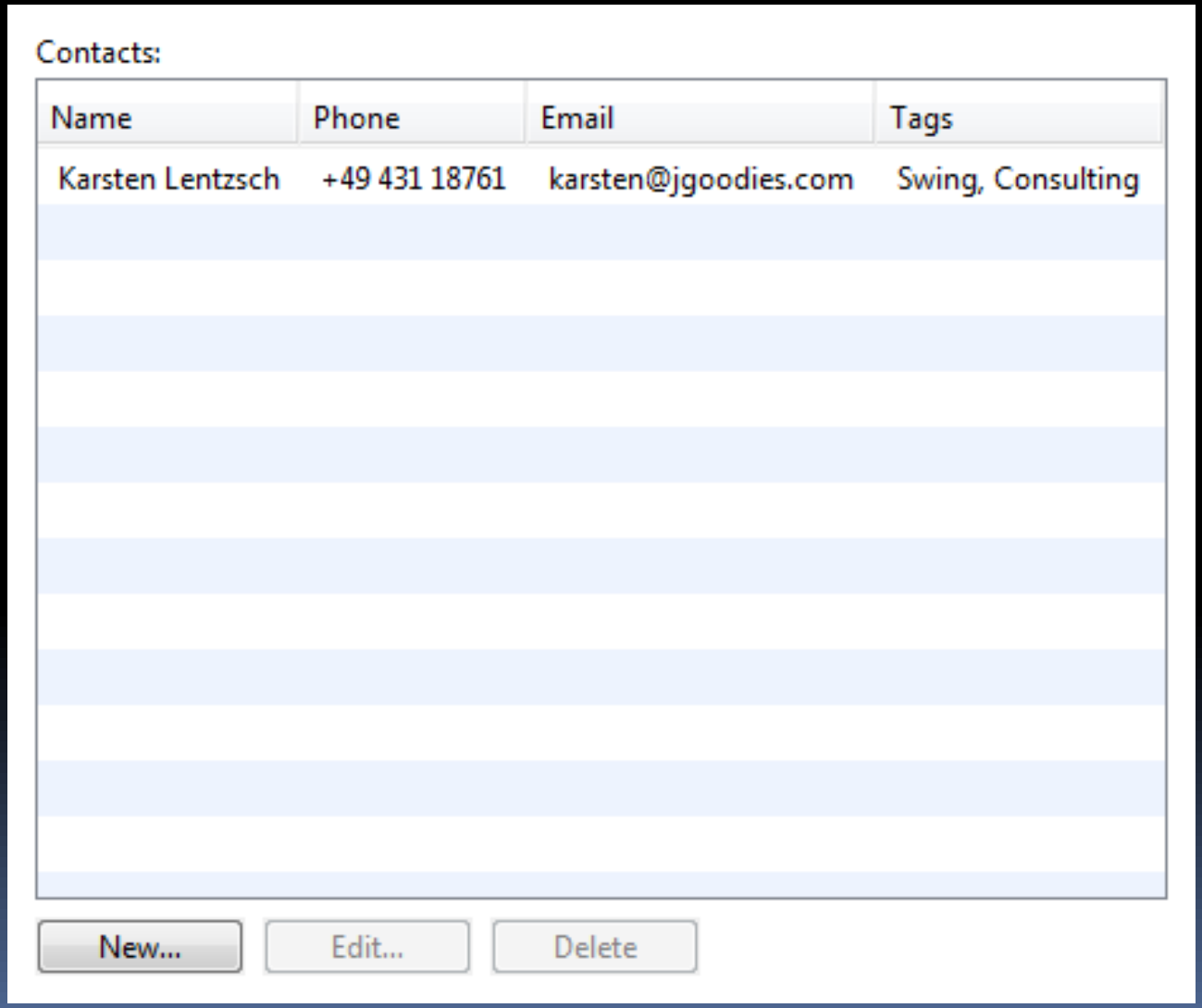
Visuelle Gebote und Verbote

Implementierungs-Muster

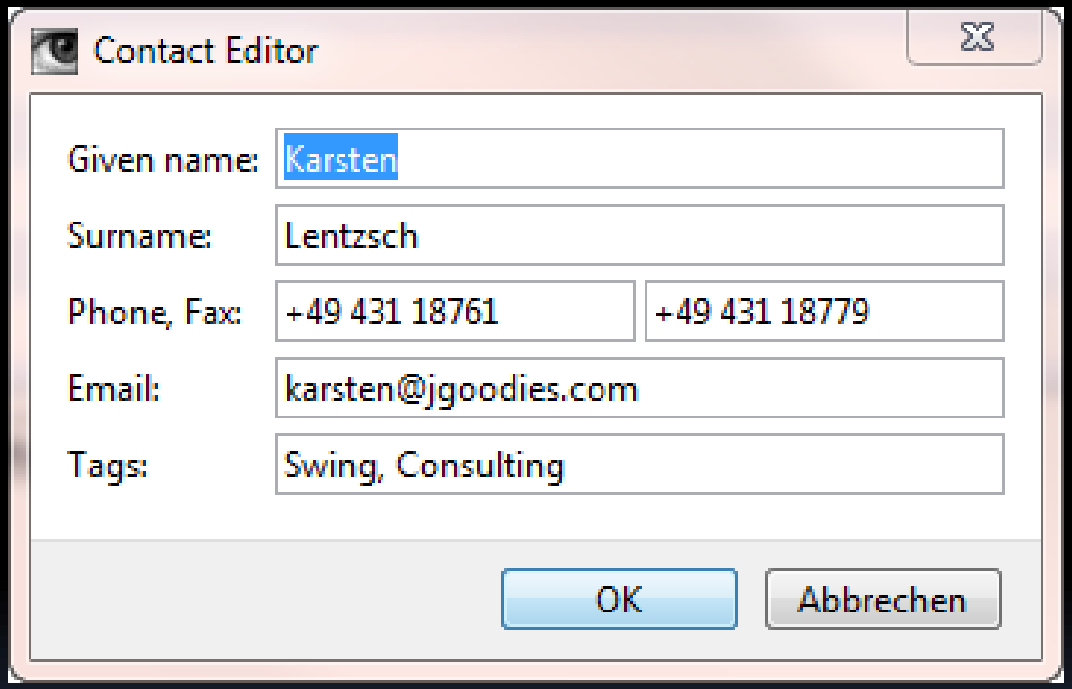
Schreibstil

Visuelle Muster

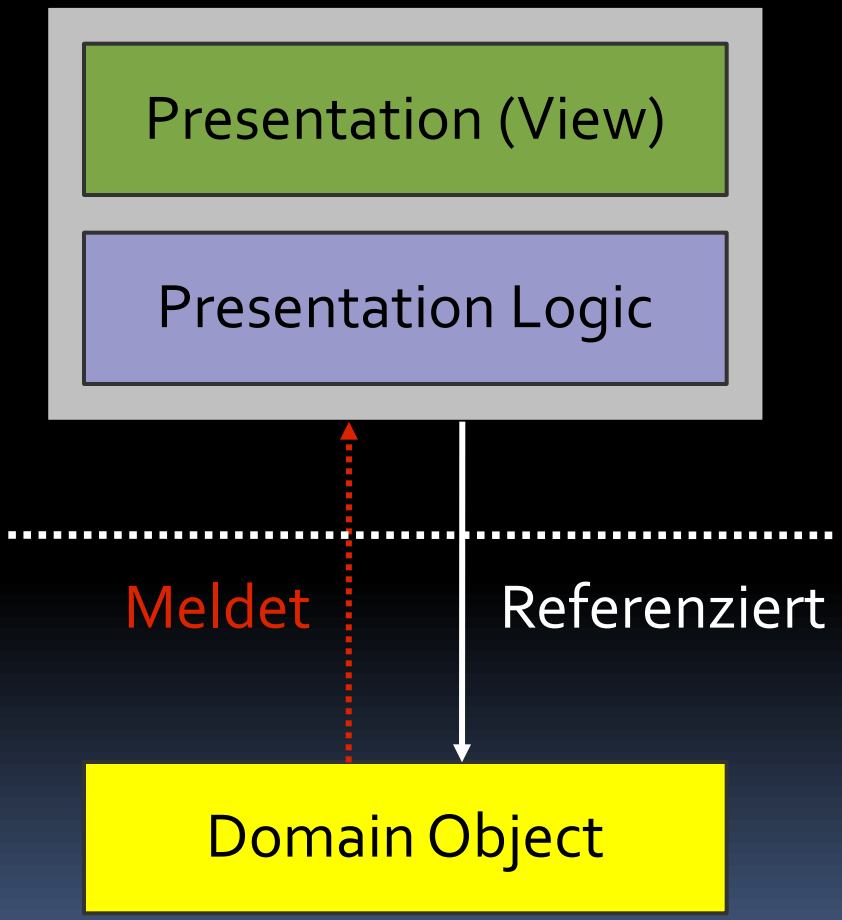
Beispiel: Kontaktverwaltung



Kontakteditor



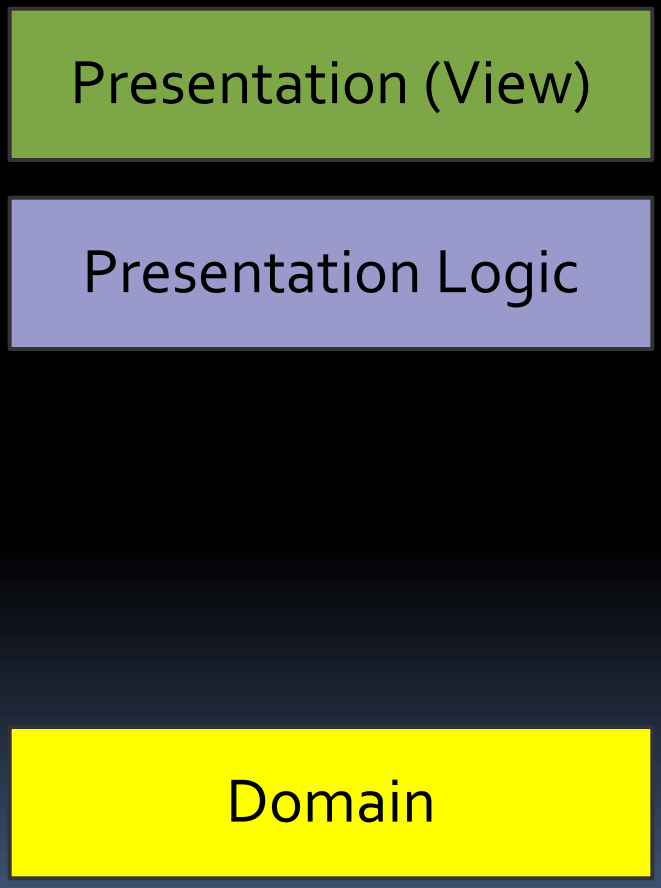
Legende


Presentation (View)
Presentation Logic
Meldet
Referenziert
Domain Object

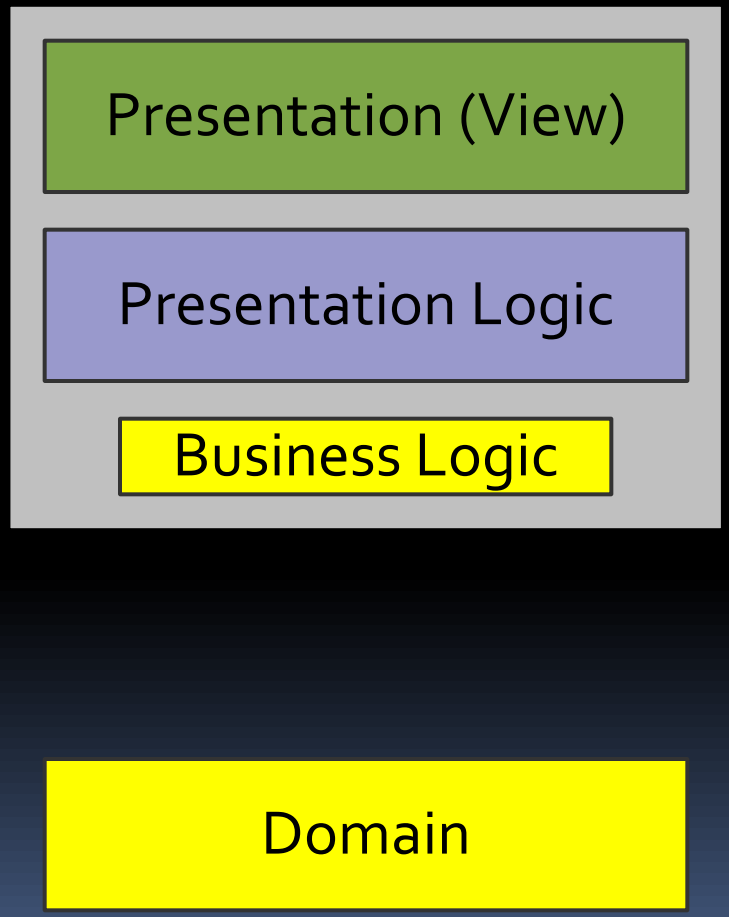
Tipp 1

- Trenne Fachdaten vom Rest

Muster: Separated Presentation



Fachlogik in der Präsentation



Tipp 1

- Trenne Fachdaten vom Rest

- Trenne Fachlogik vom Rest
- Erwäge, objektorientiert zu programmieren

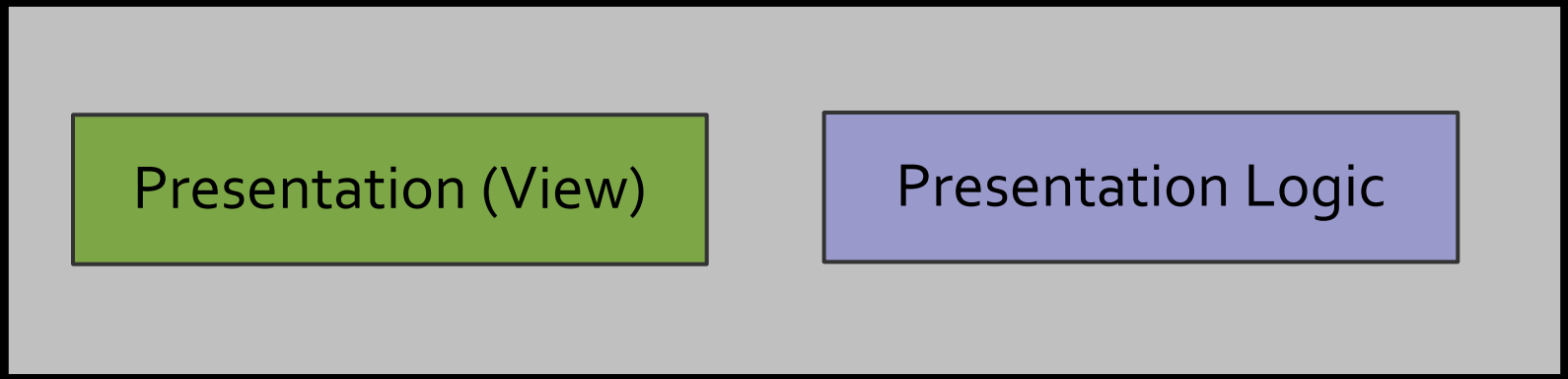
Fachklasse

```
public class Contact {

  private String givenName;
  private String surname;
  private String phone;
  …

  public String getPhone() {
    return phone;
  }


  public void setPhone(String newPhone) {
    phone = newPhone;
  }
  …
```

Mit bound-Properties

```
public class Contact extends Bean {

  public static final String PROPERTY_PHONE
      = “phone”;
  …


  public void setPhone(String newPhone) {
    String oldPhone = getPhone();
    phone = newPhone;
    firePropertyChange(
        PROPERTY_PHONE, oldPhone, newPhone);
  }

  …
```

Muster: Autonomous View



Autonomous View I

```
public class ContactEditorDialog extends JDialog {

    private final Contact contact;

    private JTextField givenNameField;
    ...

    public ContactEditorDialog(Contact contact) { ... }

    private void initComponents() { ... }

    private void initEventHandling() { ... }

    private JComponent buildContent() { ... }
```

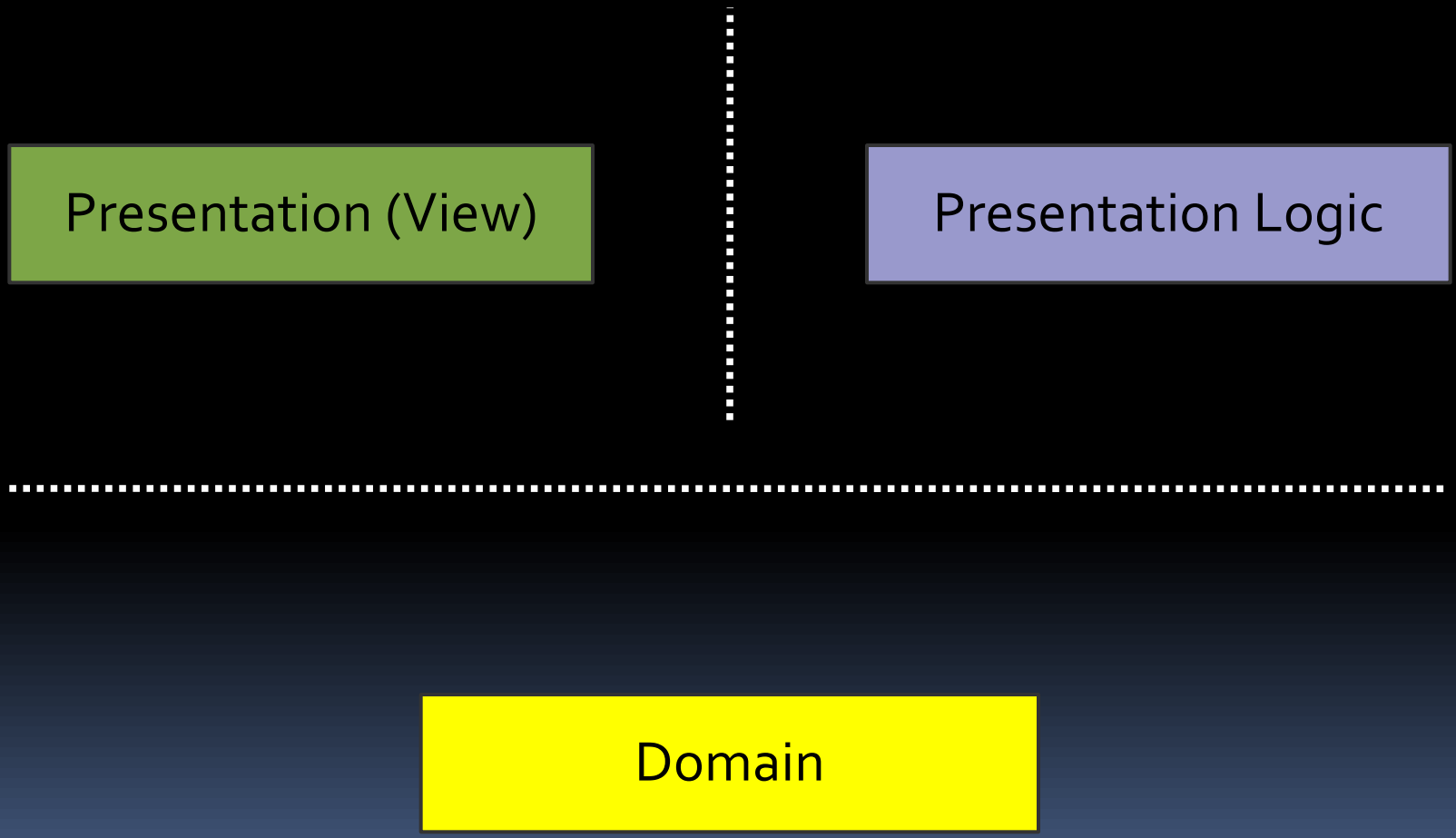
Autonomous View II

```
class OKAction extends AbstractAction {

   private OKAction() {
       super("OK");
   }

   public void actionPerformed(ActionEvent e) {
       …
   }

}
```

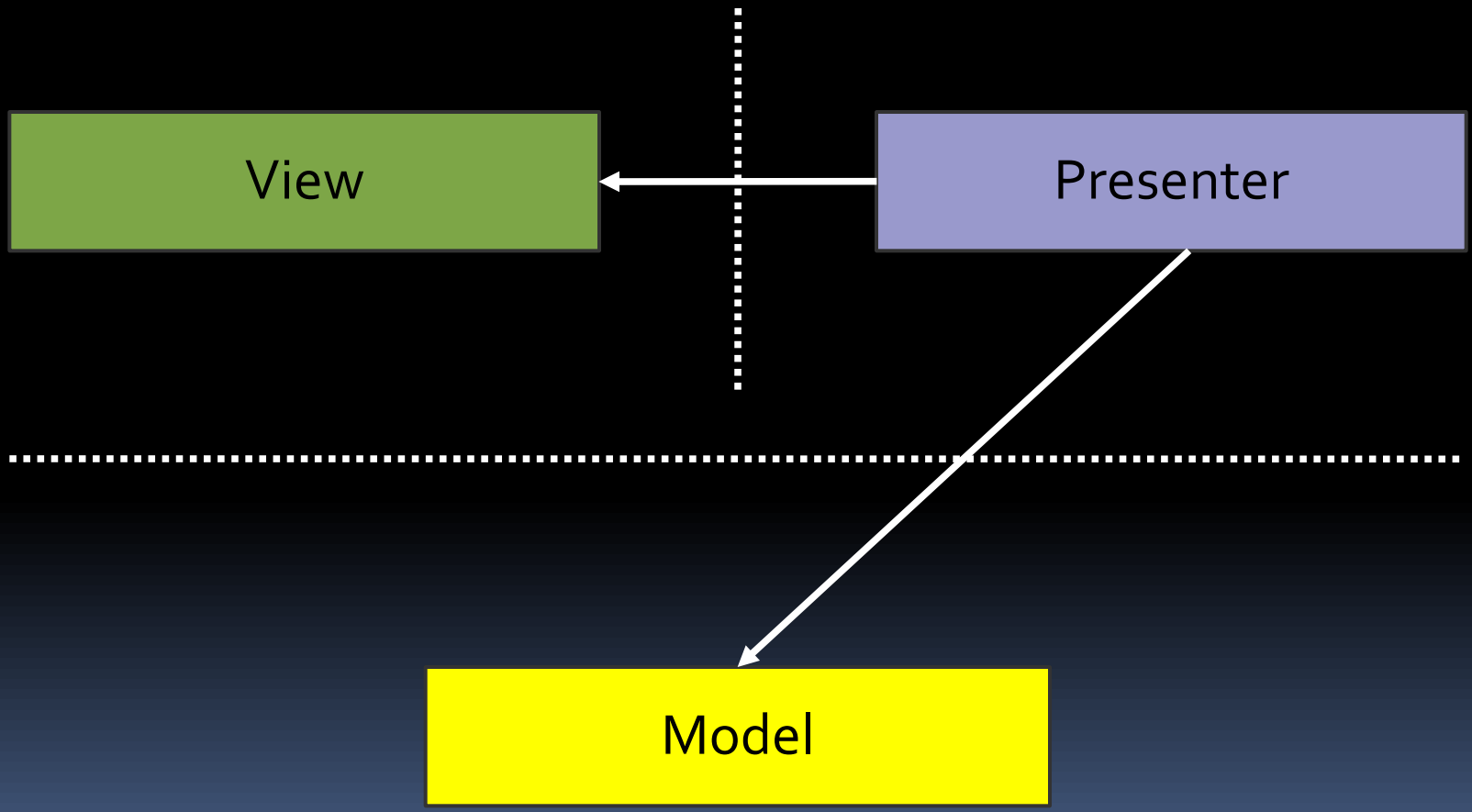
Tipp 2

- Trenne Präsentation und Präsentationslogik

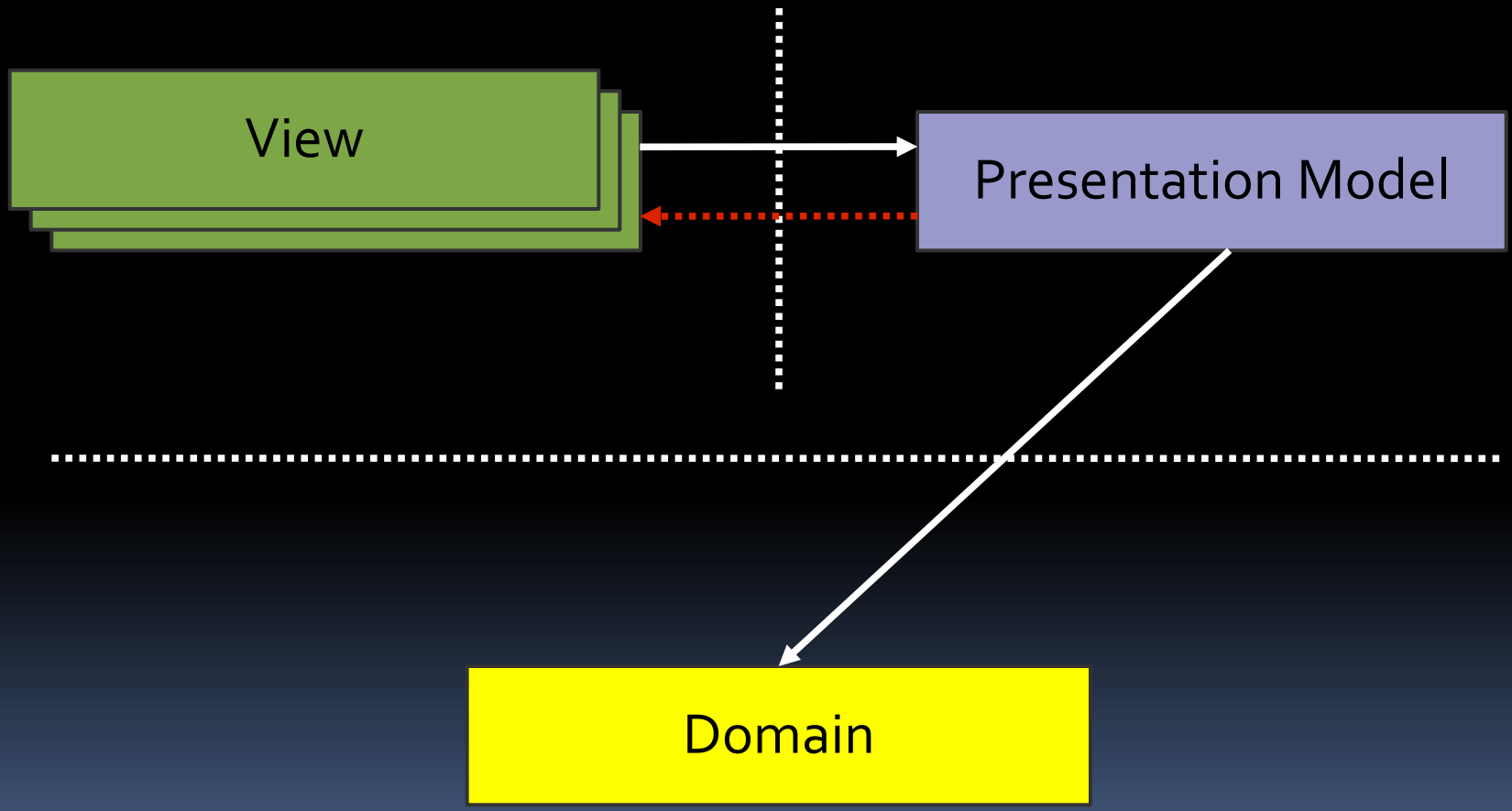
Präsentationslogik abgetrennt



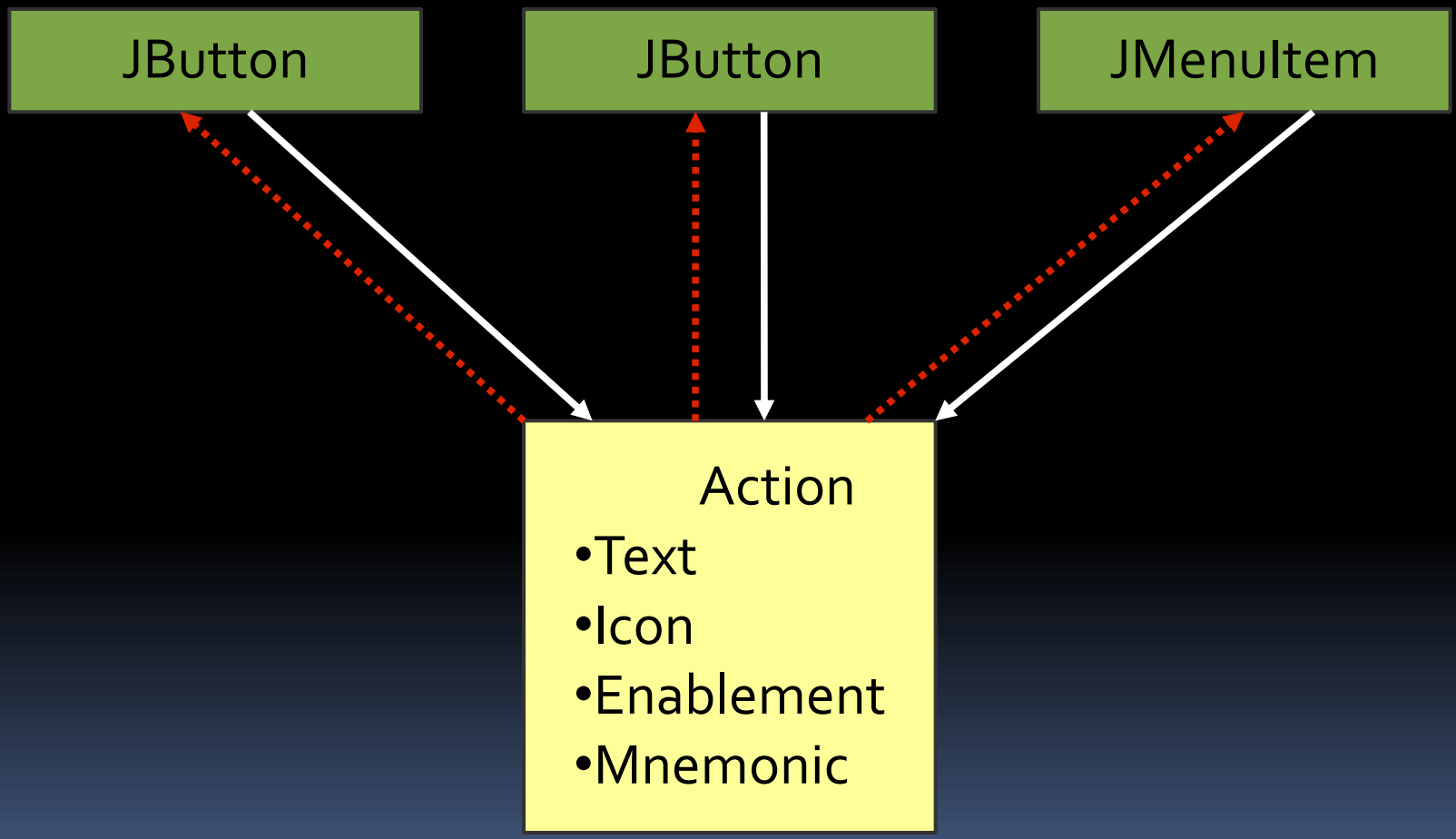
Muster: Model-View-Presenter



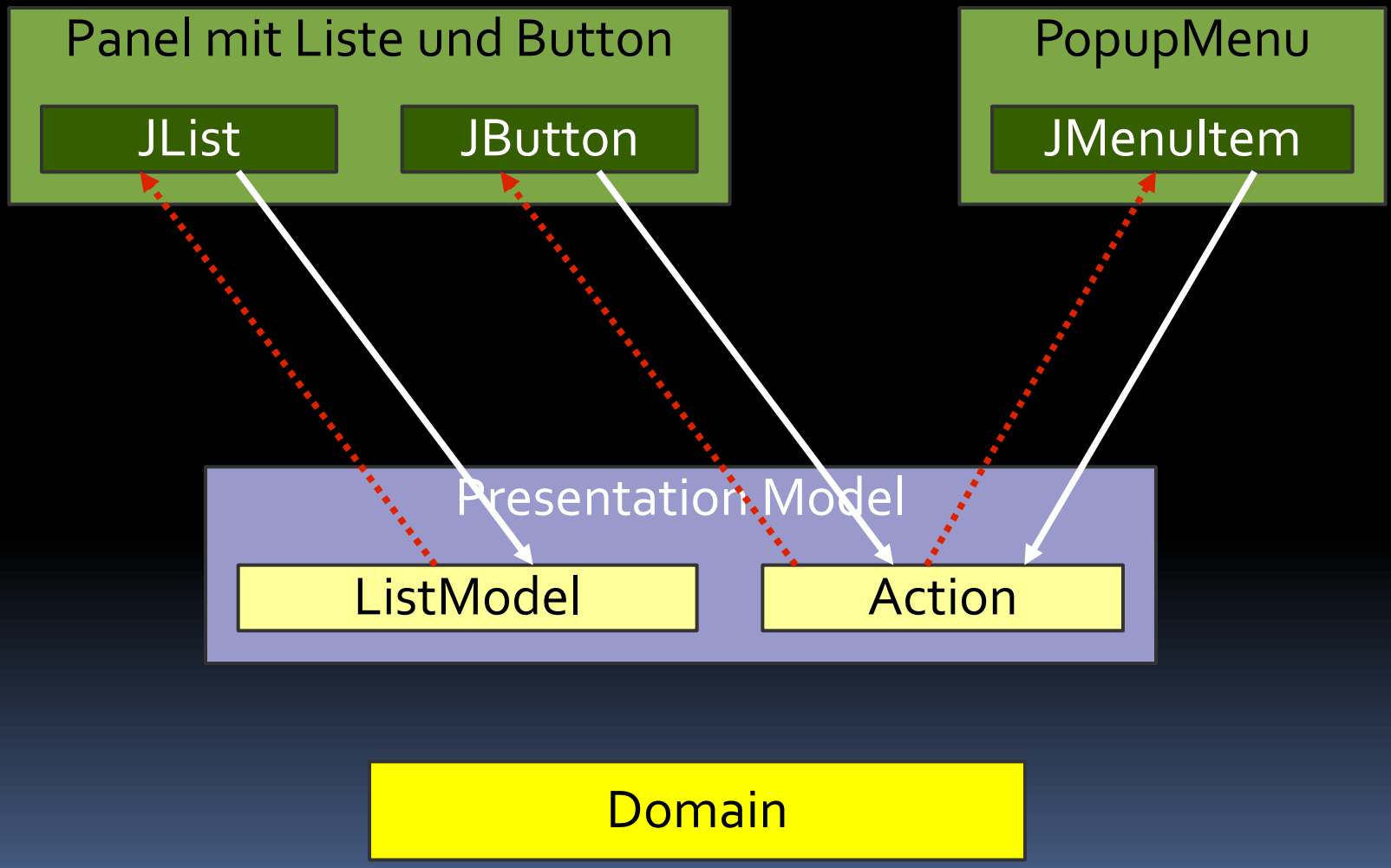
Muster: Presentation Model



Action mit mehreren Views



PM: Mehrere Views



Implementierungs-Muster

Mehr Informationen:

JGoodies.com -> Downloads -> Presentations

„Desktop-Muster und Datenbindung“

Weitere Informationen

- Martin Fowler: *Further P of EAA*
  - google "Organizing Presentation Logic"
- Chris Ramsdale:
  Large scale application development and MVP

Gliederung

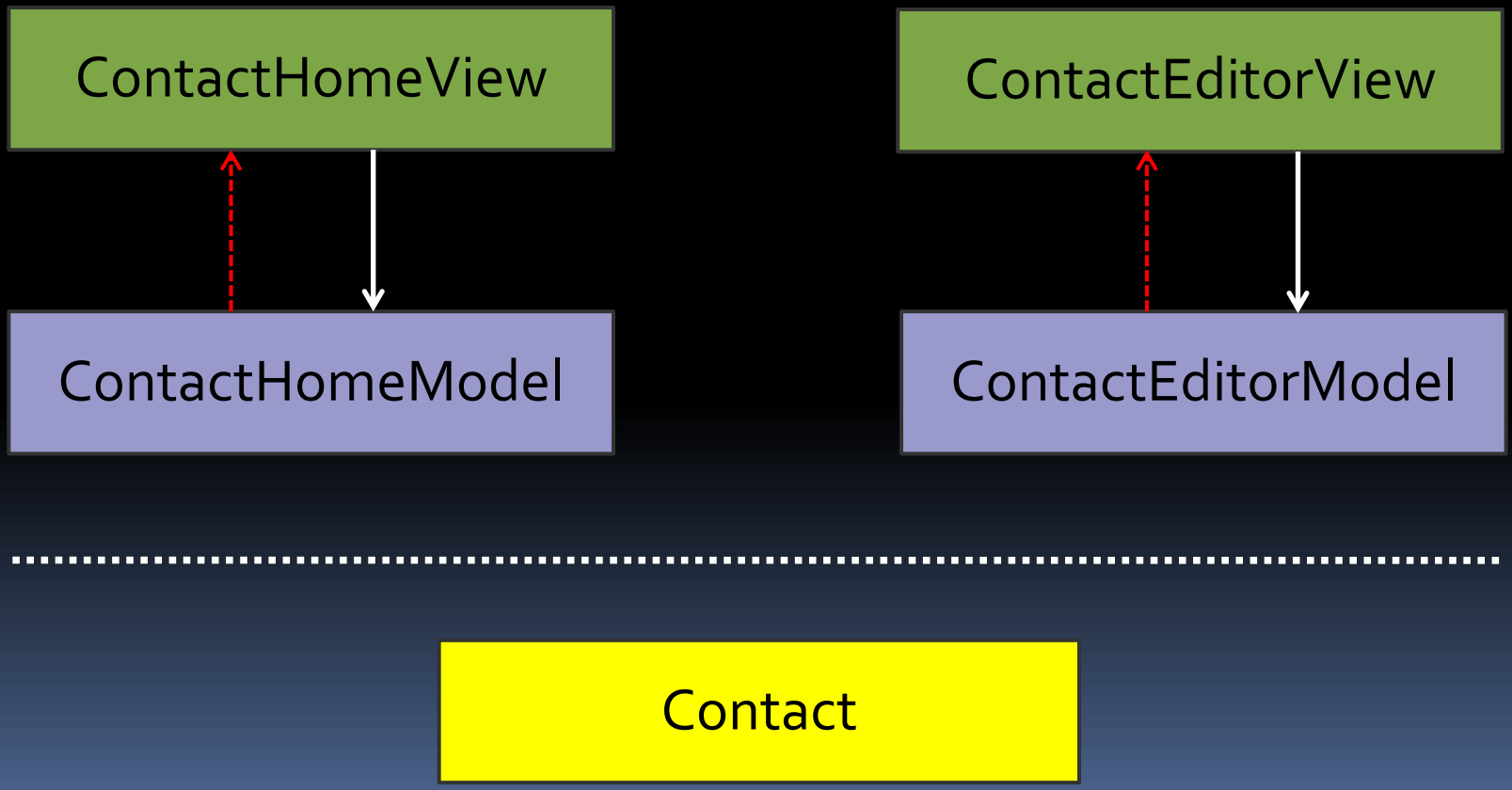
Visuelle Gebote und Verbote

Implementierungs-Muster

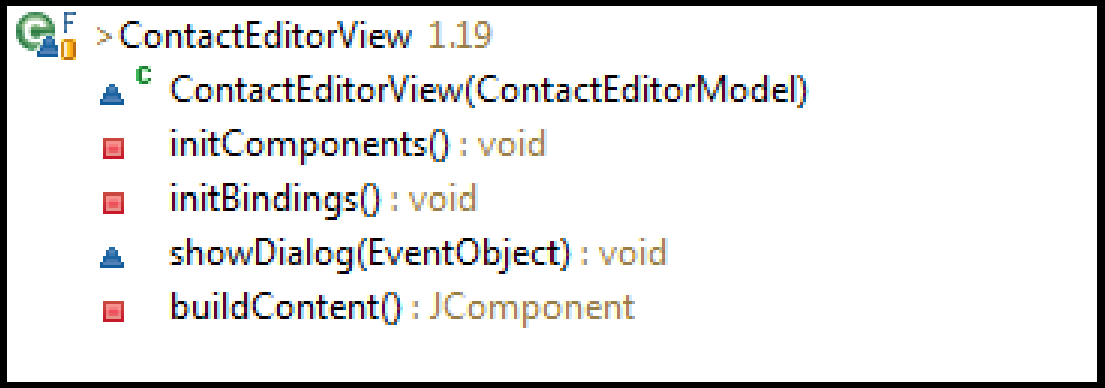
Schreibstil

Visuelle Muster

Kontakt-Beispiel



ContactEditorView



ContactEditorView (1/4)

```
final class ContactEditorView {

  private final ContactEditorModel model;

  private JTextField givenNameField;
  private JTextField surnameField;
  private JTextField phoneField;
  …
  private JButton okButton;


  ContactEditorView(ContactEditorModel model) {
    this.model = model;
    initComponents();
    initBindings();
  }
```

Tipp 3

- Baue Dialoge, Frames, Panels
- Erweitere JDialog, JFrame, JPanel wenn nötig. Erweiterst oder nutzt du HashMap?

Tipp 4

- Meide View-Oberklassen

[Zu] viele Oberklassen

JGoodiesAbstractDialog
JGoodiesDefaultDialog

CompanyAbstractDialog
CompanyDefaultDialog
CompanyValidationDialog

ProjectDefaultDialog

Tipp 4

- Meide View-Oberklassen

- Lieber Komposition als Vererbung
- Erwäge Interfaces statt Klassen für:
  - Konsistenz der View-Klassen
  - View-Lebenszyklus

Tipp 5

- Schreibe für den Leser
- Reduziere Sichtbarkeiten
  - Klassen
  - Felder
  - Methoden
- Markiere als final
- Folge Blochs Ratschlägen aus „Effective Java“

ContactEditorView (2/4)

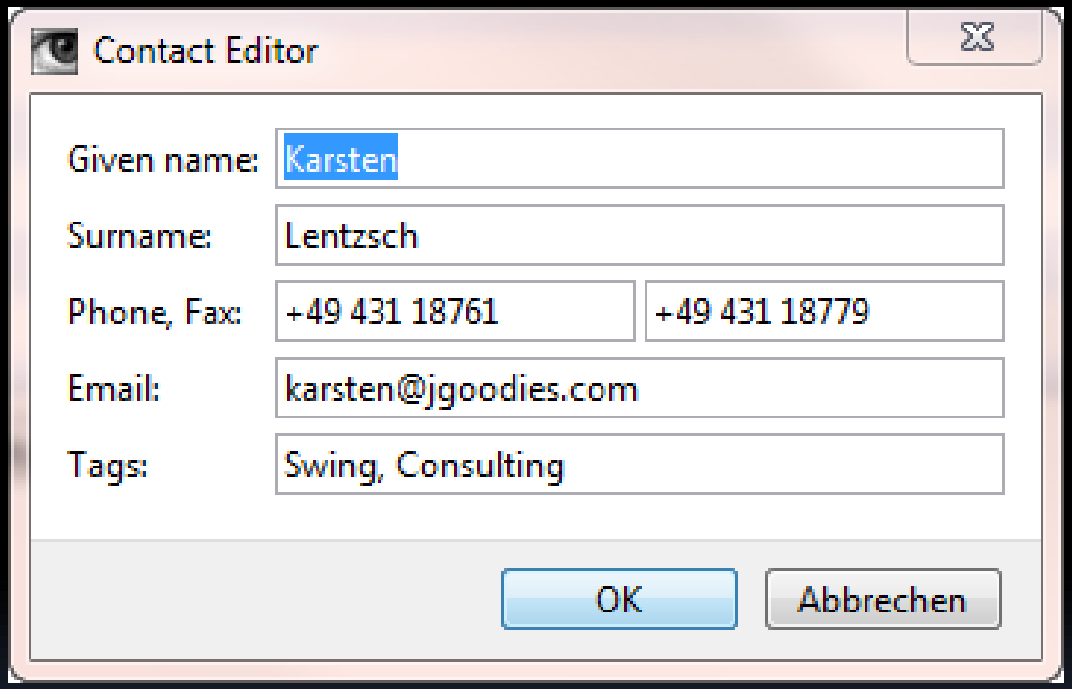
```
private void initComponents() {
  JGComponentFactory factory =
    JGComponentFactory.getCurrent();

  givenNameField = factory.createTextField();
  surnameField   = factory.createTextField();
  phoneField     = factory.createPhoneField();
  faxField       = factory.createFaxField();
  emailField     = factory.createEmailField();
  tagsField      = factory.createTextField();
  Options.setSelectOnFocusGainEnabled(
     tagsField, false);

  okButton = factory.createButton(); // No text
}
```

…

Tipp 6

- Nutze eine Komponentenfabrik
  - Erweiterbar
  - Austauschbar
  - Projektübergreifend
  - Projektspezifisch
- Erwäge spezielle Feldtypen, etwa für: Email, Geld, Strom, Gewicht, Längen, etc.

Kontakteditor



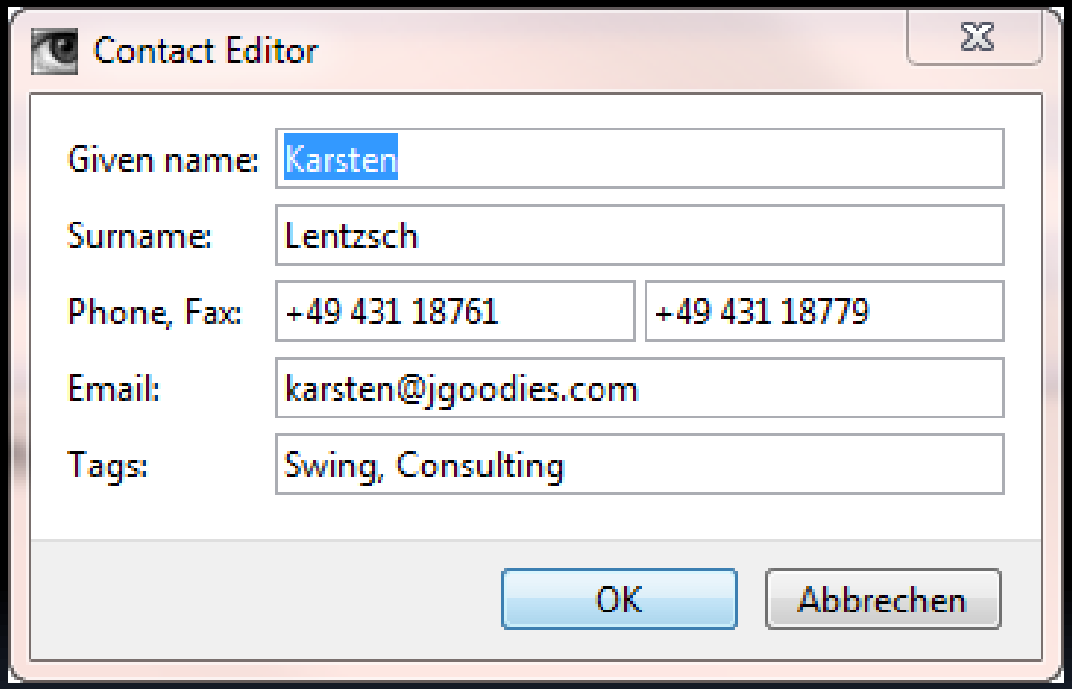
ContactEditorView (3/4)

```
private void initBindings() {
  Binder binder = Binders.binderFor(model);

  binder.bindBeanProperty("givenName")
        .to(givenNameField);
  binder.bindBeanProperty("surname")
        .to(surnameField);
  binder.bindBeanProperty("phone")
        .to(phoneField);
  …
  binder.bindAction("OK").to(okButton);
}

…
```

ContactEditorView (3a/4)

```
private void initBindings() {
  Binder binder = Binders.binderFor(model);

  binder.bindBeanProperty(PROPERTY_GIVEN_NAME)
        .to(givenNameField);
  binder.bindBeanProperty(PROPERTY_SURNAME)
        .to(surnameField);
  binder.bindBeanProperty(PROPERTY_PHONE)
        .to(phoneField);
  …
  binder.bindAction(ACTION_OK).to(okButton);
}

…
```

Contact Editor



ContactEditorView (3/4)

```
private JComponent buildContent() {
  FormLayout layout = new FormLayout(
     "pref, $lcgap, 74dlu, 2dlu, 74dlu",
     "p, $lg, p, $lg, p, $lg, p, $lg, p");

  PanelBuilder builder = new PanelBuilder(layout);
  builder.addLabel("&Given name:", CC.xy (1, 1));
  builder.add(givenNameField,      CC.xyw(3, 1, 3));
  builder.addLabel("&Surname:",    CC.xy (1, 3));
  builder.add(surnameField,        CC.xyw(3, 3, 3));
  builder.addLabel("&Phone, Fax:", CC.xy (1, 5));
  builder.add(phoneField,          CC.xy (3, 5));
  builder.add(faxField,            CC.xy (5, 5));
  …
  return builder.build();
}
```

Tipp 7

- Verwende ein gitterbasiertes Layout
  - MIGLayout
  - JGoodies FormLayout
- Meide 2-Pass-Code
  - Beschreibe erst das Gitter
  - Fülle es danach – und ohne Zustände

Tipp 8

- Lieber flache als geschachtelte Layouts
- Schachtele, wenn nicht ausgerichtet wird
- Schreibe #build-Methoden für Unterlayouts

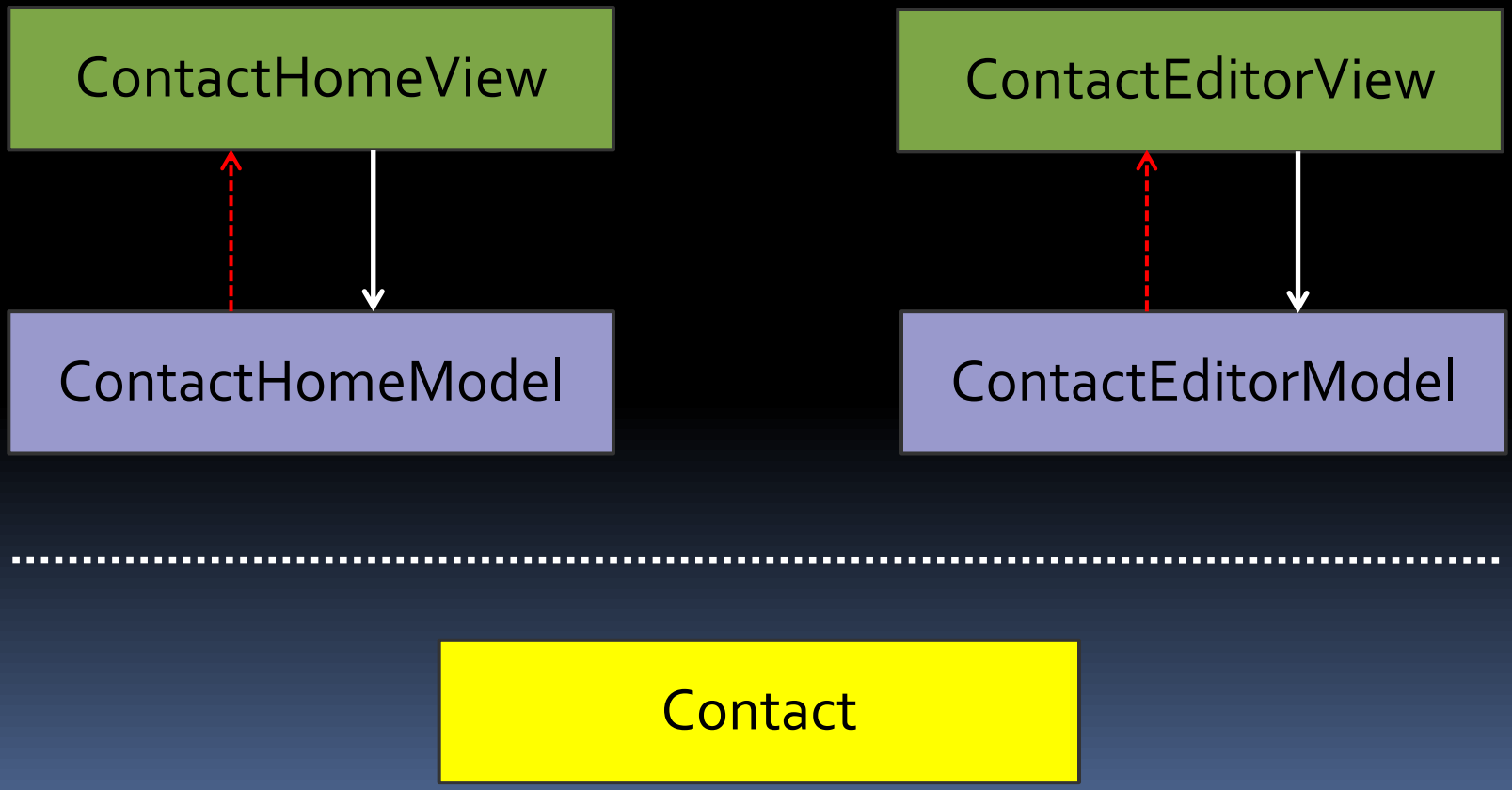
ContactEditorView (4/4)

```
void showDialog(EventObject e) {
   PropertyPane pane = new PropertyPane(
      buildContent(),
      okButton, CommandValue.CANCEL);

   pane.showDialog(e, "Contact Editor");
}
```

Tipp 9

- Trenne Inhalt von der Dekoration
- Hier:
  - Editor-Inhalt
  - Kommandobereich und Rahmen des Eigenschaftsdialoges

Kontakt-Beispiel



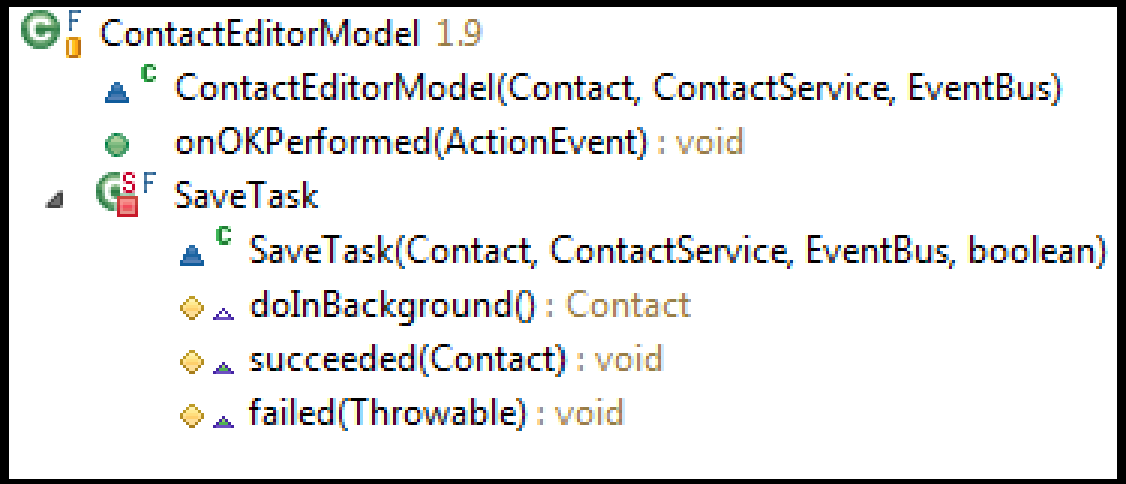
ContactEditorModel (1/2)

```
public final class ContactEditorModel
    extends ActionPresentationModel<Contact> {


  private final ContactService service;
  private final EventBus eventBus;


  public ContactEditorModel(
            Contact contact,
            ContactService service,
            EventBus eventBus) {
     super(contact);
  }
```

Tipp 10

- Verdeutliche die Zusammenhänge
  - Welche Objekte werden verwendet?
  - Womit wird gespeichert/gelesen?
  - Wem werden Änderungen gemeldet?
- Nutze etwa Constructor Injection

ContactEditorModel



Tipp 11

- Erwäge eine Kurzschreibweise für
  - Actions
  - ActionListener
  - PropertyChangeListener
  - ListSelectionListener
- Kenne den @Action-Mechanismus der JSR 296

ContactEditorModel (2/2)

```
@Action
public void onOKPerformed(ActionEvent e) {
    // Save the data
    …
}
```

ContactEditorModel (2a/2)

```
@Action(text="OK")
public void onOKPerformed(ActionEvent e) {
    // Commit pending edits
    TextComponentUtils.commitImmediately();
    // Save the data
    …
}
```

Tipp 12

- Kenne die JSR 296 – Swing App Framework
  - organisiert, vereinfacht, standardisiert
  - Ressource-Management
  - Action-Management
  - Hintergrundaufgaben

Swing Application Framework

Mehr Informationen:

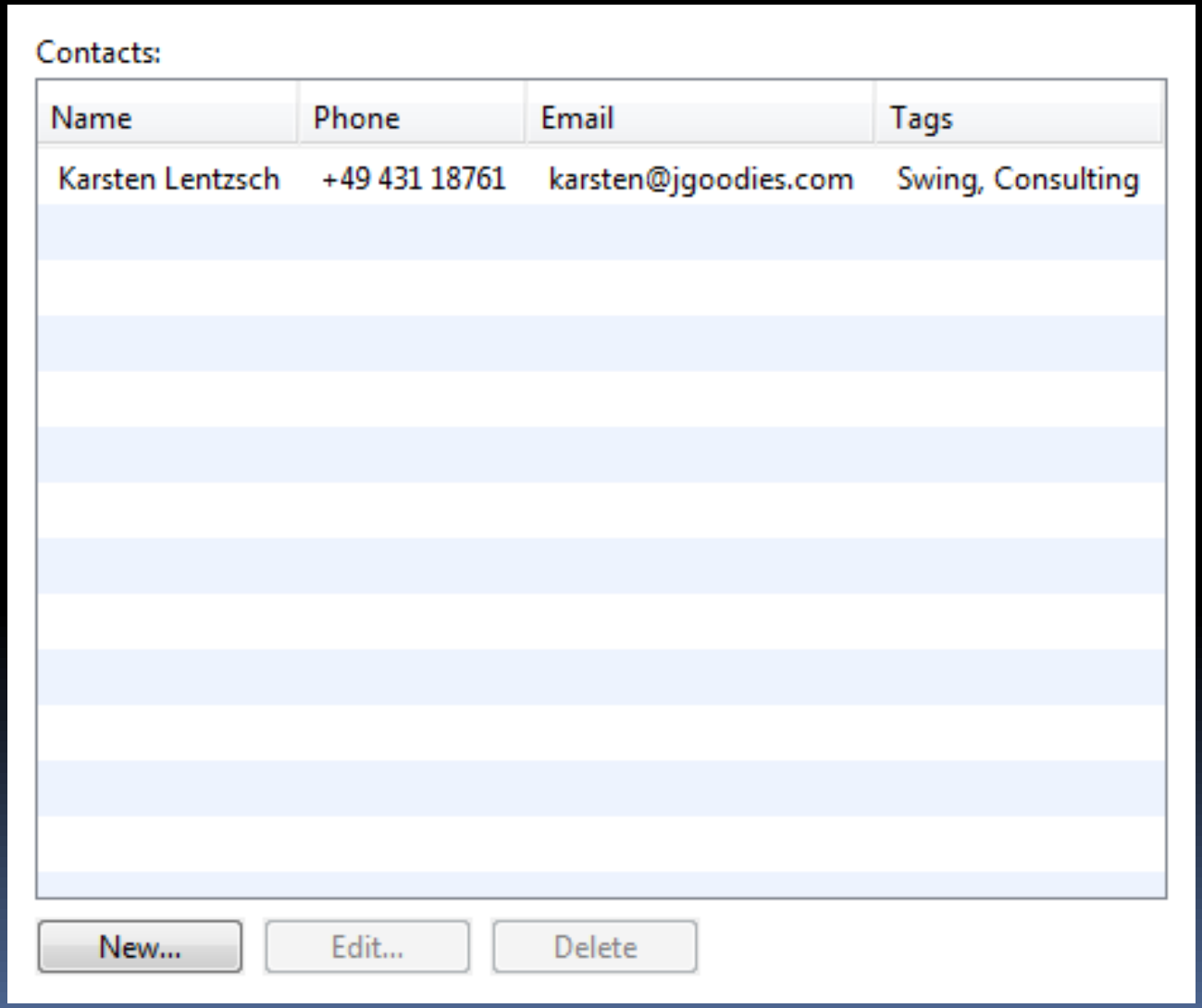
JGoodies.com -> Downloads -> Presentations

„JSR 296 –Swing App Framework“

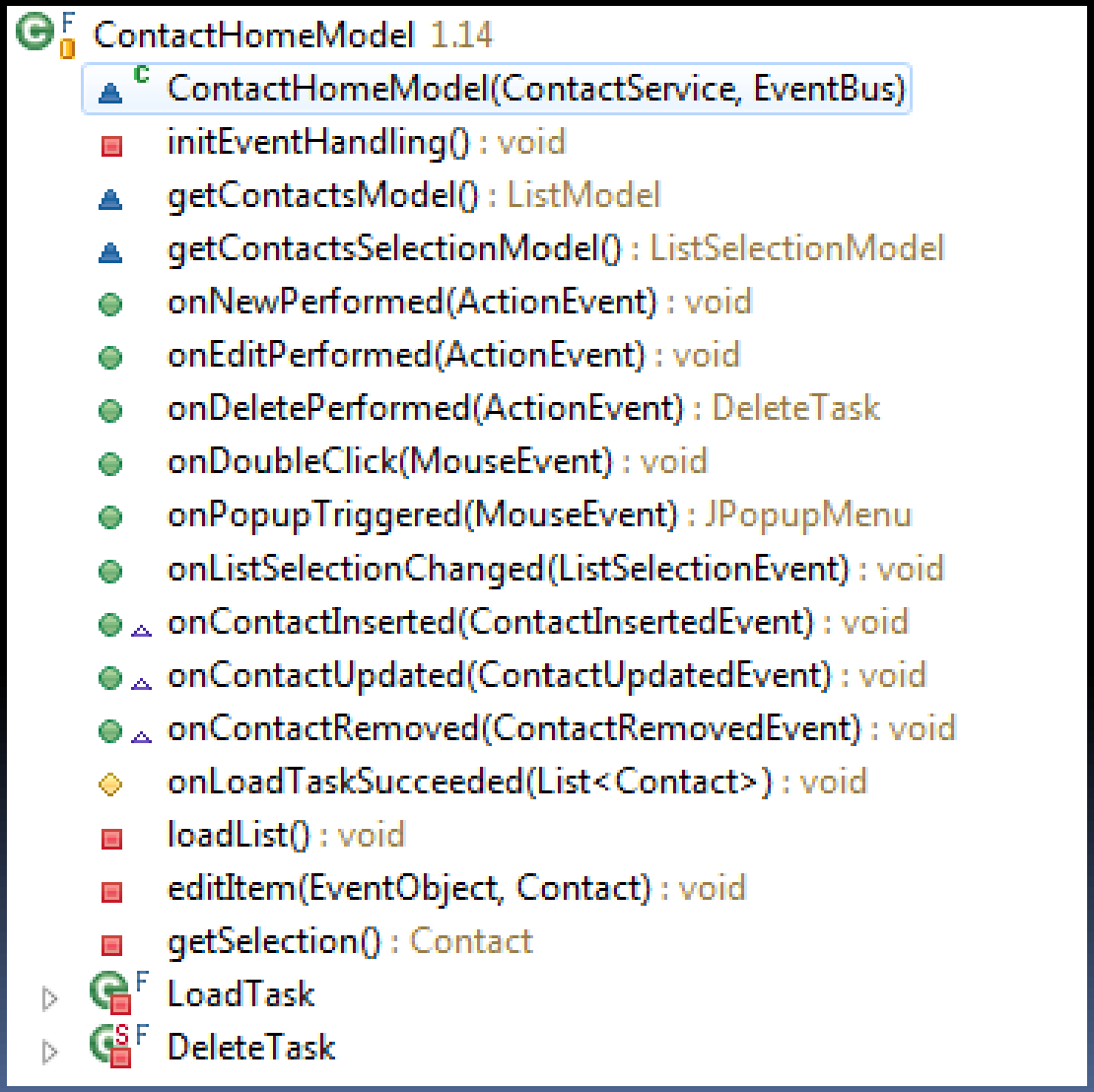
Tipp 13

- Verdeutliche, welche Ereignisse Deine Präsentationslogik behandelt

Kontaktübersicht



ContactHomeModel



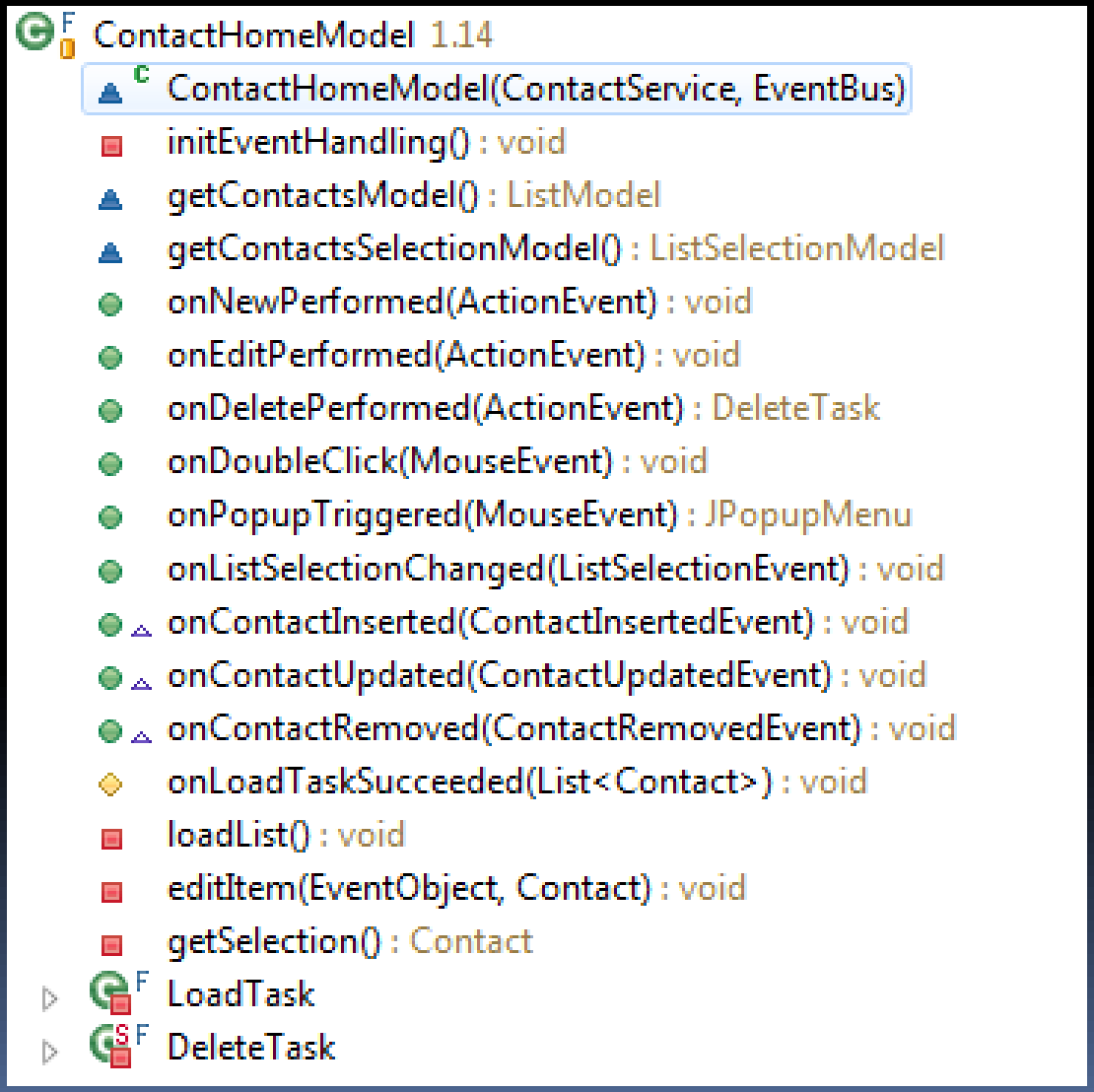
ContactHomeModel

```
@ListSelectionListener
public void onListSelectionChanged(
    ListSelectionEvent e) {
   // Enable/disable the Edit and Delete Action
}


@DoubleClickListener
public void onDoubleClickPerformed(MouseEvent e) {
   editItem(e, getSelection());
}
```

Tipp 14

- Standardisiere den Client-Klassenaufbau
  - Abschnitte
  - (Methoden-)Namen
  - Methoden-Reihenfolge
  - API

ContactHomeModel



Pflicht

- Erledige längere Aufgaben außerhalb des Event-Dispatch-Thread
  - Ein-/Ausgabe-Operationen
  - Server-Zugriffe
  - Service-Aufrufe

~~Pflicht~~

- Erledige längere Aufgaben außerhalb des Event-Dispatch-Thread
  - Ein-/Ausgabe-Operationen
  - Server-Zugriffe
  - Service-Aufrufe

Tipp 15

- Erledige längere Aufgaben eventuell innerhalb des Event-Dispatch-Thread wenn
  - Aufgaben meistens einen „Augenblick“ dauern
  - Du die Oberfläche nicht blockieren kannst
  - Dein Team Swings 1-Thread-Regel nicht beherrscht

Tipp 16

- Schreibe kurz
- Schreibe lieber klar und einfach als zu kurz

Ist kürzer besser?

```
private void initComponents() {
  …
  givenNameField = factory.createTextField();
  …
}

private void initBindings() {
  binder.bindBeanProperty(PROPERTY_GIVEN_NAME)
        .to(givenNameField);
  …
}
```

```
private void initComponents() {
  givenNameField = bindingFactory
    .createBoundTextField(PROPERTY_GIVEN_NAME);
  …
}
```

Ganz kurz

```
private JComponent buildContent() {
   return BeanBuilder.createEditor(Contact.class);
}
```

Tipp 17

- Meide implizite Operationen, die
  - keiner kennt
  - keiner versteht
  - keiner dokumentiert
  - keiner pflegt
  - Du nicht debuggen kannst
- Z. B. Binding oder Editor-Bau mittels Reflection

Tipp 18

- Nutze Views, die ohne Kontext anzeigbar sind

- Ist in MVP „geschenkt“
- Geht auch mit Presentation Model

Modell-Zugriff im PM-View

```
private void initComponents() {
   …
   okButton = new JButton(
               // NPE, wenn model == null
               model.getAction(ACTION_OK));
   …
}
```

```
ContactEditorView(ContactEditorModel model) {
    this.model = model;
    initComponents();
    if (model != null) {
       initBindings();
    }
}
```

Tipp 19

- Schreibe JavaDocs mit gesundem Verstand
  - Meide Wiederholungen des Methodennames
  - Meide leere Tags
  - Folge den Standards (schreibe in der 3. Person, usw.)
  - Frage Dich, ob der Kommentar nützt oder schadet
  - Nutze statische Prüfungen (IDE, CheckStyle)

Kriterien für guten Stil

- Kurz
- Einfach, klar
- Stabil gegenüber Änderungen
  - Muster – MVP vs. PM
  - Mit oder ohne visuellem Editor
  - Komponentenfabrik
  - Bindesystem
  - Layout
  - Toolkit
  - Sprache

Tipp 20: Entrümpeln

- Trenne ab, was sich abtrennen lässt:
  - Fachdaten, Fachlogik, Services
  - Komponentenerzeugung
  - Action-Erzeugung, Listener-Erzeugung
  - Ressourcen-Management
  - Nachrichtensystem (EventBus)
  - Mechanismus für Hintergrundaufgaben
  - Standardlayouts
  - Standarddialoge

Fehlerklassen

- Rechnungsdruck funktioniert nicht
- Eingabe des Rechnungsbetreffs vergessen
- Programmierer hat den Zahlmechanismus missverstanden
- Programmierer hat das Ausrechnen des Rechnungsbetrages missverstanden
- Oberfläche sieht schlecht aus

Gliederung

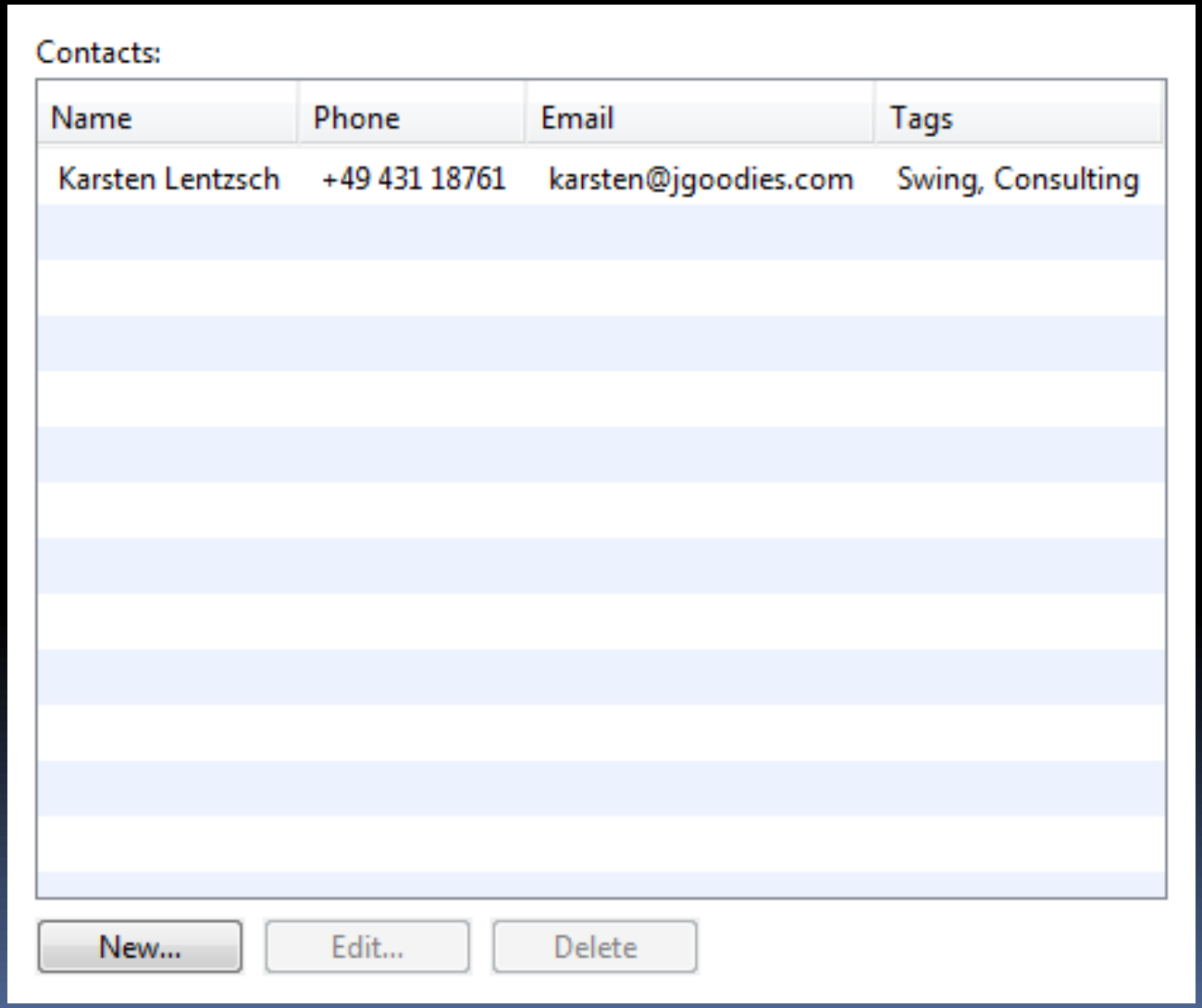
Visuelle Gebote und Verbote

Implementierungs-Muster

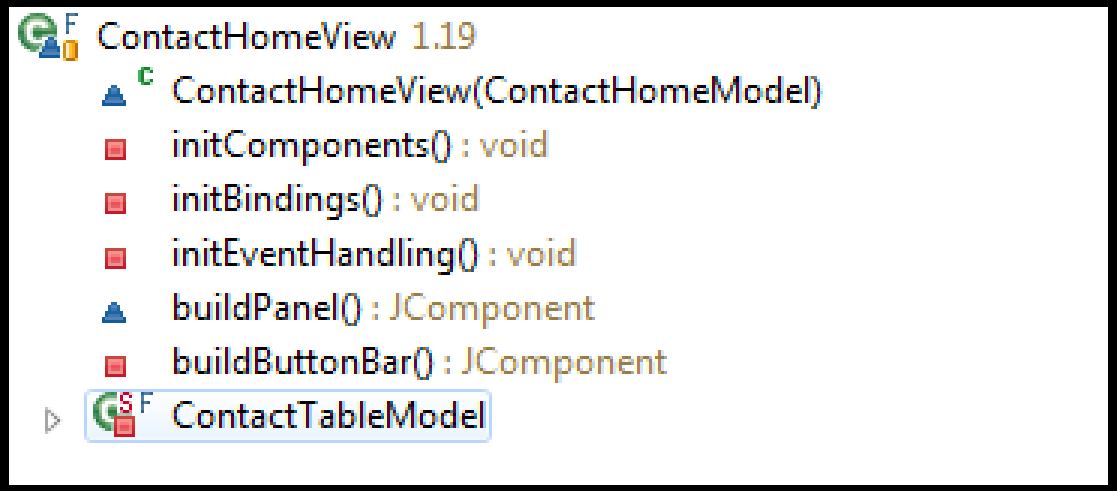
Schreibstil

Visuelle Muster

Kontaktübersicht



ContactHomeView



ContactHomeView Panel-Bau

```
private JComponent buildPanel() {
  FormLayout layout = new FormLayout(
        "fill:250dlu:grow",
        "p, $lcg, fill:200dlu, $rgap, p");

  PanelBuilder builder = new PanelBuilder(layout);
  builder.addLabel("&Contacts:",       CC.xy(1, 1));
  builder.add(
        new JScrollPane(contactsTable), CC.xy(1, 3));
  builder.add(buildButtonBar(),        CC.xy(1, 5));
  return builder.getPanel();
}
```

ContactHomeView Button Bar I

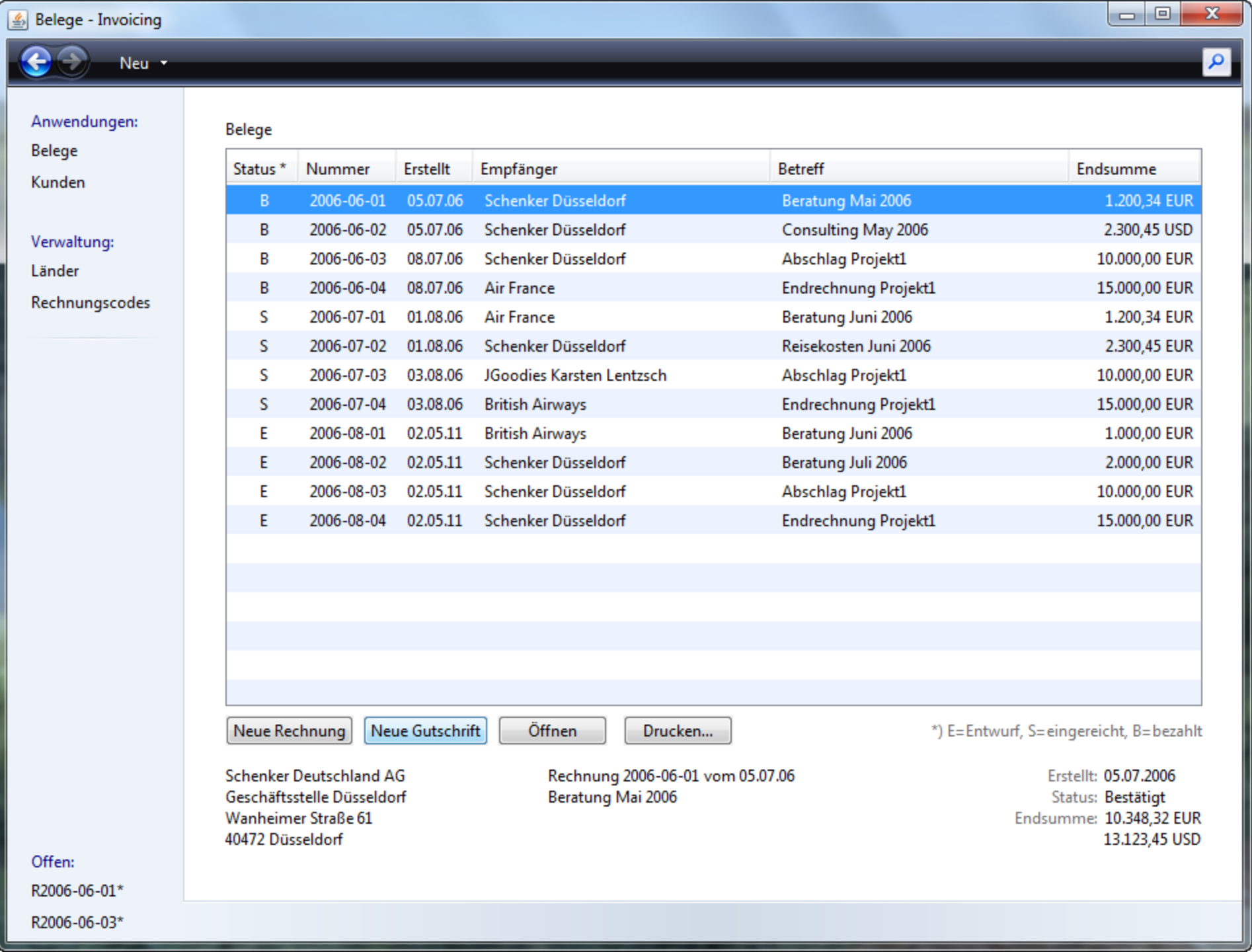
```
private JComponent buildButtonBar() {
  FormLayout layout = new FormLayout(
        "pref, 4px, pref, 4px, pref",
        "p");

  PanelBuilder builder = new PanelBuilder(layout);
  builder.add(newButton,    CC.xy(1, 1));
  builder.add(editButton,   CC.xy(3, 1));
  builder.add(deleteButton, CC.xy(5, 1));

  return builder.getPanel();
}
```

ContactHomeView Button Bar II

```
private JComponent buildButtonBar() {
  return new ButtonBarBuilder()
      .addButton(newButton, editButton, deleteButton)
      .build();
}
```

Tipp 21

- Suche nach gleichem Layout
- Baue es immer gleich, z. B. mit:
  - Panel-Bau-Code (geschachtelt)
  - Gitter-Füll-Code (eingebettet / flach)
  - visuellen Komponenten (Beans)


Belege - Invoicing
Neu
Anwendungen:
Belege
Kunden
Verwaltung:
Länder
Rechnungscodes
Offen:
R2006-06-01*
R2006-06-03*
Belege
Status *
Nummer
Erstellt
Empfänger
Betreff
Endsumme
B 2006-06-01 05.07.06 Schenker Düsseldorf Beratung Mai 2006 1.200,34 EUR
B 2006-06-02 05.07.06 Schenker Düsseldorf Consulting May 2006 2.300,45 USD
B 2006-06-03 08.07.06 Schenker Düsseldorf Abschlag Projekt1 10.000,00 EUR
B 2006-06-04 08.07.06 Air France Endrechnung Projekt1 15.000,00 EUR
S 2006-07-01 01.08.06 Air France Beratung Juni 2006 1.200,34 EUR
S 2006-07-02 01.08.06 Schenker Düsseldorf Reisekosten Juni 2006 2.300,45 EUR
S 2006-07-03 03.08.06 JGoodies Karsten Lentzsch Abschlag Projekt1 10.000,00 EUR
S 2006-07-04 03.08.06 British Airways Endrechnung Projekt1 15.000,00 EUR
E 2006-08-01 02.05.11 British Airways Beratung Juni 2006 1.000,00 EUR
E 2006-08-02 02.05.11 Schenker Düsseldorf Beratung Juli 2006 2.000,00 EUR
E 2006-08-03 02.05.11 Schenker Düsseldorf Abschlag Projekt1 10.000,00 EUR
E 2006-08-04 02.05.11 Schenker Düsseldorf Endrechnung Projekt1 15.000,00 EUR
Neue Rechnung
Neue Gutschrift
Öffnen
Drucken...
*) E=Entwurf, S=eingereicht, B=bezahlt
Schenker Deutschland AG
Geschäftsstelle Düsseldorf
Wanheimer Straße 61
40472 Düsseldorf
Rechnung 2006-06-01 vom 05.07.06
Beratung Mai 2006
Erstellt: 05.07.2006
Status: Bestätigt
Endsumme: 10.348,32 EUR
13.123,45 USD

TuneQ: Anlagen > Antrieb HW61, AVV_ZVV

TuneQ | Hilfe | Anlagen | Instandhaltung | Analyse | Wartung

Navigation

- Strecken
  - Strecke 1
    - EW 7 / 1445
      - EW 7
        - Antrieb HW61, AVV_ZVV
        - Heizung HWH
      - Steuerung HN-P 7 / 1445
      - Außenanlagen
    - EW 12 / 1449
  - Strecke 2
  - Strecke 3
  - Strecke 4
  - Strecke 5
- Datensammler
  - GUW Albertplatz
  - Btf. Gorbitz
  - Btf. Reich (wird angepasst)
  - Btf. Trachenberge
  - GUW Coswig
  - GUW Jahnstraße
  - GUW Meschwitzstrasse
  - GUW Postplatz
- Anlagen

Übersicht | Dokumente | Inspektionen | Meldungen | Historie

Antrieb EW 7 - Strecke 1
Friedrichstr. - Maxstr./Könneritzstr.

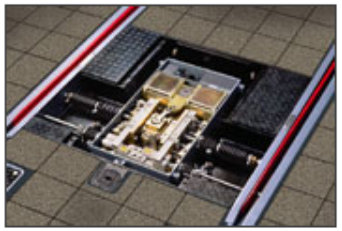
Gerätenr.:	320 513
EDV-Nr.:	109 244 318
Einbau:	12.03.2005
Inspektion:	21.02.2007

Status:	In Betrieb
Montage:	Gleismitte
Ausführung:	Flachbettweiche

Eigenschaften bearbeiten

Produkt

Baureihe:	HW 61	Stellkraft:	5000 N
Antriebsform:	Elektromagnetisch	Verschluß:	Ja
Variante:	AVV-ZVV	Feder:	Ja
Art:	Umstellweiche	Dämpfung:	Ja
Zungenaufschlag:	30 - 70 mm	Richtungsschalter:	nein
Höhe mit Kasten:	300 mm	Zungenprüfer:	Ja
Höhe ohne Kasten:	205 mm	Verschlossen:	Ja
Betriebsspannung:	600/750 V DC	Auffahrbar:	Ja



Produktdetails zeigen

Baugruppen:

Bezeichnung	EDV-Nummer	Gerätenr.	Einbau	Bemerkung
Erdkasten		320 514	12.03.2005	
Doppelzugmagnet	30032002		12.03.2005	
Antriebswelle	31051008		30.11.2006	

Neu... | Bearbeiten... | Löschen | Hoch | Runter

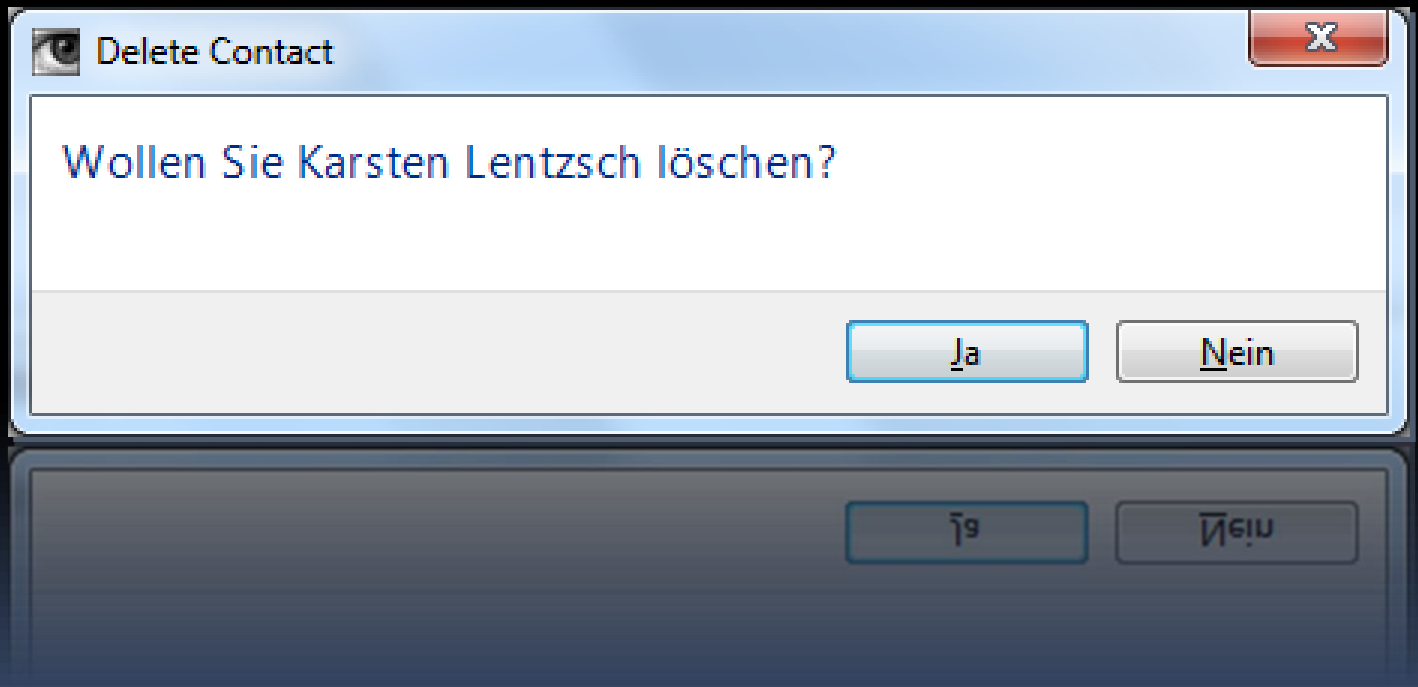
! 3 Ungelesene Nachrichten

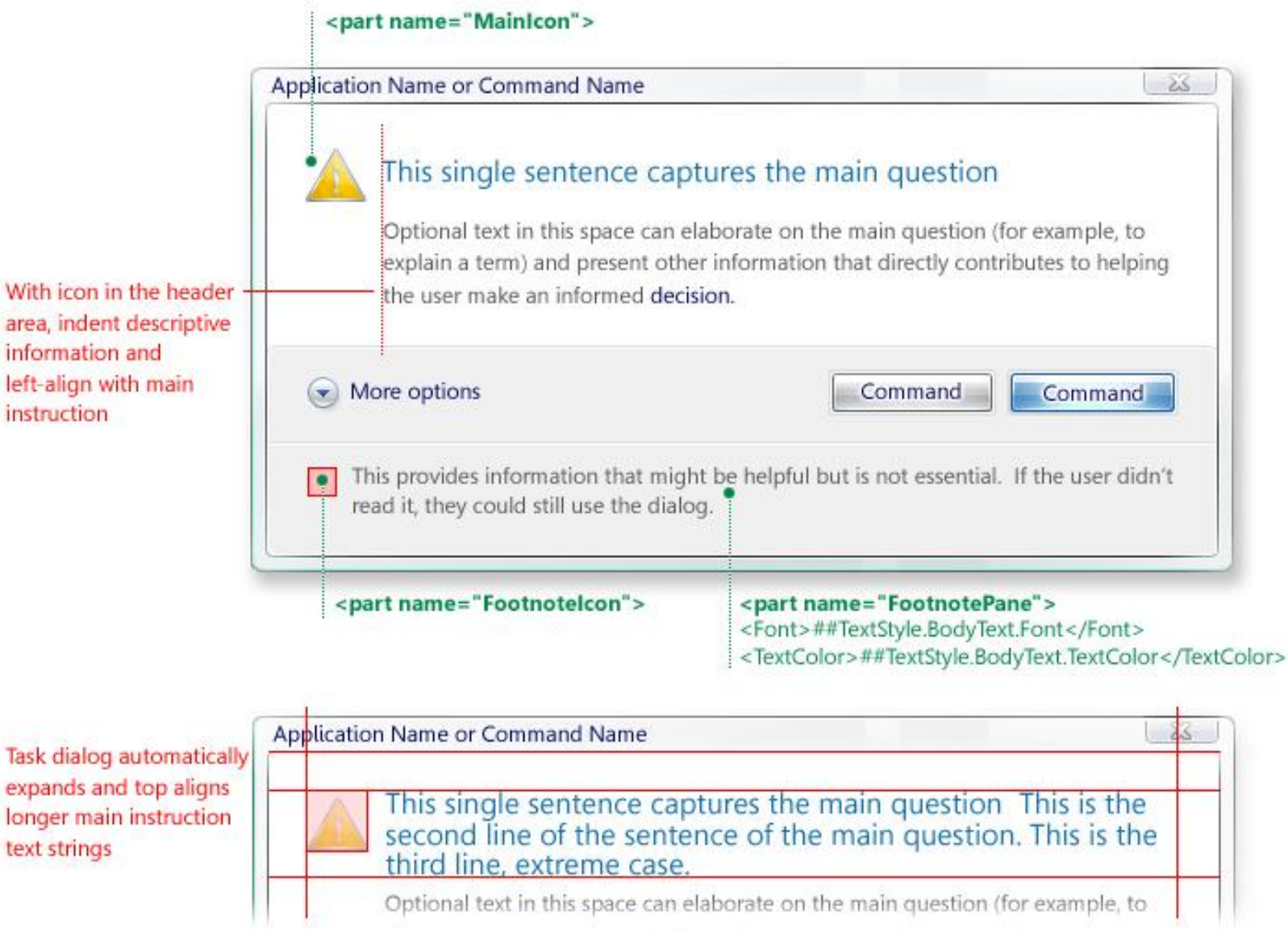
ListViewBuilder

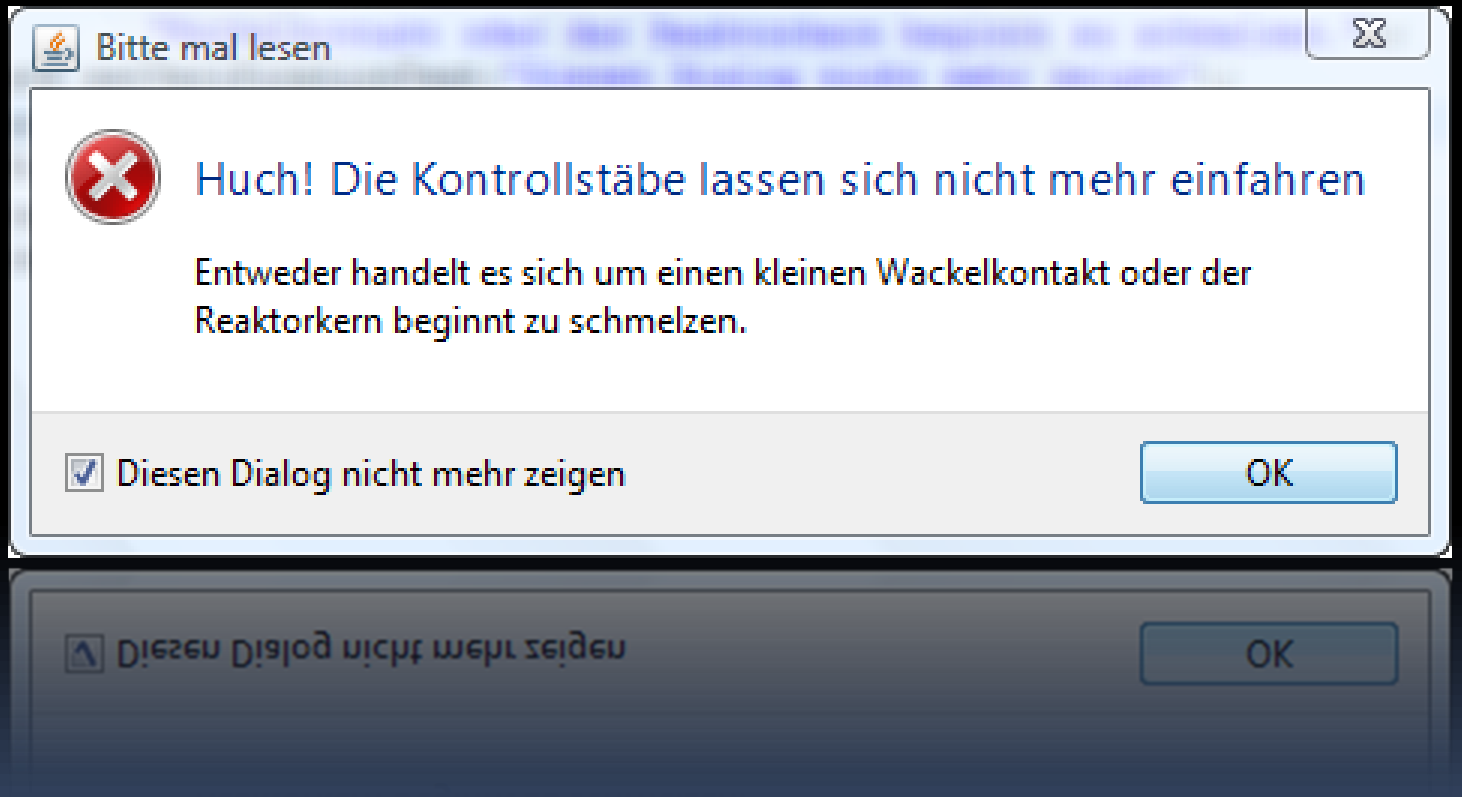
```
private JComponent buildPanel() {
  return new ListViewBuilder()
      .title("&Contacts:")
      .listView(contactsTable))
      .listBar(newButton, editButton, deleteButton)
   // .detailsView(…)
      .build();
}
```

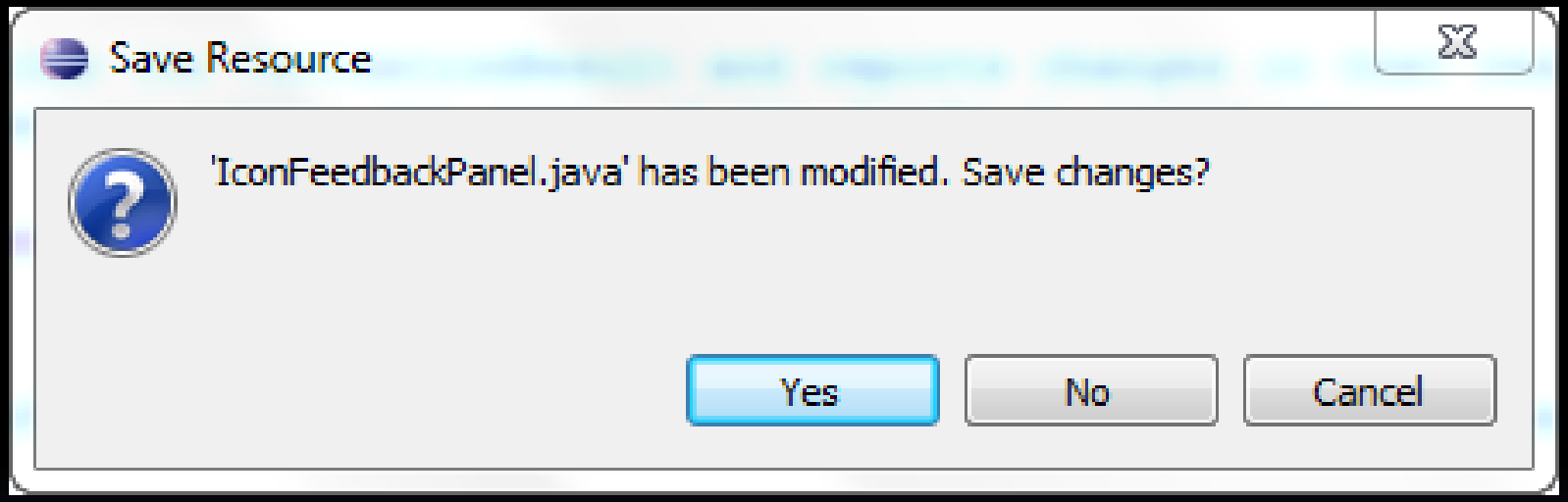
ContactHomeModel

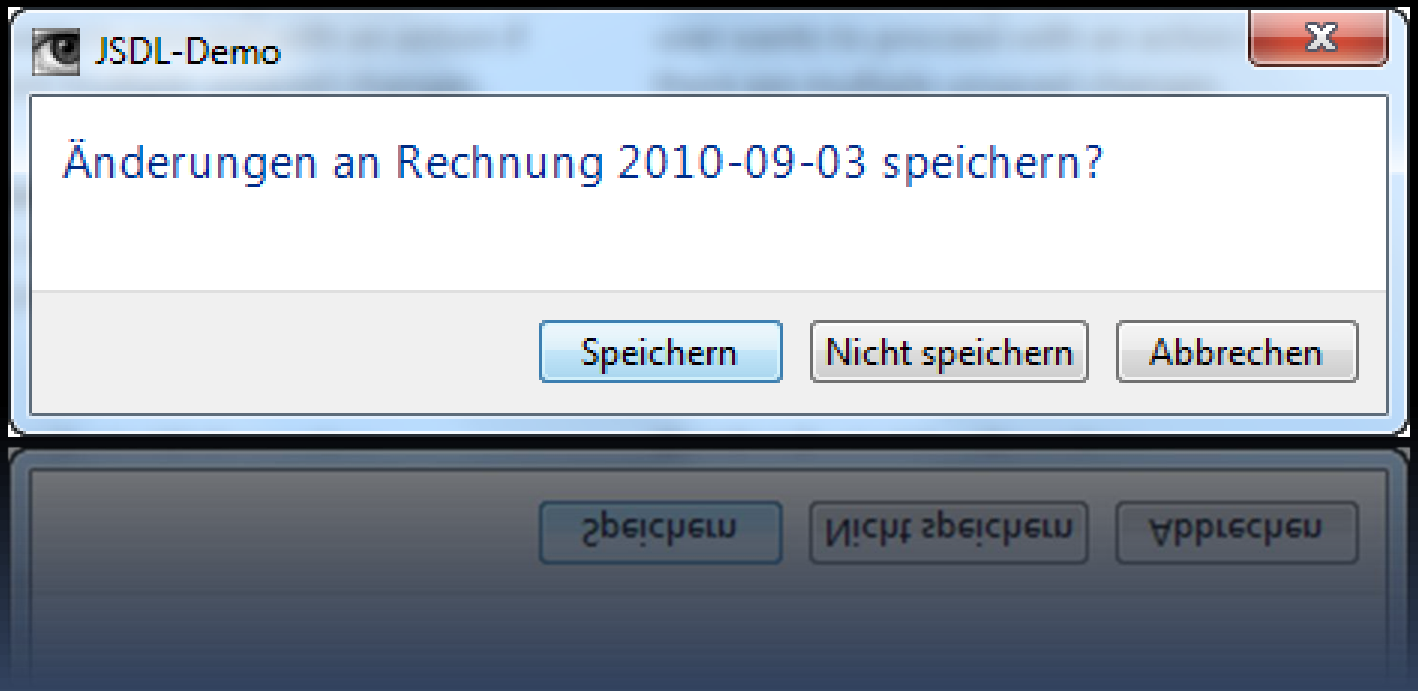
```
@Action
public void onDeletePerformed(ActionEvent e) {
  Contact selection = view.getSelection();
  String objectName = selection.getDisplayString();

  TaskPane pane = new TaskPane(
    MessageType.QUESTION,
    "Do you want to delete " + objectName + "?",
    CommandValue.YES, CommandValue.NO);

  pane.showDialog(e, "Delete Contact");

  if (pane.getCommitValue() == CommandValue.YES) {
      // Delete the selection
  }
}
```


Delete Contact
Wollen Sie Karsten Lentzsch löschen?
Ja
Nein


<part name="MainIcon">
Application Name or Command Name
This single sentence captures the main question
Optional text in this space can elaborate on the main question (for example, to explain a term) and present other information that directly contributes to helping the user make an informed decision.
With icon in the header area, indent descriptive information and left-align with main instruction
More options
Command
Command
This provides information that might be helpful but is not essential. If the user didn't read it, they could still use the dialog.
<part name="FootnoteIcon">
<part name="FootnotePane">
<Font>##TextStyle.BodyText.Font</Font>
<TextColor>##TextStyle.BodyText.TextColor</TextColor>
Task dialog automatically expands and top aligns longer main instruction text strings
Application Name or Command Name
This single sentence captures the main question This is the second line of the sentence of the main question. This is the third line, extreme case.
Optional text in this space can elaborate on the main question (for example, to


Bitte mal lesen
Huch! Die Kontrollstäbe lassen sich nicht mehr einfahren
Entweder handelt es sich um einen kleinen Wackelkontakt oder der Reaktorkern beginnt zu schmelzen.
Diesen Dialog nicht mehr zeigen
OK


Save Resource
'IconFeedbackPanel.java' has been modified. Save changes?
Yes
No
Cancel


JSDL-Demo
Änderungen an Rechnung 2010-09-03 speichern?
Speichern
Nicht speichern
Abbrechen

Style Guide im API

```
@ActionListener
public void onDeletePerformed(ActionEvent e) {
  Contact selection = view.getSelection();
  String objectName = selection.getDisplayString();

  boolean proceed = MessagePanes.getCurrent()
    .showConfirmation(e,
    “Delete Contact”,
    “Do you want to delete ” + objectName + “?”);

  if (proceed) {
      // selection löschen
  }
}
```

MessagePanes-Klasse

- `#showError(…)`
- `#showAwarenessWarning(…)`
- `#showImminentProblem(…)`
- `#showInformation(…)`
- `#showConfirmation(…)`
- `#showRiskyActionConfirmation(…)`
- `#showHelp(…)`
- …

Standarddialoge

```
@ActionListener
public void onDeletePerformed(ActionEvent e) {
  Contact selection = view.getSelection();
  String objectName = selection.getDisplayString();

  boolean proceed = StandardPanes.getCurrent()
    .showDeleteConfirmation(e, objectName);

  if (proceed) {
      // selection löschen
  }
}
```

StandardPanes-Klasse

- `#showDeleteConfirmation(…)`
- `#showRiskyDeleteConfirmation(…)`
- `#showExitConfirmation(…)`
- `#showSaveChangesConfirmation(…)`
- `#showSaveAllChangesConfirmation(…)`
- `#showNotYetImplemented(…)`
- `#showNotYetTestedConfirmation(…)`
- …

Visuelle Muster (Meta Design)

Mehr Informationen:

JGoodies.com -> Downloads -> Presentations

„Effizient gestalten mit Swing“

Arbeitsebenen

IDE

Bibliothek

Hochsprache (Java)

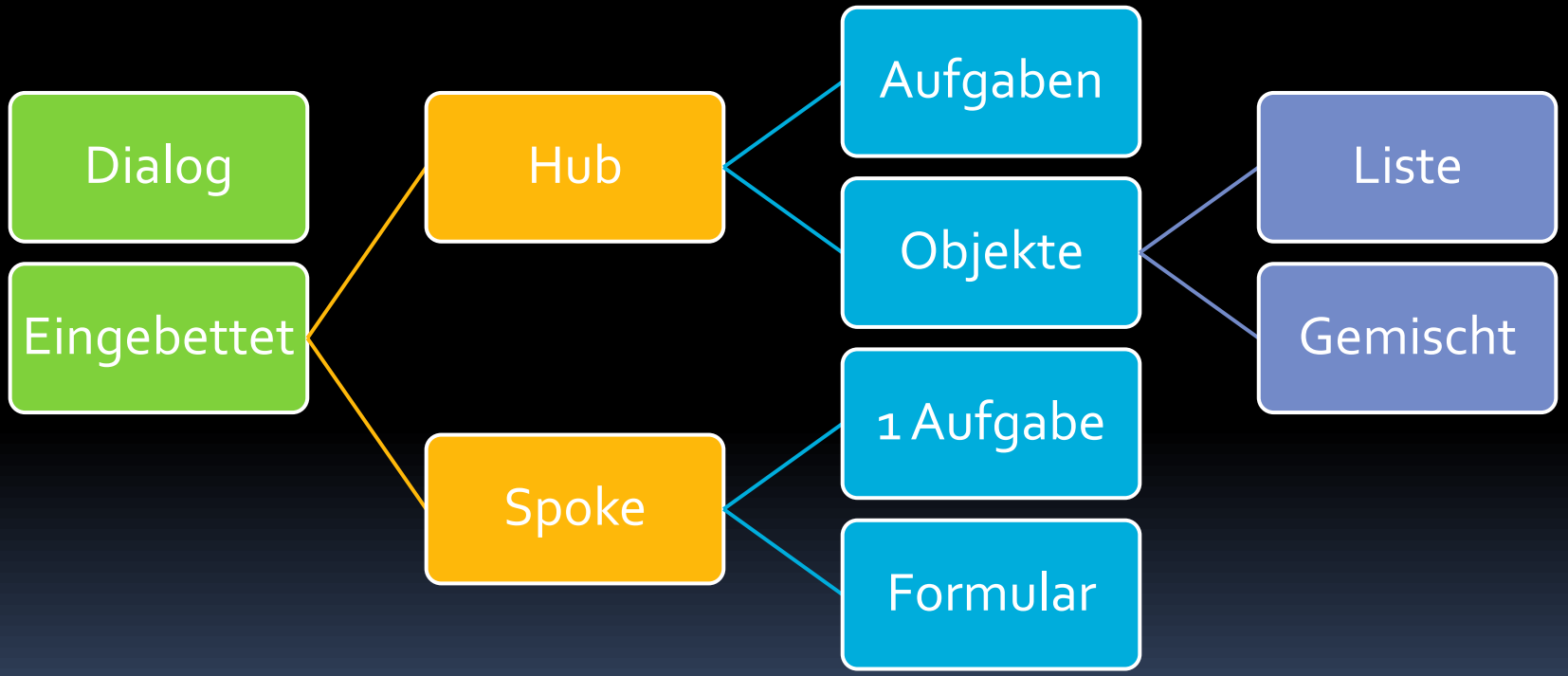
Assembler

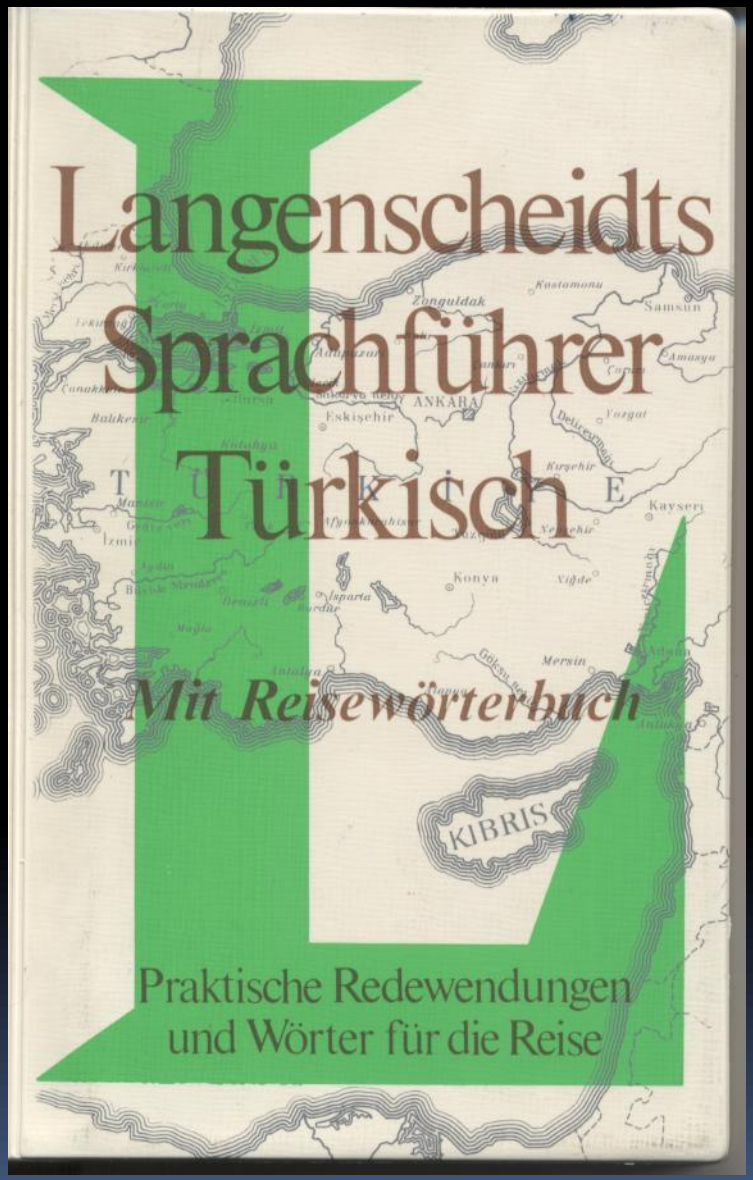
Arbeitsebenen

IDE

Bibliothek	Design-Bibliothek
Hochsprache (Java)	GUI-DSL, -Bean, -Builder
Assembler	Toolkit (GWT, Swing)

Fensterarten



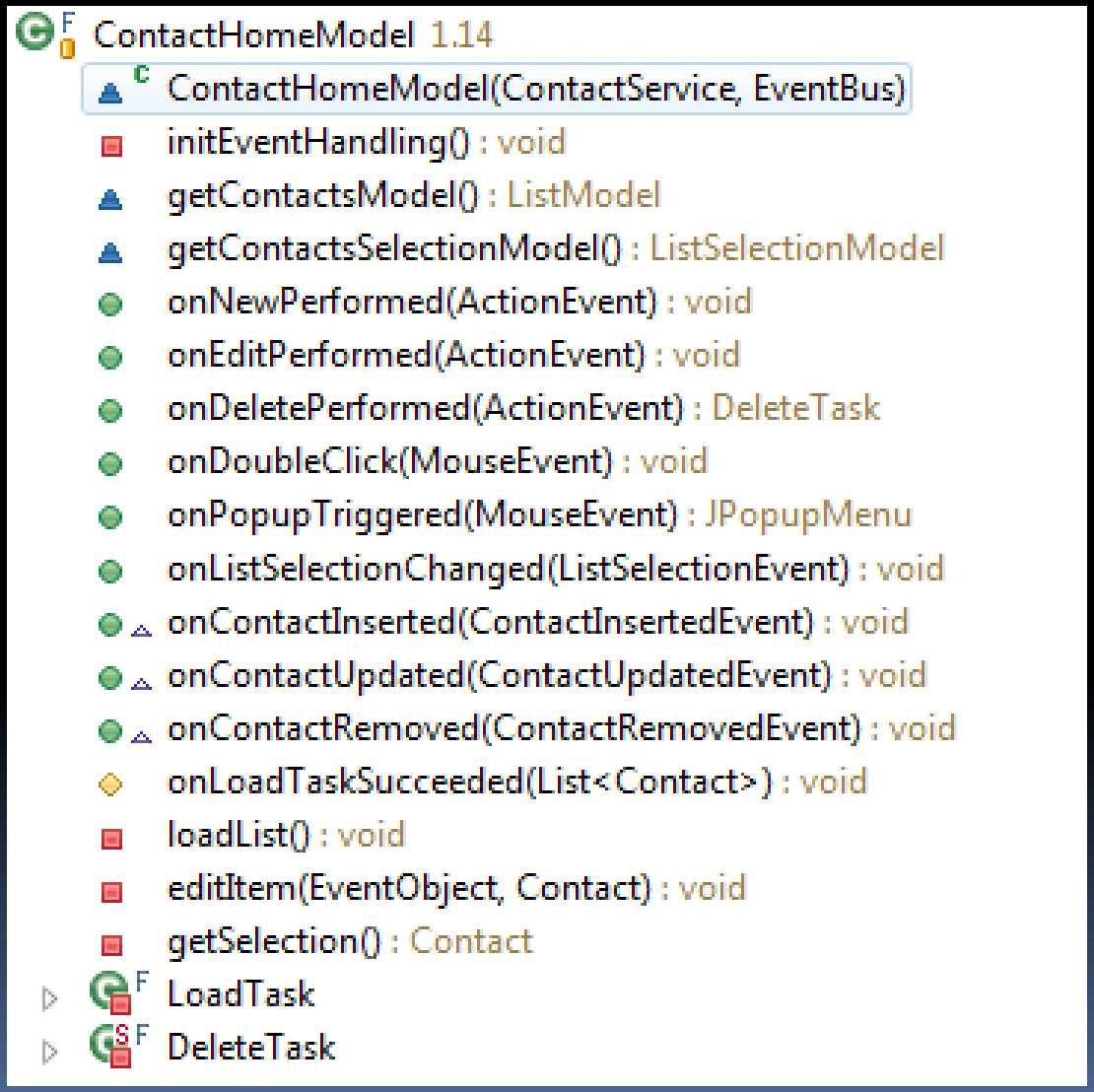

Langenscheidts
Sprachführer
Türkisch
Mit Reisewörterbuch
Praktische Redewendungen
und Wörter für die Reise
KIBRIS

Mehr zur menschlichen Seite

JAX-Video:

„Warum so viele kluge Leute so schlechte Oberflächen entwickeln“

ContactHomeModel



FRAGEN UND ANTWORTEN

Karsten Lentzsch – JGoodies

JAVA UI DESIGN WITH STYLE