BERUFSAKADEMIE SACHSEN
STAATLICHE STUDIENAKADEMIE
DRESDEN
UNIVERSITY OF COOPERATIVE EDUCATION

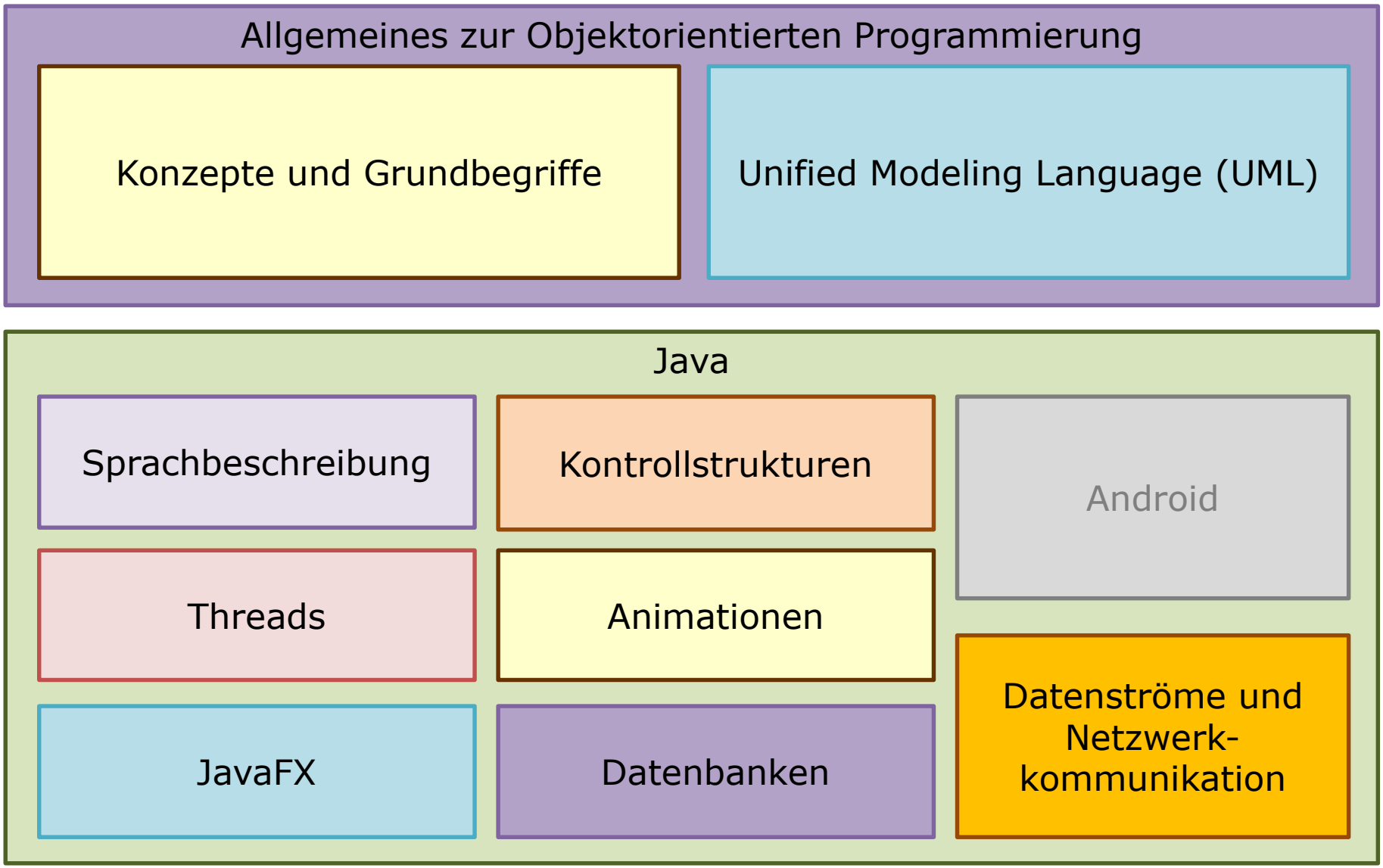
Objektorientierte Programmierung

JavaFX

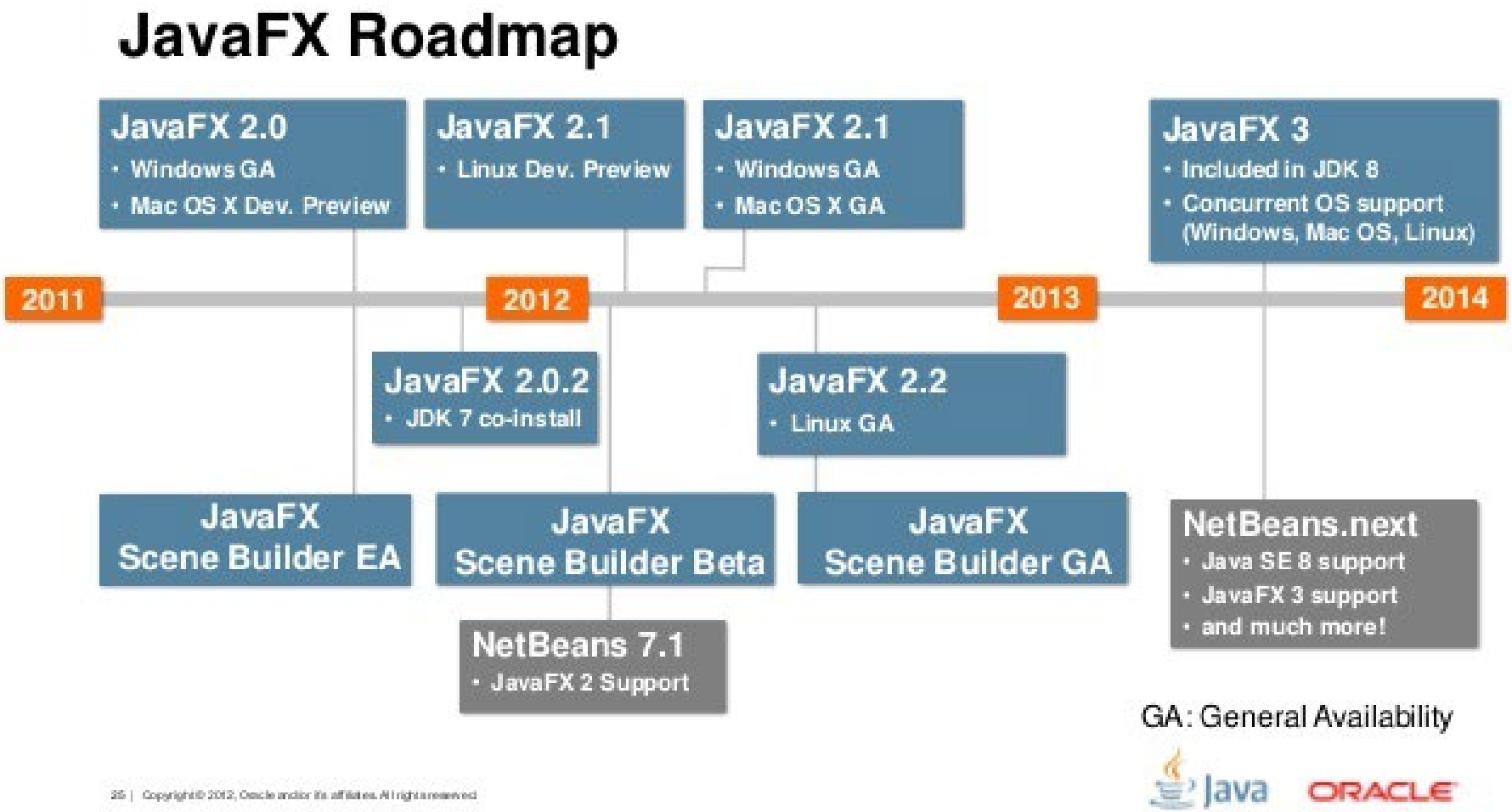
mit Material von Prof. E. Engelhardt

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-sachsen.de

Aufbau der Lehrveranstaltung



Entwicklungsstufen von JavaFX



Entwicklungsstufen von JavaFX

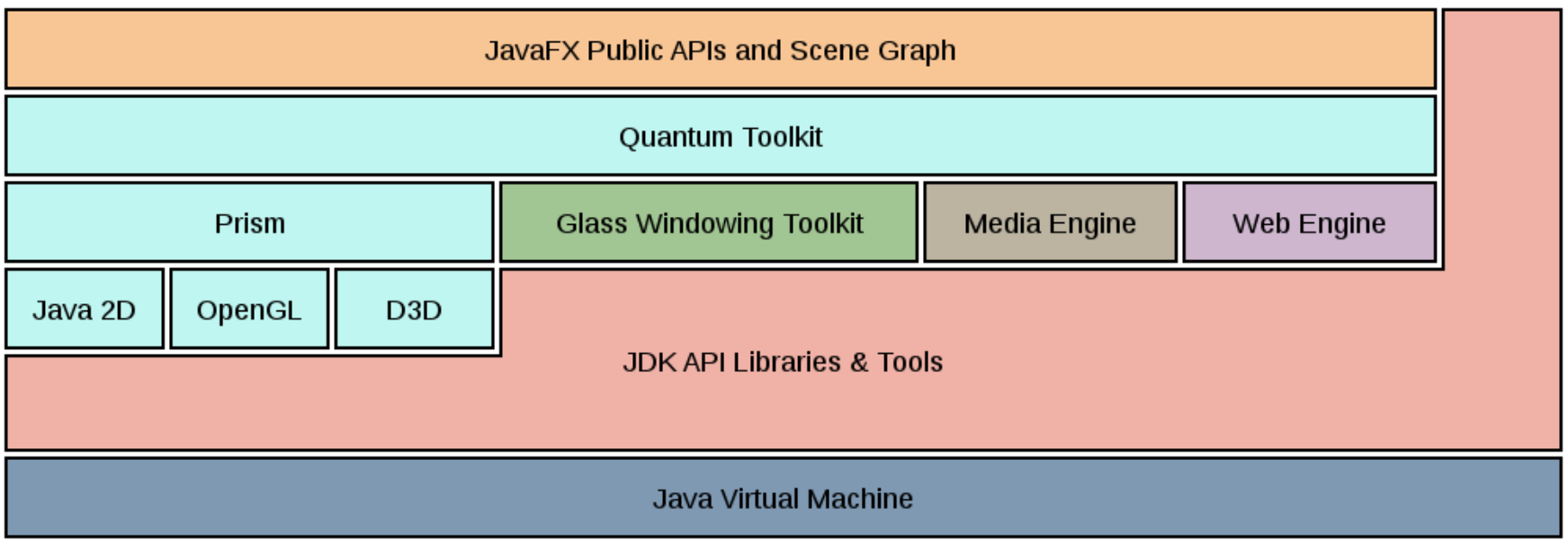
- ab Version 2.0 bezeichnet JavaFX ein Framework von Oracle zur Erstellung plattformübergreifenden Applikationen (RIA)
- Umstellung auf Java als Programmiersprache und damit moderne UI-Technologie über Klassenbibliothek
- leichtgewichtige, hardwarebeschleunigte Applikationen ohne Anwendung unterschiedlicher Technologien
- Einsatz auf Endgeräten wie: Mobilfunkgeräten, Set-Top-Boxen, Desktop-Computern und Multimediageräten
- ab Java 8 fester Bestandteil von Java (als JavaFX %JavaVersion%)
- JavaFX ist seit JDK 11 wieder aus dem JDK ausgegliedert worden

Vorteile von JavaFX

- Control-Bibliothek in Form eines hierarchischen Control-Baumes
  - Transformationen (Verschieben, Drehen, Zoomen)
  - Animationen, animierte Übergänge
  - flüssige Touch-Bedienung
- direkter Umgang der UI-Technologie mit Grafikkarte des Frontends
- HTML-Inhalte können über Browser-Control mit Schnittstelle `WebView` angezeigt werden.
- Möglichkeit, Oberflächen deklarativ mit FXML und CSS auch ohne Programmierkenntnisse zu erstellen (mittels Scene Builder)

Architektur des JavaFX

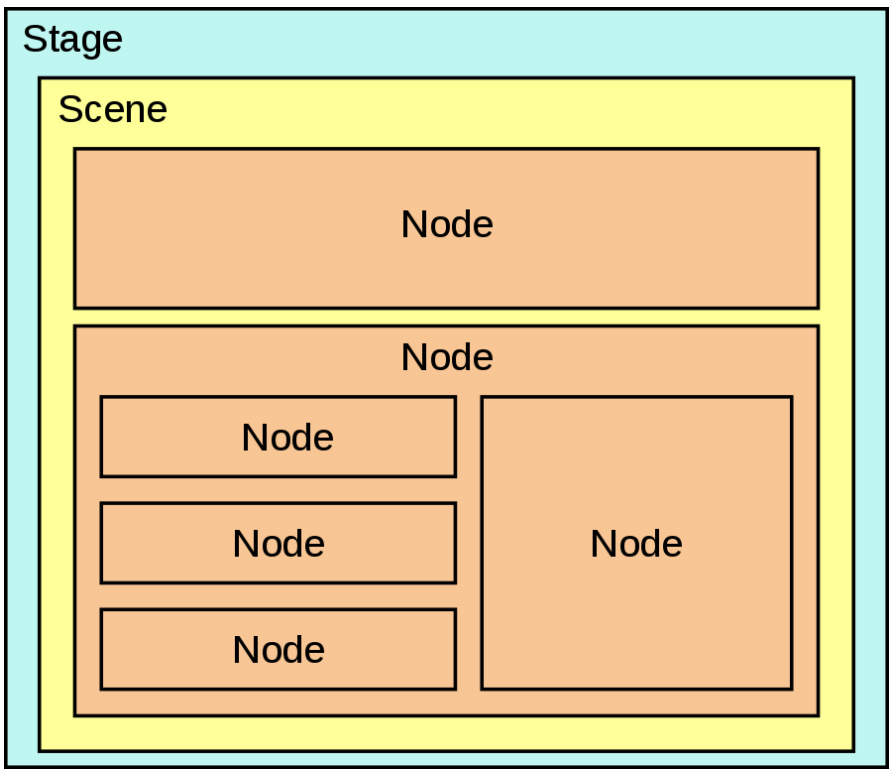
- Scene Graph enthält als Nodes alle GUI-Elemente
- Quantum Toolkit verbindet unterliegende Teile einheitlich für JavaFX-APIs
  - Rendering Engine Prism basierend auf Direct3D, OpenGL, Java 2D, …
  - Glass Windowing Toolkit greift auf Low-Level-Betriebssystemroutinen zu
  - Media Engine – Unterstützung von Audio- und Video
  - Web Engine – Einbettung von Web-Inhalten



Elemente, Struktur

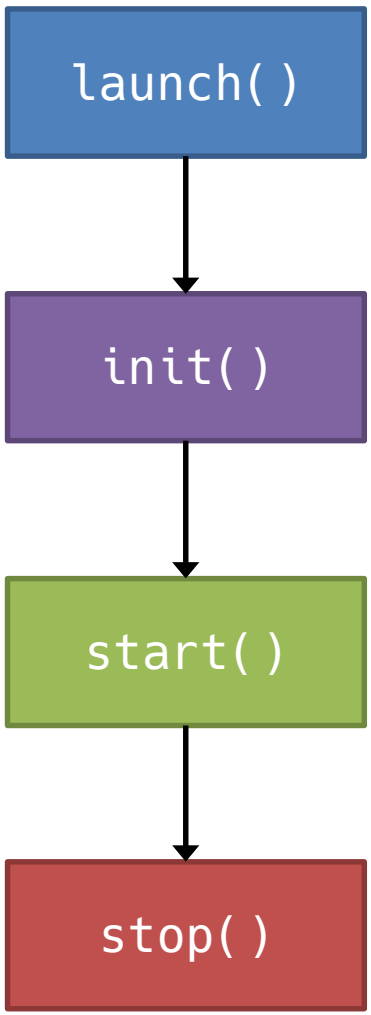
Bühne

- `Stage` ist Rahmen für eigentlichen Inhalt (weitere Stages möglich)
- Klasse `Scene` verwaltet den Szenengraph
- `Nodes` sind sichtbare GUI-Elemente

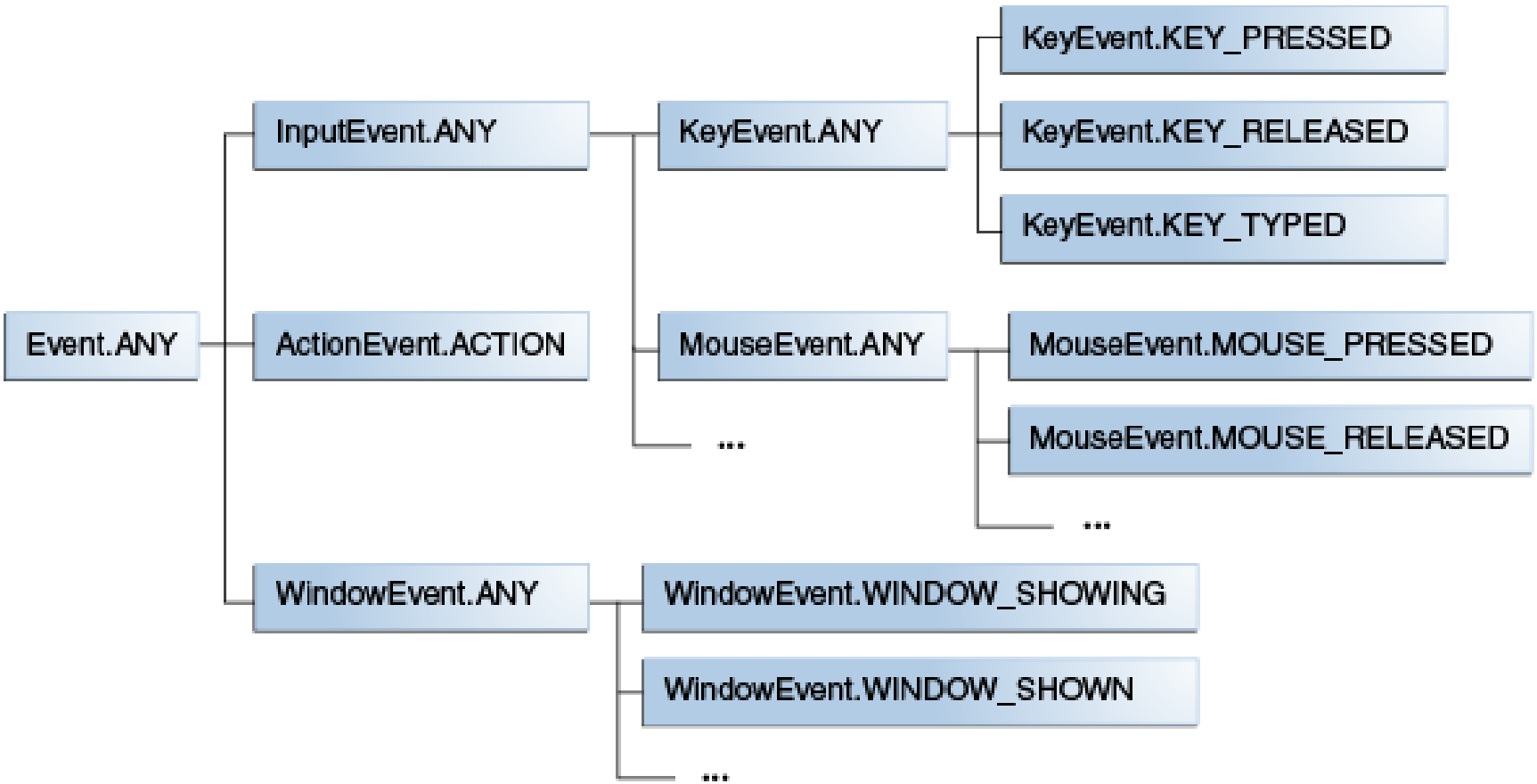


Lebenszyklus einer JavaFX-Anwendung

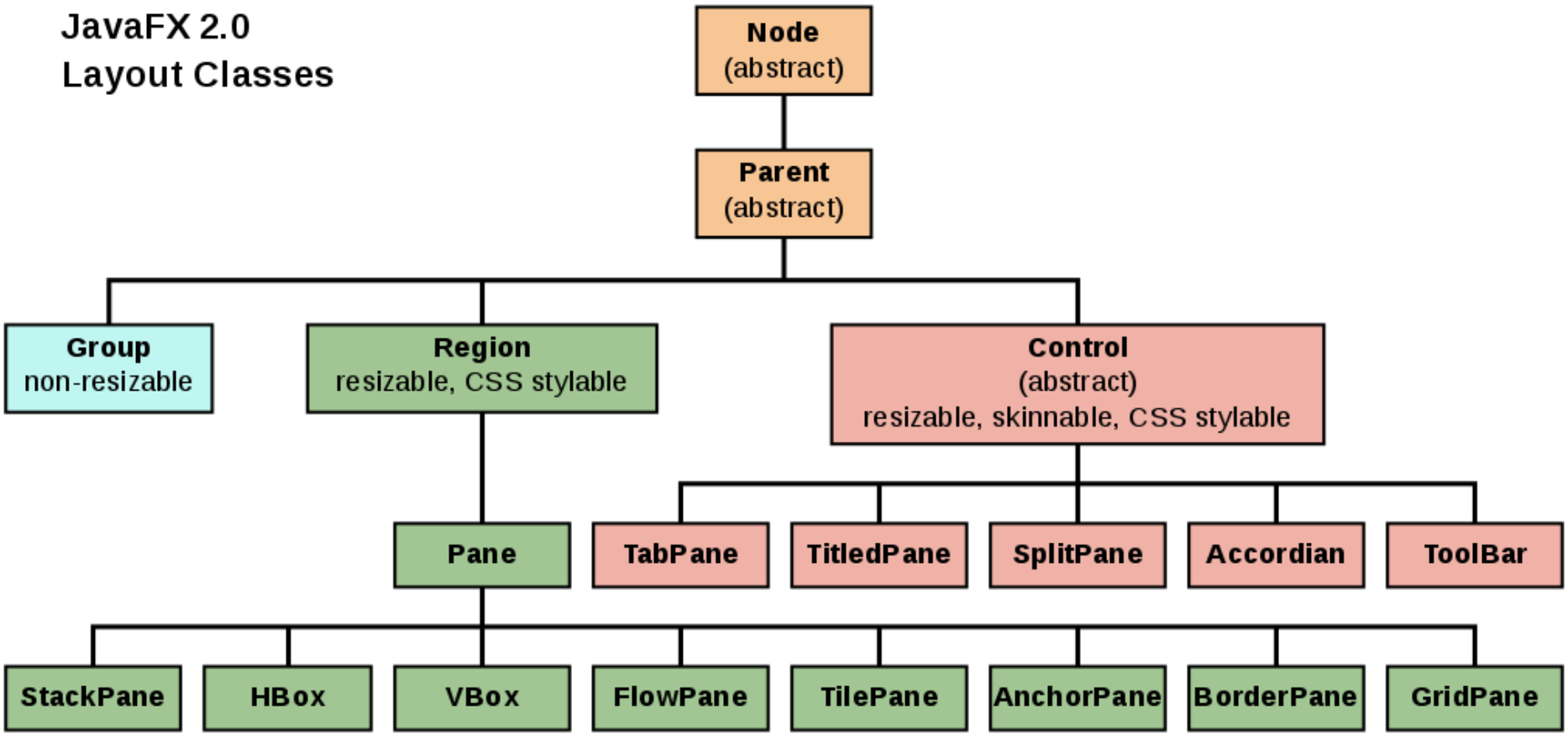
- JavaFX-Launcher organisiert die Aufrufe
- JavaFX-Applikation kann leicht in eine Webseite eingebettet werden
- `init()` und `stop()` müssen nicht zwingend überschrieben werden
- `start()` wird überschrieben
- JavaFX-Anwendung läuft bis das letzte Fenster geschlossen wird oder `Platform.exit()` aufgerufen wird



Event-Handling



NODES


JavaFX 2.0
Layout Classes
Node
(abstract)
Parent
(abstract)
Group
non-resizable
Region
resizable, CSS stylable
Control
(abstract)
resizable, skinnable, CSS stylable
Pane
TabPane
TitledPane
SplitPane
Accordian
ToolBar
StackPane
HBox
VBox
FlowPane
TilePane
AnchorPane
BorderPane
GridPane

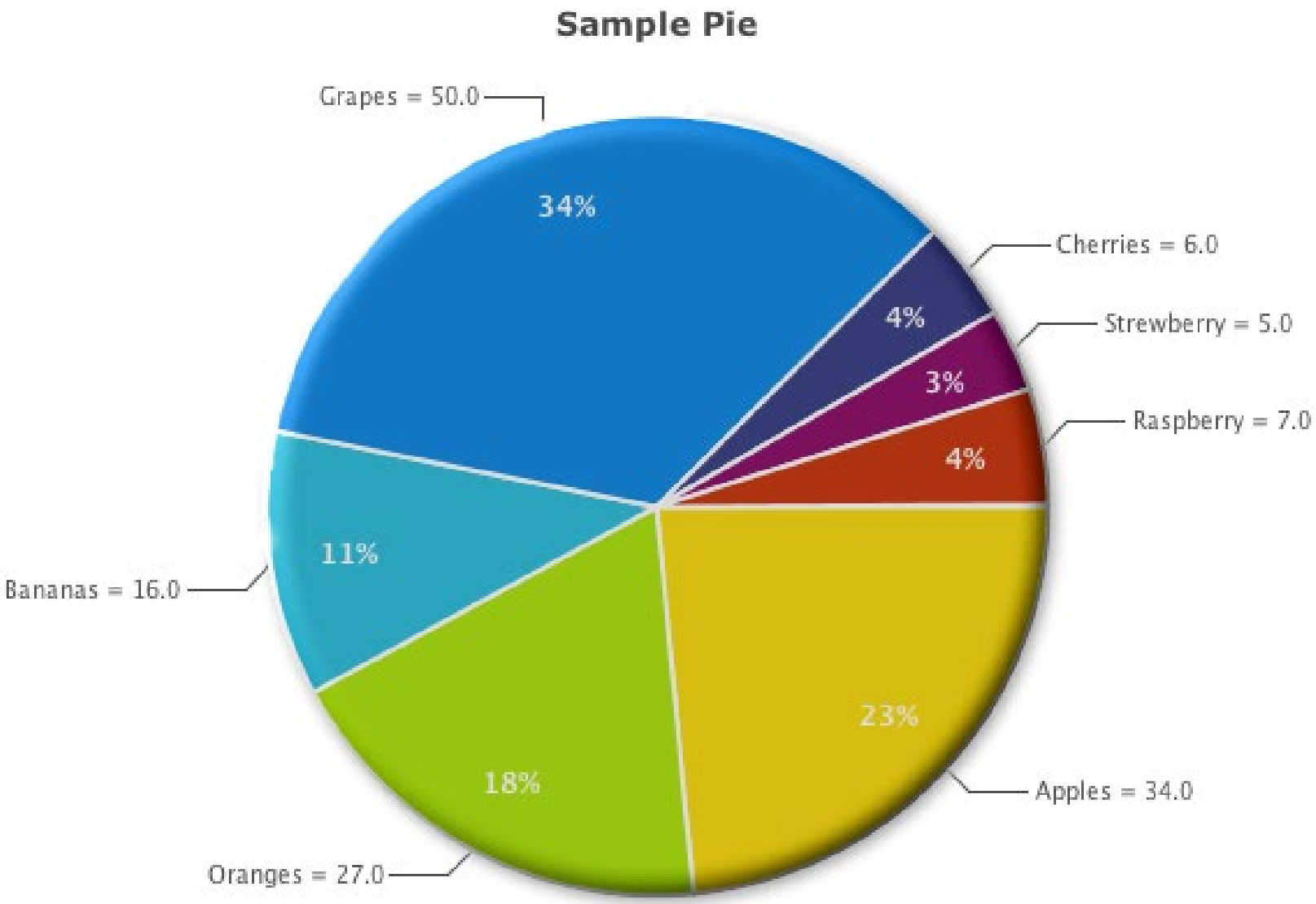
Wichtige Pakete des JavaFX

Package	Beschreibung
javafx.animation	Animationen auf Basis von Übergängen
javafx.application	grundsätzlicher Lebenszyklus einer Applikation
javafx.collections	essentielle JavaFX-Collections samt Utilities
javafx.event	Framework für JavaFX-Events & deren Behandlung
javafx.fxml	Laden einer Objekthierarchie aus einem Markup
javafx.geometry	Klassen für geometrische Operationen
javafx.scene	Die Basisklassen für das JavaFX Scene Graph
javafx.stage	Top-Level-Container-Klassen für JavaFX-Inhalte
javafx.util	Verschiedene Utilities und Hilfsklassen

neue Möglichkeiten

Hardware-integrierte Lösungen

DYNAMISCHE ECHTZEIT-INHALTE



DATENDARSTELLUNG

HelloWorld in JavaFX

HelloWorld-JavaFX-Applikation – leeres Fenster

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Fenster extends Application {
   @Override public void start(Stage primaryStage) {
      primaryStage.setTitle("Ein JavaFX-Fenster");
      primaryStage.show();
   }

   public static void main(String[] args) {
      launch(args);
   }
}
```

HelloWorld-Applikation

```
import javafx.application.Application;
import javafx.stage.Stage;

public class HelloJavaFX extends Application {
   @Override public void init() {
   }
   @Override public void start(Stage stage) {
      stage.show();
   }
   @Override public void stop() {
   }
   public static void main(String[] parameters) {
      launch(parameters);
   }
}
```

HelloWorld-Applikation

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.control.Label;

public class HelloJavaFX extends Application {
   private Label label;
   @Override public void init() {
      label = new Label("Hello World");
   }
   @Override public void start(Stage stage) {
      stage.show();
   }
   @Override public void stop() {
   }
   public static void main(String[] parameters) {
      launch(parameters);
   }
}
```

HelloWorld-JavaFX-Applikation

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;
import javafx.scene.control.Label;

   /* ... */
   @Override public void start(Stage stage) {
      StackPane root = new StackPane();
      root.getChildren().add(label);
      Scene scene= new Scene(root, 300, 200);
      stage.setTitle("Hello JavaFX Example");
      stage.setScene(scene);
      stage.show();
   }
   @Override public void stop() {
   }
   public static void main(String[] parameters) {
      launch(parameters);
   }
}
```

LITERATURANGABEN

Guido Krüger: Handbuch der Java-Programmierung, Addison-Wesley Verlag, ISBN 3-8273-2201-4

Elke Niedermair / Michael Niedermair: Internet-Programmierung mit Java, Data Becker GmbH & Co. KG, ISBN 3-8158-2086-3

Alexander Niemann: Das Einsteigerseminar, Objektorientierte Programmierung in Java, bhv Verlag Bürohandels- und Verlagsgesellschaft mbH, ISBN 3-8287-1015-8

Java 2 SDK v 1.2.2, Grundlagen Programmierung, HERDT-Verlag für Bildungsmedien GmbH, Nackenheim