Der JX - Windowmanager

Vortrag im Rahmen
des Praktikums AKBP II

am 22.03.2004

von

Johannes Ostler

j@schnitterin.de

Gliederung des Vortrages

1. Der JX - Windowmanager
2. Anbindung der AWT - Implementierung an den Windowmanager
3. Was wurde im Praktikum verändert
4. Zukunftsaussichten

1. Der JX - Windowmanager

- Der JX - Windowmanager entstand im Rahmen der Studienarbeit von Jürgen Obernolte im Jahr 2002.
- Nähere Informationen unter:
  http://www.jxos.org/publications.html

- Jürgen Obernolte:
  Entwurf und Implementierung eines Windowmanagers für das Java-Betriebssystem JX. Studienarbeit an der Universität Erlangen-Nürnberg, Februar 2002

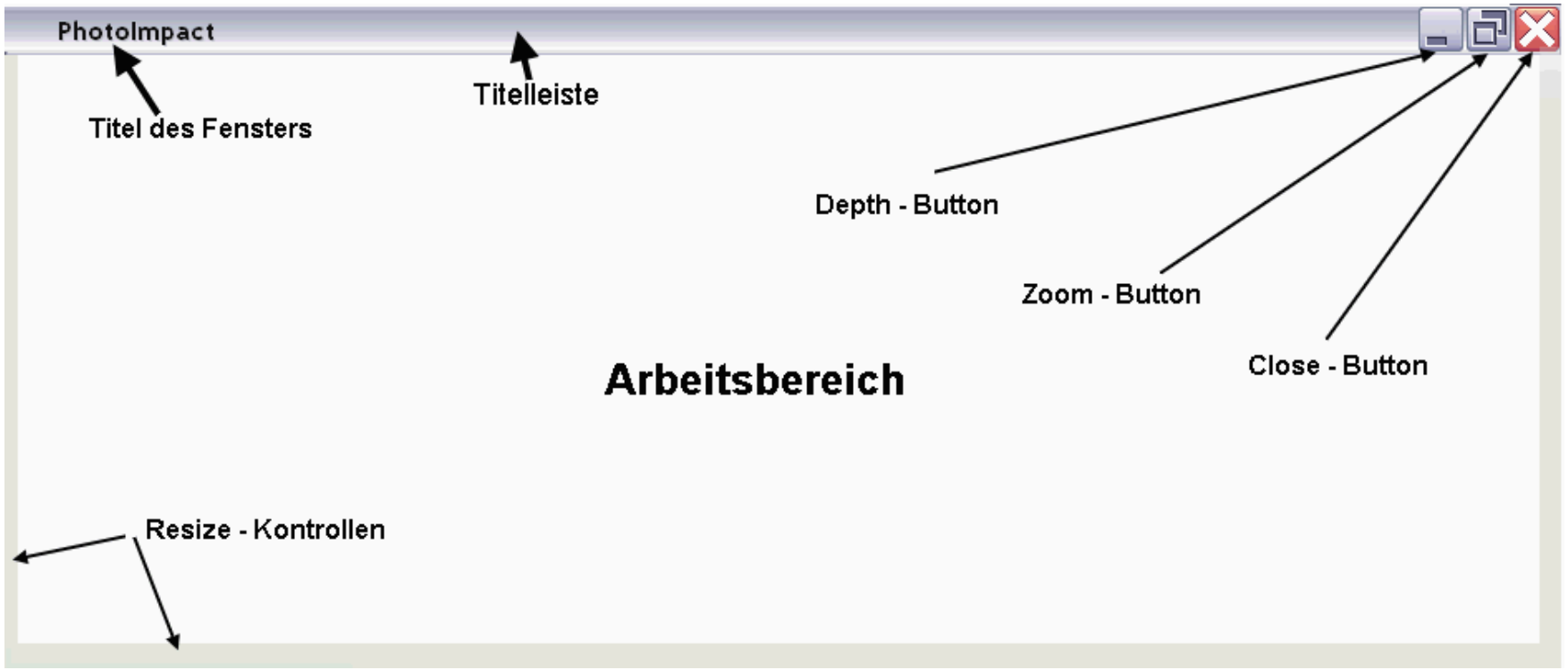
1.1 Aufgaben eines Windowmanagers

Die Aufgabe eines Windowmanagers ist die Verwaltung von Fenstern, z.B.:

- Verschieben
- Größenänderungen
- Grundlegende Zeichenfunktionen
- Aktivierung, Deaktivierung

Nicht zu den Aufgaben gehört das Zeichnen spezieller grafischer Komponenten wie Buttons oder ähnliches.

1. 2 Bereiche eines Fensters



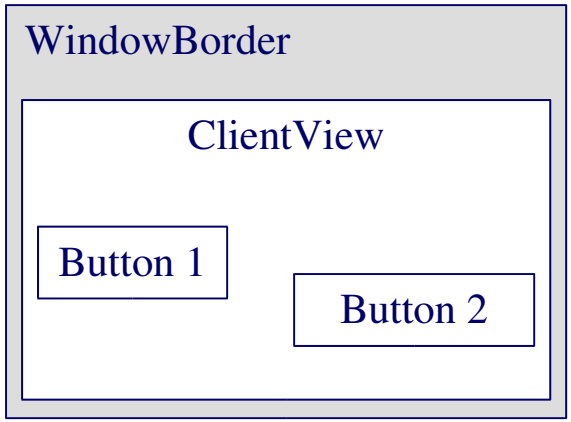
1. 3 Wichtige Klassen

- WView
  - rechteckiger Bereich auf dem Bildschirm
  - Views sind hierarchisch aufgebaut
  - Jede View speichert, welche Bereiche ihrer Fläche neu gezeichnet werden müssen

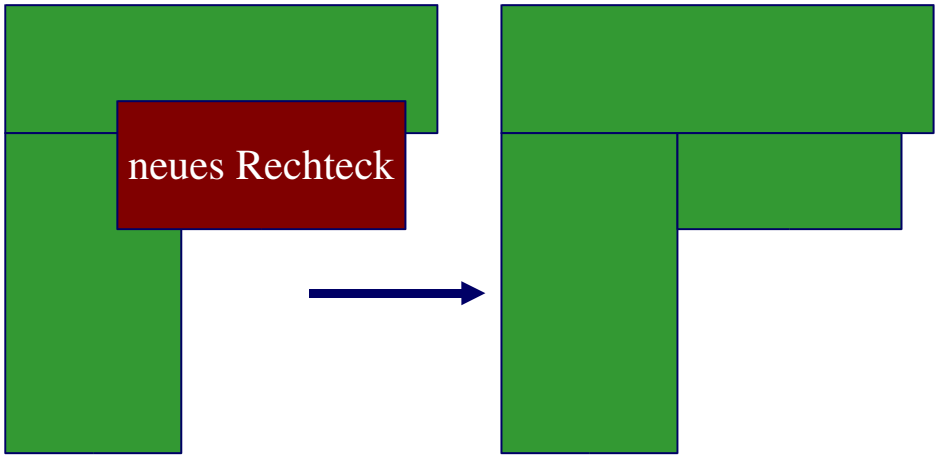
- WRegion
  - Menge von Rechtecken
  - Wird eine neues Rechteck hinzugefügt, so wird eine disjunkte minimale Überdeckung der Vereinigung aller Rechtecke gespeichert.
  - die beschädigten Bereiche einer View werden als Region gespeichert.

1. 3 Wichtige Klassen

Hierarchische Anordnung
von Views



Hinzufügen eines neuen
Rechteckes in eine Region



1. 3 Wichtige Klassen

- FrameBufferDevice
  - stellt Verbindung zur Hardware her
  - Grafikkartentreiber muss dieses Interface implementieren

- WDisplay
  - stellt Verbindung zum FrameBufferDevice her
  - WBitmap m_cScreen stellt den Bildschrim dar
  - stellt Methoden zur Mausbewegungen zur Verfügung

1. 3 Wichtige Klassen

- WWindow
  - implementiert Runnable
  - Thread arbeitet eintreffende Nachrichten ab
  - Verbindung zum Fenster durch Objekt des Typs WWindowInterface
  - Verbindung zum WindowManager, dem „Fenstererzeuger“

- WindowManager und WindowManagerImpl
  - Erzeugen neuer Fenster
  - Weitergabe von Events an die Fenster
  - Schnittstelle zum System

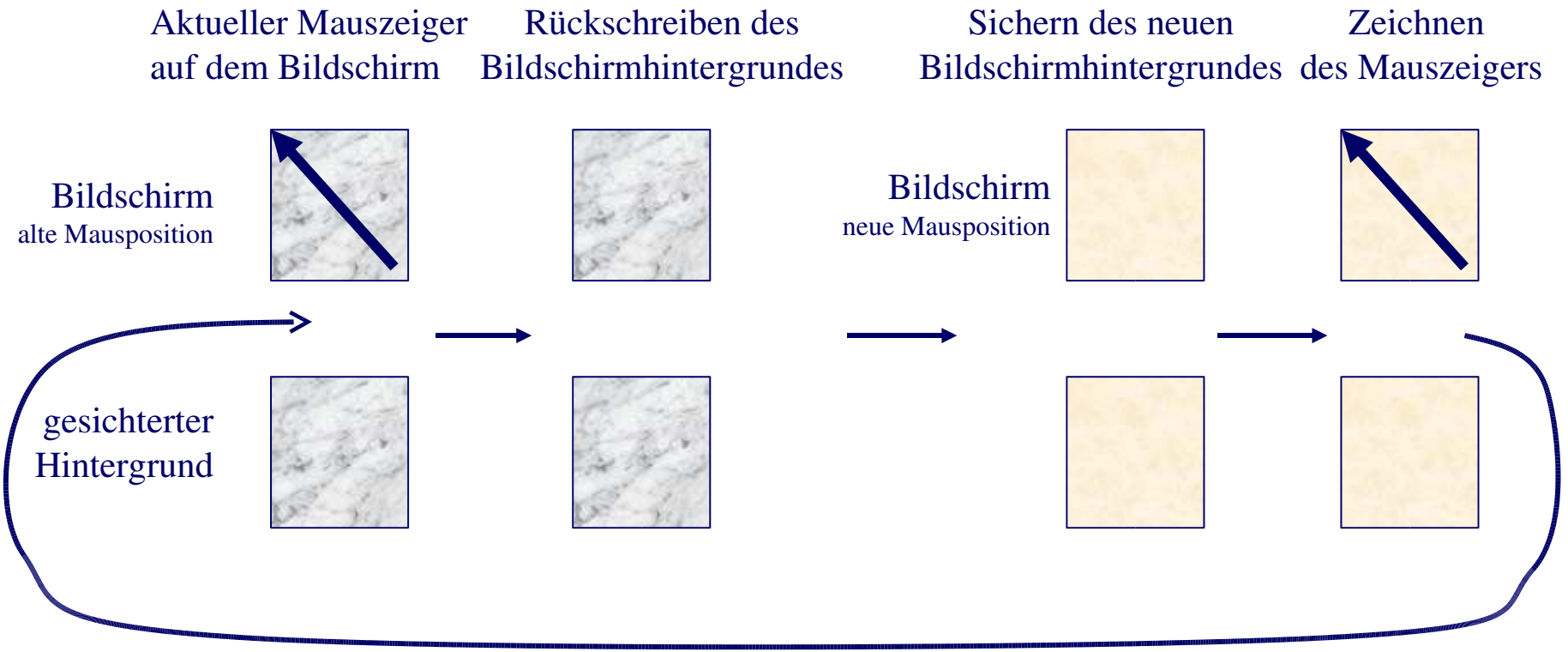
1. 3 Wichtige Klassen

- WindowBorder
  - View der Größe des Fensters, mit Arbeitsbereich als Clientview
  - ordnet den Mausereignissen ihre Bedeutung zu

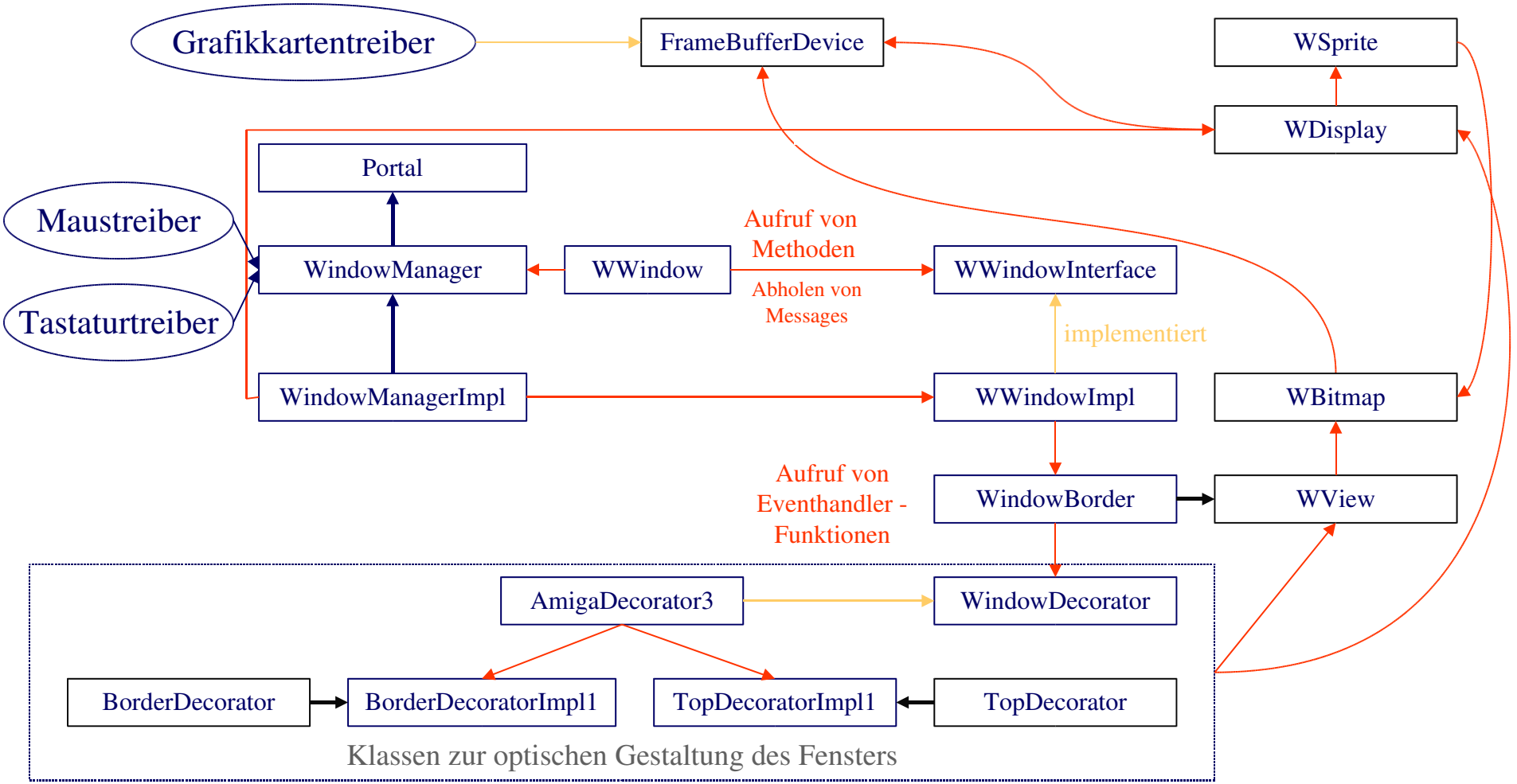
- WWindowImpl
  - bei Maus und Tastaturereignissen werden statische Handlerfunktionen dieser Klasse aufgerufen
  - weist Ereignisse dem jeweiligen Fenster, d.h. der jeweiligen Instanz zu
  - behandlet das Ereignis und sendet eventuell Message an WWindow

1. 3 Wichtige Klassen

- WSprite
  - realisiert den Mauszeiger



1. 4 Überblick über die wichtigsten Klassen



2. Anbindung des AWT an den Windowmanager

- Die AWT - Implementierung entstand im Rahmen der Studienarbeit von Marco Winter im Jahr 2002.

- Nähere Informationen unter:
  http://www.jxos.org/publications.html

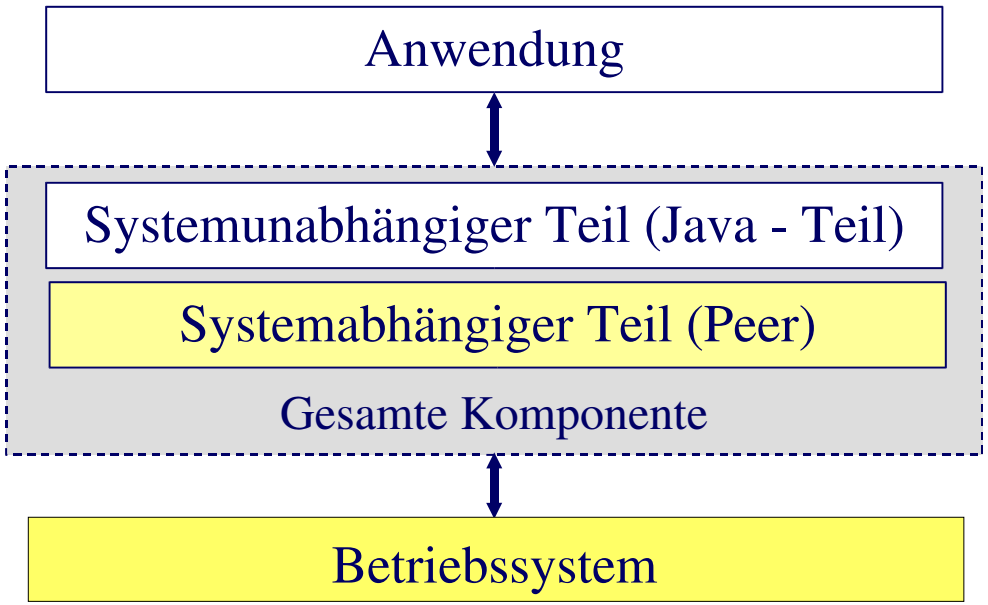
  - Marco Winter:
    Design und Implementierung der AWT - Schnittstelle für das Java - Betriebssystem JX, Oktober 2002

2.1 Was ist ein AWT

- AWT steht für Abstract Window Toolkit.
- Das AWT ist eine Klassenbibliothek für grafische Anwendungen.
- Die Klassen des AWT sind für den Anwender plattformunabhängig.
- Die Darstellung der einzelnen Komponenten hängt von der grafischen Oberfläche des Systems ab.

2.2 Peerkonzept

- Die Verbindung zwischen AWT und dem Windowmanager erfolgt durch die sogennanten Peer- und Connectorklassen



2.3 Wichtige Klassen

- JXToolkit
  - erzeugt die JX - Peer - Objekte

- GerneralConnector
  - Verbindung zum Windowmanager
  - von WWindow abgeleitet und kann somit u. a. auf die Zeichenfunktionen zugreifen

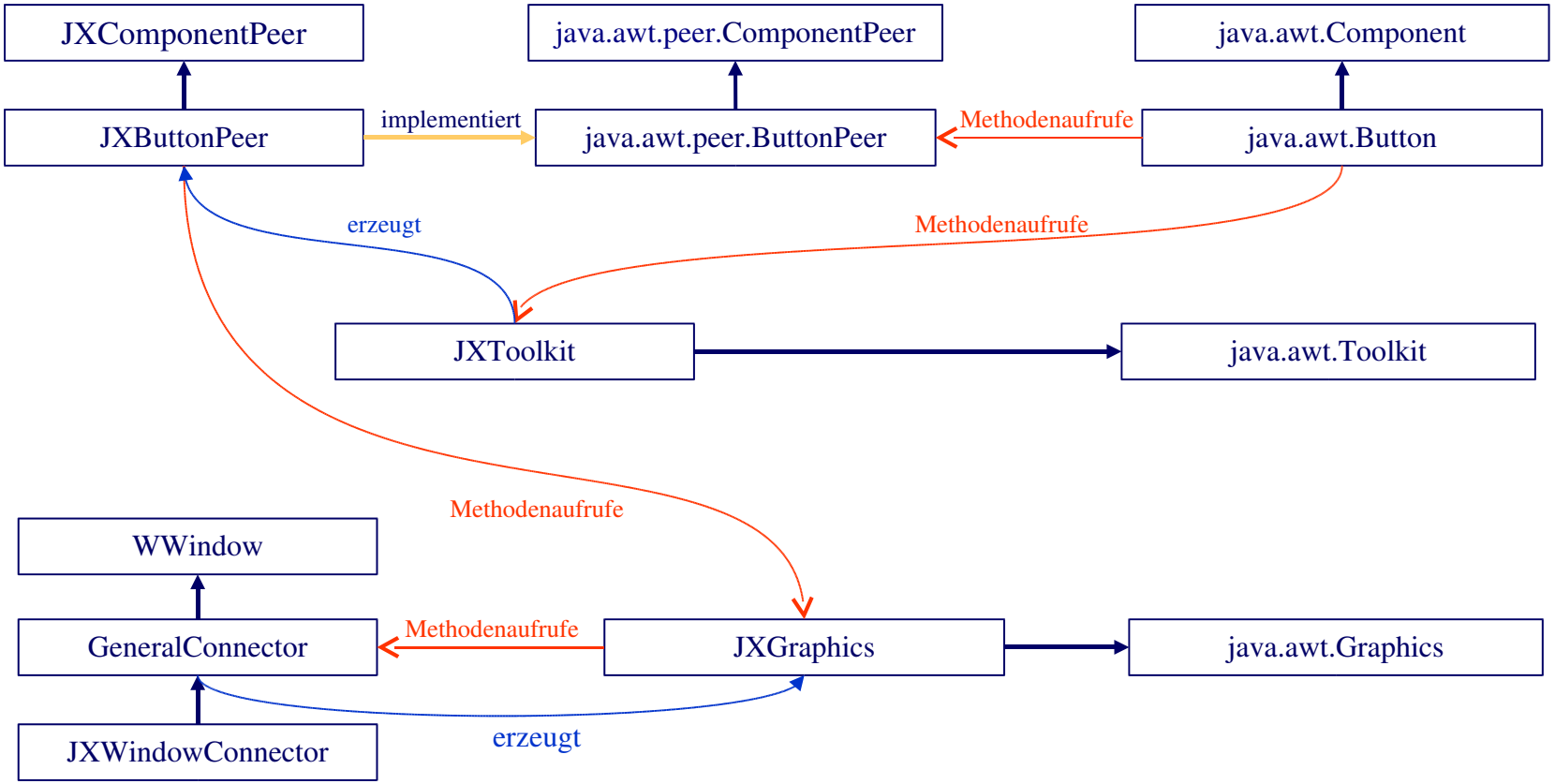
- JXWindowConnector
  - von GeneralConnector und somit auch von WWindow abgeleitet
  - überschreibt die Methoden von WWindow, die beim Auftreten von Ereignissen aufgerufen werden
  - gibt Events an die AWT - EventQueue weiter

2.3 Wichtige Klassen

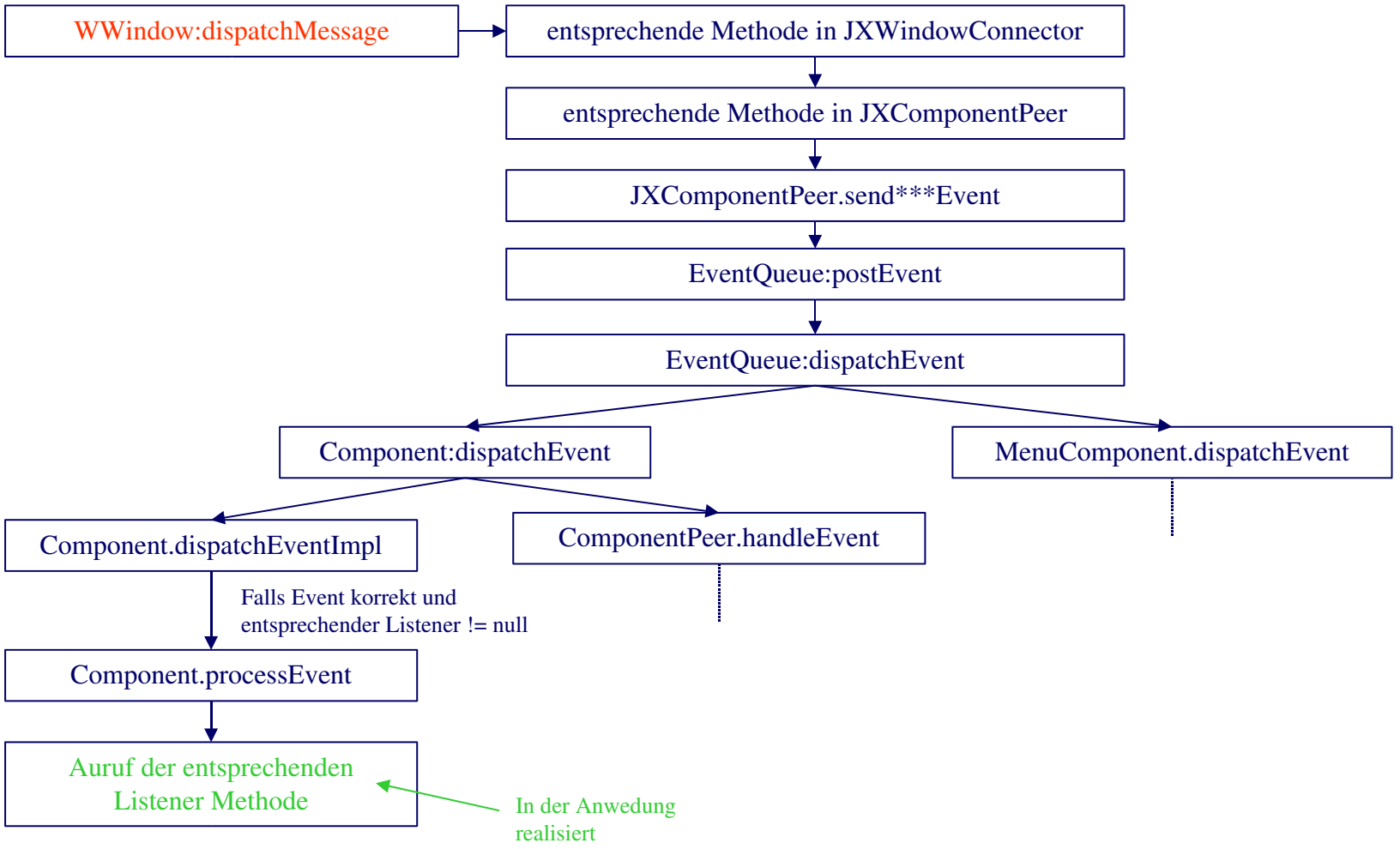
- JXGraphics
  - Verbindung zu den graphischen Funktionen des Windowmanagers
  - von java.awt.Graphics abgeleitet und kann somit als Graphics - Objekt an awt - Klassen übergeben werden
  - ist durch eine GeneralConnector - Objekt mit dem Windowmanager verbunden

- EventQueue
  - Herzstück der AWT - Eventverarbeitung
  - Erhält von JXWindowConnector die aufgetretenen Events
  - EventDispatchThread arbeitet die Events ab, indem er dispatchEvent von Component oder MenuComponent aufruft

2.4 AWT - Anbindung am Beispiel der Klasse Button



2.5 AWT - Eventhandling im Überblick



3.Was wurde im Rahmen des Praktikums verändert

3.1 Wer war beteiligt

- Marc Lörner, Marc.F.Loerner@informatik.stud.uni-erlangen.de
- Johannes Ostler, Johannes.J.Ostler@informatik.stud.uni-erlangen.de

3.2 Zielsetzung

- optische Verbesserung
- Leichteres Ändern der Optik
- Verbesserung der Performance beim Verschieben der Fenster

3.3 Optische Veränderungen

- Titelleiste
  - Klasse TopDecorator
  - Farbverläufe als Hintergrund
  - Buttons werden als Bitmap abgespeichert

- Rahmen
  - Klasse BorderDecorator
  - Rahmen können mit unterschiedlicher Breite erzeugt werden

3.4 Klassen zur Fenstergestaltung

3.5 Mauszeiger

- Mauszeiger können direkt aus ppm - Dateien eingelsen werden
- Verschiedene Mauszeiger werden beim Start des Windowmanagers in Bitmaps abgespeichert
- Bei Veränderung des Zeigers wird einfach die jeweilige Bitmap in Wsprite geändert

3.6 Verbesserung der Performance des Desktophintergrundes

- bisherige Implementierung
  - bisher war der Desktop als ein AWT - Frame implementiert
  - Nachzeichnen von Teilen des Hintergrundbildes erfordert PaintMessage an die AWT

- jetzige Implementierung
  - Hintergrundbild ist als Bitmap im Windowmanager verankert
  - Nachzeichnen wird direkt im Windowmanager, ohne Kommunikation mit der AWT vollzogen

3.7 Veringerung der Kommunikation mit dem AWT

- bisherige Implementierung
  - Bei jeder Mausbewegung wurde eine Message an das AWT geschickt.
  - Das hohe Aufkommen von Nachrichten über Mausbewegungen verschlechterte die Performance des Systems.

- jetzige Implementierung
  - Nachrichten werden nur noch geschickt, wenn sich die Maus über der Arbeitsfläche befindet, bzw. bei mouseUp vorher auf die Arbeitsfläche geklickt wurde .

4. Zukunftsaussichten

- Tool zum Einstellen von Desktopthemes, bzw Datei
- Weitere Reduktion der Kommunikation mit der AWT
- Die Möglichkeit von Views mit Speicher
- Verbesserung der AWT - Eventverarbeitung