

Scope Bereiche Øim Quelltext, in dem ein Identifier

Der Scope einer Definition eines Identifiers ist der Bereich im Quelltext, in dem der Identifier gemäß dieser Definition angesprochen und verwendet werden kann.

nur Subtypen oder auch keine

Java: iteratör oder for

Rückf^ür rekursion

begrenzte Ansicht

Basiskonzept

packages

- + Zugriffsbeschränkung
- + Gruppenen
- + Namenskonflikte

Stream gemeinsame Schnittstelle für Arrays, Listen, Dateien, ^(un-) endliche Sequenzen

Generics Einschränkungen

FOP

funktional deklarativ
referentielle Transparenz
keine Zeit

objektorientiert + imperativ

Parallogramm
zentrale Bausteine

Objekte und Klassen

Objekte haben Zustand (Zeittakt Rennfolge)

Design

Funktionen
gegenseitiger Aufruf Funktionen
Funktionen höherer Ordnung

Zerlegung in

Ablauf ist

Variation durch

Klassen und Objekte

Interaktion Objekt

Vererbung

< Konzept = Klasse // repräsentiert

Implementierung

Abbildung u. Modellieren

(Objekt eines dynamischen Typs kann in Variable statischer Supertyps gespeichert werden)

dynamische Typisierung

üblicherweise

statische Typisierung

// Typinferenz automatisch
statischen Typ erachteten

interpretiert
Racket

heterogen
eingebaut
konstant
rekursion

Listen

JVM interpretiert
Java Bytecode

kompliert

Java

homogen

Standardbibliothek

dynamisch
iterative

List
Zugriff nur durch Laufzähler
dynamisch
mehr Speicher für Referenzen

Array
direkter Elementzugriff.

konstante Größe

intermediate \leftrightarrow terminal operation

~~list & Stream~~
~~Stream~~
stream, collect()
Collectors.toList()

Stream
gemeinsame Schnittstelle
+ kann unendlich,Lazy evalvuated
nur ein Durchgang

stream.of(...)
Arrays.stream(a);
stream.ofAny(T::new);

java.util

interface Collection<T>

boolean add
int size()
boolean ~~contains~~ contains
boolean remove()

Iterator<T>
boolean hasNext()
+ next()
remove()

Collection<? extends List<T> > lst) // stabiles Sortieralgorithmen

List<T> int index(T object element)
+ set(int i, T element)
+ get(int i)

Map<K, V>
key distinct
Hashmap
put (K key, V value)
get (K key)

Optional<E> static ofNullable (E element)
E get()
E orElseGet
boolean isPresent (Consumer)

map
filter

Path Path.get (String first, String... more)

Files stream (String) Files.lines (Path path)
exists
isReadable, isWritable

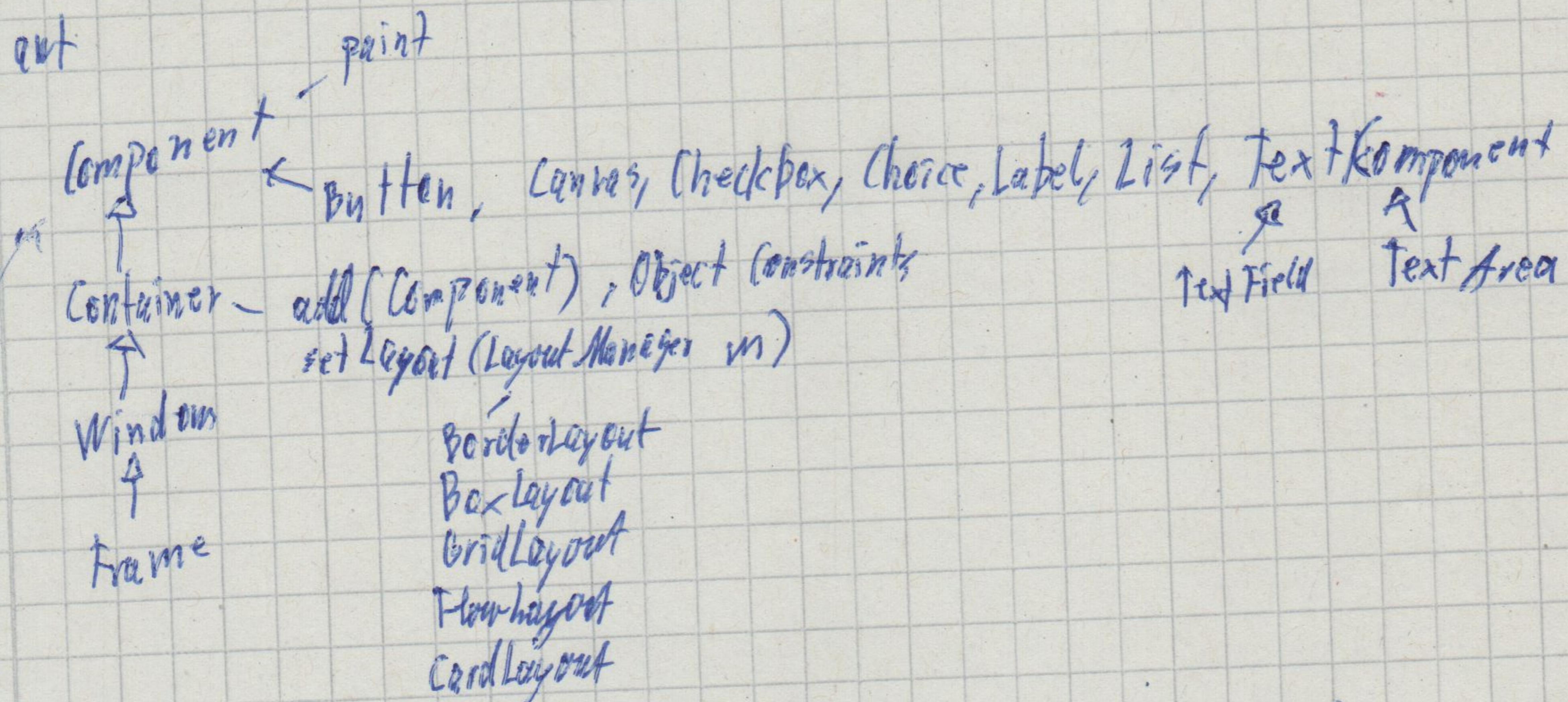
Byte

InputStream
BufferedInputStream
FileInputStream (String filename)

text

Reader r = new BufferedReader (new FileReader (String filename))
/ / InputStream
readLine()
write (String)

Window Manager, Systemprozess



Listener Event Dispatch Thread wartet permanent auf Inputs ruft registrierte Listener auf

abstract Adapter Klasse alle Methoden mit leerem Rumpf
add... Listener(--Listener l)

Graphic
getColor
fillRect
drawOval
drawString Rectangle
getFontMetrics()
getMaxAscent()
getMaxDescent()
StringW, Hh(String s)

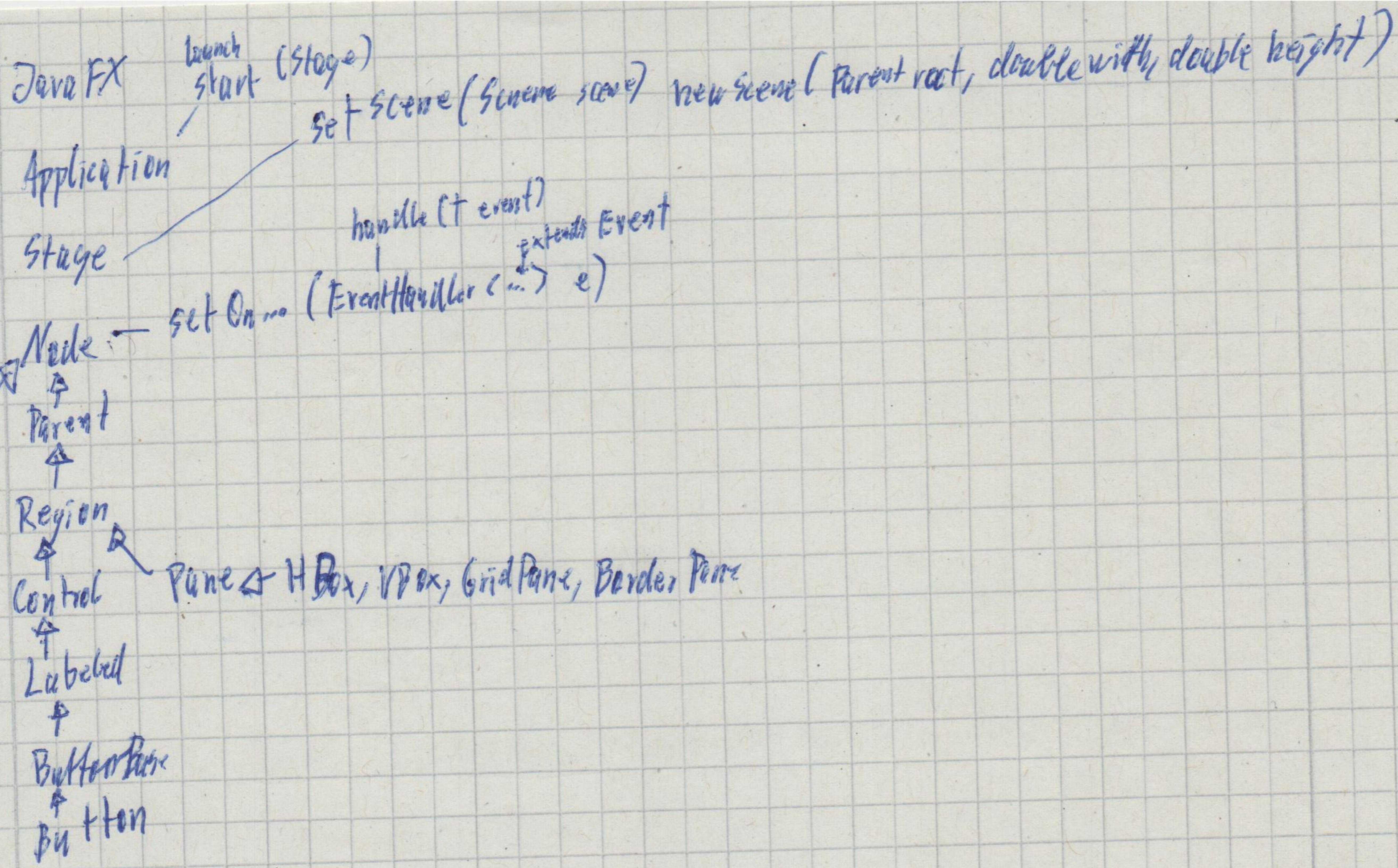
Swing UIManager static setLookAndFeel(String name)
getLookAndFeel()

Frame JFrame (+Separierung Hauptmenü von Rest des Fensters)

Container & Component
setToolTip(String)
setBorder(Border b)

JLabel
JList
JScrollBar
JToolBar
JSlider
PopupMenu
JTable
JScrollPane
getInputMap().put(KeyStroke k, Object actionMapKey)
getActionMap().put(Object actionMapKey, Action action)
void actionPerformed(ActionEvent e)

AbstractButton (< JButton, JToggleButton, JCheckBox, JRadioButton
JTextComponent (< JTextArea, JTextField, JFormattedTextField, JPasswordField



Canvas (erst mit logische Koordinaten)

- GraphicsContext getGraphicsContext2D()
- Paint each Bitmasker
- new Affine()
 - scale
 - rotate
 - translate
 - shear

Properties + Bindings

interface Property<T> \leftarrow NumberProperty

bind(ObservableValue<T> v)

NumberBinding add(...)

add
subtract
multiply
divide

ObservableValue<T>

Bindings

NumberBinding add(...)

lessThan

ObservableBooleanValue and

ObservableNumberValue Bindings.when(ObservableBooleanValue -> then(Ob.N.V.-), otherwise(Ob.N.V.-))

Korrekte Software

Kriterien:

keine Fehler

Termination (nach Aufgabe oder auf Befehl)

funktionale Anforderungen

zeitlicher

Abstraktionsebenen

Spezifikatorisch

Logisch z.B. $y = \tan x$

Semantisch z.B. $x=0; y=\frac{1}{x}$

Compiler Syntaktisch z.B. Klammerersetzung, syntaktische Konstrukte kontextfreie Regeln statement ::= ... | ...

Lexikalisch Identifier Regeln, while statt while

Klassen

Darstellungsinvariante Sicht durch public ... vermittelt

Implementierungsinvariante analog für nicht-public f:Attribut .. reutyp = hat ..

Vererbung muss private und protected übernehmen erweitern, verfeinern, aber nichts zurücknehmen

Liskov Substitution Principle

Ein Objekt der Klasse ... repräsentiert zu jedem Zeitpunkt seiner Lebenszeit (generelle Beschreibung) (wichtige Informationen, Beziehungen)

Subventionen

beidyn.

Vertrag: Typ Precondition Returns Postcondition