

Android-Apps

Gegenüberstellung

Delphi ↔ java

Grundsätzliches etc.

	Delphi	java
	Groß-/Kleinschreibung egal	... sensitiv!
	Zuweisung	
	<code>:=</code>	<code>=</code>
	Kommentare	
	<code>//</code>	<code>//</code>
	<code>{ ... }</code>	<code>/* ... */</code>
	Zeichenkettenbegrenzer:	
	<code>'text'</code>	<code>"text"</code>
	Inkrementieren / Dekrementieren	
	<code>inc(zahl); dec(zahl)</code>	<code>zahl++; zahl--</code>
	<code>inc(zahl, plus); dec(zahl, minus)</code>	<code>zahl += plus; zahl -= minus (auch *=, /=)</code>
	<code>succ(zahl); pred(zahl)</code>	<code>++zahl; --zahl</code>

Deklaration

Delphi	java
var (einleitendes Schlüsselwort zu Beginn der unit, function, procedure)	Deklaration überall möglich
ganzzahl : Integer;	int ganzzahl;
zahl : Integer = 6; (nur global)	int zahl = 6;
zusätzlich existieren Cardinal, Byte, Word, Int64	zusätzlich existieren byte, short und long und als Objektklassen die Typen Integer und BigInteger
	Integer intobj = new Integer(3);
kommazahl : Real;	float kommazahl;
zus.: Single, Double, Extended	zus.: double und die Klasse BigDecimal
zeichen : Char = 'Q';	char zeichen = "Q";
text : String; (bis 255 Zeichen)	String text = "demo"; ist eine Konstante, reicht aber häufig (z.B. zur Ausgabe)
	String text = new String("demo"); Ist eine neue Instanz der Klasse

Deklaration

Delphi	java
flag : Boolean;	boolean flag;
feld : array [von .. bis] of Typ; → feld[von], feld[von+1], ...	float[] feld = new float [anzahl]; → feld[0], ... , feld[anzahl-1]
verbund : record ... end; → verbund.komponente	class verbund { ... } → verbund.komponente
richtung : (nord,west,sued,ost);	kein Grundtyp, enum als Objektklasse
const MAX = 999;	final int MAX = 999;
	java hat eine Neigung zu double, häufig hilft nur der Hammer: float zahl = 2.1f;

Entscheidungsstrukturen

I
N
F
O
R
M
A
T
I
K

Delphi	java
If Bedingung then Anweisung;	If (Bedingung) Anweisung;
If Bed then Anw1 else Anw2;	If (Bed) Anw1; else Anw2;
<pre> case zaehlvar of wert1 : Anw1; wert2 : Anw2; wert3 .. wert4 : Anw3; ... else Anw; end; </pre>	<pre> switch (zaehlvar) { case wert1: Anw1; break; case wert2: case wert3: Anw2; break; ... default: Anw; break; } </pre>
Es wird immer genau ein Fall abgearbeitet.	Beginn bei Übereinstimmung bis zum Abbruch! <i>Beliebte</i> Fehlerquelle.
	Statt einer zählbaren Variable kann auch ein String stehen.
Statt einer Anweisung kann ein	Anweisungsblock stehen
begin ... end	{ ... }

Schleifen

	Delphi	java
I	while Bed do Anw;	while (Bed) Anw;
N	repeat Anweisungen until Abbruchbed;	do Anw; while (Bed);
F		
O		
R	for zaehler := startzahl to zielzahl do Anw;	for (init; Laufbedingung; Operation) Anw; Bsp: for (int i=0; i<n; i++) Anw;
M		
A		
T		
I		
K		

Operatoren

	Delphi	java
I		
N	+ - * /	+ - * /
F	div, mod	/, %
O	=	==
R	<>	!=
M	<, >, <=, >=	<, >, <=, >=
A	and	&& (& Langauswertung)
T	or	(Langauswertung)
I	not	!
K		Vorsicht bei Objektklassen String, Integer, ... (nicht die Objekte vergleichen, sondern ihre Werte!)
		if (integer1.equals(integer2)) ... If (text1.compareTo(text2) > 0) ...

Typenkonvertierung

I
N
F
O
R
M
A
T
I
K

Delphi	java
Fließkommatypen untereinander beliebig, Ganzzahltypen ebenso (evt. Verlust). Ganz zu Fließ direkt, zurück nur per Funktion.	Wenn Typen verwandt sind: Aufwärts direkt (implizit), abwärts casten per Klammer (explizit).
<pre>ganz := 2; kommasingle := ganz; kommadouble := kommasingle; kommasingle := kommadouble; ganz := trunc(kommasingle);</pre>	<pre>int z0 = 2; float z1 = z0; double z2 = z1; float z3 = (float) z2; int z4 = (int) z3; (ganzzahliger Anteil)</pre>
<pre>byte3 := byte1 + byte2; (verlustbehaftete) Konvertierung bei Zuweisung.</pre>	<pre>byte byte3 = (byte) (byte1 + byte2); Umwandlung muss erzwungen werden, sonst Typ mindestens int.</pre>
IntToStr(ganzzahl), FloatToStr(zahl)	String.valueOf(egalzahl)
FloatToStrF(zahl,ffFixed,8,3)	String.format("%,.3f",zahl)
StrToFloat(zahlstring)	Float.parseFloat(zahlstring)
StrToInt(zahlstring)	Integer.parseInt(zahlstring)
	Manchmal geht es auch anders herum: <pre>Integer zahl = 3; String text = zahl.toString();</pre>

Function / Procedure

I
 N
 F
 O
 R
 M
 A
 T
 I
 K

Delphi	java
<pre>function berechne(para: Integer):String; var ... begin ... result := ergebnis; end;</pre>	<pre>private String berechne (int para) { ... return ergebnis; }</pre>
<p>Alternativ kann statt result auch der Funktionsbezeichner verwendet werden.</p>	<p>Methode der Klasse mit Rückgabewert</p>
<pre>procedure tuwas(var para: Integer); var ... begin ... para := ergebnis; end;</pre>	<pre>private void tuwas (int para) { ... }</pre>
<p>Wird benutzt bei mehreren Ergebnissen oder wenn nur etwas ausgeführt werden soll</p>	<p>Call by reference nicht möglich. ggf. Rückgriff auf globale Variablen. Objekte (z.B. Felder) sind aber „Referenztypen“, deren Komponenten geändert werden können.</p>