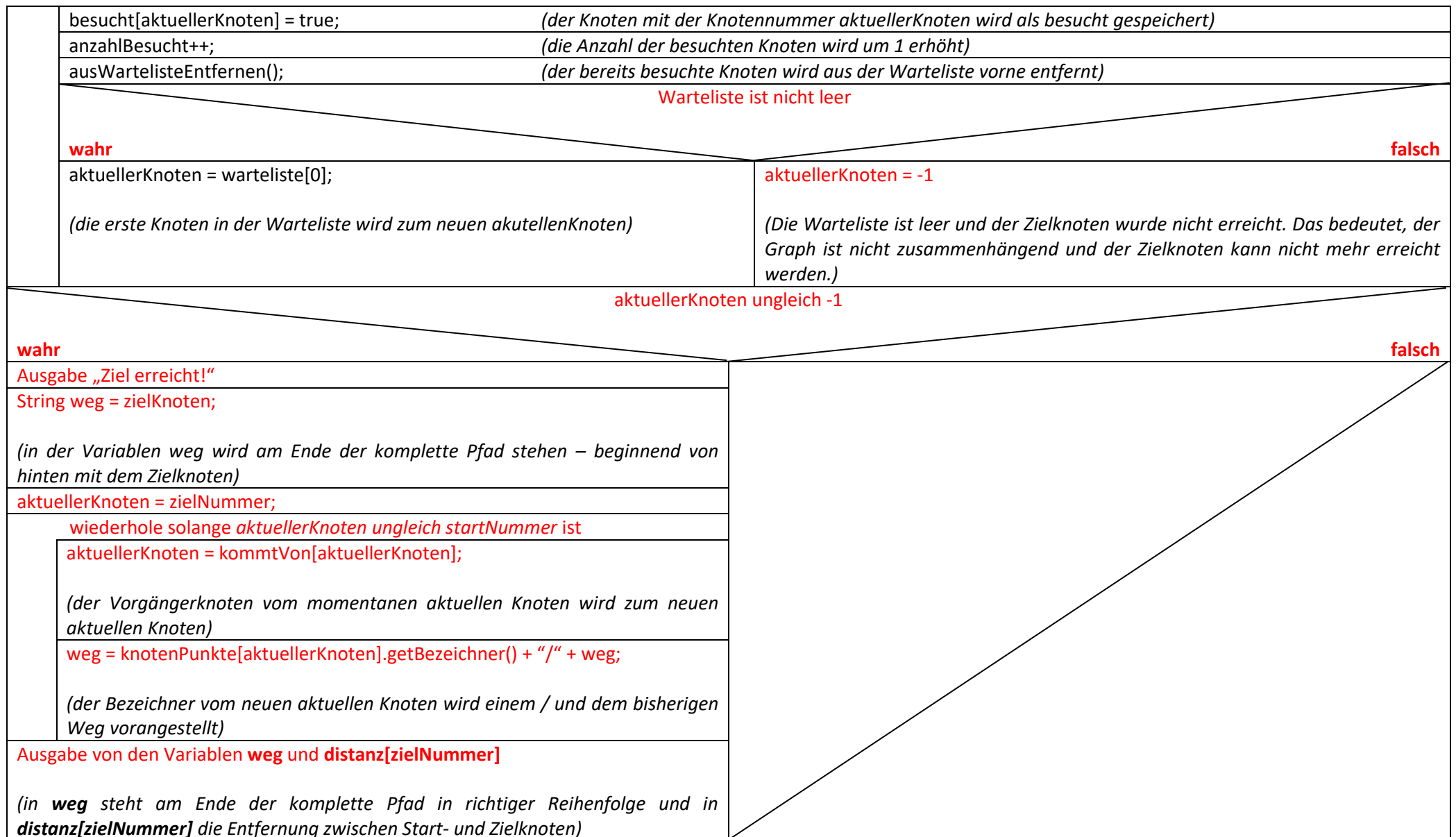


## Algorithmus für die erweiterte Breitensuche

public void **breitensucheErweitert**(String **startKnoten**, String **zielKnoten**)

int startNummer = getKnotenNummer(startKnoten);	(bestimme die Knotennummer des Startknotens und speichere diese unter startNummer ab)
<b>int zielNummer = getKnotenNummer(zielKnoten);</b>	(bestimme die Knotennummer des Zielknotens und speichere diese unter zielNummer ab)
int aktuellerKnoten = startNummer;	(setze den aktuellen Knoten auf die startNummer)
anzahlWarteliste = 0;	(setze die Anzahl der Knoten in der Warteliste auf 0)
anzahlBesucht = 0;	(setze die Anzahl der besuchten Knoten auf 0)
setze die Warteliste zurück	(alle Indixe im Feld warteliste erhalten den Wert -1)
setze die Besuchliste zurück	(alle Indixe im Feld besucht erhalten den Wert false)
<b>setze das Distanz-Feld zurück</b>	(alle Indixe im Feld distanz erhalten den Wert 0)
<b>setze das kommtVon-Feld zurück</b>	(alle Indixe im Feld kommtVon erhalten den Wert -1)
warteliste[0] = aktuellerKnoten;	(der aktuelle Knoten (= Startknoten) wird in die leere Warteliste an erster Stelle eingefügt)
<b>kommtVon[startNummer] = startNummer;</b>	(am Index startNummer im Feld kommtVon wird die startNummer gespeichert; daran erkennen wir, dass es der Startknoten ist, da er keinen Vorgängerknoten besitzt)
anzahlWarteliste++;	(die Anzahl der in der Warteliste eingefügten Knoten wird um 1 erhöht)
wiederhole solange <b>der aktuelleKnoten ungleich der zielNummer ist und der aktuelleKnoten ungleich -1 ist</b>	
zähle i von 0 bis anzahlKnoten - 1	
matrix[aktuellerKnoten][i] > 0 (es existiert eine Kante zwischen aktuellerKnoten und i) <b>und</b> der Knoten mit der Knotennummer i wurde noch nicht besucht <b>und</b> der Knoten mit der Knotennummer i ist noch nicht in der Warteliste	
<b>wahr</b>	<b>falsch</b>
warteliste[anzahlWarteliste] = i;	
(Knoten mit Knotennummer i wird in die Warteliste hinten eingefügt)	
anzahlWarteliste++;	
<b>distanz[i] = distanz[aktuellerKnoten] + matrix[aktuellerKnoten][i];</b>	
(zu der Distanz des momentan aktuellen Knotens wird das Kantengewicht zwischen aktuellerKnoten und dem Knoten i addiert)	
<b>kommtVon[i] = aktuellerKnoten;</b>	
(Knoten mit der Knotennummer i kommt vom Knoten mit der Knotennummer aktuellerKnoten – de Vorgängerknoten wird abgespeichert)	



Mit einer **erweiterten Breitensuche** kann man sowohl die Länge des Pfads (als Anzahl der auf dem Weg durchlaufenen Kanten) von einem Startknoten zu einem Knoten mit einer gewünschten Eigenschaft angeben als auch die Folge der Knoten auf diesem Pfad.