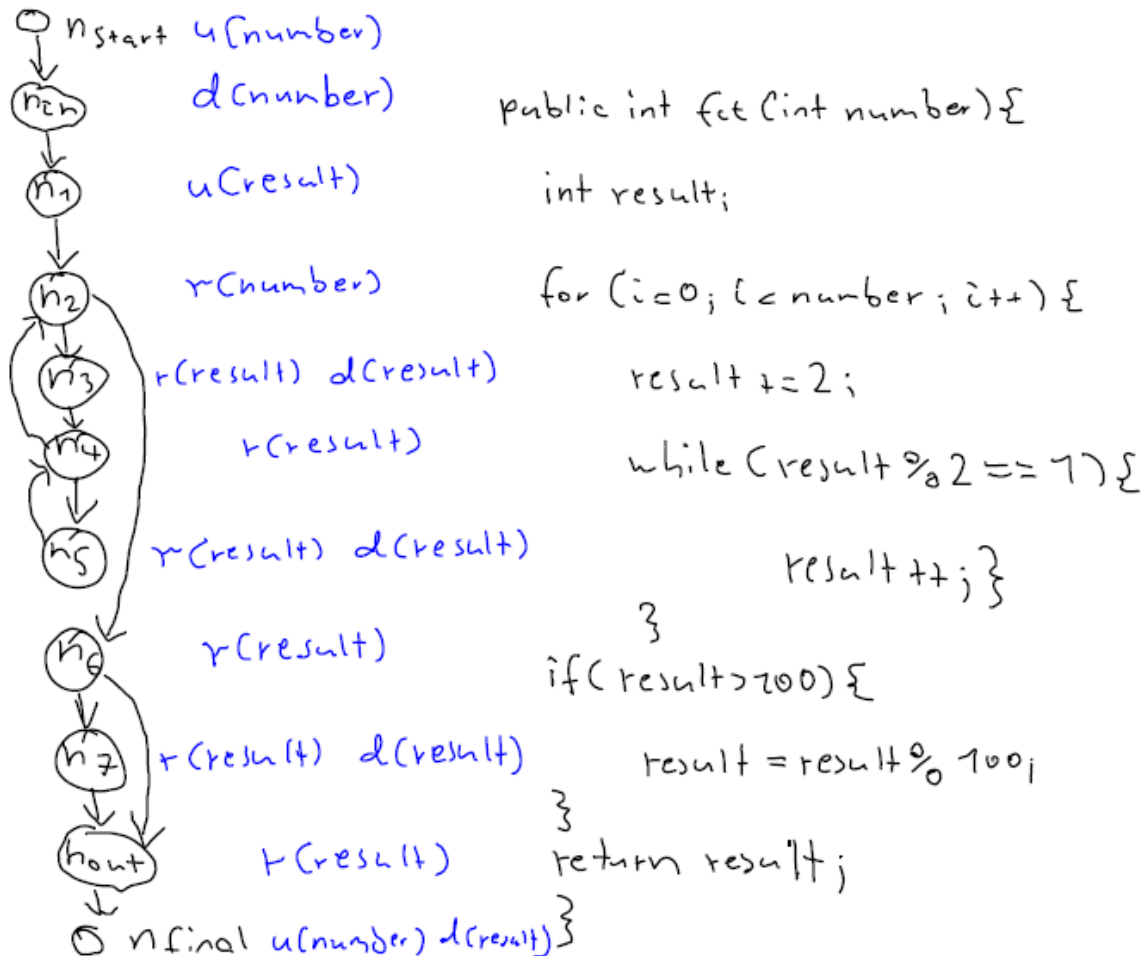


Aufgabe 2

a)



Datenflüsse

Number: $u \ d \ r \ [(r)^{\text{number}}] \ u \ d$; für $\text{number} \geq 1$

Result: $u \ [r \ d \ r \ [r \ d] \ (r \ d \ r)^{(\text{number}-1)} \ (r \ r \ d \ r \ | \ r \ r) \ u$; für $\text{number} \geq 1$

Datenanomalien

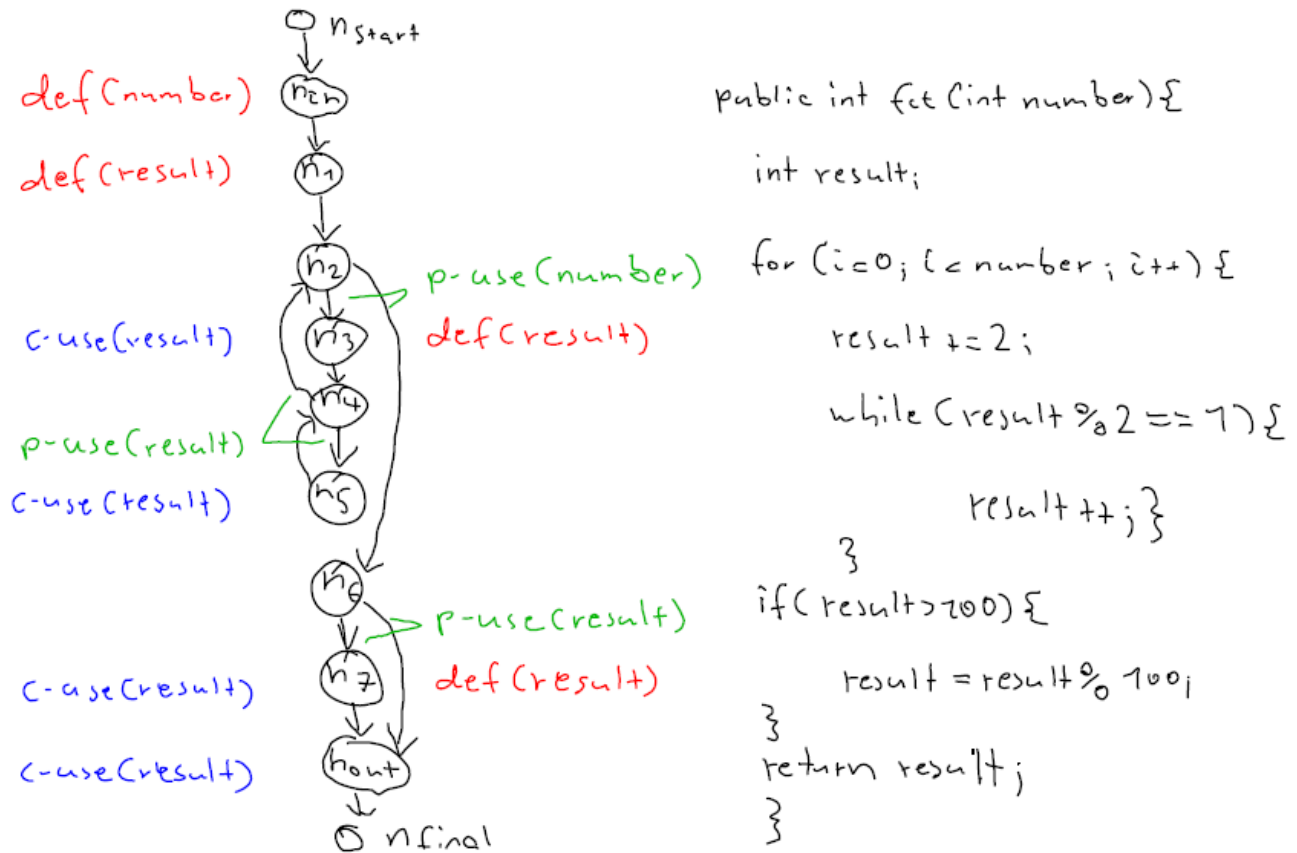
- ur-Anomalie zu Beginn des Datenflusses von *result*
--> kritisch, da in Java Variablen vor dem Zugriff initialisiert werden müssen
- falls *result* gerade ist, wird der Inhalt der while-Schleife unerreichbar (toter Code)
--> nicht kritisch, das Programm bleibt semantisch gleich

b)

- Nullpointer-Referenz
- Array-Length größer 0
- Gleiche Objekte müssen gleichen Hashcode haben
- Vergleich von Objekten (equals vs ==)
- Typkorrektheit (Casts)

Aufgabe 3

a)



b)

Variable x	Knoten n_i	$dcu(x, n_i)$	$dpu(x, n_i)$
result	n_1	$\{ n_3, n_7, n_{out} \}$	$\{ (n_6, n_7), (n_6, n_{out}) \}$
result	n_3	$\{ n_2, n_5, n_7, n_{out} \}$	$\{ (n_4, n_2), (n_4, n_5), (n_6, n_7), (n_6, n_{out}) \}$
result	n_5	$\{ n_2, n_3, n_7, n_{out} \}$	$\{ (n_4, n_2), (n_4, n_5), (n_6, n_7), (n_6, n_{out}) \}$
result	n_7	$\{ n_{out} \}$	$\{ / \}$
number	$n_{(in)}$	$\{ / \}$	$\{ (n_2, n_3), (n_2, n_6) \}$

c), d), f)

Eingabe: number = 1

result = 99

Ausgabe: result = 2

Pfad: $(n_{start}, n_{(in)}, n_2, n_3, n_4, n_5, n_4, n_2, n_6, n_7, n_{out}, n_{final})$

e)

1.

Eingabe: number = 1
result = 99
Ausgabe: result = 2

Pfad: $(n_{Start}, n_{(i\ n)}, n_2, n_3, n_4, n_5, n_4, n_2, n_6, n_7, n_{out}, n_{Final})$

2.

Eingabe: number = 1
result = 1

Pfad: $(n_{Start}, n_{(i\ n)}, n_2, n_3, n_4, n_5, n_4, n_2, n_6, n_{out}, n_{Final})$