

## 6. Übungsblatt zu Software Qualität

Michel Meyer, Manuel Schwarz

30. November 2012

### Aufgabe 6.1

(a)

|                  |                    |    |   |
|------------------|--------------------|----|---|
|                  |                    | 08 | public static double pow(double x, int n) { |
|                  |                    | 09 |   |
|                  |                    | 10 | double res = 1.0;                           |
|                  |                    | 11 | int i;                                      |
|                  | 1.<br>(08,13,16)   | 12 |   |
|                  |                    | 13 | if (n < 0) {                                |
|                  | 2.<br>(08,15,19)   | 14 | i = -n;                                     |
|                  |                    | 15 | } else {                                    |
|                  |                    | 16 | i = n;                                      |
|                  |                    | 17 | }   |
|                  | 3.<br>(16,19,24)   | 18 |   |
|                  |                    | 19 | while (i > 0) {                             |
|                  | 6.<br>(19,21,19)   | 20 | res *= x;                                   |
|                  |                    | 21 | i--;  |
|                  |                    | 22 | }   |
|                  |                    | 23 |   |
|                  |                    | 24 | if (n < 0) {                                |
|                  |                    | 25 | res = 1 / res;                              |
|                  |                    | 26 | }   |
|                  | 8.<br>(24,28,Exit) | 27 |   |
|                  |                    | 28 | return res;                                 |
|                  | 9. (28,28,Exit)    | 29 | }   |
| 4.<br>(16,21,19) |                    |    |   |

## Aufgabe 6.2

### (a) Kontrollflussgraph

1 · Pow.java · 2012-11-29 23:52 · Manuel Schwarz

```
/**
 * Stellt eine Methode zur Berechnung der n-ten Potenz zur Verfügung.
 *
 * @author Wolfgang Runte
 * @version 30.11.2010
 */
public class Pow {

    /**
     * Berechnung der n-ten Potenz.
     *
     * @param x Basis
     * @param n Exponent
     * @return n-te Potenz zur Basis x
     */
    public static double pow(double x, int n) { } n5

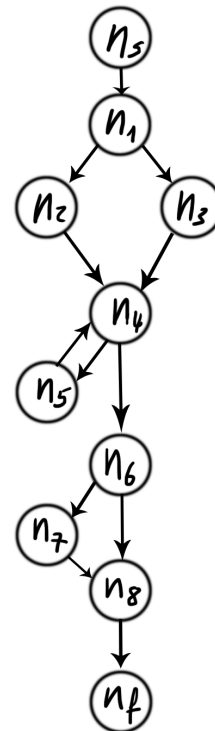
        double res = 1.0; } n1
        int i; } n2

        if (n < 0) { } n3
            i = -n; } n4
        } else { } n5
            i = n; } n6
        } } n7

        while (i > 0) { } n8
            res *= x; } n9
            i--; } n10
        } } n11

        if (n < 0) { } n12
            res = 1 / res; } n13
        } } n14

        return res; } n15
    }
}
```



### (b) zyklomatische Komplexität

Die allgemeine Formel für die zyklomatische Komplexität eines Graphen  $G$  lautet

$$Z(G) = e - n + 2 \quad (1)$$

Dabei beschreibt  $e$  die Anzahl der Kanten und  $n$  die Anzahl der Knoten von  $G$ . In unserem Fall gilt folglich:

$$Z(G) = 12 - 10 + 2$$

$$Z(G) = 4$$

### (c)