

Übung: Software-Qualität

Sommersemester 2016

swq@se.uni-hannover.de

McCabe – Berechnen über Code

```
public static int sampleMethod(int val_1, int val_2, int val_3){
    if(val_1 == val_2){
        for(int i = 0; i<10;i++){
            System.out.println("Value: "+i);
        }
        return 1;
    }
    else if(val_1 == val_3 || val_2 == val_3){
        return 2;
    } else if((val_2 == val_1) && (val_3 == val_1)){
        return 3;
    } else {
        return 4;
    }
}
```

+ 1

5 if + 1 for + 1 = McCabe 7

McCabe – Berechnen über Code

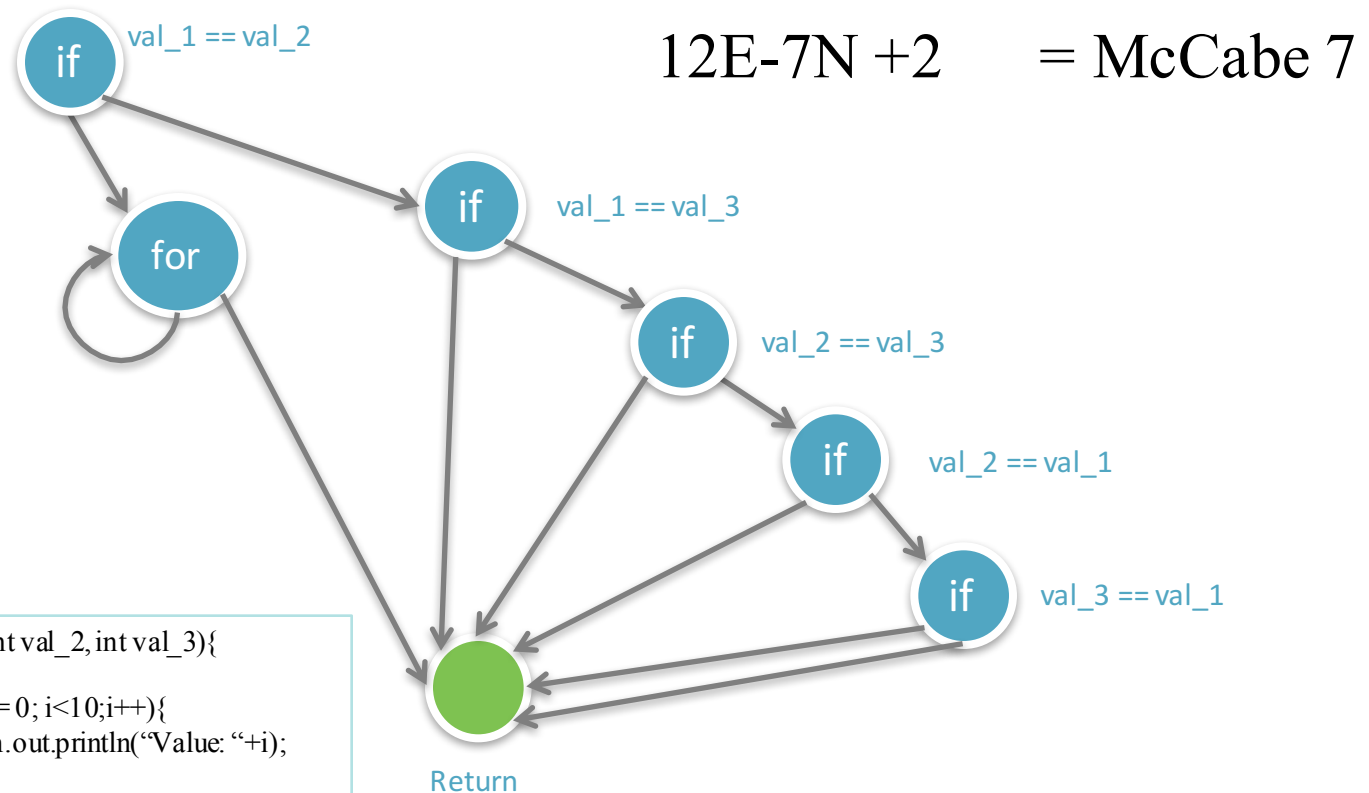
```
public static int sampleMethod(int val_1, int val_2, int val_3){  
    if(val_1 == val_2){  
        for(int i = 0; i<10;i++){  
            System.out.println("Value: "+i);  
        }  
    }  
    else if(val_1 == val_3 || val_2 == val_3){  
        return 2;  
    } else if((val_2 == val_1) && (val_3 == val_1)){  
        return 3;  
    } else {  
        return 4;  
    }  
}
```

Für jeden Äquivalenzvergleich, Bedingungsabfrage, egal ob durch && oder || verbunden wird JEWEILS 1 dazu addiert da als einzelnes IF zu betrachten!

+ 1

5 if + 1 for + 1 = McCabe 7

McCabe berechnen über Ablaufgraph



```
public static int sampleMethod(int val_1, int val_2, int val_3){  
    if(val_1 == val_2){  
        for(int i = 0; i < 10; i++){  
            System.out.println("Value: "+i);  
        }  
        return 1;  
    }  
    else if(val_1 == val_3 || val_2 == val_3){  
        return 2;  
    } else if((val_2 == val_1) && (val_3 == val_1)){  
        return 3;  
    } else {  
        return 4;  
    }  
}
```

Äquivalenzklassenmethode

- Annahme:
 - Ein Programm reagiert bei der Verarbeitung eines Wertes *aus einem bestimmten Bereich* genau so wie bei der Verarbeitung jedes anderen Wertes aus diesem Bereich.
- Ansatz:
 - Zerlege Wertebereich der Eingabeparameter oder Definitionsbereiche der Ausgabeparameter in Äquivalenzklassen
- Vorgehen:
 1. Äquivalenzklassen pro Eingabeparameter oder pro Ausgabeaspekt
 2. Davon nur je ein Repräsentant nötig
 3. Repräsentanten aller Parameter geschickt kombinieren: Testdaten

Äquivalenzklassenmethode

Die persönliche Einkommensteuer von Steuerpflichtigen soll ermittelt werden. Dazu diene die folgende Methode:

```
public Euro steuernZuZahlen(Euro einkommen, int kinderzahl) {...}
```

An die Methode werden die Anforderungen gestellt:

- R01: Unter 20.000 Euro sind keine Steuern zu bezahlen.
- R02: Zwischen 20.000 und 50.000 Euro fallen 20% an. Das ist der *Sockelbetrag*.
- R03: Für alle Einkommen über 50.000 Euro sind pauschal 40% zu zahlen.
- R04: Je Kind werden 5 Prozentpunkte vom *Sockelbetrag* abgezogen (nicht von den 40%).
Beispiel: Bei zwei Kindern ist der Sockelbetrag noch 10% bei einem Einkommen zwischen 20.000 und 50.000 Euro.
- R05: Es darf kein negativer Steuersatz entstehen.

Stellen Sie Äquivalenzklassen für die Eingabeparameter auf! Dabei brauchen Sie nur gültige Werte zu berücksichtigen, also nicht-negative Zahlen vom richtigen Typ.

Äquivalenzklassenmethode

- Eingabe-Äquivalenzklassen (nur gültige Werte):

Eingabe	Äquivalenzklasse		Anforderung
Einkommen	Ä1	Einkommen < 20.000 EUR	R01
	Ä2	20.000 EUR <= Einkommen <= 50.000 EUR	R02
	Ä3	Einkommen > 50.000 EUR	R03
Kinderzahl	Ä4	0 <= Kinderzahl < 5	R04
	Ä5	Kinderzahl >= 5	R04

Äquivalenzklassenmethode

- Kombination der Eingabe-Äquivalenzklassen:

	Ä1	Ä2	Ä3
Ä4	0 EUR Steuern zahlen	20% abzgl. 5% pro Kind vom Einkommen als Steuern zahlen	40% vom Einkommen als Steuern zahlen
Ä5			

Äquivalenzklassenmethode

	Ä1	Ä2	Ä3
Ä4	0 EUR Steuern zahlen	20% abzgl. 5% pro Kind vom Einkommen als Steuern zahlen	40% vom Einkommen als Steuern zahlen
Ä5			

Geben Sie drei vollständige Testfälle an, die zusammen möglichst viele Ihrer Äquivalenzklassen abdecken. Nennen Sie zu jedem Testfall die davon abgedeckten Äquivalenzklassen.

ID	Einkommen	Kinderzahl	Sollwert	Geprüfte ÄK
1	15.000 EUR	0	0 EUR	Ä1, Ä4
2	30.000 EUR	2	3.000 EUR	Ä2, Ä4
3	60.000 EUR	5	24.000 EUR	Ä3, Ä5