Programmieren 1 – Übung #10

Arne Schmidt

Heute

C0-, C1-Tests
mit JUnit

Live
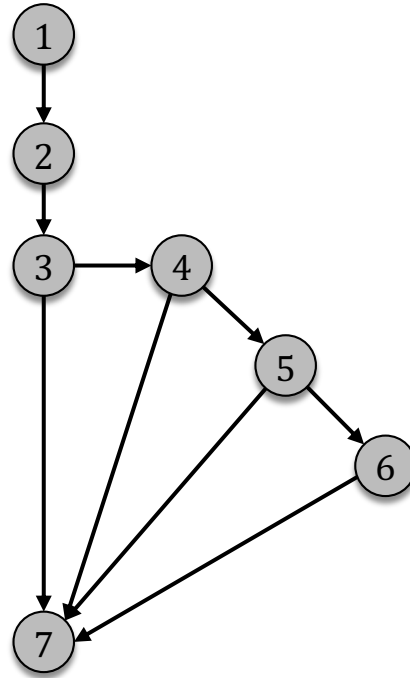Coding

Zu testender Code

```
1. public static int monus(int a, int b) {
2.     int res = 0;
3.     if (a > b
4.            && a >= 0
5.            && b >= 0) {
6.         res = a – b;
       }

7.     return res;
```

Bei mehreren Bedingungen in einem if, können diese als einzelne Anweisungen für den C0-Test betrachtet werden.

Zu testender Code

```
1. public static int monus(int a, int b) {
2.     int res = 0;
3.     if (a > b
4.              && a >= 0
5.              && b >= 0) {
6.         res = a – b;
       }

7.     return res;
```
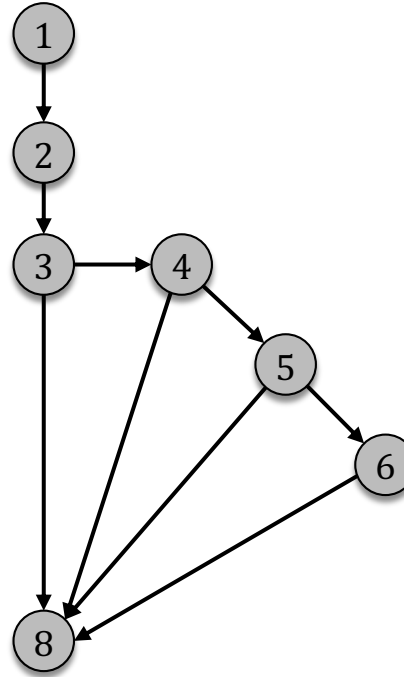

Testfälle für C0-Test?

Testfälle für C1-Test?

Zu testender Code

```
1. public static int monus(int a, int b) {
2.     int res = 0;
3.     if (a > b
4.             && a >= 0
5.             && b >= 0) {
6.         res = a – b;
       }

7.     return res;
```


Testfälle für C0-Test?

a	b	res
10	5	5

Testfälle für C1-Test?

```
1. public static int monus(int a, int b) {
2.    int res = 0;
3.    if (a > b
4.            && a >= 0
5.            && b >= 0) {
6.        res = a – b;
    }

7.    return res;
```
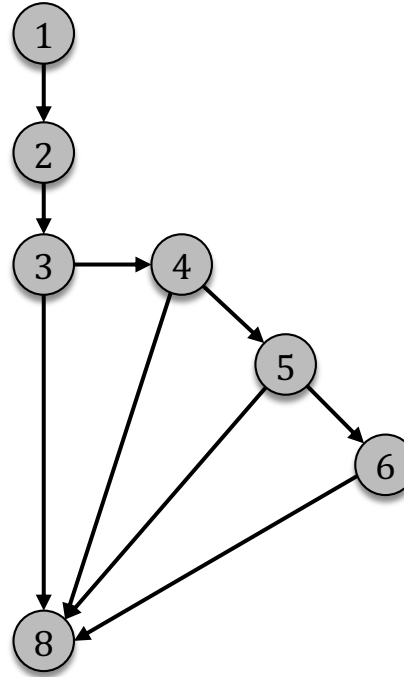

Testfälle für C0-Test?

a	b	res
10	5	5

Testfälle für C1-Test?

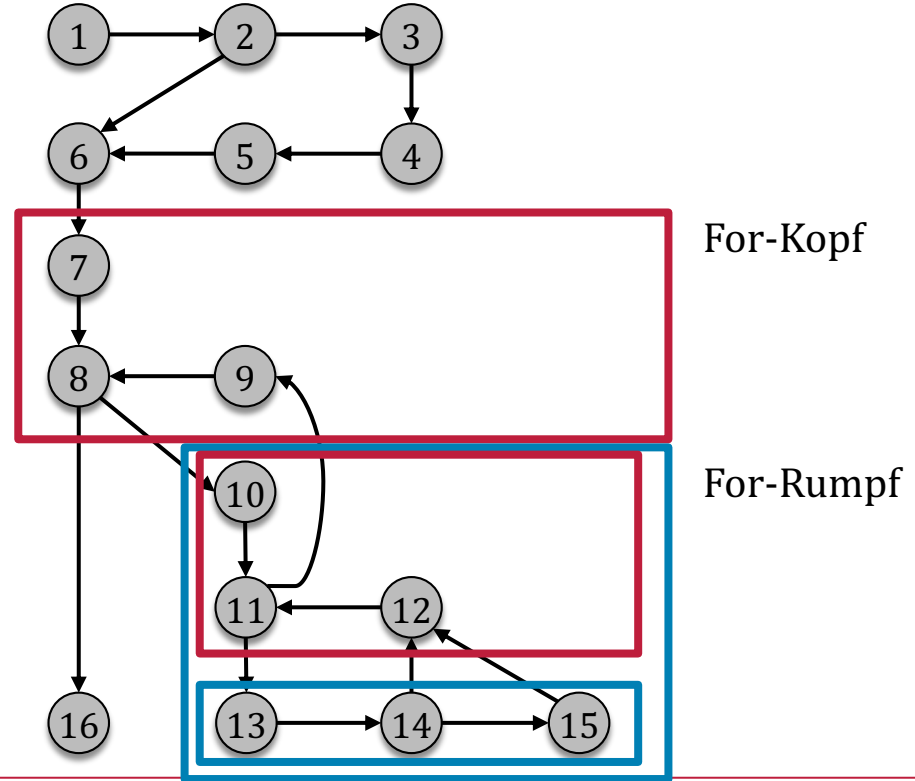| a	b	res
10	5	5
8	-1	0
-2	-3	0
3	4	0

Technische
Universität
Braunschweig

Zu testender Code

```java
1. public static int[][] buildEvenArray(int a, int b) {
2.    if (a < b) {
3.        int tmp = a;
4.        a = b;
5.        b = tmp;
    }

6.    int[][] arr = new int[a][b];

7.    for (int i = 0;
8.            i < a;
9.            ++i) {
10.       for (int j = 0;
11.               j < b;
12.               ++j) {
13.          arr[i][j] = i * j;
14.          if (i * j % 2 != 0) {
15.             arr[i][j] -= 1;
            }
        }
    }
16.  return arr;
}
```

Zu testender Code

```
1. public static int[][] buildEvenArray(int a, int b) {
2.    if (a < b) {
3.         int tmp = a;
4.         a = b;
5.         b = tmp;
      }

6.    int[][] arr = new int[a][b];

7.    for (int i = 0;
8.             i < a;
9.             ++i) {
10.       for (int j = 0;
11.              j < b;
12.              ++j) {
13.         arr[i][j] = i * j;
14.         if (i * j % 2 != 0) {
15.            arr[i][j] -= 1;
            }
         }
      }
16. return arr;
}
```
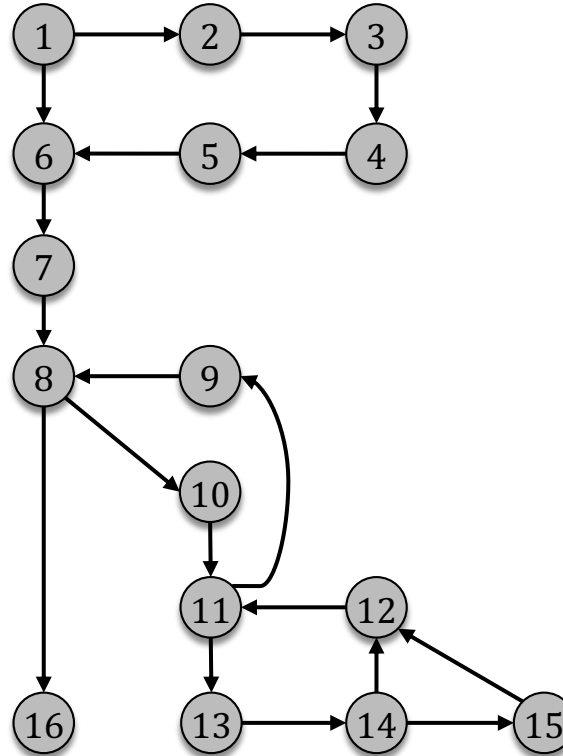

For-Kopf

For-Rumpf

Zu testender Code

```
1. public static int[][] buildEvenArray(int a, int b) {
2.    if (a < b) {
3.        int tmp = a;
4.        a = b;
5.        b = tmp;
      }

6.    int[][] arr = new int[a][b];

7.    for (int i = 0;
8.            i < a;
9.            ++i) {
10.        for (int j = 0;
11.                j < b;
12.                ++j) {
13.            arr[i][j] = i * j;
14.            if (i * j % 2 != 0) {
15.                arr[i][j] -= 1;
                }
            }
        }
16.   return arr;
}
```
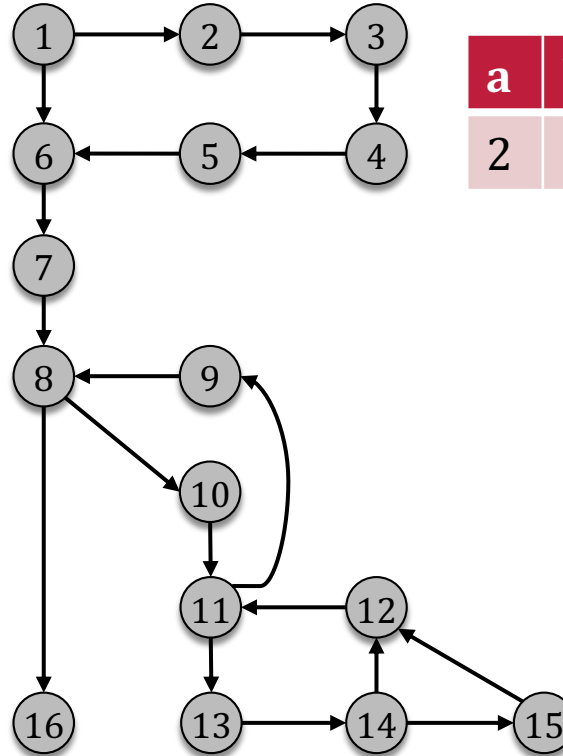

Testfälle für C0-Test?

Testfälle für C1-Test?

Zu testender Code

```
1. public static int[][] buildEvenArray(int a, int b) {
2.    if (a < b) {
3.         int tmp = a;
4.         a = b;
5.         b = tmp;
      }

6.    int[][] arr = new int[a][b];

7.    for (int i = 0;
8.            i < a;
9.            ++i) {
10.       for (int j = 0;
11.               j < b;
12.               ++j) {
13.          arr[i][j] = i * j;
14.          if (i * j % 2 != 0) {
15.               arr[i][j] -= 1;
              }
          }
      }
16.   return arr;
}
```


Testfälle für C0-Test?

a	b	res
2	3	{{0,0},{0,0},{0,2}}

Testfälle für C1-Test?

Zu testender Code

```
1. public static int[][] buildEvenArray(int a, int b) {
2.    if (a < b) {
3.        int tmp = a;
4.        a = b;
5.        b = tmp;
      }

6.    int[][] arr = new int[a][b];

7.    for (int i = 0;
8.            i < a;
9.            ++i) {
10.       for (int j = 0;
11.               j < b;
12.               ++j) {
13.           arr[i][j] = i * j;
14.           if (i * j % 2 != 0) {
15.               arr[i][j] -= 1;
              }
          }
      }
16.   return arr;
}
```
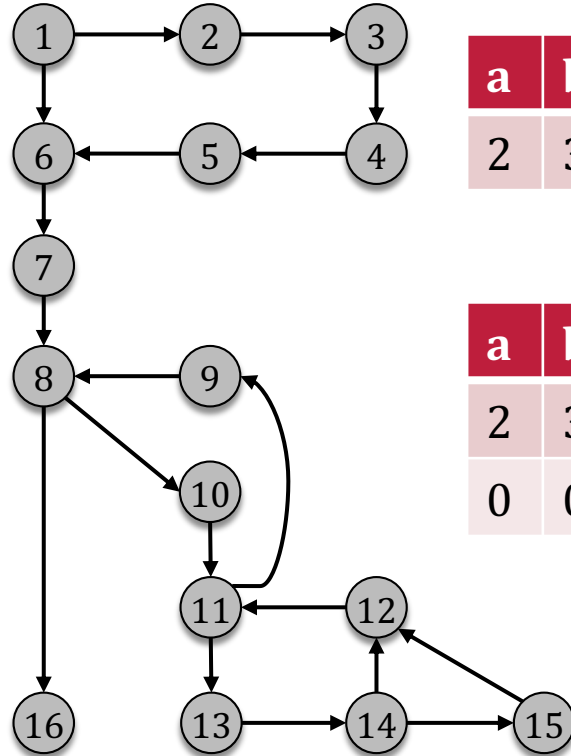
Testfälle für C0-Test?

a	b	res
2	3	{{0,0},{0,0},{0,2}}

Testfälle für C1-Test?

a	b	res
2	3	{{0,0},{0,0},{0,2}}
0	0	{ }