Aufgabe 4: Qualitätssicherung

Ein gängiger Ansatz zur Messung der Qualität von Software ist das automatisierte Testen von Programmen. Im Folgenden werden praktische Testmethoden anhand des nachstehend angegebenen Sortieralgorithmus diskutiert.
Algorithmus 1 Bubble Sort:

```
1  package org.bschlangaul.aufgaben.sosy.examen_46116_2017_09;
2
3  public class BubbleSort {
4    void bubblesort(int[] array, int len) {
5      for (int i = 0; i < len - 1; i++) {   // 1
6        for (int j = 0; j < len - 1; j++) { // 2
7          if (array[j] > array[j + 1]) {    // 3
8            int temp = array[j];            // 4
9            array[j] = array[j + 1];        // 5
10           array[j + 1] = temp;            // 6
11         }
12       }
13     }
14   }
15 }
```

(a) Nennen Sie eine Art des Black-Box-Testens und beschreiben Sie deren Durchführung anhand des vorgegebenen Algorithmus.

Beim Black-Box-Testen sind die Testfälle von Daten getrieben (Data-Driven) und beziehen sich auf die Anforderungen und das spezifizierte Verhalten.)

$\Rightarrow$ Aufruf der Methoden mit verschiedenen Eingangsparametern und Vergleich der erhaltenen Ergebnisse mit den erwarteten Ergebnissen.

Das Ziel ist dabei eine möglichst hohe Anforderungsüberdeckung, wobei man eine minimale Anzahl von Testfällen durch Äquivalenzklassenzerlegung (1) und Grenzwertanalyse (2) erhält.

zu (1): Man identifiziert Bereiche von Eingabewerten, die jeweils dieselben Ergebnisse liefern. Dies sind die sog. Äquivalenzklassen. Aus diesen wählt man nun je einen Repräsentanten und nutzen diesen für den Testfall.

zu (2): Bei der Grenzwertanalyse identifiziert man die Grenzbereiche der Eingabedaten und wählt Daten aus dem nahen Umfeld dieser für seine Testfälle.

Angewendet auf den gegebenen Bubblesort-Algorithmus würde die Grenzwertanalyse bedeuten, dass man ein bereits aufsteigend sortiertes Array und ein absteigend sortiertes Array übergibt.

(b) Zeichnen Sie ein mit Zeilennummem beschriftetes Kontrollflussdiagramm für den oben angegebenen Sortieralgorithmus.

Zur Erinnerung: Eine im Code enthaltene Wiederholung mit for muss wie folgt im Kontrollflussgraphen „zerlegt“ werden:

(c) Erklären Sie, ob eine vollständige Pfadüberdeckung für die gegebene Funktion möglich und sinnvoll ist.

Eine vollständige Pfadüberdeckung (C_1-Test) kann nicht erreicht werden, da die Bedingung der inneren Wiederholung immer wahr ist, wenn die Bedingung der äußeren Wiederholung wahr ist. D. h., der Pfad S-1-1-2-2-1“ kann nie gegangen werden. Dies wäre aber auch nicht sinnvoll, weil jeder Eintrag mit jedem anderen verglichen werden soll und im Fall true $\rightarrow$ false ein Durchgang ausgelassen.