

Übungsblatt 2

Ausgabe: 15.11.2021

Besprechung: 22.-25.11.2021 und 29.11.-02.12.

11 Punkte

Bitte laden Sie die Übungsbeispiele rechtzeitig am Sonntag 28.11.2021 bis spätestens 23:55 als Zip Datei (`<Matrikelnummer>_<Name>_SE1-ÜB<Nummer>.zip`) in Moodle hoch und tragen Sie Ihre Kreuze in ZEUS ein. Da es in ZEUS keine halben Punkte gibt, wurde die Anzahl der Punkte für ZEUS verdoppelt → laut ZEUS können Sie für diese Übung 22 Punkte bekommen. Für die Übung selbst werden aber die am Übungsblatt angegebenen Punkte gezählt → es können maximal 11 Punkte erreicht werden (= Anzahl der Punkte in ZEUS / 2). Sollten Sie Fragen zur Übung haben oder Unklarheiten auftreten, nutzen Sie bitte das Moodle-Forum.

Abgabedateien

Die Zip Datei `<Matrikelnummer>_<Name>_SE1-ÜB<Nummer>` sollte folgende Dateien beinhalten:

- 2.1_weiser.pdf
- 2.2_sdg.pdf
- 2.3_sdg_slice.pdf
- 2.4_grenzwertanalyse.pdf
- 2.5_testcases.pdf
- 2.6_testcases.pdf
- 2.7_testcases.pdf

Slicing

Slicing nach Weiser

- 2.1** Bestimmen Sie den Slice nach Weiser für das gegebene Listing 1. as Slicing Kriterium für dieses Beispiel lautet $S(14, \{\text{sqrt}\})$. **(2 Punkte)**

SDG

Gegeben ist der Code in Listing 2.

- 2.2** Bestimmen Sie den System Dependency Graph (SDG) für diesen Code. **(2 Punkte)**
- 2.3** Bestimmen mit Hilfe des System Dependency Graph (SDG) einen interprozeduralen Slice für das Kriterium $S(13, \{\text{result}\})$. **(1 Punkt)**

Testen

2.4 Beispiel Grenzwertanalyse **(2 Punkte)**

Sie werden beauftragt die Verwaltung für Dauerkunden eines Kinos mit Hilfe der Grenzwertanalyse zu testen. Von jedem Kunden werden die Kundennummer, das Alter in Jahren, sowie

die Dauer des Abos gespeichert. Die Kundennummer ist eine 4-stellige Zahl zwischen 3000 und 8000 und Kunden müssen mindestens 12 Jahre alt sein. Die Abodauer kann von 6 Monaten bis hin zu 3 Jahren variieren.

- a) Geben Sie die gültigen und ungültigen Eingabeklassen für jede Variable an.
- b) Wieviele Testfälle sind bei einer Single Fault Assumption zu berücksichtigen?
- c) Geben Sie konkrete Testfälle für diese Beispiele bei Single Fault Assumption an.

2.5 Testfälle berechnen und erstellen (2 Punkte)

Der angegebene Methodenkopf zeigt eine Schnittstelle für eine Methode, die entscheidet, ob Studierende zur Diplomprüfung antreten darf. Studierende dürfen zur Diplomprüfung antreten, wenn sie im Masterstudium inskribiert sind, mindestens 300 ECTS erfolgreich absolviert haben und eine positive Note (1-4) auf die Diplomarbeit bekommen haben.

Methodenkopf

```
1 public boolean isAdmissibleForDiplomaExamination (boolean
    isMasterStudent, int ects, int gradeThesis) throws
    IllegalArgumentException {...}
```

Berechnen Sie die Anzahl der Testfälle jeweils für

- (a) Grenzwertanalyse
- (b) Robustness Testing
- (c) Worst Case Testing
- (d) Paranoid Testing

Geben Sie die konkreten Testfälle für (a) und (c) inklusive dem erwarteten Ergebnis an.

2.6 Äquivalenzklassenzerlegung (2 Punkte)

Folgendes Beispiel beschreibt eine Methode zur Berechnung eines Aufschlagfaktors für Flugtickets. Die Berechnung erfolgt auf Basis der gebuchten Klasse, des Freigepäcks und der Sitzreihe.

Klasse	Economy	Classic	Business	Freigepäck	0	1	2
Faktor	0,65	1,0	4,9	Faktor	0,95	1,35	1,75

Sitzreihe (von - bis)	1-12	14-23	24-31	32-40
Faktor	1,6	1,3	0,95	1,5

Ein Beispiel: Für ein Flugticket Classic mit 2 Freigepäckstücken für die 3. Sitzreihe muss, ausgehend von einem Basiswert für das Ticket von 75 €, eine Gesamtpreis von $75 * 1,0 * 1,75 * 1,6 = 210,00 \text{ €}$ bezahlt werden.

1. Welche Äquivalenzklassen für gültige und ungültige Eingaben können Sie im obigen Beispiel identifizieren?
2. Wieviele Testfälle müssen Sie ausführen, um einen
 - (a) Weak ECT
 - (b) Strong ECT
 - (c) Weak Robust ECT
 - (d) Strong Robust ECTzu erhalten?
3. Erstellen Sie konkrete Testfälle mit den erwarteten Ergebnissen für einen Weak ECT.
4. Ergänzen Sie nun die Testfälle aus 3.) damit alle Testfälle für ein Weak Robust ECT vorhanden sind.

Codebeispiele

Listing 1: Codebeispiel Weiser

```
1 int simpleFloorSqrt(int x){
2     int sqrt=-1;
3     int i = 1, result = 1;
4     if (x == 0 || x == 1) {
5         sqrt=x;
6     } else {
7         while (result <= x) {
8             int help=i+1;
9             i=help;
10            result = i*i;
11        }
12        sqrt = i-1;
13    }
14    return sqrt;
15 }
```

Listing 2: Codebeispiel

```
1 public void multiplyByAdding(int a, int b) {
2     int result=0;
3     if(a>b) {
4         int help=a;
5         a=b;
6         b=help;
7     }
8     int i=0;
9     while(i<a) {
10        result=addIntegers(result ,b);
11        i++;
12    }
13    System.out.println("Result: "+result);
14 }
15 public int addIntegers(int a, int b) {
16     if(a>b) {
17         int help=a;
18         a=b;
19         b=help;
20     }
21     int i=0;
22     while(i<b) {
23         a++;
24         i++;
25     }
26     return a;
27 }
```