**Übungsblatt 3**

Lion Wagner, 3231355, st148345@stud.uni-stuttgart.de  
  
Jannis Westermann, 3288735, st153444@stud.uni-stuttgart.de  
  
Lukas Pietzsch, 3227178, st151191@stud.uni-stuttgart.de

1.)







|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Leer | Ressourcen prüfend | KapselVorhanden &  WasserVorhanden | Kaffee kochen |
| Timer auslösen | Ressourcen prüfend | Ressourcen  Prüfend | Ressourcen  Prüfend | Kaffee kochen |
| KaffeekapselFehlt || WasserFehlt | Leer | Leer | Leer | Leer |
| KaffeekapselVorhanden& WasserVorhanden | Leer | KapselVorhanden &  WasserVorhanden | KapselVorhanden &  WasserVorhanden | Kaffee kochen |
| KnopfGedrückt | Leer | Ressourcen prüfend | Kaffee kochen | Kaffee kochen |
| KaffeeFertig | Leer | Leer | Leer | Leer |



|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Beschreibung | Eingabe | Ablauf |
| 1 | Normaldurchlauf mit Wasser und Kapsel vorhanden. | 1. Timer löst aus 2. KaffeekapselVorhanden & WasserVorhanden 3. KnopfGedrückt 4. KaffeeFertig | Leer 🡪 Ressourcen prüfend 🡪 KapselVorhanden &  WasserVorhanden 🡪 Kaffe kochen 🡪 Leer |



|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Beschreibung | Eingabe | Ablauf |
| 2 | Normaldurchlauf mit Wasser und Kapsel vorhanden. | 1. Timer löst aus 2. KaffeekapselVorhanden & WasserVorhanden 3. KnopfGedrückt 4. KaffeeFertig | Leer 🡪 Ressourcen prüfend 🡪 KapselVorhanden &  WasserVorhanden 🡪 Kaffe kochen 🡪 Leer |
| 3 | Ressourcen vorhanden,  aber Kapsel wird wieder entnommen. | 1. Timer löst aus 2. KaffeekapselVorhanden & WasserVorhanden 3. Timer löst aus 4. KaffeekapselFehlt | Leer 🡪 Ressourcen prüfend 🡪 KapselVorhanden &  WasserVorhanden 🡪 Ressourcen prüfend 🡪 Leer |

1. Man müsste noch einen zusätzlichen 4. Test wie Test 3 machen bei dem das 4. Event der Eingabe „WasserFehlt“ lautet.

2.)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | B&&C | B&&D | A || (B && C) || (B && D) |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 7 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 8 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 11 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 12 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 14 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 15 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



Mögliche Testpaare die Bedingung erfüllen:

A: 1-9, 2-10, 3-11, 4-12, 5-13

B: 2-6, 3-7, 4-8

C: 5-7

D: 5-6

* 2,5,6,7,10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | B&&C | B&&D | A || (B && C) || (B && D) |
| 1 | 0 | 0 | - | - | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | - | 1 | - | 1 |
| 5 | 1 | - | - | - | - | - | 1 |



A: 1-5

B: 1-3,1-4

C: 2-4

D: 2-3

* 1,2,3,4,5

3.)



|  |  |  |  |
| --- | --- | --- | --- |
|  | Kein Schleifendurchlauf | Ein  Schleifendurchlauf | Mid. Zwei schleifendurchläufe |
| Input | {} | {1} | {-2,0} |
| Ablauf | n\_start, n\_1, n\_2, n\_final | n\_start, n\_1, n\_2, n\_3, n\_5, n\_6, n\_2, n\_final | n\_start, n\_1, n\_2, n\_3, n\_4, n\_2, n\_3, n\_5, n\_final |
| Output | 0 | 1 | -1 |



|  |  |
| --- | --- |
|  | (ürsprünglich) Kein Schleifendurchlauf |
| Input | {} |
| Ablauf | n\_start, n\_1,n\_1.5 , n\_2, n\_3, n\_5,n\_final |
| Output | 0 |

1. Zwischen Zeile 2 und 3 „list.Add(0);“. Dies sei Anweisung n\_1.5. Nun ergibt die Eingabe einer leeren Liste. Bei allen andern listen wird sowieso die Schleife mid. einmal durchlaufen.

Problem: Do-While Schleifen werden immer mindestens einmal ausgeführt.

Lsg. 1: Abändern des Kriteriums: einmaliges ausführen für Do-While Schleifen ist äquivalent zum nicht ausführen eines Schleifendurchlaufes.

Lsg. 2: Schleife umwandeln in While-do oder For-do.