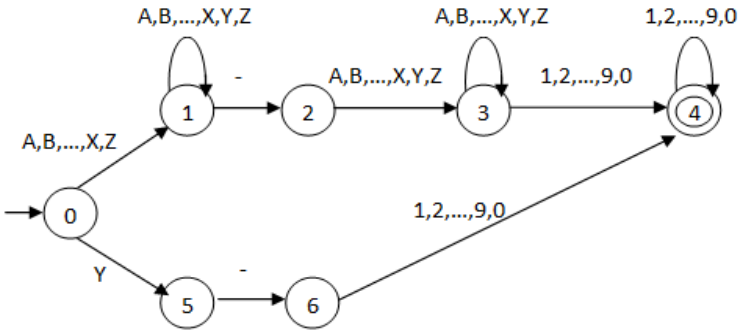
Informatik Abitur Bayern 2012 / III - Beispiellösung

Autor:
Angerer

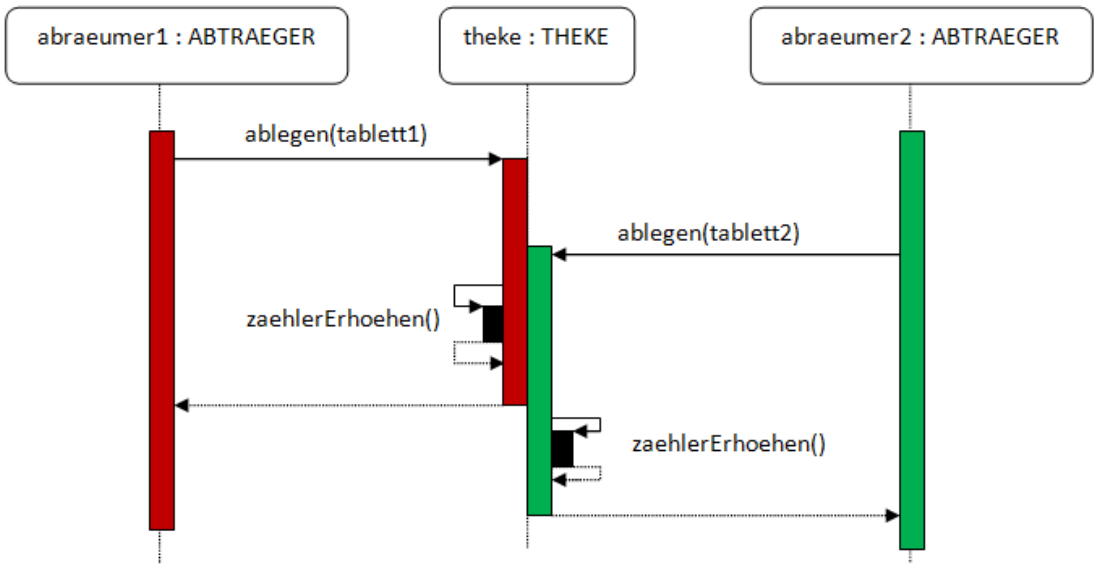
1a Kennzeichen = Verwaltungsbezirk „-“ Kennbuchstabe Kennziffer | „Y“ „-“ Kennziffer. 4
Verwaltungsbezirk = BuchstabeOhneY {Buchstabe}.
Kennbuchstabe = Buchstabe {Buchstabe}.
Kennziffer = Ziffer {Ziffer}.
BuchstabeOhneY = „A“ | … | „X“ | „Z“.
Buchstabe = BuchstabeOhneY | „Y“.
Ziffer = „1“ | „2“ | … | „9“ | „0“.

1b 5

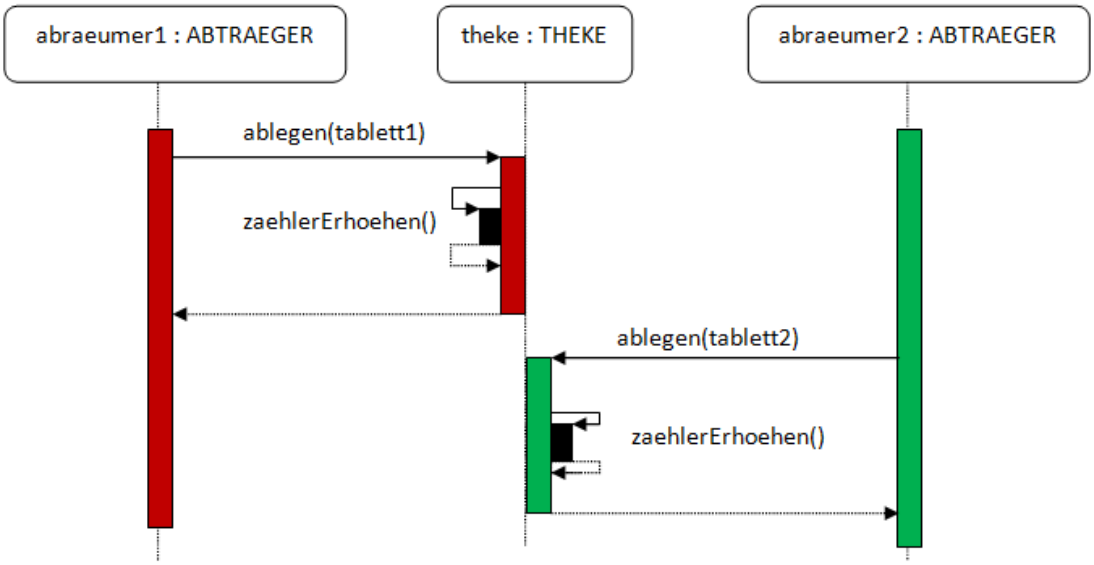


2a **Sondersituation 1:** Einer der vier Mitarbeiter, die abräumen, möchte ein Tablett ablegen, aber es ist kein freier Platz auf der Theke. Er muss also warten und es später nochmals versuchen. 10
Sondersituation 2: Der Mitarbeiter, der einräumt, möchte ein Tablett von der Theke entfernen, aber es befindet sich kein Tablett dort. Er muss ebenfalls warten.

2b Legt der erste Abräumer ein Tablett auf die Theke, könnte anschließend kein freier Platz mehr sein. Legt aber in der Zeit, in der das Tablett eingetragen und der Zähler erhöht wird, ein zweiter Abräumer ein weiteres Tablett auf die Theke, führt dies zu einem Fehler. 4



2c Das Monitorkonzept stellt sicher, dass eine Auswahl von Methoden zu einem Zeitpunkt nur von einem Prozess genutzt wird. Wird die Methode ablegen(tablett) mithilfe des Monitorkonzepts implementiert, kann der zweite Abräumer erst dann ein Tablett ablegen, wenn der erste Abräumer sein Tablett erfolgreich abgelegt hat und der Zähler erhöht wurde. 3



2d 2

```
public class THEKE
{
    private int anzahl;
    private TABLETT[] tablett;

    THEKE()
    {
        anzahl = 0;
        tablett = new TABLETT[10];
    }

    synchronized boolean ablegen(TABLETT t)
    {
        if(anzahl > 10)
        {
            return false;
        }
        else
        {
            tablett[anzahl] = t;
            anzahl = anzahl + 1;
            return true;
        }
    }

    synchronized TABLETT wegnehmen()
    {
        if(anzahl > 0)
        {
            anzahl = anzahl - 1;
            return tablett[anzahl];
        }
        else
        {
            return null;
        }
    }
}
```

3a

load 100	Akkumulator: 13	
div 101	Akkumulator: 2	*(13:5 = 2)*
mul 101	Akkumulator: 10	*(2 * 5 = 10)*
store 102	Akkumulator: 10	
load 100	Akkumulator: 13	
sub 102	Akkumulator: 3	
store 102	Akkumulator: 3	
hold		

2

Speicherzelle 100: 13
Speicherzelle 101: 5
Speicherzelle 102: 3

3b Es wird der Rest der ganzzahligen Division der Zahl in Speicherzelle 100 und der Zahl in Speicherzelle 101 berechnet. 2

3c *(Zahl / Divisor) * Divisor ist gleich Zahl* überprüft, ob Zahl / Divisor ganzzahlig ohne Rest teilbar ist. 1

3d 8

```
wiederhole :
    load    Divisor
    cmp     Zahl
    jmpge endewiederhole
    load    Zahl
    div     Divisor
    mult    Divisor
    cmp     Zahl
    jmpne wenn2
    loadi   1
    store   HatTeiler

wenn2 :
    load    Divisor
    add     Erhoehung
    store   Divisor
    loadi    2
    cmp     Erhoehung
    jmpne sonst2
    loadi    4
    store   Erhoehung
    jmp      wiederhole


sonst2 :
    loadi     2
    store     Erhoehung

endewiederhole :
    hold

//Vorbelegungen

Zahl :
    word 13   //siehe Teilaufgabe a
Divisor :
    word 5
HatTeiler :
    word 0
Erhoehung :
    word 2
```

20