

```

public class Uebung01 {
    public static void main (String[] args) {
        // Deklaration der Variablen
        int a = 7;
        int b = 6;

        // Startwert für Summe der Addition festlegen
        int start = a;

        // Startwert für Anzahl der Addition festlegen
        int anzahl = 0;
        while (start <= 30) {
            start = start + b;
            anzahl = anzahl + 1;
        }
        System.out.println("Die Anzahl der Addition ist: " + anzahl);
    }
}

```

Sehr gut. Ihr braucht aber
 nicht unbedingt `int start`
 verwenden. Aber trotzdem
 vollkommen richtig

5 P.

Aufgabe 1: Deterministische Endliche Automaten

Zustandsmenge: {aus,ein,gem_kalt,ungem_heiss,gem_heiss,kaffee} ✓

Eingabealphabet: {E,M,W,K} ✓

Startzustand: aus ✓

Endzustand: {kaffee} ✓

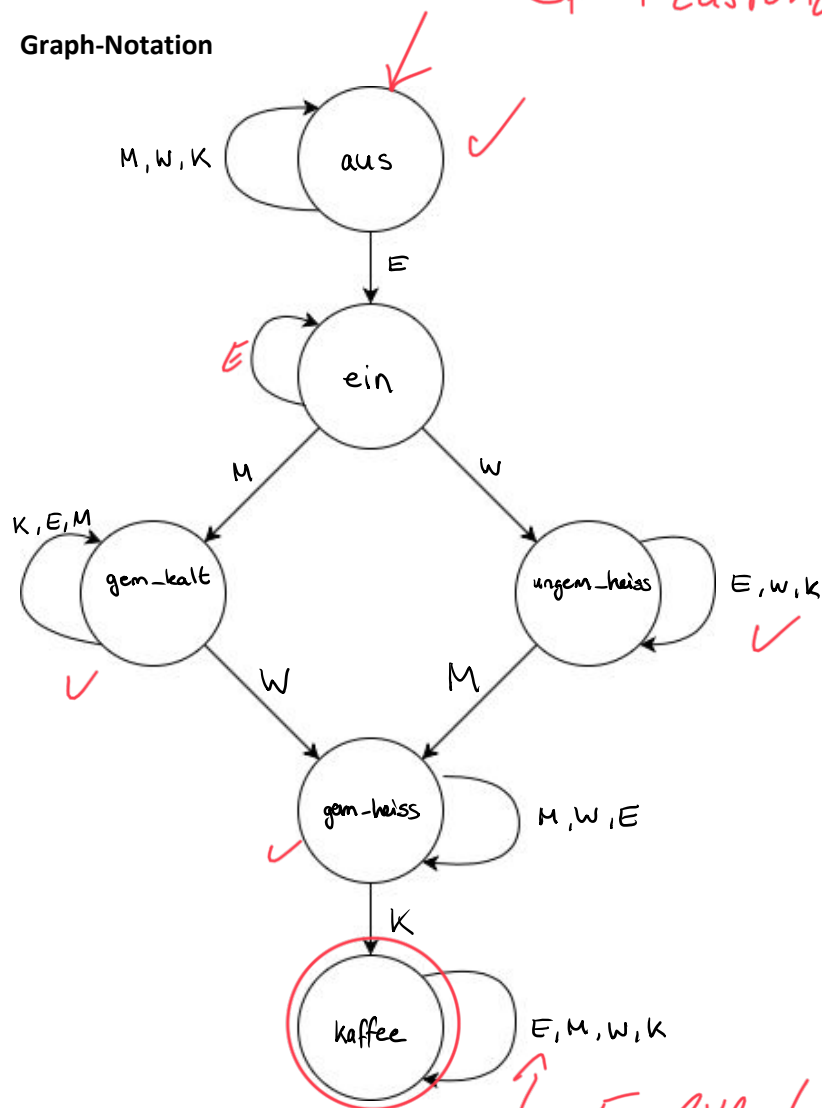
Zustands-Übergangs-Tabelle

Zustand	Gelesene Zeichen	Nächster Zustand
aus	E	ein
aus	M	aus
aus	W	aus
aus	K	aus
ein	E	ein
ein	M	gem_kalt
ein	W	ungem_kalt
ein	K	ein
gem_kalt	E	gem_kalt
gem_kalt	M	gem_kalt
gem_kalt	W	gem_heiss
gem_kalt	K	gem_kalt
ungem_heiss	E	ungem_heiss
ungem_heiss	M	gem_heiss
ungem_heiss	W	ungem_heiss
ungem_heiss	K	ungem_heiss
gem_heiss	E	gem_heiss
gem_heiss	M	gem_heiss
gem_heiss	W	gem_heiss
gem_heiss	K	kaffee
kaffee	E	kaffee
kaffee	M	kaffee
kaffee	W	kaffee
kaffee	K	kaffee

→ heiss meint
ihr bestimmt.



Graph-Notation



- Startzustand mit Eingangs-
Pfeil.

↑ E eventuell zum
ausschalten verwenden.

Endzustand immer
doppelt umkreisen.

4P.

Graph sieht sehr gut aus.

Aufgabe 2: Grammatiken in EBNF-Notation

Program	= Anweisung, {Anweisung}; ✓
Anweisung	= (VariablenDeklaration, [Wertzuweisung] Wertzuweisung); ✓
VariablenDeklaration	= "int ", Identifier; ✓
Wertzuweisung	= Identifier, "=", Number Identifier ArithmetischerAusdruck; ✓
Identifier	= Letter, LettersOrDigits; ✓
Number	= ["-", Digit, {Digit}; ✓
ArithmetischerAusdruck	= Addition Subtraktion; ✓
Addition	= Number "+" Number Number "+" Identifier Identifier "+" Number Identifier "+" Identifier; ✓
Subtraktion	= Number "-" Number Number "-" Identifier Identifier "-" Number Identifier "-" Identifier; ✓
LettersOrDigits	= {Letter Digit}; ✓
Letter	= SmallLetter CapitalLetter; ✓
SmallLetter	= "a" "b" "c" ... "z"; ✓
CapitalLetter	= "A" "B" "C" ... "Z"; ✓
Digit	= "0" "1" "2" ... "9"; ✓

5,5 P.

Sehr gut
Denkt an
das Semikolon
hinter jeder Anweisung.

Aufgabe 3: Fehler in Java

```

1 public class Power {
2
3     public static void main(String[] args) {
4         integer a = 2;
5         integer b = 4;
6         // calculate a to the power of b (a^b)
7         integer result = 1;
8         while (integer i < b) { (b != 0) {
9             result * a; result = result * a;
10            b--;
11        } // output result of 2^4
12        println(result); System.out.println(result);
13    }
14
15 }
```

→ Auch mit Zähl-
variable i möglich.
Dann müsst ihr
b=4 nicht verändern.

Korrigierte Version:

```

1 package pda.uebungszettel01;
2
3 public class Power {
4
5     public static void main (String[] args) {
6         int a = 2;
7         int b = 4;
8         // calculate to the power of b (a^b)
9         int result = 1;
10        while (b != 0) {
11            result = result * a;
12            b--;
13        }
14        //output result of 2^4
15        System.out.println(result);
16    }
17 }
```

3 P.