

## Kontrollstrukturen in Maschinensprache

- if(Bedingung){ Code }

➔ Wenn Bedingung falsch, überspringe den Code

```
if(x > 5)                //x ist in 100 gespeichert
{
    x = 3;              LOAD 100
                        SUBI 5
                        JMPNP Ende //Jump falls nicht positiv
}
                        LOADI 3
                        STORE 100
                        Ende: HOLD
```

- if(Bedingung){ Code 1 } else { Code 2 }

➔ Wenn Bedingung falsch, springe zu Code 2. Am Ende von Code 1, überspringe Code 2

```
if(x > 5)                LOAD 100
{
    x = 3;              SUBI 5
                        JMPNP Else //Jump falls nicht positiv
}
                        LOADI 3
else                    STORE 100
{
    x = 7;              JMP Ende //Jump immer
                        Else: LOADI 7
}
                        STORE 100
                        Ende: HOLD
```

- while (Bedingung) {Code}
- ➔ Sobald Bedingung falsch, springe zum Ende. Am Ende der Wiederholung, springe zum Start.

```

While(x > 5) Start: LOAD 100
{
    SUBI 5
    x--;
    JMPNP Ende //Jump falls nicht positiv
}
    LOAD 100
    SUBI 1
    STORE 100
    JMP Start //Jump immer
Ende: HOLD

```

- for (Bedingung) {Code}
- ➔ Analog zu While, zusätzlich updaten der Zählvariable

```

for(int i = 0; i < 3; i++)
{
    x--;
}

//i wird in 101 gespeichert
LOADI 0
STORE 101
Start: LOAD 101
    SUBI 3
    JMPNN Ende //Jump falls nicht negativ
    LOAD 100
    SUBI 1
    STORE 100
    LOAD 101
    ADDI 1
    STORE 101
    JMP Start //Jump immer
Ende: HOLD

```

### Beispielprogramm:

#Programm für Addition von 4 Zahlen

LOAD 100

ADD 101

ADD 102

ADD 103

STORE 105

HOLD

(Die Summanden stehen in den Zellen 100 bis 103, das Ergebnis soll in Zelle 105 gespeichert werden.)

### Beispielprogramm2:

#Programm zum positiv machen

LOAD 100

JMPNN Nichtnegativ1

MULI -1

STORE 100

Nichtnegativ1: LOAD 101

JMPNN Nichtnegativ2

MULI -1

STORE 101

Nichtnegativ2: LOAD 102

JMPNN Nichtnegativ3

MULI -1

STORE 102

Nichtnegativ3: LOAD 103

JMPNN Nichtnegativ4

MULI -1

STORE 103

Nichtnegativ4: HOLD

JAVA:

```
private int[] zahlen = {-75,-15,84,-775}
```

```
for(int i = 0; i < 4; i++)
```

```
{
```

```
    if(zahlen[i] < 0)
```

```
    {
```

```
        zahlen[i] = zahlen[i]*(-1);
```

```
    }
```

```
}
```

### Beispielprogramm3:

#Programm zur Potenzberechnung  $a^b$

```
Start:      LOAD 102
            JMPNP Ende
            LOAD 105
            MUL 101
            STORE 105
            LOAD 102
            SUBI 1
            STORE 102
            JMP Start
Ende:      HOLD
```

JAVA:

```
int a;
int b;
int ergebnis = 1;
for(int i = 0; i < b;i++)
{
    ergebnis = ergebnis * a;
}
```

## **Der Befehlszyklus einer Registermaschine**

Eine Registermaschine durchläuft solange folgende Schritte, bis der hold Befehl erreicht wurde.

- Befehl aus dem Speicher holen
- Befehl decodieren und ggf. Operandenadressen errechnen
- Operanden holen
- Befehl ausführen und ggf. Ergebnis speichern
- Befehlszähler erhöhen