

Programmierung(1)

Agenda

- FOR-Schleifen
 - **Motivation** und Beispiel
 - (Post)-**Inkrement**, -**Dekrement**
 - Darstellung im **PAP**
 - Darstellung im **Struktogramm**
 - Darstellung im **Pseudocode**
 - Syntax in **ANSI C**
- Ausführliches Training + Ergebnisbesprechung
- Fachpraktische Anwendungen

FOR-Schleifen – Motivation

Ausgangslage:

- Eine große Anzahl von Schleifen arbeiten mit einem **Schleifenzähler**, also einer Variablen die pro Durchlauf **weitergezählt** (erhöht oder verringert) wird.
- Nicht selten wird auch über diesen **Zähler gesteuert** (siehe die jeweils verwendete Bedingung), ob die Schleife weiterläuft oder beendet wird.
- Ferner muss im Falle der Verwendung eines Zählers stets dafür gesorgt werden, dass dieser mit einem (bzgl. der Aufgabenstellung geeigneten) **Startwert** initialisiert wird.

Konsequenz:

- Auf Grund der Häufigkeit dieser Ausgangslage, lohnt sich die Einführung einer eigenständigen For-Schleifen-**Syntax**, die solche Zähler-Schleifen kompakt und gut lesbar darstellt.
- Zu diesem Zweck werden die wesentlichen Elemente (Setzen des Startwertes, Überprüfung der Durchlauf-Bedingung und Hochzähl-Modus) in einer **einzigsten Zeile** notiert (sind nun also nicht mehr auf verschiedene Stellen des Codes verteilt).

(Post)-Inkrement/Dekrement – Motivation und Definition

Ausgangslage:

- Der **Modus**, mit dem bei einer Zähler-Schleife **weitergezählt** wird, kann je nach Aufgabenstellung sehr unterschiedlich aussehen (und unterliegt jedenfalls keiner technischen Beschränkung).
- Tatsächlich gilt aber wohl für die allermeisten je geschriebenen Zähler-Schleifen, dass deren Schleifenzähler pro Durchlauf **um genau 1** erhöht, oder (immer noch oft, wenn auch seltener) pro Durchlauf um genau 1 verringert wird.
- Bei der diesbezüglich zur Zeit einzigen uns bekannten Schreibweise müssten wir den Zähler also entweder in der Form **zaehler=zaehler+1** hochzählen (oder entsprechend mit **zaehler=zaehler-1** herunterzählen)

Konsequenz:

- Auch hier lohnt sich die Einführung einer Kurzschreibweise, die uns das doppelte Notieren des selben Variablenamens erspart:
 - **zaehler++** ist Kurzschreibweise für **zaehler=zaehler+1** (und wird (Post)-**Inkrement** genannt)
 - **zaehler--** ist Kurzschreibweise für **zaehler=zaehler-1** (und wird (Post)-**Dekrement** genannt)

(Post)-Inkrement/Dekrement – Motivation und Definition

Ausgangslage:

- Der **Modus**, mit dem bei einer Zähler-Schleife **weitergezählt** wird, kann je nach Aufgabenstellung sehr unterschiedlich aussehen (und unterliegt jedenfalls keiner technischen Beschränkung).
- Tatsächlich gilt aber wohl für die allermeisten je geschriebenen Zähler-Schleifen, dass deren Schleifenzähler pro Durchlauf **um genau 1** erhöht, oder (immer noch oft, wenn auch seltener) pro Durchlauf um genau 1 verringert wird.
- Bei der diesbezüglich zur Zeit einzigen uns bekannten Schreibweise müssten wir den Zähler also entweder in der Form **zaehler=zaehler+1** hochzählen (oder entsprechend mit **zaehler=zaehler-1** herunterzählen)

Konsequenz:

- Auch hier lohnt sich die Einführung einer Kurzschreibweise, die uns das doppelte Notieren des selben Variablenamens erspart:
 - **zaehler++** ist Kurzschreibweise für **zaehler=zaehler+1** (und wird (Post)-**Inkrement** genannt)
 - **zaehler--** ist Kurzschreibweise für **zaehler=zaehler-1** (und wird (Post)-**Dekrement** genannt)

Bemerkung:

Die Unterscheidung zwischen **Post**- und **Prä**-Inkrement/ –Dekrement werden wir erst später problematisieren. Im Moment können wir schlicht von **Inkrement** oder **Dekrement** sprechen. Allerdings sollten wir beides ausschließlich innerhalb des Kopfes von FOR-Schleifen verwenden, oder alleinstehend in einer einzigen Zeile. Eine Verschachtelung (etwa innerhalb von printf) sollten wir unterlassen, da genau dort der Unterschied zwischen Post- und Prä- zum Tragen kommt.

FOR-Schleifen – **Beispielaufgabe**

Aufgabenstellung:

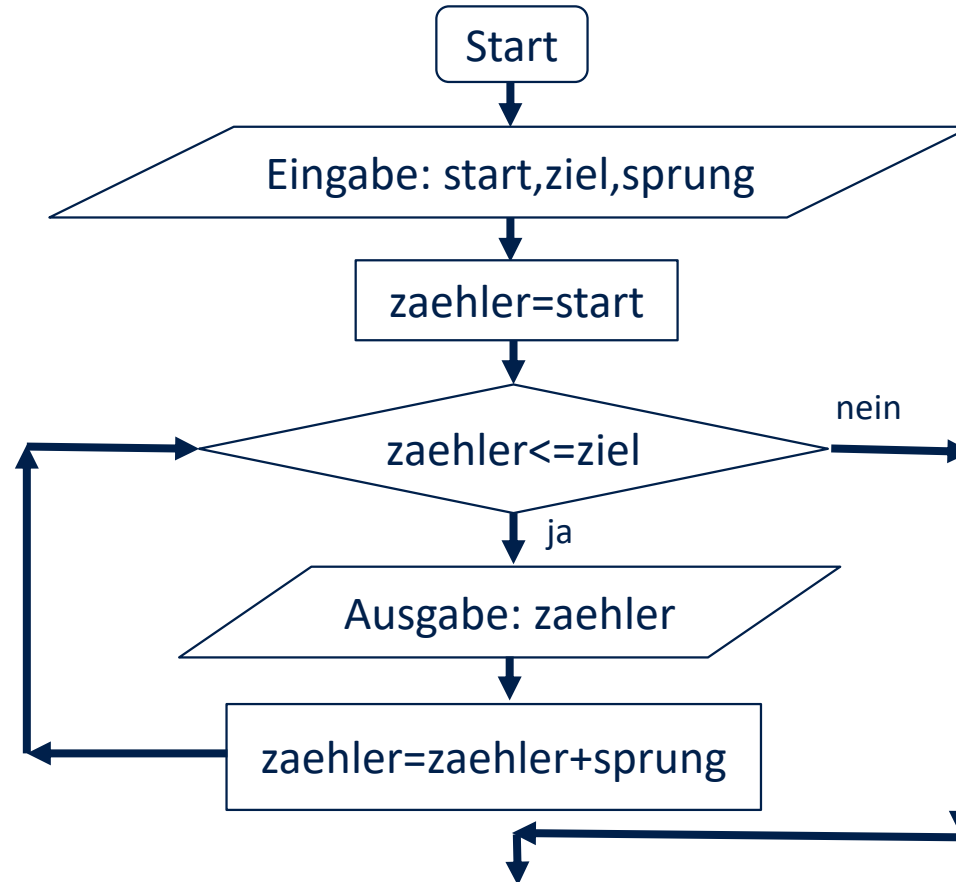
- **Vor der Schleife** werden vom User drei ganze Zahlen abgefragt:
 - Die erste Zahl soll in **start** abgespeichert werden. Sie bestimmt, von wo aus die anschließende Zähler-Schleife starten soll.
 - Die zweite Zahl soll größer als start sein, in **ziel** abgespeichert werden und bestimmen, bis zu welcher Zahl (höchstens) hochgezählt werden soll.
 - Die dritte Zahl soll in **sprung** abgespeichert werden. Sie bestimmt, um welchen Wert der Schleifenzähler pro Durchlauf hochgezählt werden soll.
- Daraufhin soll eine **Schleife starten**, die mit dem Zählerwert = **start** beginnt.
- **Pro Schleifendurchlauf** soll ...
 - (zuerst) der aktuelle Wert des Zählers ausgegeben werden
 - (anschließend) der Zähler um den Wert von **sprung** erhöht werden (zaehler=zaehler+sprung)
- Die **Schleife wird solange laufen**, solange der **Wert des Zählers kleiner (oder gleich) ziel** ist.
=> Die Schleife wird **abgebrochen**, falls **zaehler > ziel**
- **Nach der Schleife** soll die Ausgabe: „**Die Schleife endete, denn der Zähler hat den Wert:**“ **zaehler** erscheinen. Anschließend soll das Programm beendet werden.

Auch für diese Aufgabe wollen wir zunächst **PAP**, **Struktogramm** und **Pseudocode** erstellen, um erst daraufhin den entsprechenden **Quellcode** in ANSI C zu codieren.

Hinweis:

Aus didaktischen Gründen wurde diese Aufgabe möglichst allgemein gehalten.
Sie verzichtet daher zunächst auf eine Anwendung von **Inkrement** oder **Dekrement**.
Dies holen wir aber bei den folgenden Übungen nach ... Versprochen! ;-)

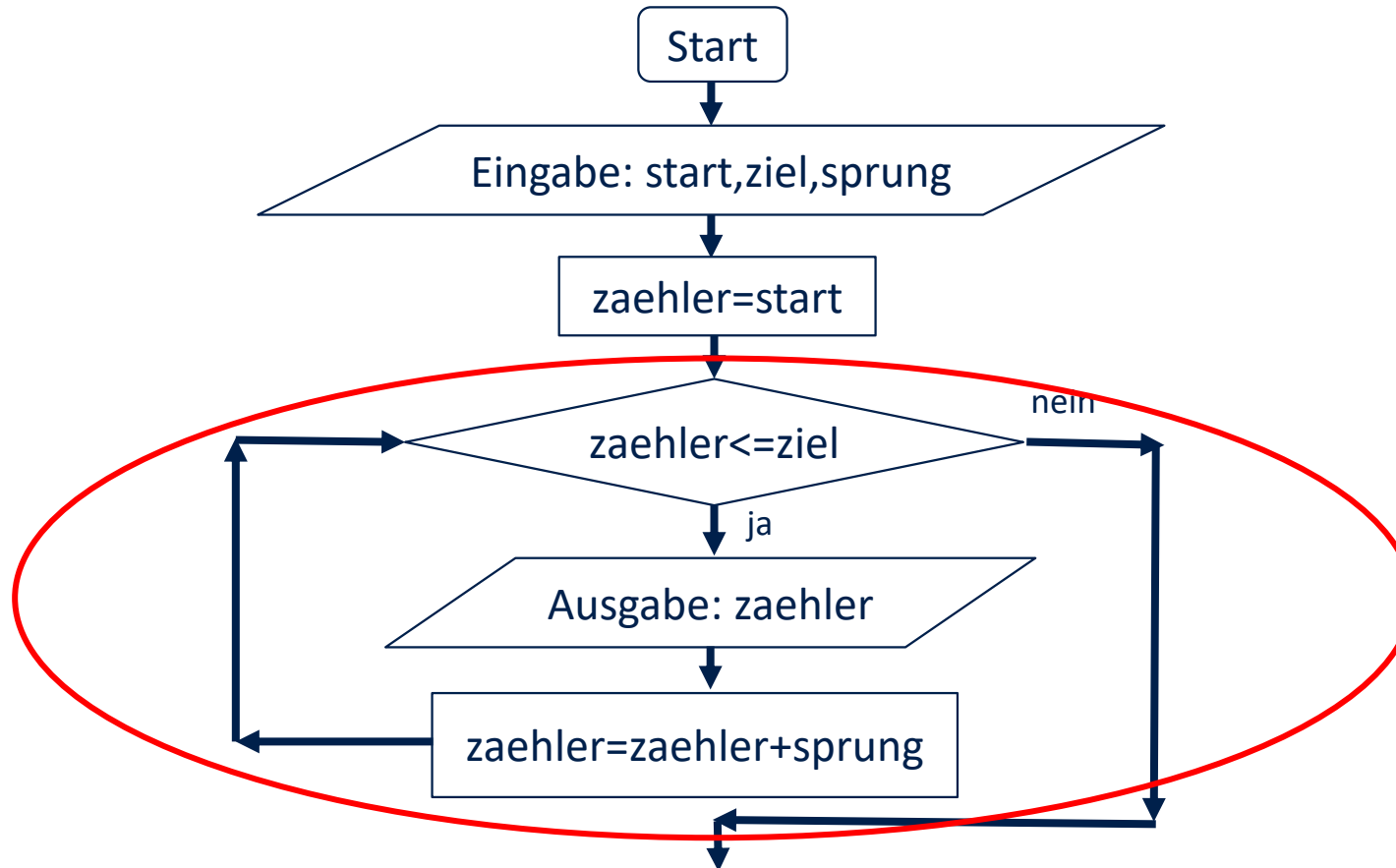
FOR-Schleifen – Beispielaufgabe – PAP



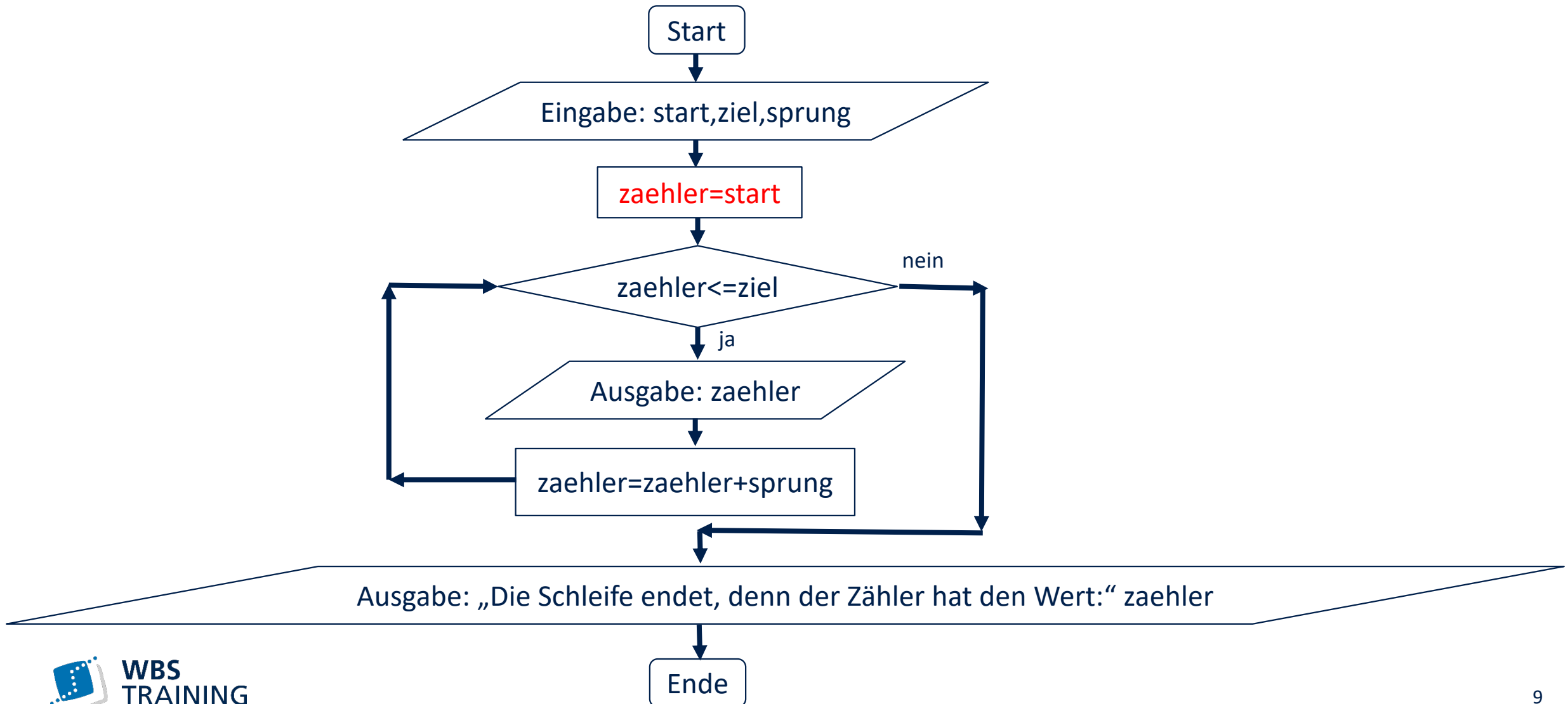
Hinweis:

PAP's besitzen kein eigenes Symbol für FOR-Schleifen. Daher ist an einer im PAP dargestellten Zähler-Schleife **nicht** ablesbar, ob diese als FOR- oder WHILE-Schleife gemeint ist.

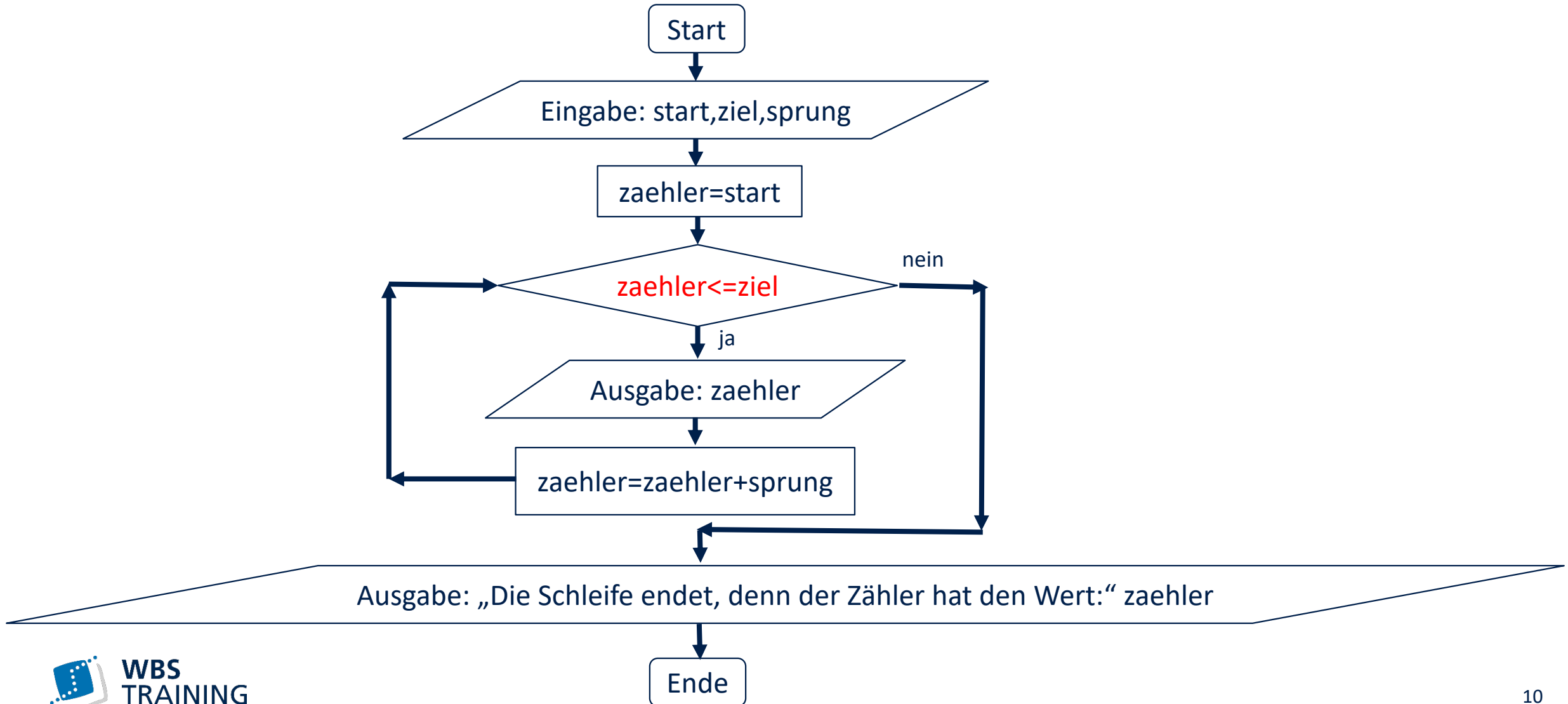
FOR-Schleifen – Beispielaufgabe – PAP – Schleife



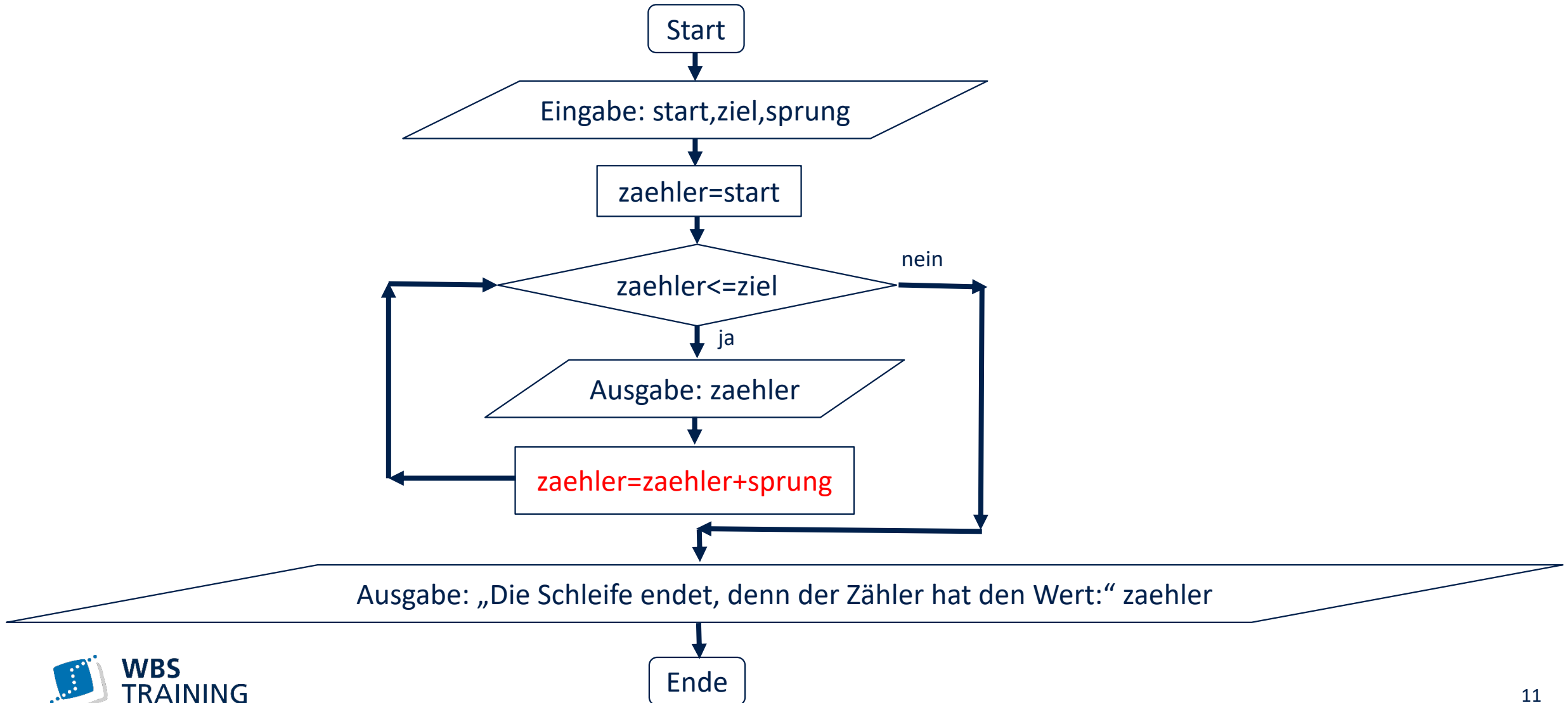
FOR-Schleifen – Beispielaufgabe – PAP – Initialisierung des Zählers



FOR-Schleifen – Beispielaufgabe – PAP – Zähler-gesteuert



FOR-Schleifen – Beispielaufgabe – PAP – Weiterzähl-Modus



FOR-Schleifen – Beispielaufgabe – Struktogramm

Eingabe: start,ziel,sprung

für(zaeher=start; zaeher<=ziel; zaeher=zaehler+sprung)

Ausgabe: zaeher

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zaeher

FOR-Schleifen – Beispielaufgabe – Struktogramm – Schleife

Eingabe: start,ziel,sprung

für(zähler=start; zähler<=ziel; zähler=zähler+sprung)

Ausgabe: zähler

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zähler

FOR-Schleifen – Beispielaufgabe – Struktogramm – Initialisierung des Startwertes

Eingabe: start,ziel,sprung

für(**zaehler=start**; zaehler<=ziel; zaehler=zaehler+sprung)

Ausgabe: zaehler

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zaehler

FOR-Schleifen – Beispielaufgabe – Struktogramm – Zähler-gesteuert

Eingabe: start,ziel,sprung

für(zähler=start; **zähler**≤**ziel**; zähler=zähler+sprung)

Ausgabe: zähler

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zähler

FOR-Schleifen – Beispielaufgabe – Struktogramm – Weiterzähl-Modus

Eingabe: start,ziel,sprung

für(zähler=start; zähler<=ziel; **zähler=zähler+sprung**)

Ausgabe: zähler

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zähler

FOR-Schleifen – Beispielaufgabe – Pseudocode

Programm „FOR-Schleifen-Beispiel“

```
{  
    Eingabe: start,ziel,sprung  
    für(zähler=start; zähler<=ziel; zähler=zähler+sprung)  
    {  
        Ausgabe: zähler  
    }  
    Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zähler  
}
```

FOR-Schleifen – Beispielaufgabe – Pseudocode – Schleife

Programm „FOR-Schleifen-Beispiel“

{

Eingabe: start,ziel,sprung

für(zaeher=start; zaeher<=ziel; zaeher=zaehler+sprung)

{

Ausgabe: zaeher

}

Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zaeher

}

FOR-Schleifen – Beispielaufgabe – Pseudocode – Initialisierung des Startwertes

Programm „FOR-Schleifen-Beispiel“

```
{  
    Eingabe: start,ziel,sprung  
    für(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)  
    {  
        Ausgabe: zaehler  
    }  
    Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zaehler  
}
```

FOR-Schleifen – Beispielaufgabe – Pseudocode – Zähler-gesteuert

Programm „FOR-Schleifen-Beispiel“

```
{  
    Eingabe: start,ziel,sprung  
    für(zähler=start; zähler<=ziel; zähler=zähler+sprung)  
    {  
        Ausgabe: zähler  
    }  
    Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zähler  
}
```

FOR-Schleifen – Beispielaufgabe – Pseudocode – Weiterzähl-Modus

Programm „FOR-Schleifen-Beispiel“

```
{  
    Eingabe: start,ziel,sprung  
    für(zaeher=start; zaeher<=ziel; zaehler=zaehler+sprung)  
    {  
        Ausgabe: zaeher  
    }  
    Ausgabe: „Die Schleife endet, denn der Zähler hat den Wert“ zaeher  
}
```

FOR-Schleifen – Beispielaufgabe – Quellcode

```
#include<stdio.h>

main()
{
    int start,ziel,sprung,zaehler;

    printf("Geben Sie bitte den Startwert ein: ");
    scanf("%d",&start);

    printf("Geben Sie bitte den Zielwert ein: ");
    fflush(stdin);
    scanf("%d",&ziel);

    printf("Geben Sie bitte den Sprungwert ein: ");
    fflush(stdin);
    scanf("%d",&sprung);

    for(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)
    {
        printf("Aktueller Zähler-Wert: %d\n",zaehler);
    }

    printf("Die Schleife endete, denn der Zähler hat den Wert %d",zaehler);
}
```

FOR-Schleifen – Beispielaufgabe – Quellcode – Schleife

```
#include<stdio.h>

main()
{
    int start,ziel,sprung,zaehler;

    printf("Geben Sie bitte den Startwert ein: ");
    scanf("%d",&start);

    printf("Geben Sie bitte den Zielwert ein: ");
    fflush(stdin);
    scanf("%d",&ziel);

    printf("Geben Sie bitte den Sprungwert ein: ");
    fflush(stdin);
    scanf("%d",&sprung);

    for(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)
    {
        printf("Aktueller Zähler-Wert: %d\n",zaehler);
    }

    printf("Die Schleife endete, denn der Zähler hat den Wert %d",zaehler);
}
```

FOR-Schleifen – Beispielaufgabe – Quellcode – Initialisierung des Startwertes

```
#include<stdio.h>

main()
{
    int start,ziel,sprung,zaehler;

    printf("Geben Sie bitte den Startwert ein: ");
    scanf("%d",&start);

    printf("Geben Sie bitte den Zielwert ein: ");
    fflush(stdin);
    scanf("%d",&ziel);

    printf("Geben Sie bitte den Sprungwert ein: ");
    fflush(stdin);
    scanf("%d",&sprung);

    for(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)
    {
        printf("Aktueller Zähler-Wert: %d\n",zaehler);
    }

    printf("Die Schleife endete, denn der Zähler hat den Wert %d",zaehler);
}
```


FOR-Schleifen – Beispielaufgabe – Quellcode – Zähler-gesteuert

```
#include<stdio.h>

main()
{
    int start,ziel,sprung,zaehler;

    printf("Geben Sie bitte den Startwert ein: ");
    scanf("%d",&start);

    printf("Geben Sie bitte den Zielwert ein: ");
    fflush(stdin);
    scanf("%d",&ziel);

    printf("Geben Sie bitte den Sprungwert ein: ");
    fflush(stdin);
    scanf("%d",&sprung);

    for(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)
    {
        printf("Aktueller Zähler-Wert: %d\n",zaehler);
    }

    printf("Die Schleife endete, denn der Zähler hat den Wert %d",zaehler);
}
```

FOR-Schleifen – Beispielaufgabe – Quellcode – Weiterzähl-Modus

```
#include<stdio.h>

main()
{
    int start,ziel,sprung,zaehler;

    printf("Geben Sie bitte den Startwert ein: ");
    scanf("%d",&start);

    printf("Geben Sie bitte den Zielwert ein: ");
    fflush(stdin);
    scanf("%d",&ziel);

    printf("Geben Sie bitte den Sprungwert ein: ");
    fflush(stdin);
    scanf("%d",&sprung);

    for(zaehler=start; zaehler<=ziel; zaehler=zaehler+sprung)
    {
        printf("Aktueller Zähler-Wert: %d\n",zaehler);
    }

    printf("Die Schleife endete, denn der Zähler hat den Wert %d",zaehler);
}
```

FOR-Schleifen – Gemeinsame Übung A_02_02_01



Aufgabe_02_02_01

Gegeben sei das folgende Struktogramm:

ML_02_02_01



Aufgabenstellung:

Bitte erstellen Sie dazu einen geeigneten **Quellcode** in ANSI C.

WBS TRAINING AG
Lorenzweg 5
D-12099 Berlin
Amtsgericht Berlin HRB 68531
Sitz der Gesellschaft: Berlin

Vorstand:
Heinrich Kronbichler,
Joachim Giese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IdNr.: DE 209 768 248

GLS Gemeinschaftsbank eG
IBAN: DE18 4306 0967 1146 1814 00
BIC: GENODEM33GLS



100% zertifiziert nach
DIN EN ISO 9001:2015 Reg. Nr. 0111046001
Zulassung nach DIN EN ISO 14001:2015 Reg. Nr. 0111046002

**VIELEN DANK
FÜR IHRE
AUFMERKSAMKEIT!**