



**WBS
TRAINING**

Programmierung(1)

Agenda

- Schleifen (Do-While)
 - **Motivation** und Beispiel
 - Darstellung im **PAP**
 - Darstellung im **Struktogramm**
 - Darstellung im **Pseudocode**
 - Syntax in **ANSI C**
 - **Bedingungen / Vergleichsoperatoren**
- Ausführliches Training + Ergebnisbesprechung
- Fachpraktische Anwendungen

Do-While – Motivation

- Wir hatten zuletzt mit den sogenannten „While-Schleifen“ bereits erste Schleifen kennengelernt. Der Schleifen-Typ, mit dem wir uns heute beschäftigen werden, wird in vielen Programmiersprachen als **Do-While-Schleife** bezeichnet.
- Für beide Schleifentypen gilt, dass diese einen zugewiesenen Abschnitt des Quellcodes solange wiederholen, solange eine gegebene Bedingung zutrifft.
- Der einzige Unterschied besteht in der **Position** der jeweiligen Bedingungsüberprüfung:
 - While-Schleifen überprüfen die Bedingung VOR jedem Durchlauf
 - **Do-While-Schleifen** überprüfen die Bedingung erst NACH einem Durchlauf
- Dieser Unterschied hat die folgende Konsequenz:
 - While-Schleifen haben unter Umständen keinen einzigen Durchlauf.
(sofern bereits die erste Überprüfung ergibt, dass die Bedingung nicht zutrifft)
 - **Do-While-Schleifen** haben **stets mindestens einen Durchlauf**.
(da die Bedingung immer erst nach diesem ersten Durchlauf überprüft wird)

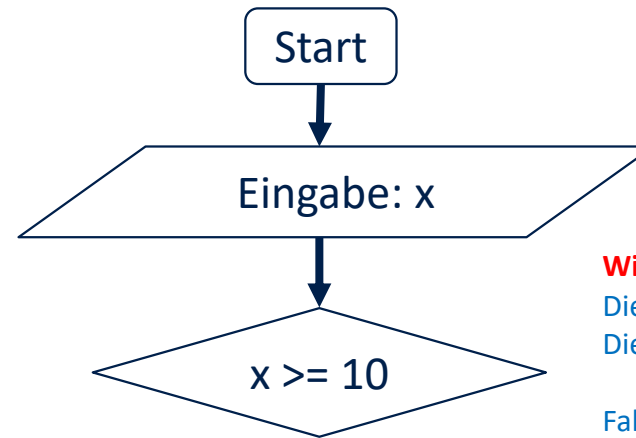
Do-While – Beispielaufgabe

Aufgabenstellung:

- Das Programm startet mit einer Schleife, in der pro Durchlauf vom User eine ganze Zahl **kleiner 10** abgefragt wird.
- **Erst nach dem ersten Durchlauf** soll (*bzw. kann!*) überprüft werden, ob die User-Eingabe korrekt war.
 - Falls der User keinen korrekten Wert eingab, so soll die Schleife den nächsten Durchlauf starten
 - Falls die Eingabe korrekt war, so soll die Schleife abgebrochen werden.
- Nach der Schleife soll auf der Konsole „*Glückwunsch!*“ erscheinen und das Programm enden.

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

Do-While – Beispielaufgabe – PAP



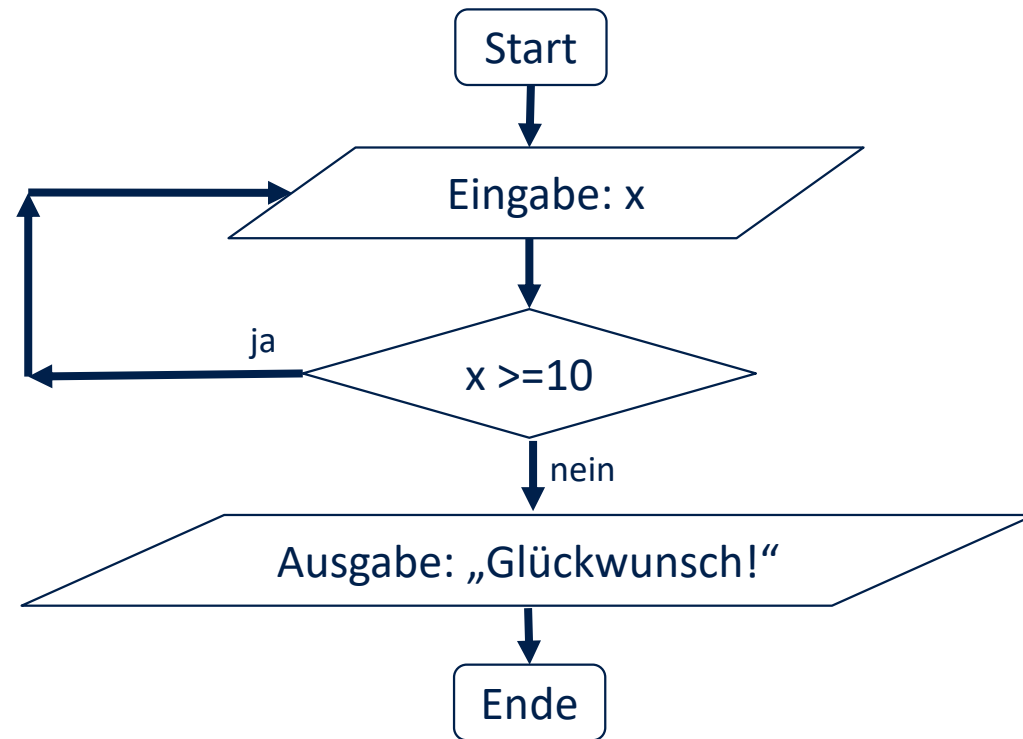
Wichtig:

Die Bedingung muss so formuliert werden, dass bei „ja“ bzw. „wahr“
Die Schleife weiterläuft. In diesem Fall also:

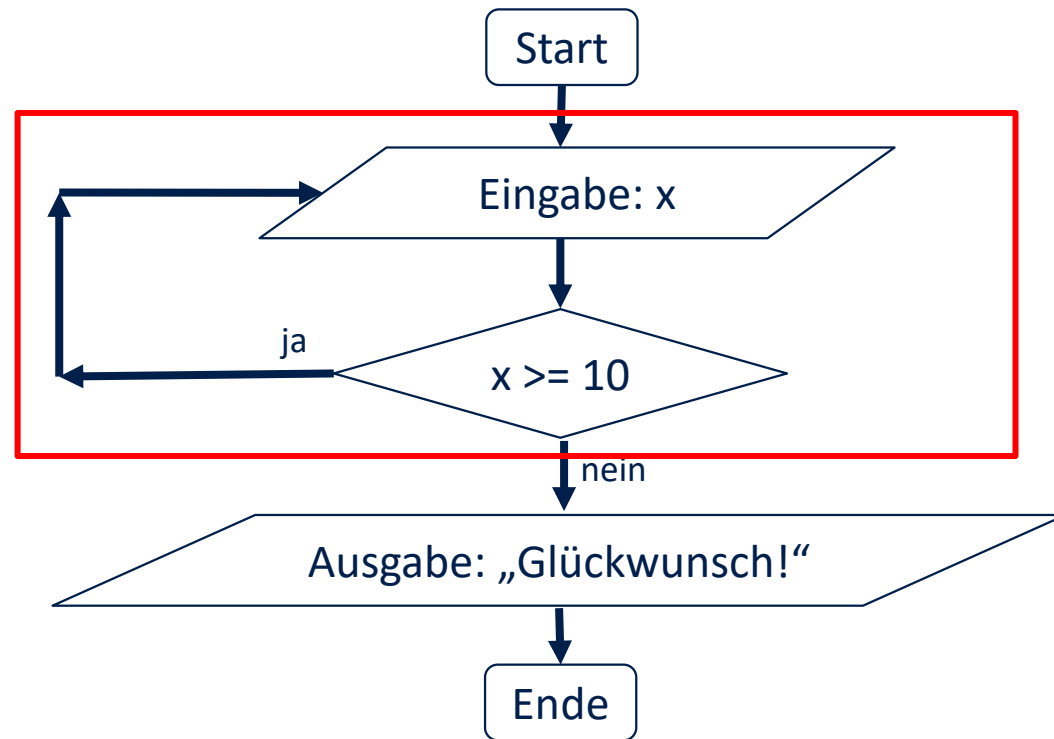
Falls zutrifft, dass $x \geq 10$, so war die Eingabe inkorrekt und die Schleife soll
ihren nächsten Durchlauf starten.

(Stichwort: „**Durchführungsbedingung**“, KEINE „Abbruchbedingung“)

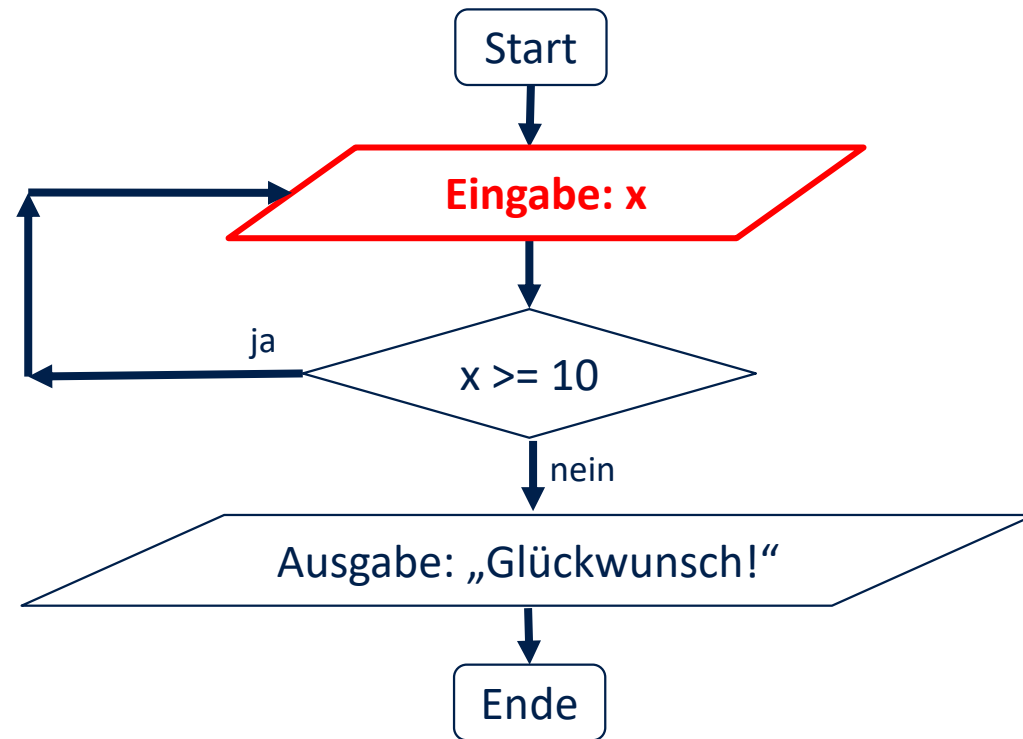
Do-While – Beispielaufgabe – PAP



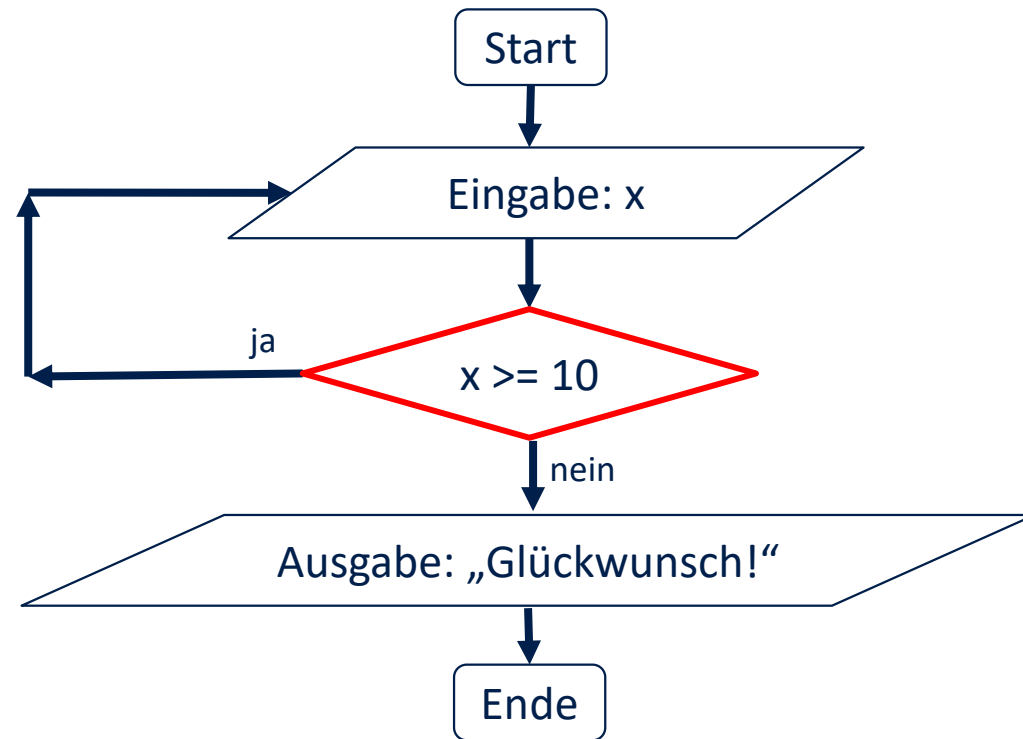
Do-While – Beispielaufgabe – PAP – Schleife



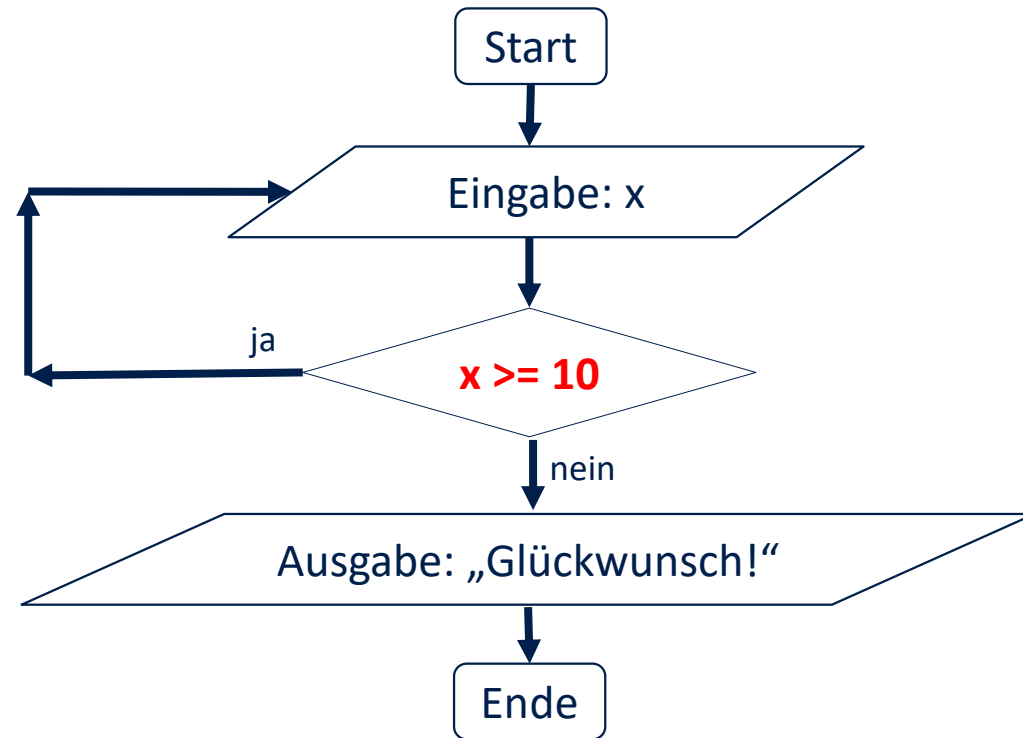
Do-While – Beispielaufgabe – PAP – Schleifenrumpf



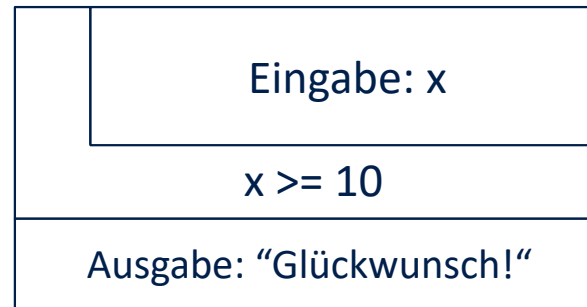
Do-While – Beispielaufgabe – PAP – Schleifenfuß



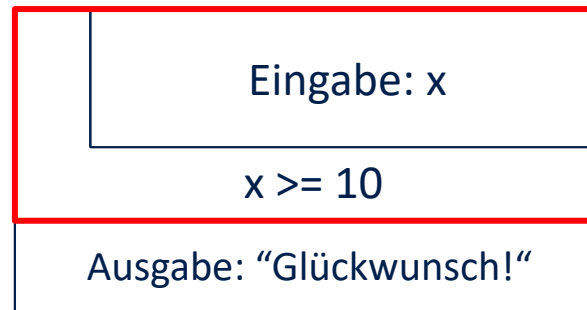
Do-While – Beispielaufgabe – PAP – Bedingung



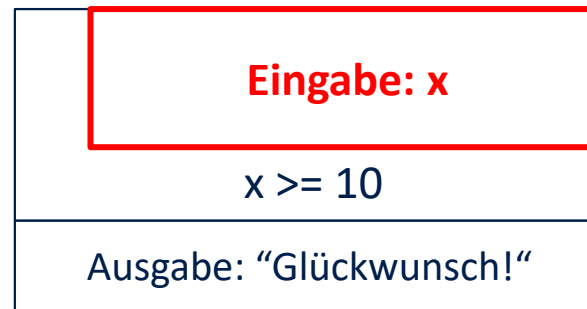
Do-While – Beispielaufgabe – Struktogramm



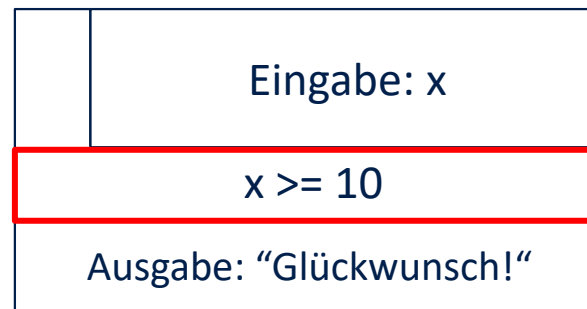
Do-While – Beispielaufgabe – Struktogramm – Schleife



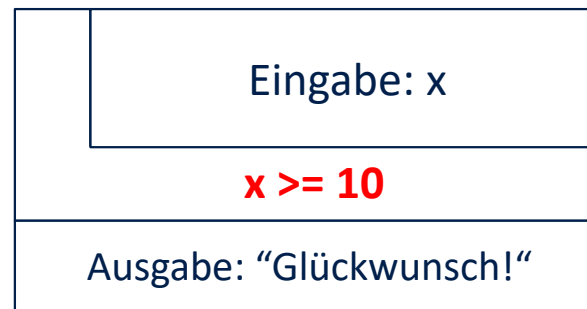
Do-While – Beispielaufgabe – Struktogramm – Schleifenrumpf



Do-While – Beispielaufgabe – Struktogramm – Schleifenfuß



Do-While – Beispielaufgabe – Struktogramm – Bedingung



Do-While – Beispielaufgabe – Pseudocode

```
Programm „Do-While-Beispiel“  
{  
    {  
        Eingabe: x  
    }  
    solange(x>=10)  
    Ausgabe: „Gratulation!“  
}
```


Do-While – Beispielaufgabe – Pseudocode – Schleife

Programm „Do-While-Beispiel“

```
{  
    {  
        Eingabe: x  
    }  
    solange(x>=10)  
    Ausgabe: "Gratulation!"  
}
```

Do-While – Beispielaufgabe – Pseudocode – Schleifenrumpf

```
Programm „Do-While-Beispiel“  
{  
    {  
        Eingabe: x  
    }  
    solange(x>=10)  
    Ausgabe: „Gratulation!“  
}
```

Do-While – Beispielaufgabe – Pseudocode – Schleifenfuß

```
Programm „Do-While-Beispiel“  
{  
    {  
        Eingabe: x  
    }  
    solange(x>=10)  
    Ausgabe: „Gratulation!“  
}
```

Do-While – Beispielaufgabe – Pseudocode – Bedingung

```
Programm „Do-While-Beispiel“  
{  
    {  
        Eingabe: x  
    }  
    solange(x>=10)  
    Ausgabe: „Gratulation!“  
}
```

Do-While – Beispielaufgabe – Quellcode

```
#include<stdio.h>

main()
{
    int x;

    do
    {
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
        scanf("%d",&x);
    }
    while(x>=10);

    printf("Glückwunsch!");
}
```

Do-While – Beispielaufgabe – Quellcode – Schleife

```
#include<stdio.h>
```

```
main()  
{
```

```
    int x;
```

```
    do
```

```
    {
```

```
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
```

```
        scanf("%d",&x);
```

```
    }
```

```
    while(x>=10);
```

```
    printf("Glückwunsch!");
```

```
}
```

Do-While – Beispielaufgabe – Quellcode – Schleifenrumpf

```
#include<stdio.h>

main()
{
    int x;

    do
    {
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
        scanf("%d",&x);
    }
    while(x>=10);

    printf("Glückwunsch!");
}
```

Do-While – Beispielaufgabe – Quellcode – Schleifenfuß

```
#include<stdio.h>

main()
{
    int x;

    do
    {
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
        scanf("%d",&x);
    }
    while(x>=10);

    printf("Glückwunsch!");
}
```


Do-While – Beispielaufgabe – Quellcode – Bedingung

```
#include<stdio.h>

main()
{
    int x;

    do
    {
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
        scanf("%d",&x);
    }
    while(x>=10);

    printf("Glückwunsch!");
}
```

Do-While – Beispielaufgabe – Quellcode – Syntax

```
#include<stdio.h>

main()
{
    int x;

    do
    {
        printf("Geben Sie bitte eine ganze Zahl kleiner 10 ein: ");
        scanf("%d",&x);
    }
    while(x>=10);

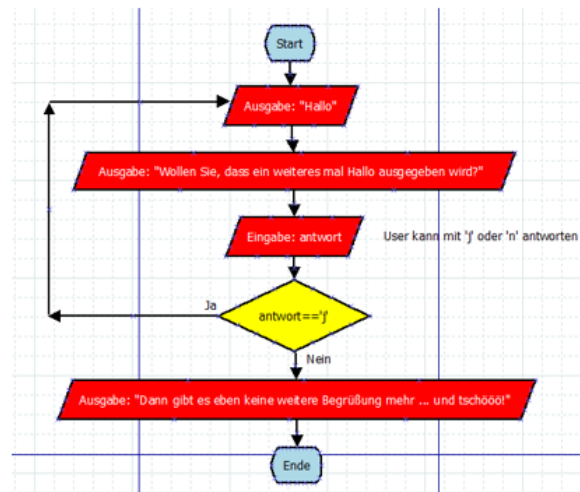
    printf("Glückwunsch!");
}
```

Do-While – Gemeinsame Übung A_02_01_01



Aufgabe_02_01_01

Gegeben sei der folgende PAP:



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 Gliese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IdNr.: DE 209 768 248

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



GLS zertifiziert nach
DIN EN ISO 9001:2015 Reg. Nr. 015284-00015
Zulassung nach AGN Reg. Nr. 015284-00015

**VIELEN DANK
FÜR IHRE
AUFMERKSAMKEIT!**