



**WBS  
TRAINING**

## Programmierung(1)

# Agenda

- Interaktive Programme
  - Eingabefunktion in ANSI C
  - Adressoperator
  - Löschen des Tastaturpuffers
- Ausführliches Training + Ergebnisbesprechung
- Fachpraktische Anwendungen

# Eingabe (kurze Wiederholung) — Darstellungen (im PAP / Struktogramm / Pseudocode)

- Im PAP gibt es ein eigenes Symbol für Eingaben:

(wenngleich es sich von Ausgaben nur durch seine Beschriftung unterscheidet)



(hinter dem Doppelpunkt wird der Name jener **Variable** notiert, die den **Eingabewert abspeichern** soll)

- Im Struktogramm wird eine Eingabe als „Vorgang“ betrachtet und in einem Rechteck notiert:



- Im Pseudocode wird dann einfach nur der Text (natürlich ohne Grafik) übernommen:

Eingabe: a

# Eingabe – 1. Beispiel für ein „interaktives“ Programm

- Wir haben bisher das bereits angesprochene **EVA-Prinzip** (Eingabe / Verarbeitung / Ausgabe) nur simuliert, indem wir die Eingabe durch eine im Programm festgelegte Initialisierung ersetzten. Solche Programme sind dann natürlich nicht „interaktiv“, da sie dem User offensichtlich keine Möglichkeit geben, Einfluss auf das Programm zu nehmen.
- Die folgende Beispielsaufgabe soll dies ändern. Wie üblich werden wir zunächst deren Lösung mittels **PAP**, **Struktogramm** und **Pseudocode** betrachten.
- Erst im Anschluss wollen wir den entsprechenden **Quellcode** erstellen und hierbei die Syntax von Eingaben in ANSI C kennenlernen.

## Aufgabenstellung

Zu Beginn des Programmes soll ein User ...

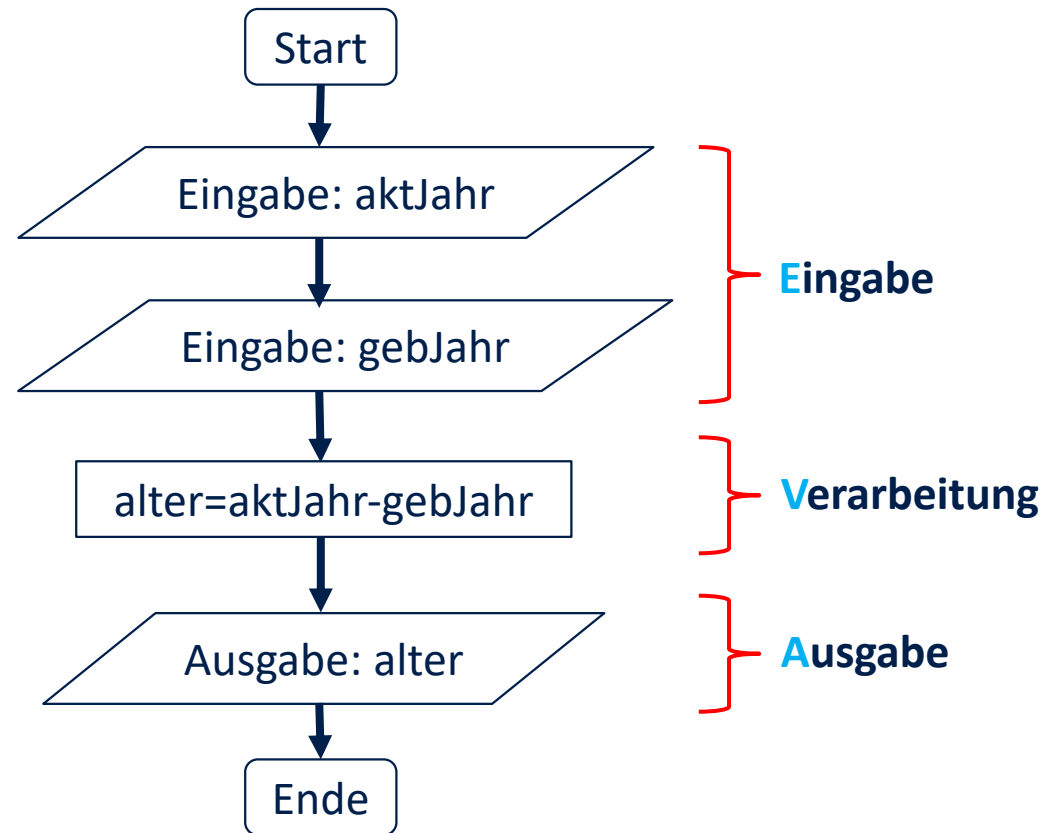
- a) die **aktuelle Jahreszahl** und
- b) sein **Geburtsjahr** eingeben.

Daraufhin soll das Programm errechnen, **wie alt** der User im aktuellen Jahr wird (oder bereits wurde).

Dieses Rechenergebnis wird zunächst **abgespeichert** und dann auf der Konsole **ausgegeben**.

Anschließend endet das Programm.

# Musterlösung - PAP



# Musterlösung - Struktogramm

Eingabe: aktJahr
Eingabe: gebJahr
$\text{alter} = \text{aktJahr} - \text{gebJahr}$
Ausgabe: alter

# Musterlösung - Pseudocode

```
Programm „Altersbestimmung“  
{  
    Eingabe: aktJahr  
    Eingabe: gebJahr  
    alter=aktJahr-gebJahr  
    Ausgabe: alter  
}
```

# Musterlösung (provisorisch) – ANSI C – Eingabebefehl

```
#include<stdio.h>

main()
{
    int aktJahr;
    int gebJahr;
    int alter;

    scanf("%d",&aktJahr);
    scanf("%d",&gebJahr);

    alter=aktJahr-gebJahr;

    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
}
```



# Musterlösung (provisorisch) – ANSI C – Typ (der zu füllenden Variable)

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

```
    scanf("%d",&aktJahr);  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

Für die **Typen** werden die selben Symbole wie bei printf verwendet:

# Musterlösung (provisorisch) – ANSI C – Typ (der zu füllenden Variable)

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

Die Notation zum Typ geschieht stets in Form eines **Strings**  
=> also in (doppelten) Anführungszeichen

```
    scanf("%d",&aktJahr);  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

# Musterlösung (provisorisch) – ANSI C – **Name** (der zu füllenden Variable)

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

Der Name der zu füllenden Variable erscheint  
nach dem String (durch Komma getrennt)

```
    scanf("%d",&aktJahr);  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

# Musterlösung (provisorisch) – ANSI C – Adressoperator

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

```
    scanf("%d",&aktJahr);  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

Vor dem Namen muss stets ein „kaufmännisches UND“ notiert werden. Es handelt sich dabei um den sogenannten **Adressoperator**, der dem Compiler mitteilt, dass wir hier nicht von dem Wert der Variable *aktJahr* sprechen, sondern von deren Speicherort, also jenem Ort, an dem die User-Eingabe für *aktJahr* abgespeichert werden soll.

# Musterlösung (provisorisch) – ANSI C – Bibliothek

printf und scanf sind beide in der selben Bibliothek **stdio.h**

```
#include<stdio.h>

main()
{
    int aktJahr;
    int gebJahr;
    int alter;

    scanf("%d",&aktJahr);
    scanf("%d",&gebJahr);

    alter=aktJahr-gebJahr;

    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
}
```

# Musterlösung (provisorisch) – warum provisorisch? – 2 Gründe

```
#include<stdio.h>

main()
{
    int aktJahr;
    int gebJahr;
    int alter;

    scanf("%d",&aktJahr);
    scanf("%d",&gebJahr);

    alter=aktJahr-gebJahr;

    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
}
```

# Musterlösung (provisorisch) – warum provisorisch? – 1. Grund

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

```
    scanf("%d",&aktJahr);  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

Wenn wir den scanf-Befehl ausführen lassen, so erscheint auf der Konsole ein „blinkender“ Cursor, der auf die Eingabe des Users „wartet“. Es fehlt also noch ein „**Aufforderungs-Text**“, etwa in der Form „*Geben Sie bitte das aktuelle Jahr ein:* “. Genau dies können wir aber natürlich durch einen entsprechenden printf-Befehl vor jeder scanf-Anweisung nachholen:

# Musterlösung (provisorisch) – warum provisorisch? – 1. Grund

```
#include<stdio.h>

main()
{
    int aktJahr;
    int gebJahr;
    int alter;

    printf("Geben Sie bitte das aktuelle Jahr ein: ");
    scanf("%d",&aktJahr);
    printf("Geben Sie bitte Ihr Geburtsjahr ein: ");
    scanf("%d",&gebJahr);

    alter=aktJahr-gebJahr;

    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
}
```



# Musterlösung (provisorisch) – warum provisorisch? – 2. Grund

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

```
    printf("Geben Sie bitte das aktuelle Jahr ein: ");  
    scanf("%d",&aktJahr);  
    printf("Geben Sie bitte Ihr Geburtsjahr ein: ");  
    scanf("%d",&gebJahr);
```

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

Die Eingabe des Users wird zunächst im **Tastatur-Puffer** abgelegt, von wo sie durch scanf ausgelesen wird. Nun ergibt sich aber das Problem, dass nach dem ersten scanf-Aufruf dieser Tastaturpuffer bereits gefüllt ist und alle weiteren scanf-Aufrufe Gefahr laufen, diesen Tastaturpuffer-Inhalt erneut auszulesen (und also nicht auf die neue Eingabe des Users zu warten). Um dies zu umgehen, sollten wir vor jeder scanf-Anweisung (spätestens aber vor der zweiten und allen folgenden) den Tastaturpuffer löschen. Dies geschieht mittels des Funktionsaufrufes: **fflush(stdin)**.

# Musterlösung (provisorisch) – warum provisorisch? – 2. Grund

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

```
main()  
{
```

```
    int aktJahr;  
    int gebJahr;  
    int alter;
```

```
    printf("Geben Sie bitte das aktuelle Jahr ein: ");
```

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

```
    printf("Geben Sie bitte Ihr Geburtsjahr ein: ");
```

```
    fflush(stdin);
```

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

```
    alter=aktJahr-gebJahr;
```

```
    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
```

```
}
```

Die Funktion **fflush()** hat Vor- und Nachteile.

Ein wichtiger Vorteil ist, dass sie eine komfortable Möglichkeit bietet, das geschilderte Problem zu lösen. Daher wird diese Funktion in Einstiegs-Lehrbüchern für ANSI C oft verwendet.

Ein Nachteil ist aber, dass sie nicht zum offiziellen ANSI C-Standard gehört und nicht auf jedem Betriebssystem anwendbar ist. Für einen Einstiegs-Kurs im Fach der prozeduralen Programmierung können wir diesen Nachteil aber vernachlässigen.

# Musterlösung (provisorisch)

```
#include<stdio.h>

main()
{
    int aktJahr;
    int gebJahr;
    int alter;

    printf("Geben Sie bitte das aktuelle Jahr ein: ");
    scanf("%d",&aktJahr);
    printf("Geben Sie bitte Ihr Geburtsjahr ein: ");
    fflush(stdin);
    scanf("%d",&gebJahr);

    alter=aktJahr-gebJahr;

    printf("Dieses Jahr werden (oder wurden) Sie %d",alter);
}
```

# Interaktiver Code – Gemeinsame Übung A\_01\_03\_01

## Allgemeiner Hinweis:

Wir werden in diesem Baustein in der Regel vom „**gutmütigen User**“ ausgehen und daher üblicherweise nicht kontrollieren, ob seine Eingaben zulässig sind, oder ob er (z.B.) Buchstaben eingab, wo eigentlich Zahlen gefordert waren.

Auf diese Weise wird uns die Konzentration auf die eigentliche Aufgabenstellung leichter fallen (zumal wir zur Zeit eine Eingabe-Kontrolle auch noch nicht technisch umsetzen könnten).

Dennoch werden wir natürlich auch dies lernen, es dann aber nicht in jeder Aufgabe wiederholen, da die immer gleichen Programmabschnitte wertvolle Unterrichtszeit kosten würden.



### Aufgabe\_01\_03\_01

Gegeben sei das folgende Struktogramm:

#### ML\_01\_03\_01

Eingabe: breiteRechteck
Eingabe: hoeheRechteck
$\text{flaecheRechteck} = \text{breiteRechteck} * \text{hoeheRechteck}$
Ausgabe: flaecheRechteck

**Konsolenausgabe** (eingabe Daten sind nur Beispiele):  
Geben Sie bitte die Breite des Rechteckes ein: 2  
Geben Sie bitte die Hoehe des Rechteckes ein: 5  
Die Flaeche des Rechteckes betraegt: 10.000000

#### 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: GENODEM1GLS



2020 zertifiziert nach  
ISO 9001:2015 und ISO 14001:2015  
Zertifizierungsstelle: KZP Reg. Nr. 0100000000

**VIELEN DANK  
FÜR IHRE  
AUFMERKSAMKEIT!**