



---

# Prozedurale Programmierung

## Selektionen

**Hochschule Rosenheim - University of Applied Sciences**

**WS 2018/19**

**Prof. Dr. F.J. Schmitt**



# Kontrollstrukturen

---

- Sprachmittel zur Steuerung des Ablaufs eines Computerprogramms
- Visualisierung des Programmablaufs mittels spezieller Diagramme
  - ⊞ Struktogramme (**Nassi-Shneiderman-Diagramme**)
    - ⊞ Grafische Darstellung von **Sequenz**, **Selektion** und **Iteration** (Mittel der **Strukturierten Programmierung**)
  - ⊞ Programmablaufpläne

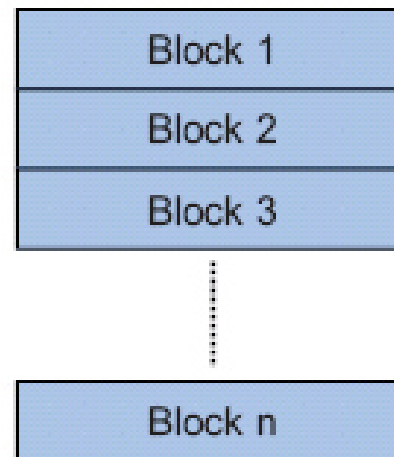


# Sequenz

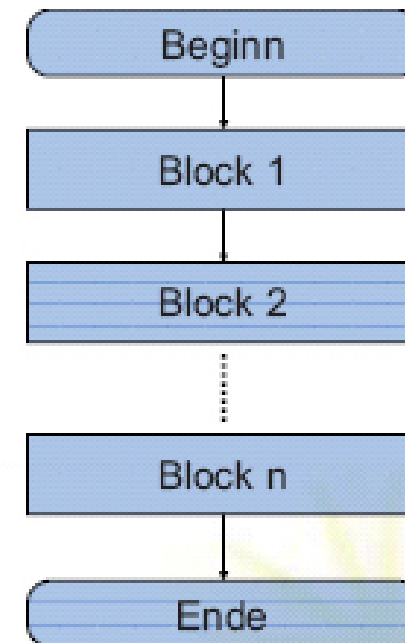
- Folge von Anweisungen oder Anweisungs-Blöcken

```
int main(void)
{
    printf(" A\n ");
    printf(" B\n ");
    printf(" C\n ");
    .
    .
    .
    printf(" Z\n ");
}
```

Struktogramm



Programmablauf





# Selektionen

---

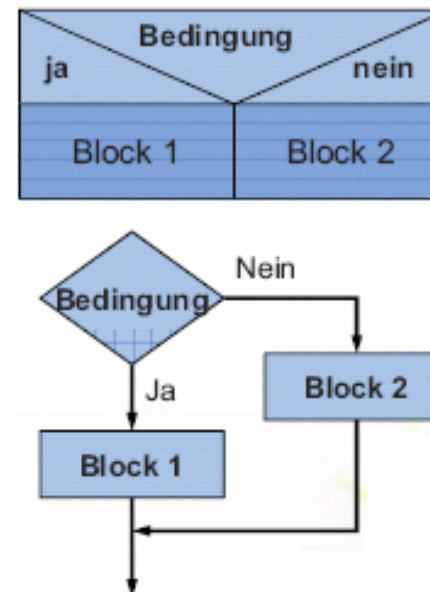
- Zielsetzung: Verschiedene Programmteile werden in Abhängigkeit von einer Bedingung ausgeführt
  
- Zwei wichtige Sprachmittel in C:
  - ⊞ `if`-Anweisung
  - ⊞ `switch`-Anweisung



## if-Anweisung (1)

- Ermöglicht das alternative Ausführen zweier Programmtteile abhängig davon, ob eine Bedingung „wahr“ oder „falsch“ ergibt

```
if (Bedingung)  
    Block  
else  
    Block
```



Block:

- entweder eine **einzelne** Anweisung
- oder eine durch geschweifte Klammern gruppierte Folge von Anweisungen



## if-Anweisung (2)

---

- else-Block bzw. else-Zweig kann weggelassen werden

```
if (Bedingung)  
    Block
```

⊞ Trifft Bedingung nicht zu, so wird if-Anweisung übersprungen

- if-Zweig kann **nicht** weggelassen werden

- Soll ein Block nur ausgeführt werden, wenn die Bedingung nicht zutrifft, dann

```
if (!Bedingung)  
    Block
```

Negierung der Bedingung



## if-Anweisung (3)

---

- Verschachteln von if-Anweisungen möglich

```
if (Bedingung1)
    if (Bedingung2)
        Block 1
    else
        Block 2
else
    Block 3
```

Achten Sie auf korrekte und konsistente Einrückung!



## if-Anweisung (4)

---

- Ist folgende Formatierung korrekt?

```
if (Bedingung1)
    if (Bedingung2)
        Block 1
else
    Block 2
```

Einrücken dient der besseren Lesbarkeit –  
syntaktisch spielt es keine Rolle!





## if-Anweisung (5)

---

- Was muss man tun, damit das `else` zum ersten/äußeren `if` gehört?

```
if (Bedingung1)
{
    if (Bedingung2)
        Block 1
}
else
    Block 2
```

Gruppieren von Anweisungen in `{ }` - Blöcke



## Beispiel `if`-Anweisung (1)

---

- Es soll eine Meldung ausgegeben werden, wenn die Werte in Variable `a` und Variable `b` gleich sind

```
if ( a == b )  
    printf("a und b sind gleich groß\n");
```

- Zusätzlich Meldung am Bildschirm, die aussagt, ob `a` kleiner als `b`

```
if ( a == b )  
    printf("a und b sind gleich groß\n");  
else  
    if ( a < b )  
        printf("a ist kleiner als b\n");
```



## Beispiel `if`-Anweisung (2)

---

- Weitere Meldung in dem Fall, dass a größer ist als b

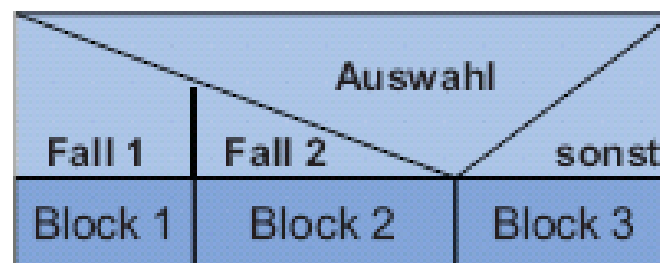
```
if ( a == b )  
    printf("a und b sind gleich groß\n");  
else  
    if ( a < b )  
        printf("a ist kleiner als b\n");  
    else  
        printf("a ist größer als b\n");
```



## Beispiel `if`-Anweisung (3)

- Verbesserte Schreibweise:

```
if ( a == b )  
    printf("a und b sind gleich groß\n");  
else if ( a < b )  
    printf("a ist kleiner als b\n");  
else  
    printf("a ist größer als b\n");
```





# Häufige Fehler (1)

---

➤ Fehlerhaft:

```
if ( a == b )  
    printf("a und b sind gleich groß\n");  
if ( a < b )  
    printf("a ist kleiner als b\n");  
else  
    printf("a ist größer als b\n");
```

Weglassen des ersten `else`

⇒ verändertes Verhalten -

Zwei Meldungen, wenn a gleich b!



## Häufige Fehler (2)

- In Bedingung erfolgt anstatt einer Abfrage auf Gleichheit (Operator ==) eine Zuweisung (=)

```
if ( a = b )  
    Block
```

- ✚ bei Gleichheit einer Variablen zu einer Konstanten → Konstante auf linke Seite schreiben
- ✚ statt `if(x == 5)` also `if(5 == x)`
- ✚ 5 ist kein Lvalue, bei Tippfehler `5 = x` gibt es einen Compilefehler
- Empfehlung:  
In MSVC 2010: Warning Level auf W4 schalten  
(Projekteigenschaften) → `if(a = b)` gibt eine Warning



## Häufige Fehler (3)

---

### ➤ Falsches Semikolon

```
if (Bedingung) ;  
    Block
```

Falls Bedingung wahr wird Leeraanweisung durchgeführt - Block wird immer ausgeführt!



# Aufgabe

---

- Schreiben Sie ein C-Programm, welches den Benutzer zur Eingabe einer ganzen Zahl (Punkte) auffordert und anschließend die den Punkten entsprechende Benotung ausgibt.

Punkte	Note
0 .. 40	Ungenügend
41 .. 55	Genügend
56 .. 70	Befriedigend
71 .. 80	Gut
81 .. 100	Sehr Gut

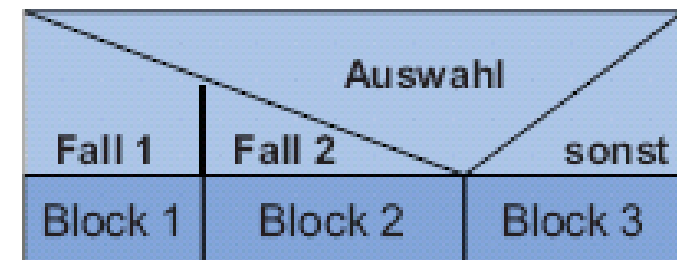
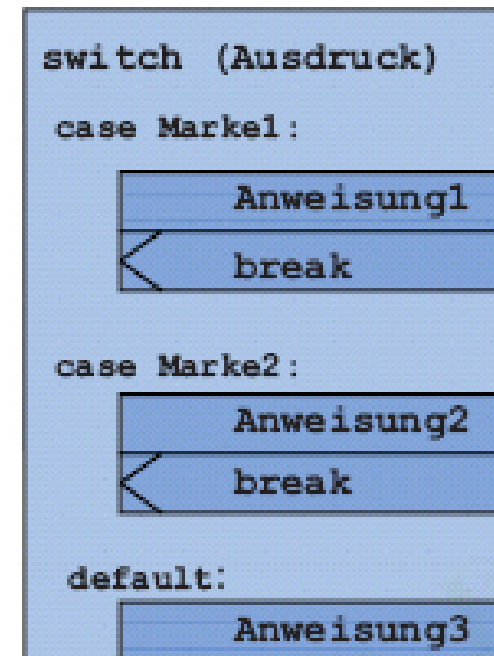




## switch-Anweisung (1)

- Ermöglicht das selektive Ausführen von beliebig vielen verschiedenen Programmteilen

```
switch (Ausdruck)
{
    case Marke1: Anweisungen
                break;
    case Marke2: Anweisungen
                break;
    . . .
    default:    Anweisungen
                break;
}
```





## switch-Anweisung (2)

---

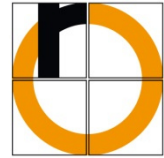
- Reihenfolge der Marken ist beliebig
- Ausdruck kann beliebiger C-Ausdruck sein
  - ⊞ Wird zu Beginn der Anweisung ausgewertet
  - ⊞ Stimmt der Wert des Ausdrucks mit dem Namen einer Sprungmarke überein  $\Rightarrow$  Verzweigung zu dieser
  - ⊞ Existiert die erforderliche Sprungmarke nicht, so wird zur `default`-Sprungmarke verzweigt
  - ⊞ Existiert auch diese nicht, so wird die gesamte `switch`-Anweisung übersprungen



## Beispiel switch-Anweisung

- Variable `zahl` ist vom Typ `long` und enthält eine beliebige ganze Zahl

```
switch (zahl)
{
    case 1:    printf(" eins\n");
               break;
    case 2:    printf(" zwei\n");
               break;
    case 3:    printf(" drei\n");
               break;
    default:   printf(" sonst\n");
               break;
}
```



# Zusammenfassung

---

- if / else Anweisung
- switch / case Anweisung