Programmieren in C
für Elektrotechniker

Kapitel 2: Einfache Beispielprogramme

- **Lernziele**
  - Programme lesen lernen
    (→ Programmiererfahrung durch abschauen/ändern)
  - Mit Variablen, Konstanten, Kontrollstrukturen, E/A vertraut machen

  → Konzepte später noch genauer
    (Präprozessor, Funktionen, Variablen etc.)
  - Ausführung von Unterprogrammen verstehen.

Ausgabe von Quadratzahlen

```
#include <stdio.h>

int main (void)
{
   int lv = 1;
   printf ("Die ersten zehn Quadratzahlen sind:\n");
   while (lv <= 10) {
      printf ("%d ", lv * lv);
      lv = lv + 1;
   }
   return 0;
}
```

- *Was macht das Programm?*

Ausgabe von:

```
Die ersten zehn Quadratzahlen sind:
1 4 9 16 25 36 49 64 81 100
```

Ausgabe von Quadratzahlen

```
#include <stdio.h>

int main (void)
{
   int lv = 1;
   printf ("Die ersten zehn Quadratzahlen sind:\n");
   while (lv <= 10) {
      printf ("%d ", lv * lv);
      lv = lv + 1;
   }
   return 0;
}
```

- **bekannt**:
  - **`#include`**: Einbindung anderer Dateien (Bibliotheken)
  - Hauptfunktion: **`main ()`**
  - Leerzeichen, Leerzeilen, Kommentare ohne Bedeutung
  - ‚;': Ende einer Anweisung oder Definition (über mehrere Zeilen möglich)
  - Ausgabe über **`printf ()`**

Ausgabe von Quadratzahlen

```
#include <stdio.h>

int main (void)
{
   int lv = 1;
   printf ("Die ersten zehn Quadratzahlen sind:\n");
   while (lv <= 10) {
      printf ("%d ", lv * lv);
      lv = lv + 1;
   }
   return 0;
}
```

- **while-Schleife**:
  - Prüfe vor jedem Schleifendurchlauf, ob Bedingung erfüllt ist (**`lv`** ≤ **`10`**).
    → Ergebnis = wahr: Schleifenrumpf wird einmal durchlaufen.
  - In jedem Schleifendurchlauf wird **`printf ()`** ausgeführt und **`lv`** um **`1`** erhöht.
    → Mehrere Anweisungen in einem Durchlauf: Anweisungen zu einem Block zusammenfassen **`{ …; …; …;}`**

```c
#include <stdio.h>

int main (void) {
   int a = 10;
   double b = 1000.00;
   double c = b;
   double d = 5.0;
   int j = 1;


   while (j <= a) {

      c = c * (1. + d/100.);

      j = j + 1;
   }


   return 0;
}
```

- *Was macht das Programm?*


```c
#include <stdio.h>

int main (void) {
   int laufzeit = 10;
   double grundkapital = 1000.00;
   double kapital = grundkapital;
   double zins = 5.0;
   int jahr = 1;


   while (jahr <= laufzeit) {

      kapital = kapital * (1. + zins/100.);

      jahr = jahr + 1;
   }


   return 0;
}
```

- *Was macht das Programm?*

Zinsberechnung

```
#include <stdio.h>

int main (void) {
   int laufzeit = 10;
   double grundkapital = 1000.00;
   double kapital = grundkapital;
   double zins = 5.0;
   int jahr = 1;
   printf ("Tabelle fuer Grundkapital %7.2f EUR\n", grundkapital);
   printf ("Kapitalstand zum Jahresende:\n");
   while (jahr <= laufzeit) {
      printf ("\nJahr: %2d\t", jahr);
      kapital = kapital * (1. + zins/100.);
      printf ("Kapital: %7.2f EUR", kapital);
      jahr = jahr + 1;
   }
   printf ("\n\nAus %7.2f EUR Grundkapital\n", grundkapital);
   printf ("wurden in %d Jahren %7.2f EUR\n", laufzeit, kapital);
   return 0;
}
```

- *Was macht das Programm?*

Zinsberechnung

```
#include <stdio.h>

int main (void) {
   int laufzeit = 10;
   double grundkapital = 1000.
   double kapital = grundkapit
   double zins = 5.0;
   int jahr = 1;
   printf ("Tabelle fuer Grund
   printf ("Kapitalstand zum J
   while (jahr <= laufzeit) {
      printf ("\nJahr: %2d\t",
      kapital = kapital * (1.
      printf ("Kapital: %7.2f
      jahr = jahr + 1;
   }
   printf ("\n\nAus %7.2f EUR
   printf ("wurden in %d Jahre
   return 0;
}
```

Ausgabe von:

```
Tabelle fuer Grundkapital 1000.00 EUR
Kapitalstand zum Jahresende:

Jahr: 1 Kapital: 1050.00 EUR
Jahr: 2 Kapital: 1102.50 EUR
Jahr: 3 Kapital: 1157.62 EUR
Jahr: 4 Kapital: 1215.51 EUR
Jahr: 5 Kapital: 1276.28 EUR
Jahr: 6 Kapital: 1340.10 EUR
Jahr: 7 Kapital: 1407.10 EUR
Jahr: 8 Kapital: 1477.46 EUR
Jahr: 9 Kapital: 1551.33 EUR
Jahr: 10 Kapital: 1628.89 EUR

Aus 1000.00 EUR Grundkapital
wurden in 10 Jahren 1628.89 EUR
```

Zinsberechnung

```
#include <stdio.h>

int main (void) {
   int laufzeit = 10;
   double grundkapital = 1000.00;
   double kapital = grundkapital;
   double zins = 5.0;
   int jahr = 1;
   printf ("Tabelle fuer Grundkapital %7.2f EUR\n", grundkapital);
   printf ("Kapitalstand zum Jahresende:\n");
   jahr = 1;
   while (jahr <= laufzeit) {
      printf ("\nJahr: %2d\t", jahr);
      ...
      jahr = jahr + 1;
   }
...
```

- Schleife: 10 Durchläufe, bis **`jahr == 10`**
  `jahr` → Laufvariable der Schleife
- Achtung: **`"="`**: Zuweisungsoperator
  `"=="`: Vergleich auf Gleichheit
- **`printf ()`**: 1. Argument " … " → kontrollierende Zeichenkette

Zinsberechnung

```
printf ("Jahr: %2d\tKapital: %7.2f EUR.", jahr, kapital);
```

- **`printf ()`**:
  - 1. Argument " … " → kontrollierende Zeichenkette
    → Formatelemente mit vorangestelltem %-Zeichen.
  - **`%2d`**: 1. Parameter (→ **`jahr`**) wird als ganze Zahl mit 2 Ziffern ausgegeben (rechtsbündig)
  - **`%-2d`**: Entsprechend linksbündig.
  - **`\t`**: Vorrücken zur nächsten Tabulator-Marke.
  - **`%7.2f`**: 7-stellige Gleitkommazahl mit 2 Dezimalziffern (4 Stellen vor dem Dezimalpunkt).
  - Für jeden Parameter muss es ein Steuerelement geben
    → Typen müssen übereinstimmen.
  - Ausgabe von Variablenwerten und Konstanten.

Euklidischer Algorithmus

```
#include <stdio.h>

int main (void)
{
   int x, y;
   printf ("\nGeben Sie bitte einen Wert fuer x ein: ");
   scanf ("%d", &x);
   printf ("Geben Sie bitte einen Wert fuer y ein: ");
   scanf ("%d", &y);
   printf ("Der ggT von %d und %d ist: ", x, y);
   while (x != y)
   {
      if (x < y)
         y = y - x;
      else
         x = x - y;
   }
   printf ("%d\n", x);
   return 0;
}
```

- Verwendung von E/A-Funktionen.
- Eingabe über Tastatur.

Euklidischer Algorithmus

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

- **`scanf ()`**:
  - Eingabe von der Tastatur.
  - Analog zu **`printf ()`**.
  - Eingabe nur in Variablen möglich
    → Parameter dürfen keine Konstanten oder Ausdrücke sein.
  - genauer:
    Parameter sind Adressen von Variablen. *(→ später genauer)*
  - In Bibliothek **`<stdio.h>`** definiert.