

# Dynamisches Array von Struktur-Datensätzen in C++

## 1. Praktikumstermin, X-Gruppe

*Praktika sind Prüfungsvorleistungen, die zum genannten Termin erbracht werden müssen. Bei Verhinderung durch Krankheit ist eine ärztliche Bescheinigung der Arbeitsunfähigkeit vorzulegen. Dieses Deckblatt ist zum Praktikumstermin ausgefüllt mitzubringen und unterschrieben abzugeben. Das Programm (C++-Projekt) wird während des Praktikums zum angegebenen Zeitpunkt im Informatik-Labor abgenommen.*

Name	Vorname	Matrikelnummer	Ihre Unterschrift

*Zu Beginn dieses Praktikumstermins müssen Sie Quellcode-nahe Struktogramme (siehe Hinweise am Ende) für die Funktionen `main` und `sucheMinCO2` vorlegen, um den Lösungsweg für diese Aufgabe vorzuzeichnen.*

## Ziel

Das Programm soll einige Unterschiede von C++ gegenüber der C-Programmierung aufzeigen. Alles was Sie bisher über C-Programmierung gehört haben, können Sie praktisch auch in C++-Programme einbauen, da C eine Untermenge der Programmiersprache C++ ist. Vieles geht in C++ aber eleganter als in C.

Bei dieser Aufgabe sollen einige wenige Daten von Stromkraftwerken in das Programm eingegeben und verarbeitet werden. Die zentrale Datenstruktur in Ihrem Programm ist ein *dynamisches* Array (deutsch: Feld) vom global definierten Struktur-Datentyp `TKraftwerk`:

```
struct TKraftwerk
{
    char sTyp[18];           //Typ
    double maxP_MW;         //max. Leistung in MW
    double kosten_EUR_kWh; //Gestehungskosten in EUR/kWh
    double CO2_g_kWh;       //CO2 im Betrieb in g/kWh
    char ort[18];           //Standort
};
```

Die Funktionsprototypen der von Ihnen zu implementierenden Funktionen lauten wie folgt:

```
void fuegeEin(TKraftwerk pKwFeld[], int &anzahl);
void gibAus(TKraftwerk pKwFeld[], int anzahl);
int sucheMinCO2(TKraftwerk pKwFeld[], int anzahl);
```

Frischen Sie zum Verständnis Ihr Wissen über die Arten der Parameterübergabe („call-by-value“ und „call-by-reference“) auf! Anstelle der Referenz für `anzahl` in der Funktion `fuegeEin` können Sie auch mit einem Zeiger arbeiten. Das Programm ist als Konsolen-Anwendung mit der Entwicklungsumgebung Turbo C++ zu realisieren.

## main – die Steuerzentrale des Programms

Die Eingabe läuft in einer Schleife, die verlassen wird, sobald die boolesche Variable `wirdBeendet` wahr wird:

```
int main(void)
{
    //Deklaration von sofort benötigten Variablen:
    //Zeiger: pKwFeld (für das dynamische Array der Kraftwerksdatensätze)
    //Integer-Variable: maximaleAnzahl (Zahl möglicher Datensätze im Array)
    //Integer-Variable: aktuelleAnzahl (Zahl bereits eingegebener Datensätze)
    //Benutzer-Interaktion:
    //Ausgabe: "Wie viele Kraftwerks-Datensaetze wollen Sie eingeben?"
    //Eingabe: maximaleAnzahl
    //dynamischen Speicherplatz für Feld anfordern
    //abbrechen, falls Speicher-Anforderung nicht erfolgreich war
    bool wirdBeendet = false;
    while(!wirdBeendet)
    {
        /*Ausgaben:
        "Was wollen Sie tun?"
        "<Erzeugen> (e/E) Neuen Kraftwerksdatensatz erzeugen"
        "<Ausgeben> (a/A) Kraftwerksdatensaetze ausgeben"
        "<CO2 min> (c/C) Kraftwerk mit kleinstem CO2-Ausstoss ausgeben"
        "<Beenden> (b/B) Programm beenden"
        "?"
        */
        //Zeichen-Variable: eingabe (Zeichen der Menü-Auswahl)
        //Einlesen des Zeichens der Menüauswahl in die Variable eingabe
        eingabe = tolower(eingabe); //Großbuchstaben in kleine wandeln
        switch (eingabe) //verzweigen
        {
            case 'e':
                //Fehlermeldung, wenn alle Feldelemente belegt
                //sonst:
                //Funktionsaufruf: Einfügen eines neuen Datensatzes
                //Funktionsaufruf: Ausgabe aller eingegebenen Datensätze
                break;
            case 'a':
                //Funktionsaufruf: Ausgabe aller eingegebenen Datensätze
                break;
            case 'c':
                //Fehlermeldung, wenn noch kein Datensatz vorhanden
                //sonst:
```

```
        //Funktionsaufruf: Suche des Datensatzes mit kleinstem CO2-Wert
        //Ausgabe von Standort und Kraftwerkstyp dieses Datensatzes
    break;
    case 'x':
        wirdBeendet = true;
    } //switch
} //while
//dynamisch angeforderten Speicher freigeben
return 0;
} //main
```

## Weitere zu implementierende Funktionen

```
void fuegeEin(TKraftWerk pKwFeld[], int &anzahl);
```

- Anlegen einer Variable `dasWerk` vom Struktur-Datentyp `TKraftwerk`
- Einlesen (`cin`) der Daten in die Strukturvariable `dasWerk`
- Einfügen des Datensatzes `dasWerk` am Ende (Position angegeben durch `anzahl`) der bisher eingegebenen Datensätze in das Array
- Erhöhen der Zählvariable `anzahl`

```
void gibAus(TKraftWerk pKwFeld[], int anzahl);
```

- Leeren des Terminalfensters mit `system("cls");`
- Ausgabe einer konstanten Zeichenkette als Tabellenkopf:

```
Typ      max. Leistung [MW]      CO2 [g/kWh]      Standort
-----
```

- Ausgabe der Datensätze (`for`-Schleife) in Tabellenform

```
int sucheCO2Min(TKraftWerk pKwFeld[], int anzahl);
```

- Fehlermeldung und Rücksprung ins Menü, wenn kein Datensätze angelegt wurden, also wenn `anzahl` gleich null ist
- Variable anlegen für den Index des Datensatzes mit dem minimalen CO<sub>2</sub>-Wert
- Vergleich der CO<sub>2</sub>-Werte der einzelnen Kraftwerksdatensätze des Arrays `pKwFeld` (Schleife)
- Rückgabe dieses Index per `return`, damit die aufrufende Funktion `main()` Werte dieses Datensatzes ausgeben kann

## Hinweise

Sie müssen über `#include`-Anweisungen die Bibliotheken `<iostream>` für `cin`, `cout`, `cerr` und `<cstdlib>` für `system()` in Ihr Programm einbinden. Nach dem Inkludieren der Bibliotheken öffnen Sie deren Namensraum `std` mit der Anweisung `using namespace std;`

Beispieldaten für Kraftwerkstypen können Sie der Tabelle 1 entnehmen (Quelle: Kaltschmitt, Streicher, Wiese: Erneuerbare Energien, Springer 2006). Namen für Standorte denken Sie sich selbst aus.

Kraftwerkstyp	Max. Leistung [MW]	Gestehungskosten [EUR/kWh]	CO <sub>2</sub> im Betrieb [g/kWh]
Steinkohle	600	0,035	746
Erdgas	600	0,036	348
Photovoltaik	1,0	0,340	ca. 2
Wind	2,5	0,072	ca. 4
Wasser	28,8	0,058	ca. 1

Tabelle 1: Beispieldaten für Kraftwerke (Gestehungskosten samt Bau, Betrieb und Abriss)

Abbildung 1 zeigt typische Elemente eines Struktogramms. Sie können das Struktogramm zum Beispiel online mit dem dem Java-Applet Structorizer zeichnen. Benutzen Sie Klicks mit der rechten Maustaste, um neue Elemente einzufügen. Im Untermenü `File/Export/Picture` finden Sie Möglichkeiten, das Diagramm zur Weiterbearbeitung zu speichern.

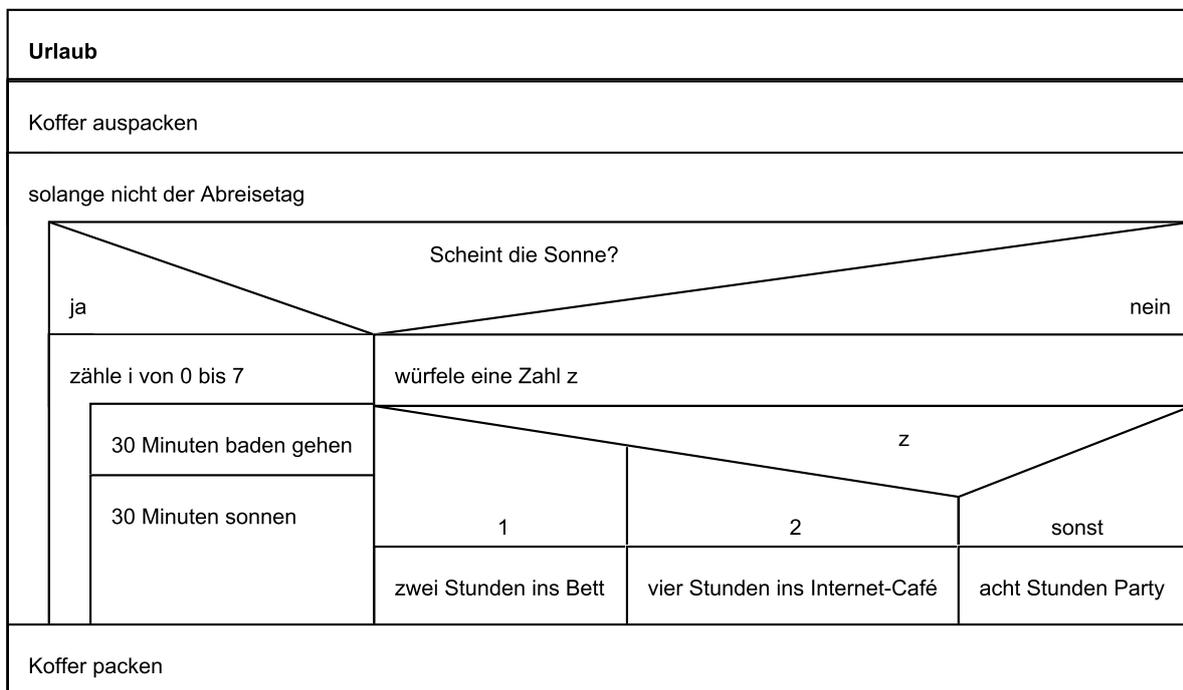


Abbildung 1: Elemente eines Struktogramms nach Nassi und Shneidermann