

Aufzählungstyp
enum

Lernziele

Ich kann den Aufzählungstyp (Enum) definieren und anwenden.

Aufzählungstyp Enum

- Aufzählungstypen sind gedacht für Integer-Variablen, die nicht jeden beliebigen Wert annehmen dürfen, sondern auf eine begrenzte Anzahl von Werten beschränkt sind.
- Diese Werte werden über Namen angesprochen.
- Jede Variable eines Aufzählungstyps darf nur Werte der Aufzählungsliste annehmen.

Beispiel:

```
enum TrafficLight { Red, Orange, Green }; //Red=0, Orange=1, Green=2
```

```
int main()
{
    enum TrafficLight light = Red;

    printf("light=%d\n", light);

    return 0;
}
```

TrafficLight.cpp

Aufzählungstyp Enum

- Der erste Enumerator hat standardmäßig den Wert 0, und der Wert jedes nachfolgenden Enumerators wird jeweils um 1 erhöht:

```
enum TrafficLight { Red, Orange, Green }; //Red=0, Orange=1, Green=2
```

- Der erste Enumerator kann auch vorgegeben werden:

```
enum TrafficLight { Red=10, Orange, Green }; //Red=10, Orange=11, Green=12
```

- Es ist möglich, jedem Enumerator einen eigenen Wert mitzugeben:

```
enum TrafficLight { Red=10, Orange=20, Green=30 }; //Red=10, Orange=20, Green=30
```

- ✓ Die Aufzählungstyp Enum macht ein C-Programm besser lesbar.

Enum als eigenständiger Datentyp

- Mit *typedef* kann ein eigenständiger Datentyp definiert werden:

```
typedef enum {Red, Orange, Green} TrafficStatus;
```

```
int main()  
{  
    TrafficStatus light = Red;  
  
    printf("light=%d\n", light);  
  
    return 0;  
}
```

TrafficStatus.cpp



- Diese Verwendungsart von Aufzählungstypen in ANSI C bietet die meiste Flexibilität und sollte wenn möglich immer so verwendet werden.

Übung 1 Adressbuch Teil 2

Adressbuch mit Enumeration:

Aufgabenstellung siehe M411_A_XCH_Adressbuch_Teil2.pdf