Deklaration und Definition einer Klasse

Klassen sind Baupläne für Programmteile, die bestimmte Eigenschaften haben und bestimmte Aufgaben erfüllen können.

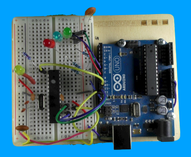
Die Deklaration der Klasse ist mit einem Prospekt vergleichbar, mit dem sie angepriesen wird. Dabei werden die Eigenschaften (oft auch Attribute genannt) und die Methoden (das sind normalerweise Funktionen) aufgelistet.

Dabei wird unterschieden, was der Benutzer selbst verwenden kann (public) und was verborgene Innere Werte sind (private), die vom Benutzer nicht direkt verwendet werden können.

Klassen können also Details vor dem Benutzer verstecken, machen eigentlich genau das, was wir mit den Modulen erreichen wollten.

Eine Klasse wird normalerweise in einer Header - Datei deklariert, zum Beispiel led.h.

```
class Led {
 public:
  void init(int ledPin);
  void ein();
  void aus();
  void blink();
 private:
  int pin;
};
```



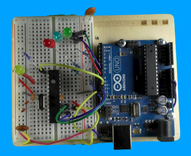
Implementierung von Klassen

Der Bauplan beschreibt bisher erst, was gemacht werden soll. Wie es gemacht werden soll ist aber noch unbekannt.

Die Implementation erfolgt normalerweise im .cpp File (hier led.cpp). Jede Funktion wird dabei ausprogrammiert.

Vor dem Funktionsnamen steht jeweils der Name der Klasse, gefolgt von zwei Doppelpunkten.

```
void Led::init(int ledPin) {
  pin = ledPin;
  pinMode(pin,OUTPUT);
  aus();
}
void Led::ein() {
  digitalWrite(pin,HIGH);
}
void Led::aus() {
  digitalWrite(pin,LOW);
}
void Led::blink() {
  digitalWrite(pin,!digitalRead(pin));
}
```



Verwendung von Klassen

Die Klasse ist ja nur ein Bauplan. Bevor die Led verwendet werden kann, muss sie also noch gebaut werden. Man bezeichnet das auch als Erstellen einer Instanz. Diese Instanz ist dann das Objekt, mit dem gearbeitet werden kann.

Aus einer Klasse können beliebig viele Objekte gebaut werden.

```
Led led1; // das Objekt led1 wird gebaut
Led led2; // das Objekt led2 wird gebaut

led1.init(5); // Led 1 ist an Pin 5
led2.init(6); // Led 2 ist an Pin 6

led1.on();    // Led 1 einschalten
led1.off();   // Led 1 ausschalten
led2.blink(); // Led 2 blinken
```

Was haben wir vereinfacht?

Eine ganze Menge! Wir sind noch weit von objektorientierter Programmierung entfernt. Stichworte wie Konstruktoren, Destruktoren, Vererbung und viele Andere wurden gar noch nicht erwähnt. Das wäre dann ein Thema für einen Kurs für Fortgeschrittene.

Wir verwenden nur die Kapselung, die es uns erlaubt das Konzept der Module ohne die dort auftretenden Probleme zu realisieren. Die neue Möglichkeit, aus einer Klasse mehrere Instanzen zu erzeugen, ist etwas, was wir gerne annehmen.