

## Objektorientierte Programmierung (OOP)

### Typ / Klasse / Objekte:

Was ist ein Typ?

Ein Typ ist eine Kategorie.

Den Menschen zeichnet unter anderem die Fähigkeit aus, zu kategorisieren.

Wir klassifizieren, ordnen, gruppieren und teilen ein.

Kurz gesagt: wir denken in *Typen* von Dingen.

Eine Orange ist eine Frucht, eine Frucht ist eine Pflanze, und eine Pflanze ist ein Lebewesen welches auf unserer Erde wächst.

Programme schreibt man in der Regel, um praxisrelevante Probleme zu lösen, wie etwa die Simulation der Abläufe eines Heizungssystems.

Diese Simulation lässt sich leichter umsetzen, wenn man Typen für die Darstellung von Räumen, Wärmesensoren, Thermostaten, Heizleitungen und Heizung erzeugen kann.

Einen neuen Typ erzeugt man durch die Deklaration einer *Klasse*.

Eine Klasse ist einfach eine Sammlung von Variablen kombiniert mit einer Gruppe von Funktionen.

### 47. Übung

Analysieren Sie bitte das nachfolgende Programm und versuchen Sie die einzelnen Abschnitte schriftlich zu beschreiben.

Arbeiten Sie dabei mit Ihren Mitschülern und/oder Ihrem Lehrer zusammen.

Was wird bei dem Programm ausgegeben und warum?

```
// Zeigt die Deklaration einer Klasse und
// die Definition von Klassenmethoden

#include <iostream.h>           // für cout

class Cat                      // Beginn der Klassendeklaration
{
    public:                    // Beginn d. öffentl. Abschnitts
    int GetAge();              // Zugriffsfunktion
    void SetAge (int age);     // Zugriffsfunktion
    void Meow();               // Allgemeine Funktion
    private:                   // Beginn des privaten Abschnitts
    int itsAge;                // Elementvariable
};
```

```
// Die öffentliche Zugriffsfunktion GetAge gibt
// den Wert des Datenelements itsAge zurück.

int Cat::GetAge()
{
    return itsAge;
}

// Definition von SetAge als öffentliche
// Zugriffsfunktion.
// Gibt Datenelement itsAge zurück.

void Cat::SetAge(int age)
{
    // Elementvariable itsAge auf den als
    // Parameter age übergebenen Wert setzen.
    itsAge = age;
}

// Definition der Methode Meow
// Rückgabe: void
// Parameter: Keine
// Aktion: Gibt "meow" auf dem Bildschirm aus

void Cat::Meow()
{
    cout << "Miau.\n";
}

// Eine Katze erzeugen(deklarieren), Alter festlegen, Miauen lassen
// Alter mitteilen lassen, dann erneut miauen lassen.

int main()
{
    Cat Frisky;           // Cat - Objekt deklarieren
    Frisky.SetAge(5);      // Cat - Objekt einen Wert zuweisen
    Frisky.Meow();         // Cat - Funktion aufrufen
    cout << "Frisky ist eine Katze, die " ;
    cout << Frisky.GetAge() << " Jahre alt ist.\n";
    Frisky.Meow();
    getchar();
    return 0;
}
```

## 48. Übung

Versuchen Sie eine der bisherigen Übungen mithilfe einer Klasse zu programmieren. Berichten Sie bitte über ihre Erfahrungen, Schwierigkeiten und Lösungsansätze.