Kapitel 18

Konstruktoren einer Klasse

Konstruktoren

- Zur Initialisierung der Eigenschaften einer Klasse werden Konstruktoren verwendet.
- Der Konstruktor darf nicht *private* sein.
- Der Konstruktor wird bei der Definition eines Objekts der Klasse automatisch aufgerufen.
- Der Konstruktor besitzt den gleichen Namen wie die Klasse.

```
methods (Access = public)
    function obj = CKonto(name, nummer, stand)
    ...
end
```

Konstruktoren

- Der Konstruktor kann mit einer unterschiedlichen Anzahl an Parametern aufgerufen werden.
- Das bedeutet, dass bei der Objektdefinition eine unterschiedliche Anzahl an Parametern übergeben werden können.

```
konto = CKonto;
konto = CKonto();
konto = CKonto("Tom Schultz");
konto = CKonto("Hugo Maier", 123456, 920.95);
```

Konstruktoren

- Im Konstruktor fragt man dann unter Verwendung von *nargin*, wieviele Parameter übermittelt wurden und reagiert entsprechend.
  (siehe nächste Folie.)

```
function obj = CKonto(name, nummer, stand)
    switch nargin
        case 1
            obj.m_name = name;
            obj.m_nummer = 0;
            obj.m_stand = 0.0;
        case 2
            obj.m_name = name;
            obj.m_nummer = nummer;
            obj.m_stand = 0.0;
        case 3
            obj.m_name = name;
            obj.m_nummer = nummer;
            obj.m_stand = stand;
        otherwise
            obj.m_name = "-----";
            obj.m_nummer = 0;
            obj.m_stand = 0.0;
    end
end
```

Standardkonstruktor

- Ist für eine Klasse kein Konstruktor definiert, so hat sie automatisch einen parameterlosen Standardkonstruktor.

```
methods (Access = public)
    function obj = CKonto()
    ...
end
```

- Auch der Aufruf des Standardkonstruktors erfolgt bei der Definition eines Objekts automatisch:

```
konto = CKonto();
```

Beispiel der Klasse CKonto mit Konstruktor

- Ein komplettes Beispiel der Klasse CKonto mit Konstruktor nebst Hauptprogramm finde Sie unter Beispiel 19 (auf der Webseite zu finden).