

Klassen, Objekte und ihre Speicherung

Gegeben seien zwei Klassen durch:

```
unit mOberklasse;.....interface
USES SysUtils;
TYPE
  TOberklasse = class(TObject)
    private
      x: INTEGER;
      A: ARRAY[1..1000] of INTEGER;
    public
      y: INTEGER;
      constructor create;
      function schreibXY: STRING;
    end;

implementation
  constructor TOberklasse.create;
  begin
    x := 17;
    y := 18
  end;

  function TOberklasse.schreibXY: STRING;
  begin
    RESULT := ('x='+IntToStr(x)+' y = '+ IntToStr(y))
  end;

end.
```

```
unit mUnterklasse;
interface.....
USES mOberklasse;
Type
  TUnterklasse = class(TOberklasse)
    private
      z: INTEGER;
    public
      constructor create;
  end;

implementation
constructor TUnterklasse.create;
begin
  //inherited create;
  y := 50
end;
end.
```

Desweiteren sei ein Hauptprogramm gegeben durch:

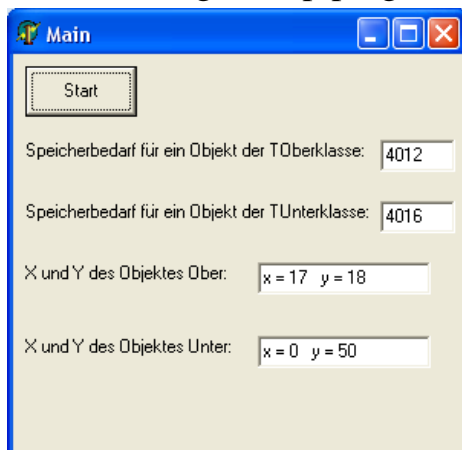
```
unit mHaupt;
interface
uses ....., mOberklasse, mUnterklasse;
type
  TMain = class(TForm)
    .....
  end;

var
  Main: TMain;
  Ober: TOberklasse;
  Unter: TUnterklasse;

implementation

procedure TMain.BtStartClick(Sender: TObject);
begin
  Ober := TOberklasse.create;
  EdOber.Text := IntToStr(TOberklasse.InstanceSize);
  Unter := TUnterklasse.create;
  EdUnter.Text := IntToStr(TUnterklasse.InstanceSize);
  EdXYOber.Text := Ober.schreibXY;
  EdXYUnter.Text := Unter.schreibXY;
  // Showmessage(IntToStr(Unter.X)) führt zu einer Fehlermeldung, weil die
  // Variable X in der Unterklasse nicht existiert.
end;
end.
```

Wird das obige Hauptprogramm ausgeführt, so erhält man folgendes Resultat:



Erklärung:

Zur Speicherung einer INTEGER-Variablen benutzt Delphi 6 vier Byte. Für das Array werden also 4000 Byte benötigt; 8 weitere Byte für die beiden Variablen x und y. Die Instanz (=Objekt) **Ober** zeigt also auf einen Speicherbereich von 4008 Byte. Alle Variablen, die eine Instanz darstellen, sind sog. Zeiger auf den entsprechenden Speicherbereich. Dieser Zeiger wird selbst durch eine 4 Byte – Zahl realisiert (diese Zahl entspricht der Nummer der Speicherzelle, an der der Speicherbereich für diese Variable beginnt). Insgesamt benötigt das Objekt **Ober** also 4012 Byte.

Jedes Objekt der Klasse **TOberklasse** besitzt Attribute (x und y), deren Werte eben für dieses Objekt charakteristisch sind, d.h. andere Objekte dieser Klasse haben zwar dieselben Attribute, aber üblicherweise andere Werte für diese Attribute.

Alle Objekte (=Instanzen) dieser Klasse **TOberklasse** besitzen jedoch dieselben Methoden (Hier nämlich den *constructor create* und die *function schreibXY: STRING*).

Es wäre unsinnig, für jede Instanz auch den Programmcode der Methoden extra zu speichern, weil schließlich alle Instanzen denselben Methoden-Programmcode besitzen. Deshalb werden die Programmcodes aller Methoden einer Klasse nur ein einziges Mal (an einer anderen Speicherstelle) gespeichert.

Es kann jedoch durchaus sein, dass diese Methoden auch auf die Attributwerte der jeweiligen Instanz zugreifen müssen. Beispiel: *Ober1.schreibXY* liefert normalerweise einen anderen Wert als *Ober2.schreibXY*. Um diesen Unterschied realisieren zu können, wird beim Aufruf derartiger Methoden (die auf Attributwerte der Instanzen zugreifen müssen) noch ein zusätzlicher Parameter mit übergeben, welcher den Speicherort der Instanzvariablen angibt. Aber dies organisiert das Delphi 6 – Programmiersystem von alleine. Der Programmierer braucht sich darum nicht zu kümmern.

Aus dem Resultat des obigen Programmes lassen sich 3 wesentliche Dinge erkennen:

1. Als *private* deklarierte Attribute (und auch Methoden) sind in der Unterklasse nicht vorhanden. Sie werden nicht vererbt, d.h. die Unterklasse besitzt diese Attribute (und Methoden) nicht!
Für obiges Beispiel bedeutet dies: die Unterklasse besitzt kein Array!
2. Trotzdem wird (in Delphi 6 !) für ein Objekt der Unterklasse mindestens derselbe Speicherplatz benötigt wie für ein Objekt der Oberklasse. Zusätzlich wird noch Speicherplatz benötigt für zusätzliche Variablen, welche die Oberklasse noch nicht besitzt. Das heißt: Eine Instanzvariable der Unterklasse besitzt (in Delphi 6) mindestens genauso viel Speicherplatz wie eine Instanz der Oberklasse, auch wenn sie diesen Platz überhaupt nicht benötigen sollte.
3. Wenn ein Objekt erzeugt werden soll, wird (falls die Klasse keinen eigenen Konstruktor besitzt) zunächst der Konstruktor der zugehörigen Oberklasse aufgerufen.

Aufgaben

1. Was ändert sich, wenn man die Bemerkungsklammern in dem Konstruktor der Unterklasse entfernt? Erklärung!
2. Was ändert sich, wenn man in der Oberklasse den Konstruktor als *virtual* und in der Unterklasse als *override* deklariert?