

Das Billardprojekt : Wir entwerfen und codieren Klassen selbst.

Lösungen und wichtige Hinweise

Die Aufgabenstellung war:

Schritt 1: Eine Kugel (kleiner Kreis), soll sich langsam von links nach rechts über den Bildschirm bewegen können.

Entwirf eine Klasse TKugel, mit der dies realisiert werden kann.

Die Projektdatei könnte folgendermaßen aussehen:

```
program Billard;

uses
  mSum,
  mUhr,
  mTKugel in 'mTKugel.pas';

var
  derBildschirm : Bildschirm;
  dieMaus       : Maus;
  meinStift     : Stift;
  meineUhr      : Uhr;
  // xPos, yPos : Zahl; wird nicht mehr benötigt,
  // die Kugel hat ein eigenes Gedächtnis
  ersteKugel    : TKugel;

begin
  // Initialisierung : Objekte Erzeugen
  derBildschirm := Bildschirm.init;
  dieMaus       := Maus.init;
  meinStift     := Stift.init;
  meineUhr      := Uhr.init;
  ersteKugel    := TKugel.init;

  // Kugel auf Anfangsposition setzen
  ersteKugel.setzePosition (50,200);

  // Kugel über den Bildschirm laufen lassen
  repeat
    ersteKugel.bewege;
    meineUhr.warte(1)
  until dieMaus.istGedrueckt;

  // Aufräumen
  ersteKugel.GibFrei;
  meineUhr.GibFrei;
  meinStift.GibFrei;
  dieMaus.gibFrei;
  derBildschirm.gibFrei
end.
```

Die Kugelklasse und die zugehörige Unit:

TKugel
- meinStift: Stift - AktuelleXPos: Zahl - AktuelleYPos: Zahl
+ ! init () + ! bewege () + ! setzePosition (pxPos: Zahl; pYPos: Zahl) + ! gibFrei ()

```
UNIT mTKugel;
```

```
interface
```

```
uses mSum;
```

```
type
```

```
TKugel = CLASS
```

```
  // Attribute
```

```
  private
```

```
    meinStift : Stift;    // Gehilfe  
    AktuelleXPos : Zahl; // Gedächtniszelle  
    AktuelleYPos : Zahl; // Gedächtniszelle
```

```
  // Methoden
```

```
  public
```

```
    constructor init; virtual;  
    procedure bewege; virtual;  
    procedure setzePosition (pxPos: Zahl; pYPos: Zahl); virtual;  
    destructor gibFrei; virtual;
```

```
end;
```

```
implementation
```

```
constructor TKugel.init;
```

```
begin
```

```
  meinStift := Stift.init;
```

```
end;
```

```
procedure TKugel.bewege;
```

```
begin
```

```
  // Kugel an alter Position löschen
```

```
  meinStift.radiere;
```

```
  meinStift.bewegeBis (AktuelleXPos,AktuelleYPos);
```

```
  meinStift.zeichneKreis (10);
```

```
  // Kugel an neue Position bringen
```

```
  AktuelleXPos := AktuelleXPos + 0.5;
```

```
  // und dort zeichnen
```

```
  meinStift.normal;
```

```
  meinStift.bewegeBis (AktuelleXPos,AktuelleYPos);
```

```
  meinStift.zeichneKreis (10);
```

```
end;
```

```
procedure TKugel.setzePosition (pxPos: Zahl; pYPos: Zahl);  
begin  
  //ggf Kugel an alter Stelle Löschen  
  meinStift.radiere;  
  meinStift.bewegeBis (AktuelleXPos,AktuelleYPos);  
  meinStift.zeichneKreis (10);  
  
  //Neue Position im Gedächtnis merken  
  AktuelleXPos := pxPos;  
  AktuelleYPos := pYPos;  
  
  // Kugel an neuer Stelle zeichnen  
  meinStift.normal;  
  meinStift.bewegeBis (AktuelleXPos,AktuelleYPos);  
  meinStift.zeichneKreis (10);  
end;  
  
destructor TKugel.gibFrei;  
begin  
  meinStift.gibFrei;  
end;  
  
end.
```

Das Zusammenspiel zwischen dem Kugelgedächtnis und der Projektdatei

soll nun verdeutlicht werden. Wir analysieren dazu die Zeilen

```
(0) VAR ersteKugel : TKugel;  
.....  
(1) ersteKugel := TKugel.init;  
(2) ersteKugel.setzePosition (50,200);
```

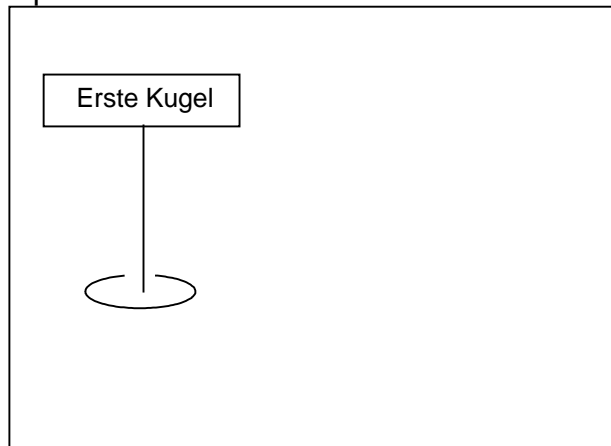
und ihre Wirkungen in der Kugelunit und im Speicher des Rechners:

(0) Die Zeile

```
Var ersteKugel : TKugel
```

haben wir als Anlegen eines Namensschildes im Speicher des Rechners interpretiert:

Speicher des Rechners

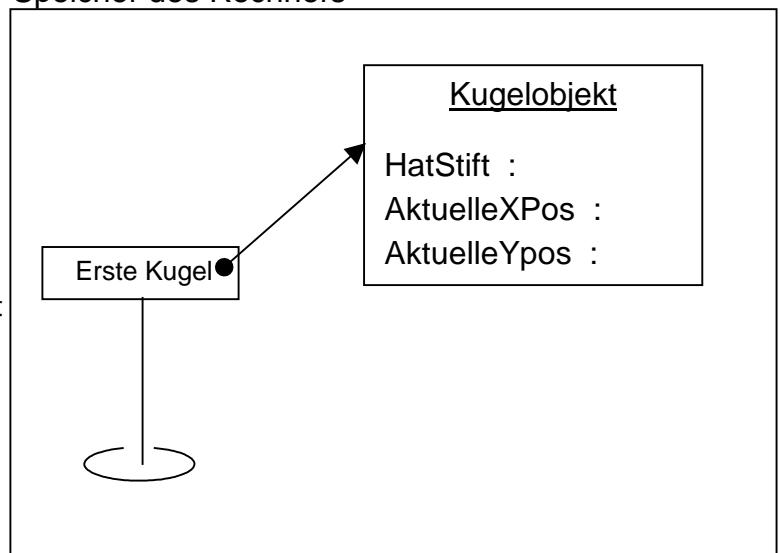


(1) Die Zeile

```
ersteKugel := TKugel.init;
```

hat zuerst einmal zur Folge, dass im Speicher des Rechners ein Kugelobjekt angelegt wird und das Namensschild auf dieses Kugelobjekt zeigt (verweist):

Speicher des Rechners



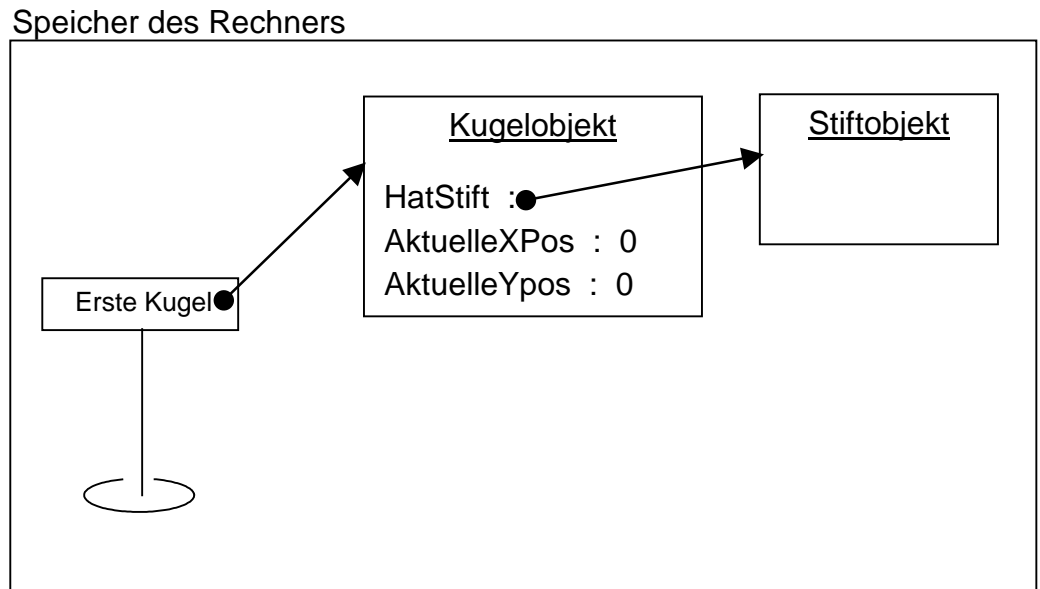
Nun werden die Zeilen der Init-Methode der Kugelklasse ausgeführt:

```

constructor TKugel.init;
begin
  meinStift := Stift.init;
  AktuellexPos := 0;
  AktuelleyPos := 0;
end;

```

Es wird zuerst ein Stiftobjekt erzeugt, auf welches das Attribut hatStift der Kugel verweist. Danach werden die Gedächtniszellen der Kugel benutzt. Die Kugel merkt sich ihre erste Aktuelle Position:



Das Ergebnis ist hierüber zu sehen

In der Projektdatei folgt nun die Zeile:

```
(2) ersteKugel.setzePosition (50,200);
```

Die Kugel wird beauftragt ihre Position neu festzulegen und tut dies, indem sie folgenden Code ausführt:

```

(2.1) procedure TKugel.setzePosition (pXPos: Zahl; pYPos: Zahl);
(2.2) begin
      //Kugel an alter Stelle löschen
(2.3)   meinStift.radiere;
(2.4)   meinStift.bewegeBis(AktuelleXPos,AktuelleYPos);
(2.5)   meinStift.zeichneKreis(10);

      //Gedächtnis benutzen !! : Position merken
(2.6)   AktuelleXPos := pXPos;
(2.7)   AktuelleYPos := pYPos;

      //neu Zeichnen an neuer Position
(2.8)   meinStift.bewegeBis(AktuelleXPos,AktuelleYPos);
(2.9)   meinStift.zeichneKreis(10);
(2.10) end;

```

Für die Lebensdauer der Prozedur hat pXpos den Wert 50 und pYPos den Wert 200.

Beim Beenden der Prozedur werden die Speicherstellen pXPos und pYPos gelöscht!
Die entsprechenden Werte sind nicht mehr vorhanden.
Aber im Gedächtnis der Kugel bleiben sie in **AktuelleXPos**, **AktuelleYPos** gespeichert.