

Speicherung der Referenz in einer Zettelliste

Um im Zettellistenprogramm Zugriff auf früher erzeugte Zettel zu haben, müssen die Referenzen auf die Speicheradressen in einer Liste gespeichert werden. Dazu greifen wir auf die von Delphi bereits vorgegebene Klasse `TList` zurück:

Im Hauptprogramm `TMain` wird dazu eine Zettelliste als privates Attribut deklariert:

```
private
  Zettelliste: TList;
```

Zu Start des Programms wird der Konstruktor aufgerufen, um die Zettelliste zu erzeugen:

```
Zettelliste := TList.create();
```

Als Attribut besitzt `TList` unter anderem das Attribut `count`, das die Anzahl der aktuellen Einträge angibt.

Wichtige Methoden von `TList` sind:

- `add (object: TObject)` z.B.: `Zettelliste.add(Zettel);`
- `delete (index: integer)` z.B.: `Zettelliste.delete(3);`
Dies löscht Eintrag Nummer 3 aus der Zettelliste.
- Zuordnung: z.B. `Zettel := Zettelliste[2];`
Dies lädt Zettel Nummer 2 in den Speicher.

Insgesamt erhalten wir folgenden Quelltext:

```
procedure ...
begin
  if Zettelliste=nil then        {wenn es noch keine Zettelliste gibt}
    Zettelliste:=liste.create();
  ...
  Zettelliste.add(Zettel);        {Zettel wird hinzugefügt}
  Dialog.destroy();
  index:=Zettelliste.count-1;    {wird von 0 an nummeriert}
  ...
end;
```

```
procedure ...                    {Löschen eines Zettels}
begin
  if Zettel <>nil then
  begin
    Zettel.destroy();
    Zettelliste.delete(Index);    {Eintrag in der Zettelliste wird
                                   gelöscht}
```

```
if Zettel.count>0 then
begin
  Index:=Index-1;           {Index wird um eins heruntermgesetzt}
  Zettel:=Zettelliste[Index] {Zettelzeiger zeigt auf den vorher
                             vorletzten Zettel}
end
else
  lbleer.caption:='kein Zettel vorhanden'
end;
```

Problem: Man kann nur von dem letzten bis zum ersten Zettel löschen.