

Stundenprotokoll vom 25.02, 04.03, 11.03.2014

In der Informatikstunde am Dienstag den 25.02.2014 ging es um die Erstellung eines Stapels (Stack). Auf diesen Stapel soll man Objekte drauflegen können (push) und nur das oberste Element sollte heruntergenommen werden können (pop). Dieses Prinzip wird „Lifo“-Prinzip genannt (Last in First out). Außerdem sollte die Klasse über eine Funktion verfügen, die das oberste Element als Rückgabewert liefert. Sinnvoll könnte diese Klasse zum Beispiel für ein Kartenspiel, einen Güterbahnhof oder zum Spiegeln von Wörtern sein. Es gilt folgende Dokumentation (Quelle Kursseite):

Dokumentation der Klasse TStack

TStack verwaltet Objekte in einer linearen Liste (*Stapel*) nach dem LIFO-Prinzip (Last In First Out).

Oberklasse: TList

Konstruktor create geerbt von TList

Anfrage isEmpty: Boolean geerbt von TList

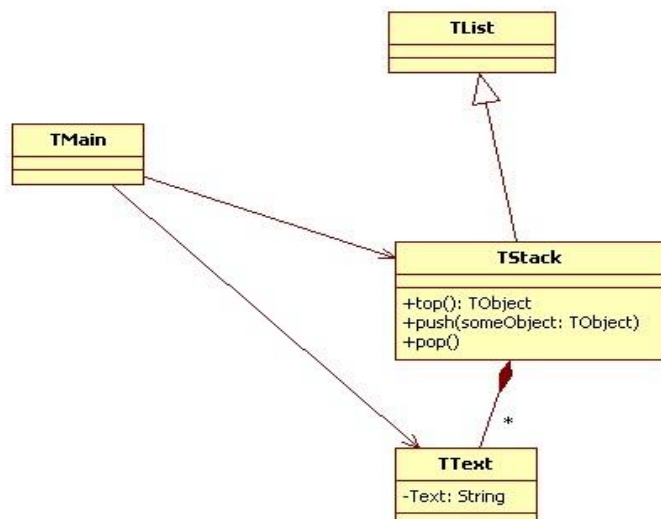
Auftrag push (pObject: TObject)
Vorher: Der Stapel ist erzeugt.
Nachher: pObject liegt oben auf dem Stapel.

Auftrag pop
Vorher: Der Stapel ist nicht leer.
Nachher: Das zuletzt eingefügte Element ist vom Stapel entfernt.

Anfrage top: TObject
Vorher: Der Stapel ist nicht leer.
Nachher: Die Anfrage liefert das oberste Stapelement. Der Stapel ist unverändert.

Destruktor destroy geerbt von TList

Nun sollte zuerst ein Klassendiagramm erstellt werden. Auf folgende Version hatten wir uns dann im Unterricht geeinigt (Quelle: Kursseite):



Der nächste Arbeitsschritt war es nun diese Klasse dann in Delphi umzusetzen. Die *push*-Prozedur musste zuerst mit einer *if*-Abfrage sicherstellen, dass der Stapel nicht leer ist. Danach wird zum Anfang des Stapels gegangen, und mit der Prozedur *insertBefore()*, der Oberklasse das Element eingefügt. Für die *pop*-Prozedur galt dasselbe, bloß wurde hier statt des Einfügens die *delete()*-Prozedur verwendet, um das oberste Element aus dem Stapel zu entfernen. Für die *top*-Funktion muss man zum Anfang der Liste gehen und dann das Ergebnis der Funktion gleich *getItem()*(Funktion der Oberklasse) setzen.

Nun sollten wir die Klasse nutzen um einen „Wortspiegel“ zu entwickeln. Das zu spiegelnde Wort wird über ein Editfenster als String eingelesen. Mit der *length*-Methode, die Delphi standardmäßig zur Verfügung stellt, kann die Länge des Strings bestimmt werden und in eine Variable eingetragen werde. Diese wird später wichtig für eine *for*-Schleife. Mit dieser Schleife nehmen wir jeden Buchstaben einzeln von hinten aus dem String und fügen ihn in einen neuen ein. Die Schleife hat so viele Durchläufe wie der *string* lang ist.

Die Quelltexte finden sich auf der Kursseite.

In der Stunde von 04.03.2014 ging es darum eine Liste numerisch oder alphabetisch zu sortieren. Zunächst wollen wir uns dabei jedoch auf das numerische Verfahren beschränken. Gegeben wurde uns dafür die Wrapper-Klasse TInteger, die Zahlen in Objekte einpackt. Nun muss nur noch ein Klasse erstellt werden, die die Zahlen sortiert. Dafür überlegten wir uns folgenden Grobalgorithmus:

Falls Liste leer **dann**
 ergänze Objekt

sonst

 gehe an den Anfang der Liste

 lese aktuelle Zahl aus der Liste

solange aktuelle Zahl kleiner ist als einzufügende Zahl

 gehe ein Element weiter

 lies aktuelle Listenzahl ein