

PROJEKTARBEIT ZUM MORSEALPHABET

Im Morsealphabet ist jeder Buchstabe durch eine Aneinanderreihung von Punkten (·) und Strichen (-) codiert. Die Codierung der Buchstaben lautet wie folgt:

Morsealphabet

A	·-	B	-···	C	-··-	D	-···	E	·	F	···-	G	··-	H	····	I	··	J	··-·	K	-·-	L	····	M	--
N	--	O	---	P	··-·	Q	··-·	R	··-	S	···	T	-	U	··-	V	····	W	··-	X	··-·	Y	-··-	Z	····

Eine Möglichkeit, den Morse-Code zu verwalten, ist die Speicherung der Codes in einem Binärbaum. Der Morsepunkt bewirkt dabei eine Verzweigung nach links, der Morsestrich eine Verzweigung nach rechts. In jedem Knoten des Baumes steht der durch den im Baum vorausgegangenen Punkt-Strich-Code repräsentierte Buchstabe.

Für die Umsetzung dieser Speichermöglichkeit in DELPHI bietet sich das folgende **interface** an:

```
uses BinBaum;
const CodeTabelle: array['A'..'Z'] of string[4]=
    ('·-', '····', '··-·', '····', '·', '···-', '··-', '····', '··', '··-·', '·-', '····', '····',
     '····', '····', '··-·', '··-·', '··-·', '··-·', '··-·', '··-·', '··-·', '··-·',
     '··-·', '··-·', '··-·', '··-·', '··-·', '··-·', '··-·');
type TMorsebaum = class(TBinBaum)
public
    Inhalt: char;
    constructor Create(c: char);
    function WurzelinhaltToString: string; override;
    function Einlesen: TMorsebaum;
    function MorsetextToKlartext (Morse: string): char;
    function KlartextToMorsetext (Klar: char): string;
end;
```

- Aufgabe:** Es ist ein Programm zu entwickeln, welches
- den Binärbaum zum Morse-Code aufbaut (**function** Einlesen),
 - den Binärbaum auf dem Bildschirm ausgibt,
 - einen beliebigen Klartext mit Hilfe der Konstanten (**const** CodeTabelle) in einen Morsetext umwandelt und
 - einen beliebigen Morsetext mit Hilfe des Binärbaums in einen Klartext umwandelt.

Als Anregung für Ihr Layout sollte der unten abgebildete Screenshot dienen.

