

UNTERRICHTSREIHE CODIERVERFAHREN

1. Stunde: Einführung in die Cäsar-Methode

Mit Hilfe des Arbeitsblattes (01 Arbeitsblatt Einführung) werden die Schüler an die Thematik Codierverfahren herangeführt.

Das Verfahren wird umgangssprachlich beschrieben.

2. + 3. Stunde: Implementierung eines kleinen Übersetzungsprogramms

Die Schüler entwickeln ein Delphi-Programm, mit Hilfe dessen man einen Text in ein Editfeld eingeben kann, welcher anschließend in einem anderen Editfeld codiert ausgegeben wird. Das Layout könnte wie folgt aussehen:



Dazu werden die folgenden Funktionen entwickelt:

```
function Caesar_Codieren_c(c, key: char): char;
var hilf_c: char;
    hilf_i: integer;
begin
    hilf_c:= UpCase(c);
    if hilf_c in ['A'..'Z']
    then begin
        hilf_i:= Ord(hilf_c);
        hilf_i:= hilf_i + (ord(key) - ord('A'));
        if hilf_i > ord('Z')
        then hilf_i:= hilf_i - 26;
        hilf_c:= Chr(hilf_i);
    end;
    Caesar_Codieren_c:= hilf_c;
end;
```

```
function Caesar_Decodieren_c(c, key: char): char;
var hilf_c: char;
    hilf_i: integer;
begin
    hilf_c:= UpCase(c);
    if hilf_c in ['A'..'Z']
    then begin
        hilf_i:= Ord(hilf_c);
        hilf_i:= hilf_i - (ord(key) - ord('A'));
        if hilf_i < ord('A')
        then hilf_i:= hilf_i + 26;
        hilf_c:= Chr(hilf_i);
    end;
    Caesar_Decodieren_c:= hilf_c;
end;
```

```

function Caesar_Codieren_str(s: string; key: char): string;
var i: integer;
    hilf_str: string;
begin
    hilf_str:= '';
    for i:= 1 to length(s)
        do hilf_str:= hilf_str + Caesar_Codieren_c(s[i],key);
    Caesar_Codieren_str:= hilf_str;
end;

function Caesar_Decodieren_str(s: string; key: char): string;
var i: integer;
    hilf_str: string;
begin
    hilf_str:= '';
    for i:= 1 to length(s)
        do hilf_str:= hilf_str + Caesar_Decodieren_c(s[i],key);
    Caesar_Decodieren_str:= hilf_str;
end;

```

Der Aufruf der Funktionen kann mit Hilfe der folgenden Button-Prozedur geschehen:

```

procedure TF_Codierer.B_CodierenClick(Sender: TObject);
var Input: string;
begin
    Input:= 'E';
    if InputQuery('Cäsar-Verschiebung', 'Cäsar Schlüssel: A -> ?', Input)
        then E_Ausgabe.Text:= Caesar_Codieren_str(E_Eingabe.Text,Input[1]);
end;

```

Hinweis: Die Schüler werden wahrscheinlich mit dem Projekt nicht fertig, müssen aber in der Lage sein, dies zur Hausaufgabe fertig zu stellen, da ähnliche Routinen bereits in der 11.1 implementiert wurden.

4. Stunde: Vorstellung der Open- und Save-Dialoge

Die Verwendung des Open- und Savedialogs von Delphi werden vorgestellt. Die Schüler erweitern Ihr Projekt um diese Dialoge. Das Layout hat dann folgendes Design:



Durch die Dialoge werden lediglich die Dateinamen spezifiziert und in den Editfeldern ausgegeben. Eine Bearbeitung der Dateien wird erst in der Folgestunde durchgeführt. Der Aufruf des Open- bzw. Savedialogs kann mit Hilfe folgender Button-Prozedur durchgeführt werden:

```

procedure TF_Codierer.Button1Click(Sender: TObject);
begin
    if Opendialog1.Execute
        then E_Eingabe.Text:= ExtractFilename(Opendialog1.FileName);
end;

```

5. + 6. Stunde: Verwendung von Textfiles.

Das Prinzip von Textdateien wird vorgestellt. Beispiel:

I	N	F	O	R	M	A	T	I	K	#32	I	S	T	#32	T	O	L	L	EOF
↑										↑									↑
Filemarke am Anfang										Filemarke mitten drin					Filemarke am Ende				

Die folgenden Befehle zur Verwaltung von Textdateien werden vorgestellt und am obigen Beispiel erläutert.

<code>var Dateivariabile: TEXTFILE;</code>	Eine neue Dateivariabile wird erzeugt. Mit Hilfe dieser Variablen ist ein Zugriff auf Dateien möglich.
<code>ASSIGNFILE(Dateivariabile, Dateiname)</code>	Die Dateivariabile wird mit einer konkreten Textdatei verknüpft. Ab jetzt ist ein Zugriff auf die Dateivariabile auch ein Zugriff auf die Datei auf der Festplatte.
<code>RESET(Dateivariabile)</code> <code>REWRITE(Dateivariabile)</code>	Die Datei wird geöffnet. Bei RESET wird die Datei zum bearbeiten geöffnet, d. h. die Datei muss schon vorher auf der Festplatte vorhanden sein. Bei REWRITE wird eine neue Datei erstellt, d. h. die Datei muss zuvor nicht auf der Festplatte existieren - ansonsten wird sie überschrieben. Beide Befehle positionieren die Filemarke auf die erste Position der Datei.
<code>READ(Dateivariabile, Zeichen)</code> <code>WRITE(Dateivariabile, Zeichen)</code>	Lesen bzw. Schreiben von/auf der Textdatei. Beide Befehle bewegen die sogenannte Filemarke eine Position nach rechts.
<code>EOF(Dateivariabile)</code>	Stellt fest, ob die Datei zu Ende ist. Sobald die Filemarke das letzte Zeichen gelesen hat, ergibt der Funktionsaufruf EOF(...) den Wert true.
<code>CLOSEFILE(Dateivariabile)</code>	Schließt die mit RESET bzw. REWRITE geöffnete Datei. Das Schließen einer Datei ist wichtig, da sonst die Datei nicht mehr ansprechbar ist.

Die Implementierung von Prozeduren, welche Textdateien codieren bzw. Decodieren werden besprochen. Eine Implementierung könnte wie folgt aussehen:

```

procedure Caesar_Codieren_file(eingabe, ausgabe: string; key: char);
var Klar_Datei, Code_Datei: Textfile;
    Zeichen: char;
begin
    assignfile(Klar_Datei, eingabe);
    assignfile(Code_Datei, ausgabe);
    Reset(Klar_Datei);
    Rewrite(Code_Datei);
    while not eof(Klar_Datei)
    do begin
        Read(Klar_Datei, Zeichen);
        Write(Code_Datei, Caesar_Codieren_c(Zeichen, key));
    end;
end;

```

```

        end;
        Closefile(Klar_Datei);
        CloseFile(Code_Datei);
end;

procedure Caesar_Decodieren_file(eingabe, ausgabe: string; key: char);
var Klar_Datei, Code_Datei: Textfile;
    Zeichen: char;
begin
    assignfile(Klar_Datei, eingabe);
    assignfile(Code_Datei, ausgabe);
    Rewrite(Klar_Datei);
    Reset(Code_Datei);
    while not eof(Code_Datei)
    do begin
        Read(Code_Datei, Zeichen);
        Write(Klar_Datei, Caesar_Decodieren_c(Zeichen, key));
    end;
    Closefile(Klar_Datei);
    CloseFile(Code_Datei);
end;

```

Der Aufruf der Prozeduren würde mit folgender Button-Prozedur geschehen:

```

procedure TF_Codierer.B_CodierenClick(Sender: TObject);
var Input: string;
begin
    if (E_Ausgabe.Text<>'') and (E_Eingabe.Text<>'')
    then begin
        Input:= 'E';
        if InputQuery('Cäsar-Verschiebung', 'Cäsar Schlüssel: A -> ?', Input)
        then Caesar_Codieren_file(E_Eingabe.Text, E_Ausgabe.Text, Input[1]);
    end;
end;

```

Hinweis: Wichtig ist, dass in den Editfelder Dateinamen stehen, da sonst die ASSIGNFILE-Routine eine Fehlermeldung produziert.

7. Stunde: Implementierung der Routinen

Die Schüler implementieren die Routinen zur Codierung und Decodierung eines Textfiles mit der Cäsar-Methode.

8. + 9. Stunde: Erweiterung des Cäsar- auf das Schlüsselwort-Verfahren

Die Cäsar-Methode wird auf das Verfahren der Schlüsselwort-Methode verfeinert. Die Schüler erweitern das Projekt um diese Art der Codierung.

10. Stunde: Implementierung der Schlüsselwort-Methode

Fortführung der Implementierung.