

Quelltext zum Lazarus-Programm „Vigenère-Chiffre“

Für den eigentlichen Algorithmus nicht notwendigen Passagen wurden größtenteils entfernt (...). Einige weitere Passagen wurden gekürzt (→ case-Anweisungen: Zeichen c bis _ bzw. 3 bis 70; ebenfalls ...).

Teil 1

```
...
type
  { TForm1 }

TForm1 = class(TForm)
  ...
  procedure Einlesen;
  procedure Ausgeben;
  procedure StringZuArray;
  procedure ArrayZuString;
  procedure Encode;
  procedure Decode;
private
  { private declarations }
public
  { public declarations }
end;

var
  Form1: TForm1;
  eingabe, key, keylang, ausgabe: string;
  eingabezahl, keyzahl, ausgabezahl: array of integer;
  eingabelaenge, keylaenge, i, j: integer;
  zeichen: char;
  rechnen: boolean;

implementation
  {$R *.lfm}

  { TForm1 }

  procedure TForm1.Button1Click(Sender: TObject);
  begin
    //Verschluesseln
    label6.Caption:='';
    label7.Caption:='';
    edit3.Text:='';
    rechnen:=true;
    Einlesen;
    if (eingabe = '') or (key = '') then label6.Caption:='Eingabe unvollständig!' else
      begin
        StringZuArray;
        if rechnen then
          begin
            Encode;
            ArrayZuString;
            Ausgeben
          end;
        end;
      end;
  end;

  procedure TForm1.Button2Click(Sender: TObject);
  begin
    //Entschluesseln
    label6.Caption:='';
    label7.Caption:='';
    edit3.Text:='';
    rechnen:=true;
    Einlesen;
    if (eingabe = '') or (key = '') then label6.Caption:='Eingabe unvollständig!' else
      begin
        StringZuArray;
        if rechnen then
          begin
            Decode;
            ArrayZuString;
            Ausgeben
          end;
        end;
      end;
  end;

  procedure TForm1.Button3Click(Sender: TObject);
  begin
    //Programm beenden
    close
  end;

  ...

  procedure TForm1.Einlesen;
  begin
    //Einlesen
    eingabe:=edit1.Text;
    key:=edit2.Text;
  end;

  procedure TForm1.Ausgeben;
  begin
    //Ausgabe
    edit3.Text:=ausgabe;
  end;
end;
```

Quelltext zum Lazarus-Programm „Vigenère-Chiffre“

Für den eigentlichen Algorithmus nicht notwendigen Passagen wurden größtenteils entfernt (...). Einige weitere Passagen wurden gekürzt (→ case-Anweisungen: Zeichen c bis _ bzw. 3 bis 70; ebenfalls ...).

Teil 2: Fortsetzung

```
procedure TForm1.StringZuArray;
begin
  //String auf Array abbilden
  eingabelaenge:=length(eingabe);
  keylaenge:=length(key);
  SetLength(eingabezahl, eingabelaenge);
  SetLength(keylang, eingabelaenge);
  keylang:=key;
  //Schlüssellänge anpassen
  if keylaenge < eingabelaenge then
  begin
    j:=(eingabelaenge div keylaenge) +1;
    for i:=0 to j do
      keylang:=keylang+key
    end;
  SetLength(keyzahl, length(keylang));
  //Eingabestring auf Array abbilden
  for i:=1 to eingabelaenge do
  begin
    zeichen:=eingabe[i];
    case zeichen of
      'a': eingabezahl[i]:=1;
      'b': eingabezahl[i]:=2;
      ...
      '#': eingabezahl[i]:=71;
      '^': eingabezahl[i]:=72;
      ' ': eingabezahl[i]:=73;
    else
      begin
        rechnen:=false;
        label6.Caption:='Klar-/Schlüsseltext: nicht unterstütztes Zeichen!'
      end;
    end;
  end;
  //verlängerten Schlüsselstring auf Array abbilden
  for i:=1 to length(keylang) do
  begin
    zeichen:=keylang[i];
    case zeichen of
      'a': keyzahl[i]:=1;
      'b': keyzahl[i]:=2;
      ...
      '#': keyzahl[i]:=71;
      '^': keyzahl[i]:=72;
      ' ': keyzahl[i]:=73;
    else
      begin
        rechnen:=false;
        label7.Caption:='Schlüssel: nicht unterstütztes Zeichen!'
      end;
    end;
  end;
end;

procedure TForm1.ArrayZuString;
//Ausgabearray auf Ausgabestring abbilden
begin
  SetLength(ausgabe, length(ausgabebezahl));
  for i:=1 to length(ausgabebezahl) do
  begin
    j:=ausgabebezahl[i];
    case j of
      1: ausgabe[i]:='a';
      2: ausgabe[i]:='b';
      ...
      71: ausgabe[i]:='#';
      72: ausgabe[i]:='^';
      73: ausgabe[i]:=' ';
    end;
  end;
end;

procedure TForm1.Encode;
//Codieren
begin
  SetLength(ausgabebezahl, eingabelaenge);
  for i:=0 to eingabelaenge do
  begin
    ausgabebezahl[i]:=eingabezahl[i]+keyzahl[i];
    if ausgabebezahl[i] > 73 then ausgabebezahl[i]:=ausgabebezahl[i]-73
  end;
end;

procedure TForm1.Decode;
//Decodieren
begin
  SetLength(ausgabebezahl, eingabelaenge);
  for i:=0 to eingabelaenge do
  begin
    ausgabebezahl[i]:=eingabezahl[i]-keyzahl[i];
    if ausgabebezahl[i] < 1 then ausgabebezahl[i]:=ausgabebezahl[i]+73
  end;
end;
end.
```