Rekursion – Lösungen zu den Übungsaufgaben

Übungsaufgabe 1 und 2

Berechnung der Fakultät einer natürlichen Zahl rekursiv und iterativ

```
function fak_iterativ(n: integer): integer;
var
  i, fak: integer;
begin
  fak := 1;
  for i := 1 to n do begin
    fak := i * fak;
  end;
  fak_iterativ := fak;
end;

function fak_rekursiv(n: integer): integer;
begin
  if n = 1 then fak_rekursiv := 1
    else fak_rekursiv := n * fak_rekursiv(n - 1);
end;
```

Übungsaufgabe 3

Berechnen Sie die Summe der ersten n natürlichen Zahlen s(n) = 1 + 2 + 3 + ... + n rekursiv. Orientieren Sie sich dabei an dem rekursiven Ansatz der Fakultätsfunktion!

```
function summe(a: integer): integer;
begin
  if a = 0 then summe := 0
    else summe := a + summe(a - 1);
end;
```

Übungsaufgabe 4

Berechnen Sie das Summe der ersten n Quadratzahlen q_summe(n) = $1^2 + 2^2 + ... + n^2$ mit Hilfe einer rekursiven Funktion!

```
function quad_summe(a: integer): integer;
begin
  if a = 1 then quad_summe := 1
    else quad_summe := a * a + quad_summe(a - 1);
end;
```

Übungsaufgabe 5 c) und d)

c) Schreiben Sie ein Programm, das nach Eingabe von n den Bestand an Kaninchenpaaren ausgibt!

```
function fibonacci(a: integer): integer;
begin
  if a < 2 then fibonacci := 1
    else fibonacci := fibonacci(a - 1) + fibonacci(a - 2);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  n, fibo: integer;
begin
  n := StrToInt(Edit1.Text);
  fibo := fibonacci(n);
  Edit2.Text := IntToStr(fibo);
end;
```

d) Schreiben Sie ein Programm, das nach Eingabe einer Kaninchenzahl (in Pärchen) ausgibt, wann diese Zahl erreicht oder überschritten ist!

```
procedure TForm1.Button2Click(Sender: TObject);
var
  paerchen, fibo, zaehler: integer;
begin
  paerchen := StrToInt(Edit3.Text);
  zaehler := 0;
  repeat
    fibo := fibonacci(zaehler);
    inc(zaehler);
  until fibo > paerchen;
  Edit4.Text := IntToStr(zaehler-2);
end;
```