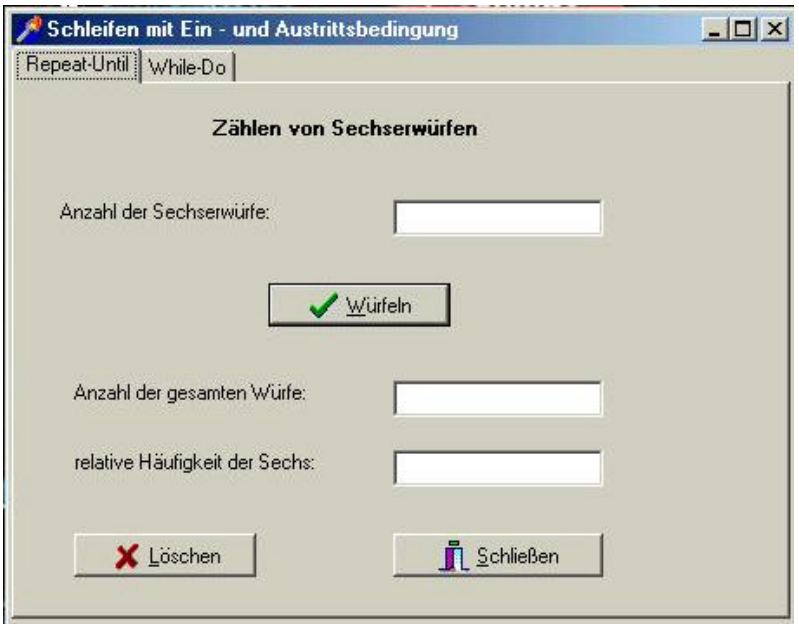
Aufgaben zu Wiederholungsanweisungen mit Ein – und Austrittsbedingung

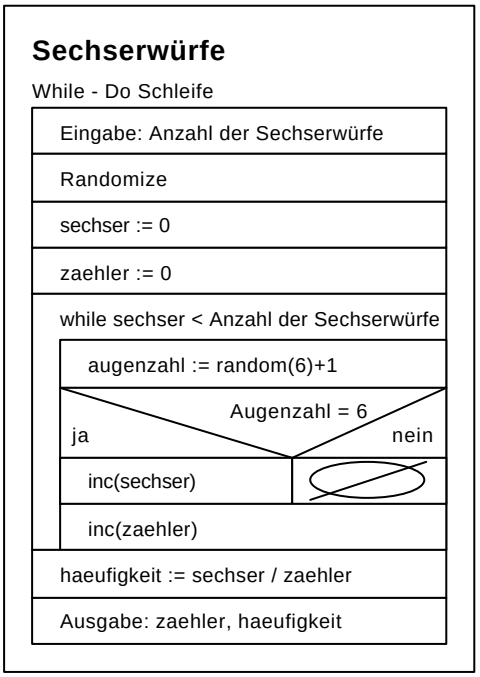
Beispiel:

Es soll solange mit einem Würfel gewürfelt werden, bis eine zuvor eingegebene Zahl von Sechsen erreicht ist. Danach soll die Gesamtanzahl der Würfe und die relative Häufigkeit der Augenzahl 6 ausgegeben werden.

Benutzeroberfläche:

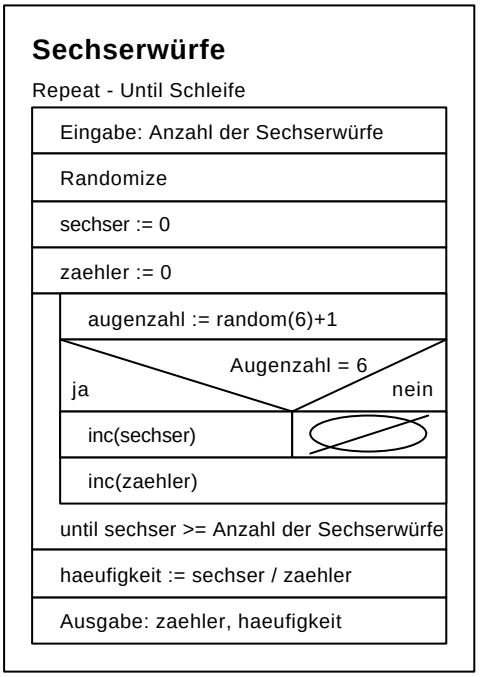


Struktogramm und Quelltext zur Variante mit While – Do:




```
procedure TForm1.BitBtn4Click(Sender:
TObject);
var
  sechser, wurfzahl, zaehler, augenzahl:
integer;
  haeufigkeit: double;
begin
  Randomize;
  wurfzahl := StrToInt(Edit4.Text);
  sechser := 0;
 // Zähler für Sechsen initialisieren
  zaehler := 0;
 // Zähler für Würfe initialisieren
  while sechser < wurfzahl do
  begin
    augenzahl := Random(6)+1;
  // Zufallszahl ermitteln (Würfeln)
    if (augenzahl = 6) then inc(sechser);
// Zähler für Sechsen erhöhen
    inc(zaehler);
   // Zähler für Wurfanzahl erhöhen
  end;
  haeufigkeit := sechser / zaehler;
   // relative Häufigkeit berechnen
  Edit5.Text := IntToStr(zaehler);
  Edit6.Text := FloatToStrF(haeufigkeit,
ffFixed, 6, 2);
end;
```

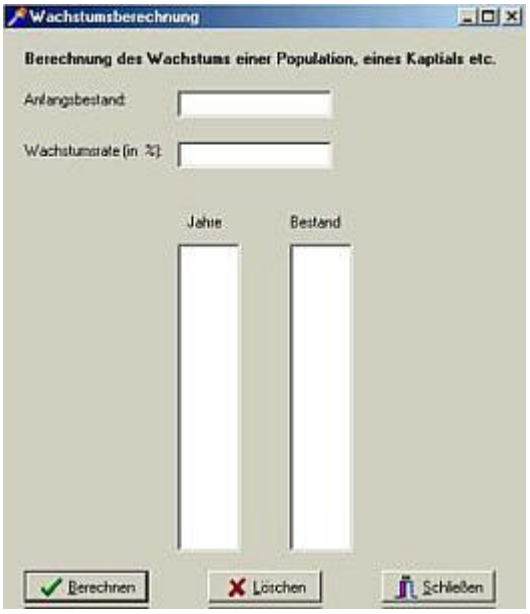
Struktogramm und Quelltext zur Variante mit Repeat - Until:




```
procedure TForm1.BitBtn1Click(Sender:
TObject);
var
  sechser, wurfzahl, zaehler, augenzahl:
integer;
  haeufigkeit: double;
begin
  Randomize;
  wurfzahl := StrToInt(Edit1.Text);
  sechser := 0;
 // Zähler für Sechsen initialisieren
  zaehler := 0;
 // Zähler für Würfe initialisieren
  repeat
    augenzahl := Random(6)+1;
   // Zufallszahl ermitteln (Würfeln)
    if (augenzahl = 6) then inc(sechser);
// Zähler für Sechsen erhöhen
    inc(zaehler);
  // Zähler für Wurfanzahl erhöhen
  until sechser >= wurfzahl;
  haeufigkeit := sechser / zaehler;
 // relative Häufigkeit berechnen
  Edit2.Text := IntToStr(zaehler);
  Edit3.Text := FloatToStrF(haeufigkeit,
ffFixed, 6, 2);
end;
```

Wachstumsberechnung

Benutzeroberfläche:



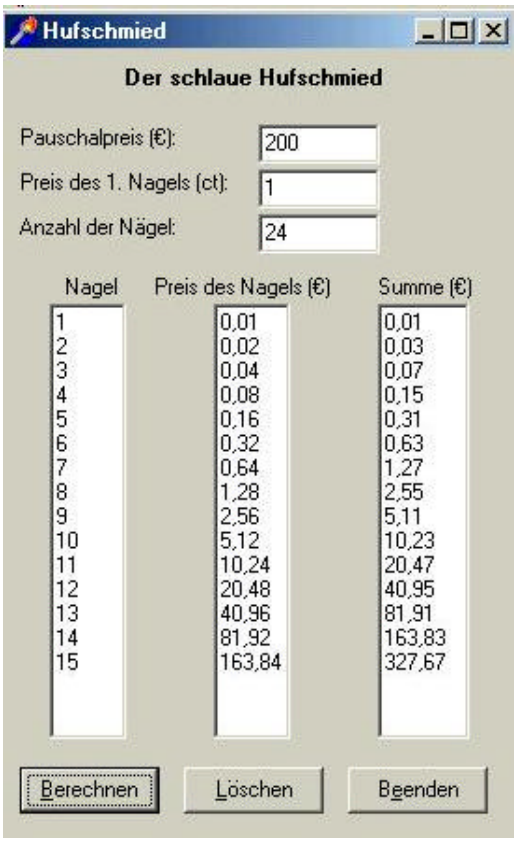
Struktogramm und Quelltext:



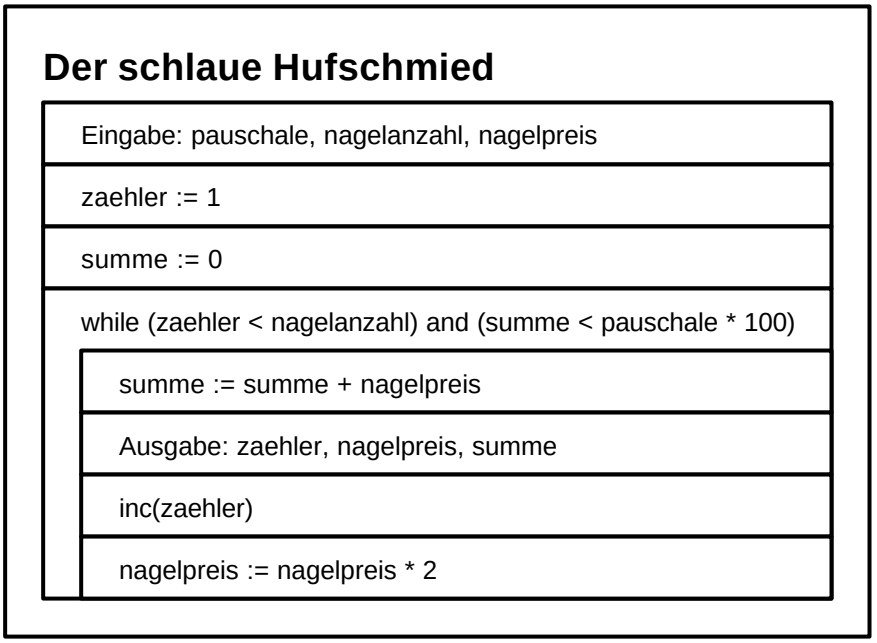

```
procedure TForm1.BitBtn1Click(Sender:
TObject);
var
  anfangsbestand, rate, zuwachs, bestand:
double;
  jahre: integer;
begin
  anfangsbestand := StrToFloat(Edit1.Text);
  bestand := anfangsbestand;
  rate := StrToFloat(Edit2.Text);
  jahre := 0;
  repeat
    Memo1.Lines.Add(IntToStr(jahre));
    Memo2.Lines.Add(FloatToStrF(bestand,
ffFixed, 6, 2));
    zuwachs := bestand * rate / 100;
    bestand := bestand + zuwachs;
    inc(jahre);
  until bestand > 2 * anfangsbestand;
end;
```

Der schlaue Hufschmied

Benutzeroberfläche:



Struktogramm und Quelltext:




```
procedure TForm1.Button1Click(Sender: TObject);
var
  nagelanzahl, zaehler: integer;
  nagelpreis, pauschale, summe: double;
begin
  pauschale := StrToFloat(Edit1.Text);
  nagelpreis := StrToFloat(Edit2.Text);
  nagelanzahl := StrToInt(Edit3.Text);
  zaehler := 1;
  summe := 0;
  while (zaehler < nagelanzahl) and (summe < pauschale * 100) do
  begin
    summe := summe + nagelpreis;
    Memo1.Lines.Add(IntToStr(zaehler));
    Memo2.Lines.Add(FloatToStrF(nagelpreis/100,ffFixed, 6, 2));
    Memo3.Lines.Add(FloatToStrF(summe/100, ffFixed, 6, 2));
    inc(zaehler);
    nagelpreis := nagelpreis * 2;
  end;
end;
```