

Visual Pascal 3.2.

ist ein grafisch orientiertes Unterrichtssystem für TurboPascal und Borland Pascal.

Petr Hejl

Visual Pascal ist vor allem für Unterrichtszwecke bestimmt. Dem **Programmieranfänger** bietet Visual Pascal eine anschauliche Darstellung der Algorithmen. Dem **fortgeschrittenen Programmierer** bietet Visual Pascal moderne Methoden zur Programmentwicklung, indem das Quellprogramm erst im Anschluß an ein Struktogramm entsteht.

Während des Editierens kann man interaktiv zwischen Grafik- und Textansicht wechseln.

VisPascal ist ein graphischer Editor für

- Flußdiagramme,
- Nassi-Shneidermann-Struktogramme und
- Jackson Struktogramme

Da der Ausgangspunkt entweder Quellcode oder Struktogramm sein kann, eignet sich VisPascal gleichermaßen zum Entwickeln neuer Programme als auch zur nachträglichen Dokumentation und Analyse alter Programme („reengineering“).

Die Strukturelemente werden durch VisPascal selbständig auf dem Bildschirm plziert. Man sieht in einem Fenster die Grafik und in einem anderen das Quellprogramm. Die Konversion von der Grafik zum Programm erfolgt automatisch. Wenn man die Grafik durchblättert, scrollt gleichzeitig das Programm, und wenn man das Programm durchblättert, scrollt gleichzeitig die Grafik. Es werden alle Pascal-Steuerelemente (WHILE, REPEAT-UNTIL, FOR, WITH, IF, CASE) generiert. Die graphischen Symbole für die Steuerstrukturen kann man wählen. Selbstverständlich kann die Grafik auch gedruckt werden.

Darüberhinaus eignet sich VisPascal auch zum grafischen Programm-entwurf (für TurboVision) von

- Menüs,

- Datenbanken und
- Vererbungshierarchien

Visual Pascal zeigt den Vorgang der Programmierung Schritt für Schritt. Beispiele können in 4 Fenstern grafisch gleichzeitig verfolgt werden

- Struktogramm,
- Quellprogramm,
- Variablen und
- Aktivitäten.

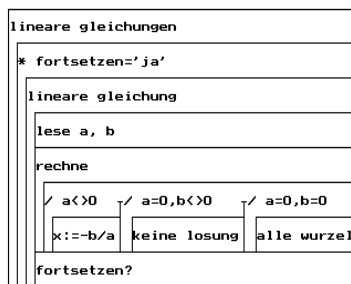
VisualPascal ist auch eine Hilfe für den Unterricht, da der Lehrer seine eigenen Programme und Studientexte, neben den vielen mitgelieferten Beispielen einbauen kann. VisualPascal unterrichtet dann nach diesen Programmen.

Eigenschaften

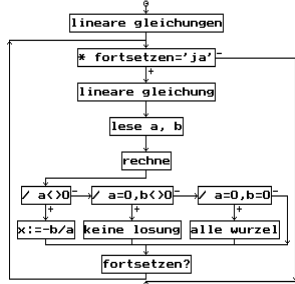
- DOS-Programm
- Für Netze geeignet (Novell, Lantastic ...)
- 640 kByte Hauptspeicher
- 10 MByte Festplattenplatz zuzüglich 1 MByte pro Benutzer in Netzen
- CPU 286 aufwärts, in Netzen ab 386
- Dokumentation etwa 200 Seiten, englisch
- 70 kommentierte Beispiele: Einleitung, Ketten, Nummer, Gleichungen, Dateien, Lernprogramme, Maus, Grafik, Editoren, Syntax-Analysator für Pascal, Animation, TurboVision.

Autor: Ing. Petr Hejl, Ondrouskova 15, CZ-63500 Brno, Tschechische Republik, Tel.:+42/5/792900, Fax:+42/51411346-15 □

Beispiel: Lösung linearer Gleichungen als...



Struktogramm...



...Flußdiagramm...

Das Zeichen "*" steht für eine Schleife (WHILE), das Zeichen "/" für eine Entscheidung (IF). Diese Zeichen haben zwei Aufgaben:

- Damit man auch sequentielle Blöcke ineinander schachteln kann (lineare Gleichung besteht aus Lesen, rechnen und fragen of fortsetzen).
- Damit man Flußdiagramme und Struktogramme editieren und dabei unterscheiden kann, wo eine Sequenz (Blockname ohne Zeichen), und wo eine Schleife (* gefolgt von Bedingung) und wo eine Entscheidung (/ gefolgt von Bedingung) gegeben ist.

Diese Zeichen können frei gewählt werden. Es gibt auch Zeichen für REPEAT, FOR, WITH und CASE.

```

program linear;
var
  a: real;
  b: real;
  x: real;
  fortsetzen: string;
begin { lineare gleichungen }
  fortsetzen:='ja';
  while ((fortsetzen='ja') or { fortsetzen='ja' }
  (fortsetzen='JA')) do
  begin { lineare gleichung }
    begin { lese a, b }
      wri tel n(' a: ');
      readln(a);
      wri tel n(' b: ');
      readln(b);
    end { lese a, b };
    begin { rechne }
      if (a<0.0) then { a<0 }
      begin { x:=-b/a }
        x:=-b/a;
        wri tel n('One root:');
        wri tel n(' x = ',x);
        wri tel n(' Press " Enter ".');
        readln;
      end { x:=-b/a }
      else if ((a=0.0) and (b<>0.0)) then { a=0,b<0 }
      begin { keine losung }
        wri tel n('Kei n Wurzel. ');
        wri tel n(' Drucken Sie " Enter ".');
        readln;
      end { keine losung }
      else if ((a=0.0) and (b=0.0)) then { a=0,b=0 }
      begin { alle wurzel }
        wri tel n('Jede reale Nummer ist Wurzel. ');
        wri tel n(' Drucken Sie " Enter ".');
        readln;
      end { alle wurzel };
    end { rechne };
  end { fortsetzen? }
  wri tel n(' Fortsetzen (nei n-ja und " Enter ") ?');
  readln(fortsetzen);
  end { fortsetzen? };
end { lineare gleichung };
end { lineare gleichungen }.
    
```

...und generiertes Listing