Übungsblock Pascal

Der Variant-RECORD-Typ:

- Unterschiedl. Anzahl Elemente unterschiedlichen Typs → Erweiterung von RECORD
- Syntax: RECORD

```
        feld1: typ1;
        feld2: typ2;
        CASE tag: tagtyp OF
                wert1:( feld3: typ3;feld4:typ4);
                wert2:(feld5:typ5);
                …
    END;
```

- varianter Teil: immer am Ende des Records!

Übungsblock Pascal

- Beispiel

```
TYPE
    familienstand = (ledig, verh, gesch, verw);
    person = RECORD
            name: string;
            alter: 0..120;
            CASE famstand: familienstand OF
                    verh:  (vseit: jahr; gebname:string);
                    gesch:(gseit: jahr);
                    …
        END;
```

- reservierter Speicherplatz: für größtmögliche Komponente

Übungsblock Pascal

Der Packed-Array-Typ:

- Speicherplatzsparende Version des Array-Typs
- Syntax: PACKED ARRAY [*indextyp*] OF *komponententyp*
- *indextyp:* Aufzählung, Unterbereich, boolean oder char
- *komponententyp:* bel. Typ, auch strukturierter (also auch array)
- Beispiel:

  TYPE eng = PACKED ARRAY[1..10] OF Person;
- In Pascal ohne string-Erweiterung

  string = packed array[0..255] of char;

  eigene String-Ende-Kennung nötig (oftmals chr(0))
- var name: string;

  begin

  name := ´Mueller´; {Konstantenzuweisung}

  if ´Meier´< name then …..{relationale Operatoren erlaubt}

Übungsblock Pascal

Der FILE-Typ:

- beliebig lange Folge von Elementen gleichen Typs
- Syntax: *name* = FILE OF *typ;*
- *typ:* bel. Typ möglich
- Länge kann im Programmlauf variieren
- nur sequentielles Durchlaufen möglich (kein Index)
- Elemente dürfen nur gelesen oder geschrieben werden
- Beispiel:

```
TYPE        …
            personen = FILE OF person;
VAR
            mitfahrer: personen;
```

erzeugt automatisch eine Variable mitfahrer^ vom Typ person, die Wert des aktuellen FILE-Elements aufnimmt

Übungsblock Pascal

- Beispiel (Fortsetzung):

```
VAR
        mitfahrer: personen;
        start: person;
BEGIN
    mitfahrer^.vorname := ´Paul´;
    mitfahrer^.nachname := ´Meier´;
    put(mitfahrer); {hängt neue Person (hier an leere Folge) an}
    …
    reset(mitfahrer); {positioniert mitfahrer^ auf 1. Element}
    get(mitfahrer); {liest akt. Element in mitfahrer^ ein,
                     springt zum nächsten Element}
    rewrite(mitfahrer); {löscht komplett alten Inhalt,
                         eof(mitfahrer) liefert true}
```

Übungsblock Pascal

Allgemein:

- nur put, get, reset, rewrite können auf FILE arbeiten
- steht Puffervariable *varname*^ auf letztem Element:
  - get(varname) holt Inhalt des letzten Elements
  - varname^ springt einen Platz weiter
  - eof(varname) wird zu true
  - erneuter get-Aufruf liefert Fehler, put-Aufruf erweitert FILE um ein Element