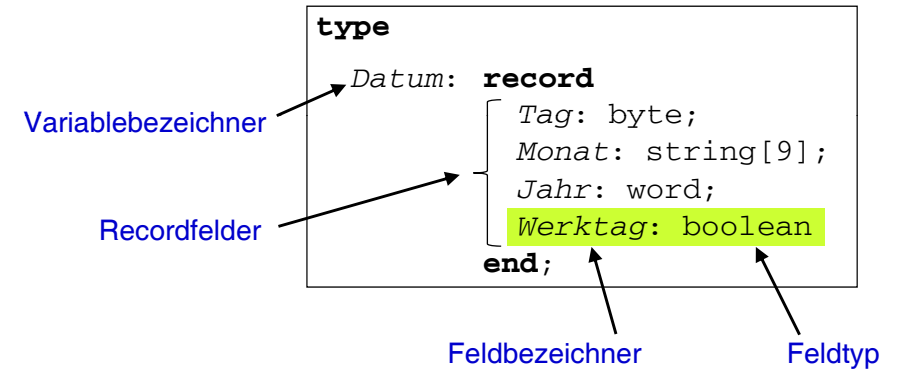


- **Strukturierte Variablen: Records**
- Datentypen: Aufzählungstypen, Mengen
- Lösungsansätze für Praxismodul 6

Lesen Sie den Begleittext Seite 60 - 67

Variablen vom Typ **Record** erlauben mehrere **Felder** von unterschiedlichem Typ.



### Zugriff auf die Felder eines Records geschieht über Namen

```
var heute, morgen: Datum;
```

```
heute.Tag:= 8;
heute.Monat:= 'Mai';
heute.Jahr:= 2008;
heute.Werktag:= true
```

```
morgen:= heute;
morgen.Tag:= 9;
```

Die Felder der Variablen **heute** und **morgen** sind mit Ausnahme des Feldes **Tag** identisch.

### Arrays und Records im Vergleich

```
var
```

```
  Punkt: array[1..3] of real;
  Pkt: array['x'..'z'] of real;
  ch: char;
```

```
for ch:= 'x' to 'z' do Pkt[ch]:= random;
```

```
var
```

```
  Punkt: record
    x, y, z: real;
  end;
```

```
Punkt.x:= random;
Punkt.y:= random;
Punkt.z:= random;
```

Keine Schleifen möglich

```
with Punkt do
begin
```

```
  x:= random;
  y:= random;
  z:= random;
```

```
end;
```

## Erfassen verschiedener Eigenschaften eines Objekts mit Records

```
var
  Punkt: record
    x, y, z: real;
    inKugel: boolean
  end;
  radius: real;
```

```
with Punkt do
begin
  x:= random * radius;
  y:= random * radius;
  z:= random * radius;
  inKugel:= sqr(x)+sqr(y)+sqr(z) <= sqr(radius)
end;
```

## Arrays von Records

```
type
  Datum = record
    Tag: byte;
    Monat: string[9];
    Jahr: word;
  end;

var
  EinJahr: array[1..366] of Datum;
  Heute: Datum;
```

```
EinJahr[1]  Tag Monat Jahr      Heute:= EinJahr[2]
EinJahr[2]  Tag Monat Jahr      Tag Monat Jahr
EinJahr[3]  Tag Monat Jahr      Heute 9 Juni 2008
```

	Tag	Monat	Jahr
EinJahr[1]	8	Juni	2008
EinJahr[2]	9	Juni	2008
EinJahr[3]	10	Juni	2008

	Tag	Monat	Jahr
Heute	9	Juni	2008

## Geschachtelte Records

```
type
  Mo = record
    Name: string[9];
    Jzeit: char;
  end;
```

```
var
  Jahr: record
    Tag: 1..366;
    Monat: Mo;
  end;
  Monat: Mo;
```

```
Monat.Name:= 'Oktober';
Monat.Jzeit:= 'H';
Jahr.Monat:= Monat;
```

```
Jahr.Tag:= 126;
Jahr.Monat.Name:= 'Mai';
Jahr.Monat.Jzeit:= 'F';
```

- Strukturierte Variablen: Records
- **Datentypen: Aufzählungstypen, Mengen**
- Lösungsansätze für Praxismodul 6

## Aufzählungstypen

Listet alle Werte, die Variablen von diesem Typ annehmen können.

### type

```
Waehrung = (CHF, Dollar, Euro, Yen);
```

### var

```
Geld: Waehrung;
```

Die Variable Geld kann nur die folgenden Werte annehmen :  
CHF, Dollar, Euro, Yen

## Mengentypen

- Definiert die Menge aller möglichen Mengen von Elementen eines Basistyps.
- Die Basis eines Mengentyps muss ein Ordinaltyp sein, z.B. ein Aufzählungstyp:

### type

```
Waehrung = (CHF, Dollar, Euro);
```

### var

```
Geld: set of Waehrung;
```

## Mengentypen: Eigenschaften

Die Werte der Variablen **Waehrung** sind die Mengen:

```
[], [CHF], [Dollar], [Euro], [CHF, Dollar],  
[CHF, Euro], [Dollar, Euro], [CHF, Dollar, Euro]
```

Der Basistyp darf nicht mehr als 256 Elemente enthalten.

Zuweisungsanweisungen:

```
Geld:= [];  
Geld:= [Euro];  
Geld:= [CHF, Dollar];
```

## Mengentypen: Operationen

### Vereinigung +

```
Geld:= [CHF] + [Dollar] ⇨ [CHF, Dollar]
```

### Durchschnitt \*

```
Geld:= [CHF, Euro] * [Euro, Dollar] ⇨ [Euro]
```

### Differenz –

```
Geld:= [CHF, Dollar, Euro] – [Euro] ⇨ [CHF, Dollar]
```

### Mengenzugehörigkeit

```
[CHF] in [CHF, Dollar] ⇨ TRUE
```

## Mengentypen: Operationen

### Vergleiche

- = Gleichheit
- <> Ungleichheit
- <= Teilmenge
- >= Obermenge

```
[CHF] = [CHF, Dollar] ⇒ FALSE  
[CHF] <> [CHF, Dollar] ⇒ TRUE  
[CHF] <= [CHF, Dollar] ⇒ TRUE  
[CHF] >= [CHF, Dollar] ⇒ FALSE
```

## Mengentypen vereinfachen Vergleiche

```
var ch: Char;  
( '0' <= ch ) and ( ch <= '9' )
```

ist wahr, wenn *ch* eine Ziffer darstellt

Der Mengentyp erlaubt einen bündigeren Ausdruck:

```
ch in [ '0'..'9' ]
```

Die folgende Anweisung überspringt Leerzeichen und Satzzeichen in der File "F". :

```
while ch in [ ' ', ',', '.', ';', ':', ':' ] do read(F, ch);
```