

# Informatik I Übung, Woche 38

Giuseppe Accaputo

17. September, 2015

## Plan für heute

1. Wer bin ich
2. Allgemeines
3. Zusammenfassung der bisherigen Vorlesungsslides
4. Kurze Vorbesprechung zur Übung 1

## Wer bin ich

- ▶ Giuseppe Accaputo
- ▶ Studium: Rechnergestützte Wissenschaften (2. Semester Master)
- ▶ E-Mail: [accaputg@student.ethz.ch](mailto:accaputg@student.ethz.ch)
- ▶ Homepage: <http://n.ethz.ch/~accaputg/>

## Allgemeines

- ▶ Versucht die Übungen so gut wie möglich zu lösen
  - ▶ Learning by doing
- ▶ Bei Fragen könnt ihr mir jederzeit schreiben
  - ▶ “There are naive questions, tedious questions, ill-phrased questions, questions put after inadequate self-criticism. But every question is a cry to understand the world. There is no such thing as a dumb question” - Carl Sagan
- ▶ Über Feedback bin ich immer froh und dankbar

## Aufbau eines Pascal Programms

```
Program Example;      {Kopf}
Var                   {Deklarationen}
    Num1 , Num2 , Sum : Integer;
Begin                 {Rumpf}
    Writeln('Input number 1: ');
    Readln(Num1);
    Writeln('Input number 2: ');
    Readln(Num2);
    Sum := Num1 + Num2;
    Writeln(Sum);
End.
```

## Die wichtigsten Anweisungen

- ▶ `Writeln(...);` → Gibt die Argumente aus (mit Zeilenende)
- ▶ `Write(...);` → Gibt die Argumente aus (ohne Zeilenende)
- ▶ `Readln(...);` → List Werte von der Tastatur (bis Zeilenende)
- ▶ `A := B;` → Weise der Variable A den Wert B zu

## Write, Writeln

- ▶ Funktion: Geben die Argumente auf der Konsole aus
  - ▶ `Writeln('Hello, world!');` → Hello, world!
  - ▶ `q11 := 12; Writeln(q11);` → 12
- ▶ Längenangaben sind auch möglich
  - ▶ Beispiel: gib 77 in 5 Zeichen aus:  
`Writeln(77:5)` → `__77`
- ▶ Längenangaben zu den Nachkommastellen sind auch möglich
  - ▶ Beispiel: Gib 1.23456 mit 2 Nachkommastellen in 5 Zeichen aus:  
`Writeln(1.23456:5:2)` → `_1.23`

## Readln

- ▶ Funktion: liest den Wert in eine Variable ein
  - ▶ `Readln(v1)` → Liest den Wert von der Tastatur in die Variable `v1` ein

## Zuweisungsoperator :=

- ▶  $A := B$  weist der Variable  $A$  das Resultat des Ausdrucks  $B$  zu
  - ▶ Der Ausdruck  $B$  wird zuerst ausgewertet falls nötig (z.B. wenn  $B$  eine Variable ist) und dann der Variablen  $A$  zugewiesen

## Variablen

- ▶ repräsentieren (evtl. wechselnde) Werte
- ▶ haben Namen und Typ (z.B. `INTEGER`)
- ▶ Regeln bezüglich der Definition von Variablennamen:
  - ▶ Erstes Zeichen: Buchstabe oder Underscore `_`
  - ▶ Weitere Zeichen: Buchstaben, Zahlen, Underscore
  - ▶ Keine Schlüsselwörter wie `BEGIN`, `PROGRAM`, etc.

```
VAR
    a    :  INTEGER ;
    b    :  REAL ;
BEGIN
    a := 10; b := 1.2345;
END .
```

## Typen, Literale, Ausdrücke

Typ	Bedeutung	Literal
INTEGER	Ganze Zahlen	42
REAL	Reelle Zahlen	3.141
STRING	Zeichenkette	'Hello, World!'
BOOLEAN	Wahrheitswert	true

## Konstanten

- ▶ repräsentieren fixe Werte
- ▶ können nicht mehr *verändert* (neuen Wert zuweisen) werden im Rumpf

```
CONST    pi := 3.141;  
VAR      twopi : REAL;  
BEGIN  
    twopi := 2 * pi;  
END.
```