# explizit vs. implizit

"explizit" heißt, dass man genau erläutert was der Compiler machen soll,

ohne ihm dabei "Interpretationsspielraum" zu geben.

"implizit" ist nur eine unvollständige Angabe von Handlungsanweisungen.

Der Compiler soll aus den nicht explizit ausgesprochenen Anweisungen

den fehlenden Teil der Informationen ergänzen,

"machen, was er für richtig hält".

Für allgemeine Programmiersprachen ist die explizite Vorgehensweise typischer,

für Skriptsprachen eher die implizite.

Manchmal hat man die Wahl.

Das explizite Vorgehen ist dabei zwar aufwändiger,

aber weniger fehleranfällig u. damit besserer Stil.

Beispiel für explizite u. implizite Deklaration einer Variablen:

Es soll eine Ganzzahlvariable "a" deklariert u. mit 5 init. werden.

explizit: int a = 5;

implizit: a = 5;

Bei der expliziten Deklaration wird der Typ "int" mit angegeben.

Der Compiler hat keinen "Interpretationsspielraum".

Anders bei der impliziten Deklaration:

Es wird nur geschrieben "a = 5".

In zwei Punkten "interpretiert" der Compiler:

1. Da die Variable "a" nicht deklariert wurde,  
   aber darauf zugegriffen wird,  
   soll sie deklariert werden.
2. Der Typ der Variablen wird aus dem zugewiesenen Wert geschlossen:  
   "5" ist vom Typ "int".  
   Also soll wohl eine int-Variable deklariert werden.

Interpretation:

Was spricht für die implizite, was für die explizite Variante?

* Die implizite ist kürzer,   
  also weniger Schreibarbeit
* kann aber zu schwer auffindbaren Fehlern führen

2 Beispiele, wie schwer auffindbare Fehler durch implizite Deklaration entstehen können:

Beispiel 1:

Sei in beiden Fällen eine Variable deklariert u. init.

Ich möchte mich schreibend auf die Variable beziehen,

verschreibe mich aber beim Variablennamen.

Was passiert:

* expizit: Fehlermeldung: "cannot find symbol"
* implizit: Compiler würde fehlinterpretieren,  
  dass ich eine neue Variable deklarieren möchte  
  u. weist ihr den Wert zu,  
  den eigentlich einer anderen, schon existenten Variablen  
  zugewiesen werden soll.  
  Es erfolgt keine Fehlermeldung!  
  Eine Berechnung auf der richtig geschriebenen Variablen wird jedoch falsch,  
  weil die gewünschte Zuweisung nicht erfolgt ist.

Der Fehler bei impliziter Deklaration ist kaum zu finden,

da keine Fehlermeldung erfolgt.

Bei der expliziten erfolgt eine Fehlermeldung,

und zwar an genau der Stelle, wo der Fehler liegt.

Beispiel 2:

Sie wollen die eine Variable "a" als int-Zahl deklarieren u. mit 0 init.

Dabei verschreiben Sie sich aber u. erwischen statt der Null das große "O".

* explizit: int a = O; // Großbuchstabe von "o"
* implizit: a = O;

Was passiert:

* explizit: Fehlermeldung "incompatible types"
* implizit: Compiler wird aus dem Großbuchstaben fehlinterpretieren,  
  dass ich eine Variable vom Typ "char" deklarieren möchte.  
  Es erfolgt keine Fehlermeldung!  
  Mglw. wird später eine Fehlermeldung erfolgen,  
  wenn ich versuche, auf die char-Variable etwas zu addieren.  
  "Mglw" deshalb,  
  weil je nach Sprache ein Buchstabe als int-Wert des ASCII-Zeichens gespeichert wird.  
  Dabei könnte man auf den Buchstaben eine int-Zahl addieren  
  u. bekäme den ASCII-Wert eines anderen Buchstaben.

Selbst die Ausgabe des Großbuchstabens "o" unterschied sich kaum

von der der Ziffer 0.  
Es gäbe auch keine Fehlermeldung.  
Die Fehlersuche würde sich sehr aufwändig gestalten!

Deshalb mein Appell:

Nach Möglichkeit implizite Mechanismen vermeiden.

Keinen "Interpretationsspielraum" dem Compiler,

da dies zu Fehlinterpretationen führen kann.