

# Modellierung und Optimierung mit OPL

## 2 Einführung in OPL

Andreas Popp



## 2.1 Aufbau eines OPL-Projekts

## 2.2 Grundlegende Datentypen und Operatoren

## Datentypen

## Operatoren

## 2.3 Mathematische Modelle in OPL-Syntax

## 2.4 Die CPLEX Studio IDE

## 2.4 Die CPLEX Studio IDE

## 2.5 Fehler und Warnungen in OPL



## 2.1 Aufbau eines OPL-Projekts

# Arten von OPL-Dateien

2 Einführung in  
OPL

CC-BY-SA  
A. Popp

2.1 Aufbau eines  
OPL-Projekts

2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen

Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL

**Modelldateien** Beschreibung des allgemeinen  
Optimierungsmodells (Endung: .mod)

**Datendateien** Daten zur Instanziierung eines OPL-Modells  
(Endung: .dat)

**Einstellungsdateien** Einstellungen für den Solver (Endung:  
.ops)



# Aufbau eines OPL-Projekts

2 Einführung in  
OPL

CC-BY-SA  
A. Popp

2.1 Aufbau eines  
OPL-Projekts

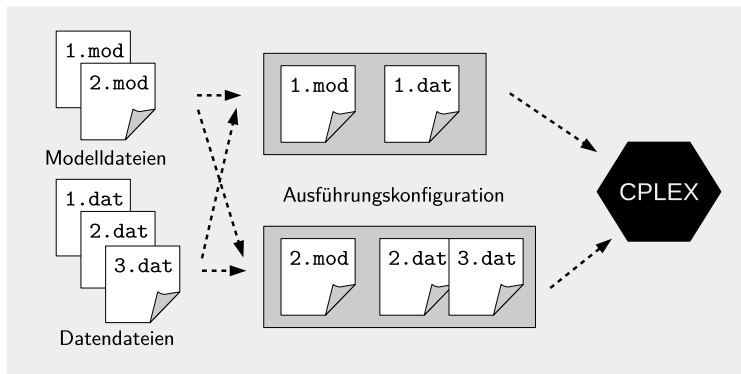
2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen  
Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL







# Aufbau einer einfachen Zuweisungsanweisung

2 Einführung in  
OPL

CC-BY-SA  
A. Popp

2.1 Aufbau eines  
OPL-Projekts

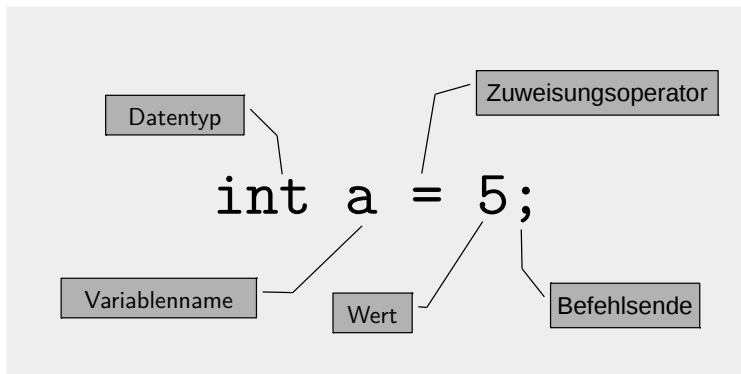
2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen  
Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL



**float** Gleitkommazahl mit beliebigem Vorzeichen.  
Beispielliterale: 0.0, 1.0, 3.14, -7.86

`boolean` eigentlich ein logischer Wahrheitswert; bei Entscheidungsvariablen eine 0-1-Variable.

9/28 ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

Datentypen  
Operatoren

## 2.4 Die CPLEX Studio IDE

Zugriff mittels Index, z.B.:

10/28 ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

- ▶ Arrays können ineinander geschachtelt werden um mehrfache Indizes abzubilden, z.B.

```
float Entf[Standorte][Standorte] =
    [[0.0, 5.05, 4.89],
     [5.05, 0.0, 1.22],
     [4.89, 1.22, 0.0]];
```

- Zuordnungsregel: von links nach rechts, von außen nach innen

- ▶ Zuweisungsoperator =
- ▶ Arithmetische Operatoren
  - + Addition
  - Subtraktion
  - \* Multiplikation
  - / Division (selten in linearen Modellen)
- ▶ Vergleichsoperatoren (für lineare Modelle)
  - == gleich
  - <= kleiner-gleich
  - >= größer-gleich

2.1 Aufbau eines  
OPL-Projekts

2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen

Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL









## Mathematisches Modell

## Parameter:

$p_i$  Preis von Produkt  $i \in I$

$c_r$  Kapazität von Ressource  $r \in R$

$v_{ri}$	Kapazitätsverbrauch von Produkt $i \in I$ auf Ressource $r \in R$
----------	---

## Modelldatei

```
//Parameter
float p[I] = ...; //Preis
float c[R] = ...; //Kapazität
float v[R][I] = ...; //Kapazitätsverbrauch
```

# Beispiel: Produktionsproblem – Parameter

2 Einführung in  
OPL

CC-BY-SA  
A. Popp

## Mathematisches Modell

### Parameter:

$p_i$  Preis von Produkt  $i \in I$

$c_r$  Kapazität von Ressource  $r \in R$

$v_{ri}$  Kapazitätsverbrauch von Produkt  $i \in I$  auf  
Ressource  $r \in R$

## Datendatei

```
//Parameter  
p = [2.9, 3.3, 2.2];  
c = [64.0, 48.0];  
v = [  
    [5.3, 2.9, 2.5],  
    [3.9, 4.8, 3.1]  
];
```

2.1 Aufbau eines  
OPL-Projekts

2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen  
Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL

# Beispiel: Produktionsproblem – Entscheidungsvariablen

## Mathematisches Modell

### Entscheidungsvariablen:

$x_i$  Produktionsmenge von Produkt  $i \in I$

[...]

$$x_i \geq 0 \quad \forall i \in I$$

## Modelldatei

```
//Entscheidungsvariablen  
dvar float+ x[I]; //Produktionsmenge
```

## Beispiel: Produktionsproblem – Zielfunktion

## Mathematisches Modell

$$\max \sum_{i \in I} p_i \cdot x_i$$

## Modelldatei

```
//Zielfunktion
maximize sum(i in I)(p[i]*x[i]);
```

CC-BY-SA  
A. Popp

Datentypen  
Operatoren

## 2.3 Mathematische Modelle in OPL-Syntax

## 2.4 Die CPLEX Studio IDE

# Beispiel: Produktionsproblem.mod

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6 float p[I] = ...; //Preis
7 float c[R] = ...; //Kapazität
8 float v[R][I] = ...; //Kapazitätsverbrauch
9
10 //Entscheidungsvariablen
11 dvar float+ x[I]; //Produktionsmenge
12
13 //Zielfunktion
14 maximize sum(i in I)(p[i] * x[i]);
15
16 //Nebenbedingungen
17 subject to{
18
19     //Kapazitätsrestriktion
20     forall(r in R)
21         sum(i in I)(v[r][i]*x[i]) <= c[r];
22
23 }
```

2 Einführung in  
OPL

CC-BY-SA  
A. Popp

2.1 Aufbau eines  
OPL-Projekts

2.2 Grundlegende  
Datentypen und  
Operatoren

Datentypen  
Operatoren

2.3 Mathematische  
Modelle in  
OPL-Syntax

2.4 Die CPLEX  
Studio IDE

2.5 Fehler und  
Warnungen in OPL

```
1 //Indextmengen
2 I = {"Produkt_1", "Produkt_2", "Produkt_3"};
3 R = {"Maschine_A", "Maschine_B"};
4
5 //Parameter
6 p = [2.9, 3.3, 2.2];
7 c = [64.0, 48.0];
8 v = [
9     [5.3, 2.9, 2.5],
10    [3.9, 4.8, 3.1]
11 ];
```

```
> oplrun -v Produktionsproblem.mod
LewigSanstetten.dat
```

• • •

OBJECTIVE: 35.61677

← Optimalwert

• • •

```
x = [11.737 0 0.71856];
```

← Optimallösung

Datentypen  
Operatoren

## 2.3 Mathematische Modelle in OPL-Syntax

## 2.4 Die CPLEX Studio IDE





# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL

IBM ILOG CPLEX Optimization Studio

Produktionsproblem.mod

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6 float p[I] = ...; //Preis
7 float c[R] = ...; //Kapazität
8 float v[R][I] = ...; //Kapazitätsverbrauch
9
10 //Entscheidungsvariablen
11 dvar float+ x[I]; //Produktionsmenge
12
13 //Zielfunktion
14 maximize sum(i in I)(p[i] * x[i]);
15
16 //Nebenbedingungen
17 subject to
18
19 //Kapazitätsrestriktion
20 forall(r in R)
21   sum(i in I)(v[r][i]*x[i]) <= c[r];
22
23 }
```

Gliederung

- Unter Verwendung von CPLEX
- Externe Daten (5)
  - c: float[R]
  - I: (string)
  - p: float[I]
  - R: (string)
  - v: float[R][I]
- Entscheidungsvariablen (1)
  - x: dvar float+[I]
  - Zielfunktion: einfach
- Nebenbedingungen (1)

Eigenschaften

Eigenschaft	Wert
0 Elemente	

Problem

Lösung mit Zielsatzung 35,617

Name	Wert
Externe Daten (5)	
c	[64 48]
I	("Produkt_1" "Produkt_2" "Prod
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

1 Elemente ausgewählt

00:00:04.5

# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL

**Projektbrowser:**  
Übersicht über die Inhalte  
des aktuellen Projekts

```
//Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6
17
18
19 //Kapazitätsrestriktion
20 forall(r in R)
21   sum(i in I) (v[r][i]*x[i]) <= c[r];
22
23 }
```

Name	Wert
Lösung mit Zielfunktion 35,617	
Daten (5)	
c	[64 48]
I	("Produkt_1" "Produkt_2" "Produkt_3")
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL

The screenshot shows the IBM ILOG CPLEX Optimization Studio IDE interface. The main window displays a project named 'Produktionsproblem.mod'. The left sidebar shows the project structure with files like 'Produktionsproblem.mod', 'Lewig\_Sanstetten.dat', and 'Produktionsproblem.mod : CPLEX'. The central editor shows the OPL model code, including declarations for products, resources, and parameters, and a constraint section. The right sidebar shows the 'Gliederung' (Outline) view with sections like 'Externe Daten', 'Entscheidungsvariablen', 'Zielfunktion', and 'Nebenbedingungen'. The bottom status bar indicates '0 Elemente'.

**Projekt**

**Ausführungskonfiguration**

**Modelldatei**

**Datendatei**

Name	Wert
c	[64 48]
r	("Produkt_1" "Produkt_2" "Prod
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

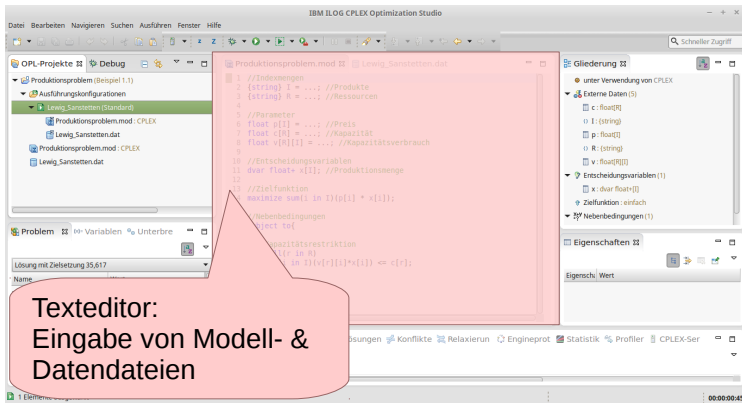
### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL



# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL

The screenshot displays the IBM ILOG CPLEX Optimization Studio interface. The main window shows a project named 'Produktionsproblem (Beispiel 1.1)' with a file 'Lewig\_Sanstetten.dat'. The 'Problem' tab is active, showing a table of results for the 'Lösung mit Zielfunktion 35,617'. The table lists variables and their values: 'c' (54.48), 'f' ('Produkt\_1' 'Produkt\_2' 'Prod'), 'p' (2.9 3.3 2.2), 'r' ('Maschine\_A' 'Maschine\_B'), and 'v' (5.3 2.9 2.5) (3.9 4.8 3.1)). A red callout bubble points to the 'Problem' tab with the text: 'Problembrowser: Hier erscheint nach erfolgreicher Lösung des Problems das Ergebnis'. The 'Fehler' (Errors) tab is also visible, showing 0 elements.

IBM ILOG CPLEX Optimization Studio

Produktionsproblem.mod Lewig\_Sanstetten.dat

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Daten
6
7 //Zielfunktion
8
9 //Nebenbedingungen
10
11 //Variablen
12
13 //Ergebnis
14
15 //Ergebnis
16
17 //Ergebnis
18
19 //Ergebnis
20
21 //Ergebnis
22
23 }
```

Problembrowser:  
Hier erscheint nach erfolgreicher  
Lösung des Problems das Ergebnis

Problem

Lösung mit Zielfunktion 35,617

Name	Wert
Externe Daten (5)	
c	[54.48]
f	("Produkt_1" "Produkt_2" "Prod")
p	[2.9 3.3 2.2]
r	("Maschine_A" "Maschine_B")
v	[5.3 2.9 2.5] [3.9 4.8 3.1]

0 Elemente

# Die CPLEX Studio IDE

## 2 Einführung in OPL

CC-BY-SA  
A. Popp

### 2.1 Aufbau eines OPL-Projekts

### 2.2 Grundlegende Datentypen und Operatoren

Datentypen  
Operatoren

### 2.3 Mathematische Modelle in OPL-Syntax

### 2.4 Die CPLEX Studio IDE

### 2.5 Fehler und Warnungen in OPL

IBM ILOG CPLEX Optimization Studio

Produktionsproblem.mod

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6
7 //Kapazitätsrestriktion
8 forall(r in R)
9   sum(i in I)(v[r][i]*x[i]) <= c[r];
10
11 //Nebenbedingungen
12 subject to
13
14 maximize sum(i in I)(p[i]*x[i]);
15
16 //Zielfunktion - einfach
17
18
19
20
21
22
23 }
```

Gliederung

- Unter Verwendung von CPLEX
- Externe Daten (5)
  - c: float[]
  - I: (string)
  - p: float[]
  - R: (string)
  - v: float[][]
- Entscheidungsvariablen (1)
  - x: float[]
- Zielfunktion - einfach
- Nebenbedingungen (1)

Eigenschaften

Eigenschaft	Wert
c	[64 48]
r	("Produkt_1" "Produkt_2" "Prod
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

Lösung mit Zielfestsetzung 35,617

1 Elemente ausgewählt

00:00:00:45

## 2.5 Fehler und Warnungen in OPL



# Arten von Fehlern

## Einteilung nach Schwere

**Fehler** Verhindert das erfolgreiche Abschließen eines Lösungslaufs

**Warnung** Behindert den Lösungslauf nicht, es kann aber zu unerwünschten Ergebnissen kommen.  
Manchmal Hinweis auf Fehler im Code.

## Einteilung nach Zeitpunkt des Auftretens

**Compilerfehler** Treten bei Übersetzung des Problems für den Solver auf. Werden von der IDE erkannt.

**Laufzeitfehler** Treten erst zur Laufzeit des Solvers auf. Werden nicht von der IDE erkannt, aber nach Lösungslauf angezeigt.

- ▶ `syntax error, unexpected ...` (Compilerfehler)
  - ▶ Compiler versteht die Anweisung nach „unexpected“ an dieser Stelle nicht
  - ▶ fehlender Strichpunkt?
- ▶ `syntax error, unexpected =` (Compilerfehler)
  - ▶ Spezialfall zu oben
  - ▶ meist Verwechslung von Zuweisungsoperator `=` und Vergleichsoperator `==`
- ▶ Der Typ `...` kann nicht für `...` verwendet werden (Compilerfehler)
  - ▶ Datentypen durcheinander gebracht
- ▶ Der Index für den Array `...` liegt außerhalb des gültigen Bereichs (Laufzeitfehler)
  - ▶ ein Array wurde mit einem ungültigen Index angesprochen