

## Príklad

### Zadanie

2 pekárne pečú denne 500 a 300 ks chleba. Rozvážajú ho do 4 predajní, pričom každá predajňa (odberateľ) požaduje 200 ks chleba. Je daná matica nákladov na prepravu 1 ks chleba medzi pekárňami a predajňami.

	O1	O2	O3	O4
P1	5	4	9	2
P2	8	5	4	7

Určte rozvoz chleba tak, aby boli náklady na dopravu minimálne.

### Model zapísaný v MOR-e

$$\min f = 5x_{11} + 4x_{12} + 9x_{13} + 2x_{14} + 8x_{21} + 5x_{22} + 4x_{23} + 7x_{24}$$

st

$$x_{11} + x_{12} + x_{13} + x_{14} < = 500$$

$$x_{21} + x_{22} + x_{23} + x_{24} < = 300$$

$$x_{11} + x_{21} => 200$$

$$x_{12} + x_{22} => 200$$

$$x_{13} + x_{23} => 200$$

$$x_{14} + x_{24} => 200$$

allinteger

boundonsum=100000

## **Model zapísaný v Xpress-Mosel**

```
model pekarne

uses "moxprs"

declarations
    x11,x12,x13,x14,x21,x22,x23,x24: mpvar
end-declarations

x11 is_integer
x12 is_integer
x13 is_integer
x14 is_integer
x21 is_integer
x22 is_integer
x23 is_integer
x24 is_integer

naklady := 5*x11+4*x12+9*x13+2*x14+8*x21+5*x22+4*x23+7*x24 !ucelova funkcia
x11 + x12 + x13 + x14 <= 500
x21 + x22 + x23 + x24 <= 300
x11 + x21 >= 200
x12 + x22 >= 200
x13 + x23 >= 200
x14 + x24 >= 200

minimise(naklady);

writeln("Hodnota ucelovej fcie je ", getobjval);
writeln("x11=", getsol(x11));
writeln("x12=", getsol(x12));
writeln("x13=", getsol(x13));
writeln("x14=", getsol(x14));
writeln("x21=", getsol(x21));
writeln("x22=", getsol(x22));
writeln("x23=", getsol(x23));
writeln("x24=", getsol(x24));

end-model
```

## **Model zapísaný v Xpress-Mosel s použitím polí, súm a cyklov**

```
model pekarne

uses "mxxprs" ! pouzi Xpress-optimizer

declarations
  PEKARNE=1..2
  OBCHODY=1..4
  KAPACITY: array(PEKARNE) of real;
  POZIADAVKY: array(OBCHODY) of real;
  C: array(PEKARNE,OBCHODY) of real; !naklady z i do j
  X: array(PEKARNE,OBCHODY) of mpvar; !mnozstvo chleba z i do j
end-declarations

KAPACITY := [500,300]
POZIADAVKY := [200,200,200,200]
C := [5,4,9,2,8,5,4,7]
forall(i in PEKARNE, j in OBCHODY) X(i,j) is_integer !Xij sú celociselne
NAKLADY := sum(i in PEKARNE, j in OBCHODY) (C(i,j)*X(i,j)) !ucelova funkcia
forall(i in PEKARNE) sum(j in OBCHODY) X(i,j) <= KAPACITY(i) !kapacity
forall(j in OBCHODY) sum (i in PEKARNE) X(i,j) = POZIADAVKY(j) !poziadavky

minimise(NAKLADY);

!vypis riesenia
writeln("Hodnota ucelovej fcie je ", getobjval);
writeln(" ");
forall(i in PEKARNE) do
  forall(j in OBCHODY) do
    writeln("X[" , i , " , " , j , " ]=" , strfmt(getsol(X(i,j)),2));
  end-do
end-do

end-model
```

## **Model zapísaný v Xpress-Mosel s využitím súboru s datami**

```
model pekarne

uses "mmxprs";

options explterm; !budeme pouzivat bodkociarku ako oddelovac prikazov

declarations
    PEKARNE: set of string;                !Nacitame zo suboru
    OBCHODY: set of string;                !Nacitame zo suboru
    KAPACITY: array(PEKARNE) of real;
    POZIADAVKY: array(OBCHODY) of real;
    C: array(PEKARNE,OBCHODY) of real;    !naklady z i do j
    X: array(PEKARNE,OBCHODY) of mpvar;   !mnozstvo chleba z i do j
end-declarations

!Inicializacia zo suboru
initializations from "pekarne.dat"
    PEKARNE;OBCHODY; KAPACITY; POZIADAVKY; C;
end-initializations

forall(i in PEKARNE, j in OBCHODY) create(X(i,j)); !prem. sa nevytvorili
forall(i in PEKARNE, j in OBCHODY) X(i,j) is_integer; !Xij sú celociselne

NAKLADY := sum(i in PEKARNE, j in OBCHODY) (C(i,j)*X(i,j)); !UF
forall(i in PEKARNE) sum(j in OBCHODY) X(i,j) <= KAPACITY(i); !kapacity
forall(j in OBCHODY) sum (i in PEKARNE) X(i,j) = POZIADAVKY(j); !poziadavky

minimise(NAKLADY);

!vypis riesenia
writeln("Hodnota ucelovej fcie je ", getobjval);
writeln(" ");
forall(i in PEKARNE) do
    forall(j in OBCHODY) do
        writeln("X[" , i , " , " , j , "]=",strfmt(getsol(X(i,j)),2));
        !writeln(strfmt(getsol(X(i,j)),2));
    end-do
end-do

end-model
```

## **Štruktúra súboru "pekarne.dat"**

```
PEKARNE: [ "P1" , "P2" ]
OBCHODY: [ "O1" , "O2" , "O3" , "O4" ]
KAPACITY: [ 500 , 300 ]
POZIADAVKY: [ 200 , 200 , 200 , 200 ]
C: [ 5 , 4 , 9 , 2 , 8 , 5 , 4 , 7 ]
```