

Dominik Matoulek  
Student AI2  
FIM UHK

# Fortran

Naučme se pracovat s dědečkem

# Obsah

Historie

Programujeme

Tipy

## Úvod

## Co je Fortran?

Fortran je vysokoúrovňový programovací jazyk

- FORMula TRANslator
- je kompilovaný
- existují různé verze
- poslední je Fortran 2008
- OOP a generika

## Jak to začalo?

Píše se rok 1953...

- PC? Neexistuje
- sálové počítače jsou programovatelná pouze assemblerem
- Backusův tým v IBM navrhuje Fortran

## Jak to pokračuje?

- Fortran 77 přidává strukturované programování
- Fortran 95 přidává pole
- Fortran 2003 přidává OOP a generika
- Fortran 2008 přidává paralelní zpracování

# Dnes?

## Fortran...

- předpovídá počasí
- analyzuje konečné prvky
- počítá fyziku, i částicovou
- testuje superpočítače
- ovládá družice (Voyager I & II)

# Kompilátor

Jaký je nejvhodnější?

- IBM Fortran (VS a XL)
- Intel Fortran
- Cray Fortran
- PGI Fortran
- Oracle Solaris Fortran
- Open64 Fortran (SGI) - GPL
- PGI Fortran (CUDA)
- GNU Fortran - GPL

# Jak začít?

## Definujeme program

```
program MyProgram
  implicit none
  ! kod programu
contains
  ! deklarace a kod subrutin pouzitych v programu
  MyProgram
end program MyProgram
```



## Definujeme proměnné

```
! typ [, parametry] :: jmeno [= hodnota]  
character(len=32) :: slovo  
character :: znak = 'a'  
integer, dimension(1,3) :: vektor = 1
```

## Opakujeme

```
! do cyklus  
integer :: i  
  
do i=1,10  
    ! kod cyklu  
end do
```

## Implikovaný do-loop

```
integer :: j  
write (*,*) (j, j=1,10)
```

## Větvíme

```
if a .eq. 4 then
    ! kod
else if a .lt. 4 then
    ! kod
else
    ! kod
end if
```

.eq. (==), .ne. (!=), .lt. (<), .gt. (>), .le. (<=), .ge. (>=), .and. , .or. ,  
.true. , .false.

## Dělíme kód - funkce

```
fuction twice(i) result(j)
  integer, intent(in) :: i
  integer :: j
  j = 2 * i
end function
```

## Jak se volá?

```
integer :: doubleJ
doubleJ = twice(6)
```

## Dělíme kód - subrutina

```
subroutine twice(j, doubleJ)
  integer, intent(in) :: j
  integer, intent(out) :: doubleJ

  doubleJ = 2 * j
end subroutine twice
```

## Jak se volá?

```
call twice(j, doubleJ)
```

## Interakce s uživatelem

```
write (*,*) "Hello World"  
read (*,*) a  
  
print *, "Hello ", "world"  
read (*,*) a,b,c
```

## data.csv

1,5,3,8  
6,7,9,2

## Jak se přečte?

```
integer, dimension(4,2) :: a = 0  
open(10,file="data.csv", status=old, action=read)  
read (10,*) a
```

## a zápis?

```
open(20,file="res.csv",status=replace,action=write)
do i = 1,row
  write (20,*) (a(j,i), ",", j=1, col-1), a(col,i)
end do
```



## read ještě jednou, přetypování

```
character(len=10) :: numberAsString  
integer :: a = 0  
  
read (numberAsString, *) a
```

## "Dynamická" matice

```
integer, allocatable, dimension(:,:) :: mat
integer :: rows, columns = 0

allocate( mat(columns, rows) )
```

## Užitečné knihovny

- **BLAS** - ATLAS, cuBLAS (CUDA), clBLAS (OpenCL), Intel MKL, SCSL (SGI), HP MLIB,
- **FortCUDA**
- **FortranCL**
- **FGSL** - Fortran  $\Leftrightarrow$  GNU Scientific Library
- **LAPACK**
- **GNUFOR2** - Fortran  $\Leftrightarrow$  Gnuplot
- **g2** - 2D grafická knihovna
- **ODEPACK**

Otázky?

{CSG} {=}

Díky za pozornost

{CSG} {=}