

## Exemple simple cu fapte si reguli

### Exercitiu:

Sa se afiseze bunicul, folosind faptele de la exercitiul anterior. (tati.txt)

Predicatul il vom introduce direct in **main.pro**:

```
implement main
open core

class facts - fapteTati
    tata : (string Tata , string Copil ).

class predicates
    bunic : (string Bunic, string Copil ) nondeterm anyflow.

clauses
    bunic( Bunic , Copil ) :- tata( Tata, Copil ) , tata(Bunic, Tata ).

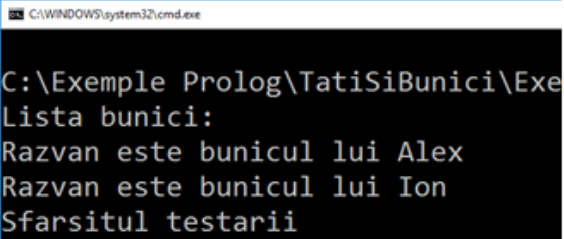
clauses
    run():-
        console::init(),
        file::consult("../tati.txt", fapteTati),

        stdIO::write("Lista bunici:\n"),
        bunic(X, Y),
        stdIO::write(Y, " este bunicul lui ", X, "\n"),
        fail.

    run():-
        stdIO::write("Sfarsitul testarii\n").
end implement main

goal
    mainExe::run(main::run).
```

La rulare avem:



```
C:\WINDOWS\system32\cmd.exe
C:\Exemple Prolog\TatiSiBunici\Exe
Lista bunici:
Razvan este bunicul lui Alex
Razvan este bunicul lui Ion
Sfarsitul testarii
```

## Genuri – simplu

### Exercitiu:

Sa se afiseze genul fiecarei persoane

```
implement main
  open core

  class facts - fapteFamilia
    persoana : (string Nume, string Gen).

  clauses

    run():-
      console::init(),
      file::consult("../familia.txt", fapteFamilia),

      stdIO::write("Lista persoane:\n"),
      persoana(Persoana, Gen),
      stdIO::write(Persoana, " are genul ", Gen, "\n"),
      fail.

    run():-
      stdIO::write("-----\n"),
      stdIO::write("Gen feminin:\n"),

      persoana(Persoana, "fata"),
      stdIO::write(Persoana, " , "),
      fail.

    run():-
      stdIO::write("\n-----\n"),
      stdIO::write("Gen masculin:\n"),
      persoana(Persoana, "baiat"),
      stdIO::write(Persoana, " , "),
      fail.

    run():-
      stdIO::write("\nSfarsitul testarii\n").
end implement main

goal
  mainExe::run(main::run).
```

Avem fisierul cu fapte, **familia.txt**:

clauses

```
persoana("Gabriel","baiat").
persoana("Ion","baiat").
persoana("Adriana","fata").
persoana("Monica","fata").
persoana("George","barbat").
```

```
Lista persoane:
Gabriel are genul baiat
Ion are genul baiat
Adriana are genul fata
Monica are genul fata
George are genul barbat
```

-----

Gen feminin:

Adriana , Monica ,

-----

Gen masculin:

Gabriel, Ion,  
Sfarsitul testarii

### ▲ Genuri – utilizand un domeniu

#### Exercitiu:

Sa se afiseze genul fiecarei persoane folosind domeniile:

implement main

open core

domains

gen = fata(); baiat().

class facts - fapteFamilia

persoana : (string Nume, gen Gen).

clauses

run():-

console::init(),

file::consult("../familia.txt", fapteFamilia),

stdIO::write("Lista persoane:\n"),

persoana(Persoana, Gen),

stdIO::write(Persoana, " are genul ", Gen, "\n"),

fail.

run():-

stdIO::write("-----\n"),

stdIO::write("Gen feminin:\n"),

persoana(Persoana, fata),

stdIO::write(Persoana, " , "),

fail.

```

run():-
  stdIO::write("\n-----\n"),
  stdIO::write("Gen masculin:\n"),
  persoana(Persoana, baiat),
  stdIO::write(Persoana, " "),
  fail.

```

```

run():-
  stdIO::write("\nSfarsitul testarii\n").
end implement main

```

```

goal
mainExe::run(main::run).

```

Fisierul **familia.txt**:

```

clauses
  persoana("Gabriel", baiat).
  persoana("Ion", baiat).
  persoana("Adriana", fata).
  persoana("Monica", fata).
  persoana("George", baiat).

```

```

Lista persoane:
Gabriel are genul baiat
Ion are genul baiat
Adriana are genul fata
Monica are genul fata
George are genul barbat
-----
Gen feminin:
Adriana , Monica ,
-----
Gen masculin:
Gabriel, Ion,
Sfarsitul testarii

```