

Variabile globale vs. variabile locale

```
use strict;
my $var_globala = 2;
my $var_locala = 4;

sub procedura
{
    my $var_locala += 6;
    $var_globala += 10;
}

print '$var_globala'," are valoarea $var_globala\n";
print '$var_locala'," are valoarea $var_locala\n";
print "\nSe executa procedura care modifica valoarea variabilelor\n";
print "Dupa aceasta, variabilele au urmatoarele valori:\n\n";

procedura;

print '$var_globala'," are valoarea $var_globala\n";
print '$var_locala'," are valoarea $var_locala\n";

use strict;
my $var_globala = 2;
my $var_locala = 4;

sub procedura
{
    my $var_locala += 6;
    my $var_globala += 10;
}

print '$var_globala'," are valoarea $var_globala\n";
print '$var_locala'," are valoarea $var_locala\n";
print "\nSe executa procedura care modifica valoarea variabilelor\n";
print "Dupa aceasta, variabilele au urmatoarele valori:\n\n";

procedura;

print '$var_globala'," are valoarea $var_globala\n";
print '$var_locala'," are valoarea $var_locala\n";
```

RECOMANDARE

Ori de câte ori este posibil, este bine de evitat utilizarea numelor identice pentru variabile diferite declarate în diverse blocuri ale programului.

Commented [A1]: În acest exemplu s-au introdus în blocul principal două variabile. Identificatorul `my` precizează că aceste variabile aparțin blocului principal al programului. Din acest punct de vedere ele sunt vizibile atât în programul principal cât și în oricare sub-program (subrutină) care le poate accesa și altera valoarea. Pentru a putea demonstra ușor acest aspect, în sub-procedură am accesat variabila `$var_globala` în cadrul subrutinei. În același timp variabila `$var_locala` este și în blocul principal și în subrutină, cu același nume! ATENȚIE! Cum își dă seama programul că este vorba de două variabile diferite? Modulul `strict` obligă declararea tuturor variabilelor în variantă locală. Indicativul `my` precizează acest aspect. Variabila `my $var_locala` este inițializată cu 4 în blocul principal, iar variabilei `my $var_locala` din subrutină i se adaugă valoarea 6. Faptul că ambele au `my` la inițializare indică faptul că este vorba de variabile diferite. Aceasta din urmă fiind declarată în subrutină, nu este vizibilă în programul principal. Pe de altă parte `$var_globala` NU! este inițializată cu `my`, deci! este vorba de aceeași variabilă inițializată în blocul principal. Conform restricțiilor impuse de modulul `strict`, declararea unei variabile se face prin utilizarea lui `my`. De fiecare dată când numele unei variabile este precedat de `my` atunci aceasta trebuie să fie o variabilă nouă. În același bloc de program nu se pot declara două variabile identice, însă acest lucru este posibil în blocuri diferite, așa cum este arătat în acest exemplu. Cele două variabile `$var_locala` sunt declarate în două blocuri diferite ale programului, ele au nume identice dar reprezintă locații diferite de memorie, de aceea conținutul lor nu este același.

Commented [A2]: Acest exemplu este identic cu cel anterior cu excepția faptului că în subrutină a fost introdus `my` în fața `$var_globala`. Astfel aceasta devine o variabilă locală nouă în subrutină și este diferită față de cea declarată în blocul principal. Din această cauză, rezultatele pe care le produc cele două programe sunt diferite.