

Declarații. Compilare separată

30 mai 2006

Variabile globale și locale

Variabilele pot fi declarate și în afara funcțiilor.

Dacă în declarația de variabile nu apar alți specificatori înainte de tip:

Variabile globale

= o variabilă declarată în afara oricărei funcții

– are spațiu de memorie alocat pe întreaga execuție a programului

– e inițializată o singură dată (cu valoarea dată explicit în declarație, sau implicit cu zero)

– e vizibilă în întreg textul programului începând cu declarația ei

Variabile locale (interne)

= o variabilă declarată în interiorul unui bloc (inclusiv de funcție)

– există doar atât timp cât programul execută blocul respectiv

– sunt inițializate cu valoarea dată la orice intrare în blocul respectiv

(sau au o valoare nedefinită dacă declarația nu specifică inițializare)

– sunt vizibile doar în interiorul blocului respectiv

Domeniul de vizibilitate al identificatorilor

Pt. orice identificator, compilatorul trebuie să-i decidă semnificația
Identificatorii obișnuiți: variabile, tipuri, funcții, constante enumerare
sau un *spațiu de nume* comun (NU: variabilă și funcție cu același nume)

Q1: Un identificator poate fi folosit într-un punct de program ?

R: *Domeniul de vizibilitate* (al unei declarații / al unui identificator)

– domeniu de vizibilitate la nivel de *fișier* (*file scope*)

– pentru identificatori declarați în afara oricărui bloc (oricărei funcții)
din punctul de declarație până la sfârșitul fișierului compilat

– domeniu de vizibilitate la nivel de *bloc* (*block scope*)

– pentru identificatori declarați într-un bloc { } (corp de funcție,
instrucțiune compusă) și pentru parametrii unei funcții

– din punctul de declarație până la acolada } care închide blocul

Un identificator poate fi *redeclarat* într-un bloc interior și își recapătă
vechea semnificație când blocul ia sfârșit.

Legătura dintre identificatori (linkage)

Q2: Două declarații ale unui identificator se referă la aceeași entitate?

R: Tipul de legătură (*linkage*) al unui identificator (obiect/funcție)

– *extern*: toate declarațiile identificatorului din toate fișierele care

compun un program se referă la același obiect sau funcție

– pentru declarațiile *la nivel de fișier* fără specificator de memorare

sau declarația cu specificatorul *extern* a unui identificator

care nu a fost deja declarat cu tipul de legătură *intern*

– *intern*: toate declarațiile identificatorului din fișierul curent se referă

la același obiect sau funcție; nu se propagă în exteriorul fișierului

pt. declarațiile *la nivel de fișier* cu specificatorul de memorare *static*

– *fără legături* (*no linkage*): fiecare declarație denotă o entitate unică

– pentru declarațiile *la nivel de bloc* fără specificatorul *extern*

Durata de memorare a obiectelor

Q3: Ce timp de viață/dură de memorare are un obiect în program?

R: 3 feluri diferite: *static*, *automatic* și *alocat* (discutat ulterior)

Pe întreaga durată de viață, un obiect are o *adresă constantă*

și își *păstrează ultima valoare* memorată.

Durată de memorare *statică*:

– pentru obiecte declarate cu tipul de legătură *extern* sau *intern*,

sau declarate cu specificatorul de memorare *static*

– timp de viață: *întreaga execuție* a programului.

– obiectul e *inițializat o singură dată*, înainte de lansarea în execuție.

Durată de memorare *automată*: pentru obiecte fără legătură

– timp de viață: de la intrarea în blocul asociat până la încheierea sa

– la fiecare apel recursiv, se crează o nouă instanță a obiectului

– *valoarea inițială: nedeterminată*;

– o eventuală inițializare în declarație e repetată de câte ori e atinsă

Structura programelor din mai multe fișiere

– fiecare fișier poate fi compilat separat în format obiect

– apoi fișierele sunt legate (linkeditate) pentru a crea executabilul

– orice identificator trebuie declarat în fiecare fișier sursă unde e folosit

– o variabilă va fi *definită* (evtl. cu inițializare) într-un singur fișier, și
declarată (evtl. cu specificatorul *extern*) în celelalte

Realizarea propriilor biblioteci (ex. tipuri de date cu funcțiile lor)

– într-un fișier *.h* se *declară* tipurile și funcțiile necesare

– într-un fișier *.c* se *definesc* (implementează) funcțiile

– programul care le folosește va include fișierul *.h* și va fi compilat cu
fișierul *.c* (linkeditat cu fișierul obiect rezultat din el)