

M. Caramihai, © 2020

**PROGRAMAREA
ORIENTATA
OBIECT**

A gandi “orientat obiect”

Evolutia programarii

- Cod masina
- Limbaj asamblare
- Limbaje independente de masina
- Proceduri si functii
- Obiecte
- Servicii

Istoric

- 1960

- 1980 – Byte

 - Smaltalk (1970) – XEROX

 - Simula I ('62 – '65)

 - Simula II ('67)

- Portland, Oregon, USA

- C++ (1980)

Generalitati

- POO a evoluat din *programarea structurata (PS)*
- POO este:
 - O idee revolutionara
 - O evolutie normala
- Aplicabilitate
 - POO: sisteme distribuite
 - PS: fluxuri de date, descompunere de procese

Limbaaj si gandire (1)

- M. Sapir (1956): “Oamenii se gasesc la cheremul unui limbaaj particular care reprezinta mediul de comunicare in cadrul unei societati”
 - ⇒ Limbaajul influenteaza gandirea
- d.e. vocabularul extensiv (eschimosii si zapada)
- **Concluzie:** un limbaaj OO poate simplifica dezvoltarea unei solutii OO

Limbaaj si gandire (2)

- Exemplu: analiza secventei DNA: fie un vector cu N variabile in cadrul careia trebuie determinata o secventa repetitiva de lungime M
- Rezolvare intr'un limbaaj standard:
 $M \times N \times N$ operatii
- Rezolvare intr'un limbaaj OO
 $M \times N \times \log(N)$ operatii
- *Observatie:*
Limbaajul OO NU este mai rapid decat limbaajul structurat

Postulatul lui Church & ipoteza lui Whorf

- **Ipoteza lui Church (1936):** orice rationament pentru care exista o “procedura efectiva” (i.e. are la baza un algoritm) poate fi realizat de o masina *Turing*.
- **Postulatul lui Whorf (1956):** este posibil ca un individ, ce comunica intr'un limbaj, sa nu poata comunica ganduri / idei intr'un alt limbaj

Consecinte

- Pentru **Ipoteza lui Church**

- ➔ Orice limbaj in cadrul caruia poate fisimulata o masina *Turing*, este suficient de performant pentru a permite implementarea oricarui algoritm
- ➔ Toate limbajele de programare sunt identice

- Pentru **Postulatul lui Whorf:**

- ➔ Pot exista idei ce pot sa fie exprimate mai bine intr'un limbaj decat intr'altul

- **Consecinta**

- ➔ George Steiner 1975: orice act de comunicare umana poate fi vazut ca implica o traducere

O noua paradigma

- Programul OO creaza obiecte ce incapsuleaza date si procedure ce vor actiona asupra acelor date
- Un POO este structurat ca o comunitate de agenti / obiecte ce interactioneaza intre ele
- Fiecare obiect produce un serviciu sau realizeaza actiuni
 - ➔ Exemplul florarului

Caracteristicile POO

- O noua metoda de proiectare a aplicatiilor pornind de la o reprezentare naturala
- Programele sunt gandite a avea *componente* ce functioneaza independent, fiecare cu propriile proprietati si functionalitati
- Un sistem este proiectat ca o interconexiune de asemenea *componente*.

Caracteristicile programarii structurate

- Programele sunt gandite strict secvential
- Se utilizeaza *branching conditions* in cazul implementarii deciziilor
- Se utilizeaza functii / proceduri pentru implementarea secventelor *repetabile*.
- **Sinteza:** programele definesc date, dupa care apeleaza subprograme pentru a prelucra acele date

Limitările programării structurate

- În viața reală evenimentele nu sunt întotdeauna secvențiale
- Nu este simplu de găsit soluții secvențiale pentru problemele reale
- Codul aplicației nu poate fi re-utilizat și depanat independent
- Codul nu se poate “adapta” la situațiile în schimbare
- → Abordări OO sunt benefice în cadrul aplicațiilor de dimensiuni mari (>1000 linii de cod)

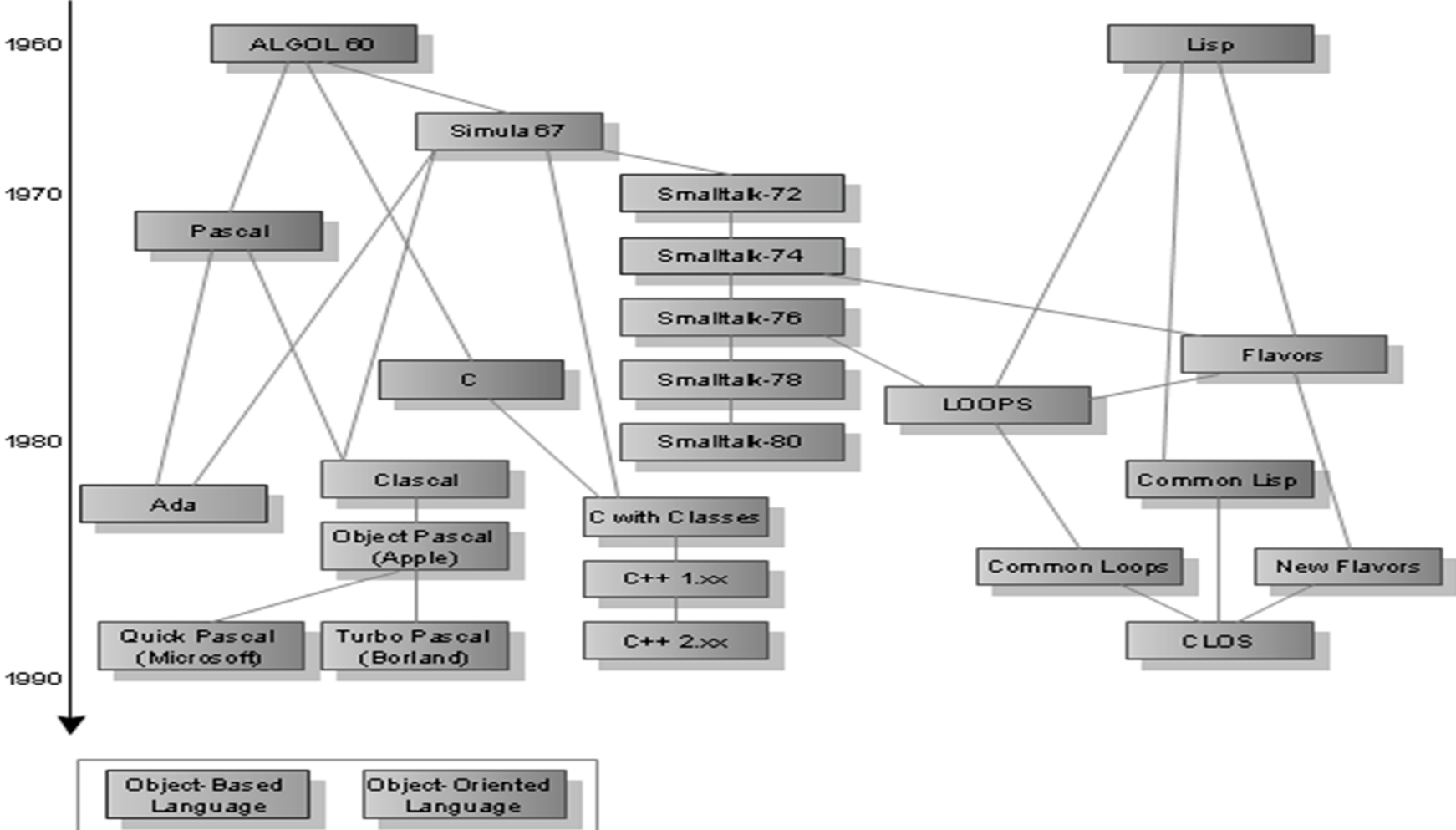
POO – un nou model de gandire

- Actiunile sunt initiate in POO prin transmitere de mesaje de catre un agent (obiect) responsabil pentru o actiune
- Programele sunt gandite ca interactiuni / asocieri de obiecte
- In cadrul POO nu trebuie “anticipat” orice – ci trebuie doar utilizate serviciile obiectelor

Evolutia limbajelor de programare (1)

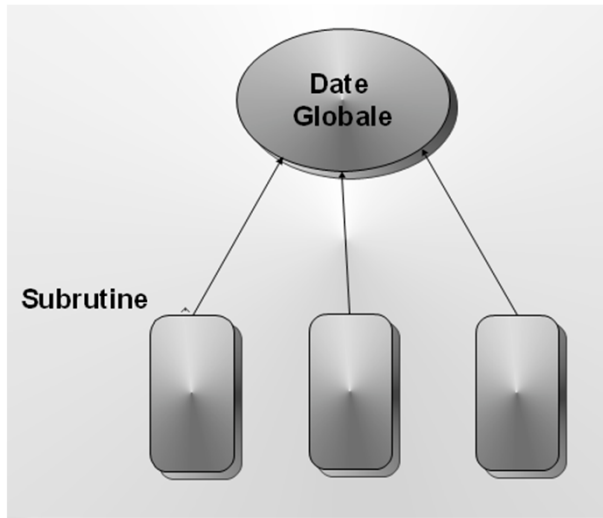
- Generatia I (1954 - 1958) – formule matematice
 - ➔ Fortran I, Algol 58, IPL
- Generatia II (1959 -1961) – algoritmi
 - ➔ Fortran II, Algol, Cobol
- Generatia III (1962 - 1970) – abstractizare date
 - ➔ PL/I, Pascal, Algol 68, C, Fortran77
- Limbaje OO
 - ➔ Smalltalk, C++, Eiffel
- Limbaje actuale: Java, Python, C#

programare (2)

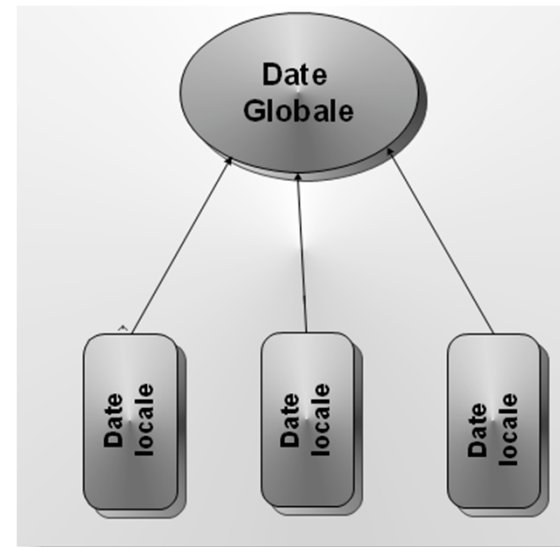


Evolutia limbajelor de programare (3)

Generatia I



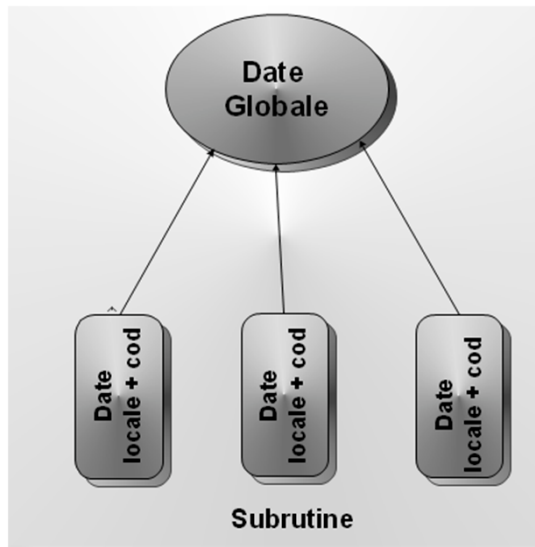
Generatia II



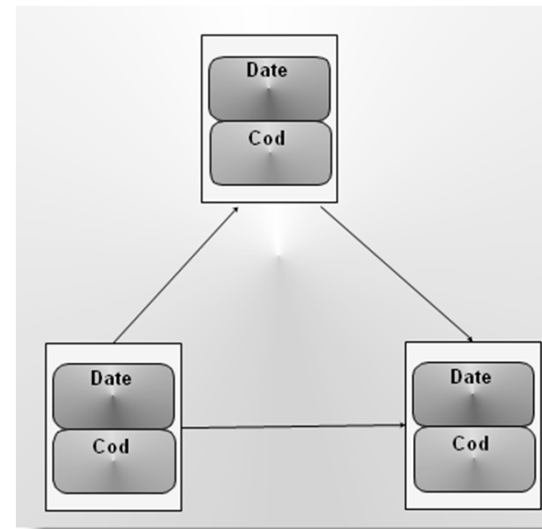
- Mecanisme de “transfer parametrii”
- Dezvoltare structuri de control
- Subprograme

Evolutia limbajelor de programare (4)

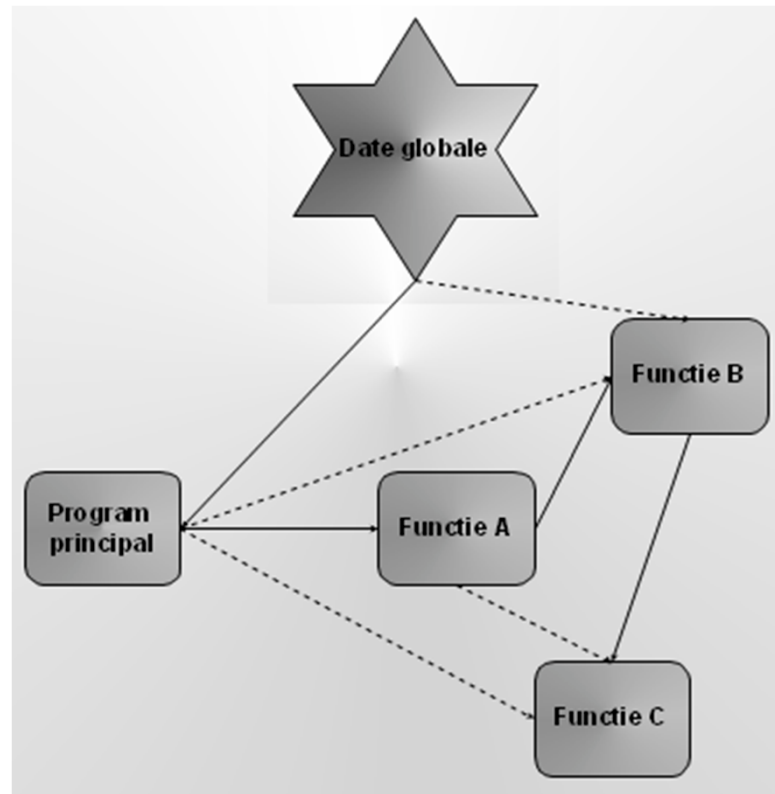
Generatia III



Generatia IV



Programarea structurata



Mesaje (1)

- Mesajele au 3 componente
 - Un obiect caruia ii este adresat
 - Numele unei metode ce trebuie sa fie executate in obiectul destinatie
 - Orice parametru / variabila aferente acelei metode
- Observatie: pentru a interpreta un mesaj, obiectul destinatie trebuie sa aiba o metoda definita corespunzator

Mesaje (2)

- Prin ce difera un mesaj de un apel de procedura?
 - Apel de procedura: este cunoscuta adresa de memorie aferenta codului dorit → salt la adresa;
 - Mesaj: se transmite mesajul catre obiectul destinatie
Mesajul este *interpretat* (i.e. se alege o metoda pentru a raspunde mesajului) de catre receptor; aceasta interpretare are loc in momentul rularii.

Mesaje (3)

- Prin ce difera un mesaj de un apel de procedura?
 - Apelul de procedura este o subrutina ne-asociata cu datele asupra carora actioneaza
 - Metodele (apelate prin mesaje) sunt definite la nivelul obiectelor → sunt asociate datelor incapsulate in obiectul respectiv.
- Observatie: in C++ metoda (de la destinatie) trebuie sa existe, altfel se poate semnala eroare

Avantajele POO

- Analiza si proiectare mai facila
- Reutilizare cod
- O mai buna mentenanta a programelor
- Reducerea codului recurrent
- Mascarea informatiilor
- Robustete sporita
- O mai buna securitate a aplicatiilor

Concluzii ... istorice

- **“I think there is a world market for maybe five computers.”**
 - Thomas Watson, chairman of IBM, 1943
- **“Computers in the future may weigh no more than 1.5 tons.”**
 - Popular Mechanics, 1949
- **“There is no reason anyone would want a computer in their home.”**
 - Ken Olson, president, chairman and founder of Digital Equipment Corp., 1977