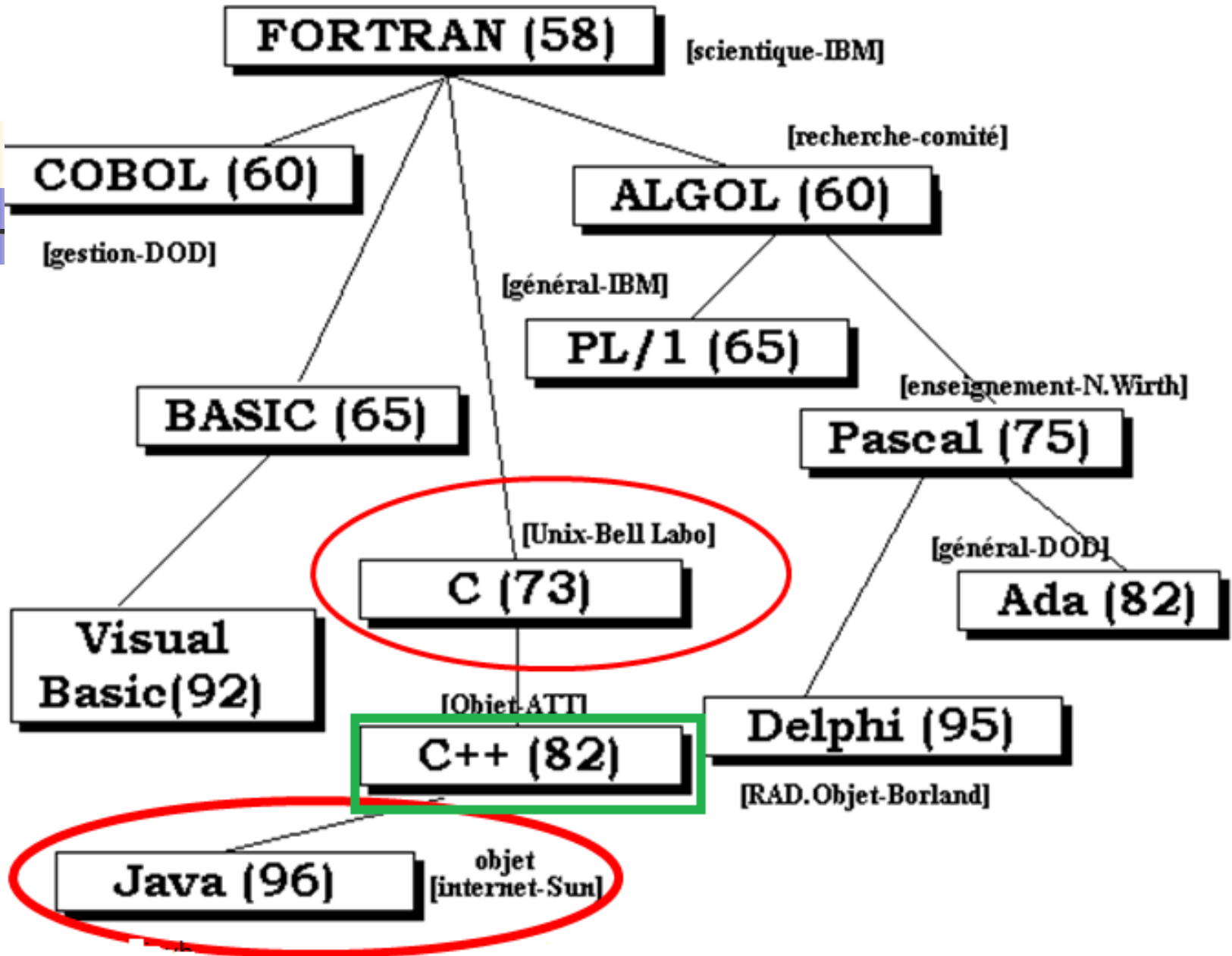
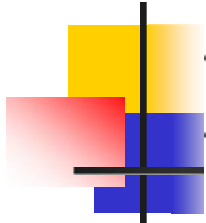


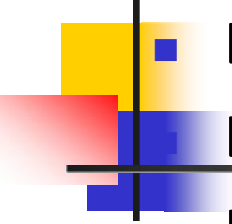


# Programmation Orientée Objet

---

## Concepts de base



- 
- Les années 1950 : **Fortran, Lisp, Cobol, Algol**
  - Les années 1960 : **Apl, Basic, PL/1**
  - Les années 1970 : **Pascal, Prolog, SmallTalk, C, Ada**
  - Les années 1980 : **Dbase, C++, Eiffel, Perl, Tcl/Tk**
  - Les années 1990 : **Java[script], Php, Mysql, VisualBasic**
  - Les années 2000 : **Java, Delphi, Php, Perl, C#**
  - Les années 2010 : **Matlab, Scilab, R, Ruby, Python, Javascript**
  - Les 500 prochaines années !

Il est impossible de prévoir ce que seront les prochaines années... La technologie en général et donc l'informatique et les langages de programmation en particulier évoluent suffisamment vite pour qu'il y ait encore des surprises. Toutefois, on peut remarquer que :

- C++ et Java sont "installés" comme langages de grosses applications,
- Perl, Php, Mysql sont très utilisés pour la programmation Web,
- tous les langages ne fonctionnent pas forcément sur tous les ordinateurs,
- le diction "à objectif précis, langage imposé" reste encore valide,



# Types de programmation

---

- Programmation procédurale : P.P.  
(Pascal, C, etc.)
- Programmation orientée objet :  
P.O.O.(C++, Java, Delphi)
- Programmation déclarative Prolog, SQL, ..
- Programmation de balisage HTML, XML, ..

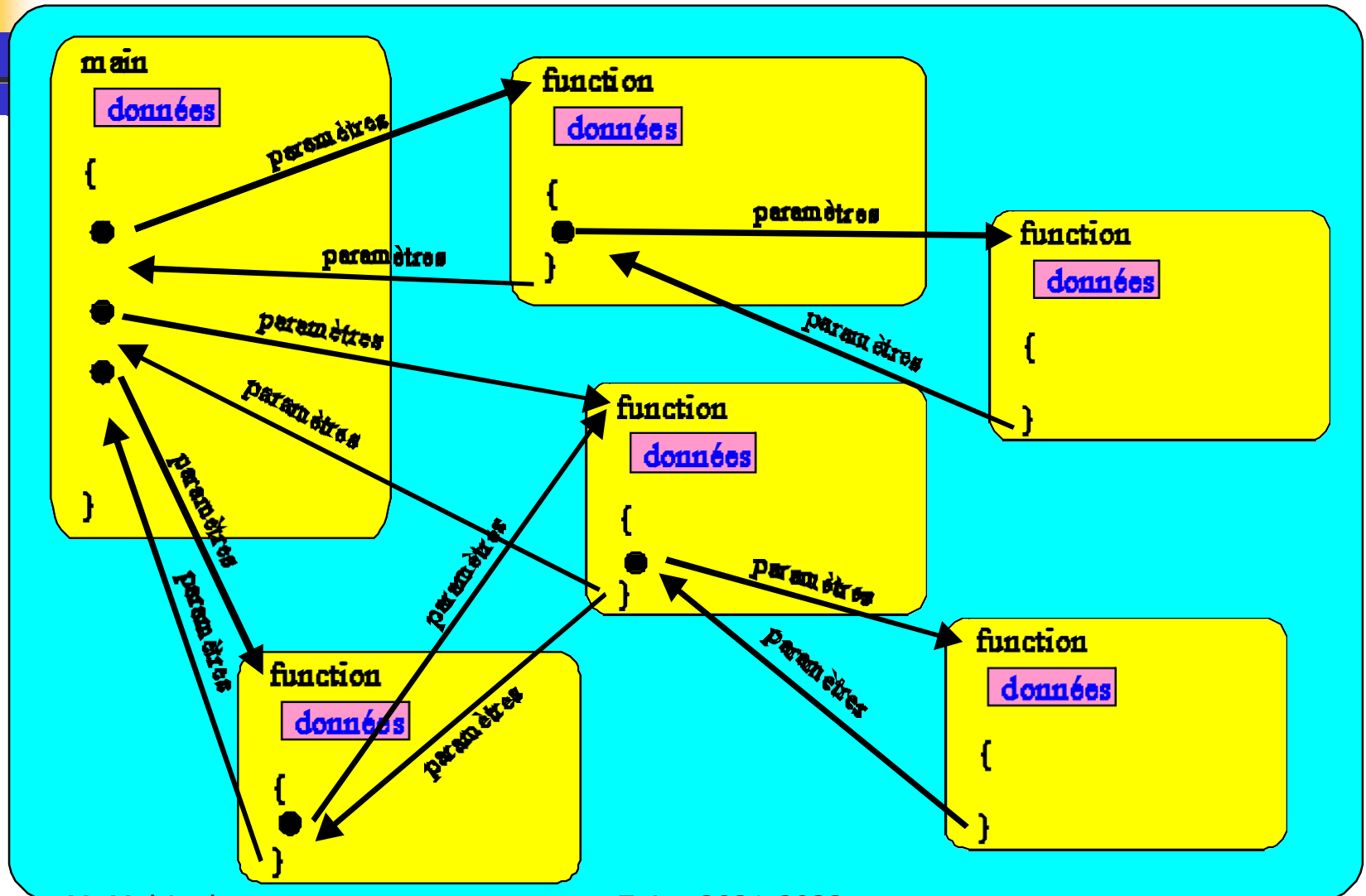


# Programmation procédurale (rappel de C)

---

- Le programme est composé des fonctions
- Les données (variables) sont créées à l'intérieure des fonction ou bien passées comme paramètres
- Il y a un programme principal (main)

# Un programme en C



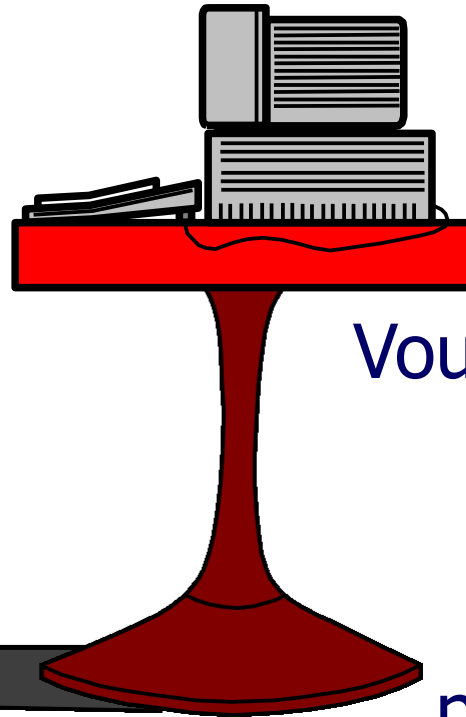


# Limitations

---

- Il n'y a pas de méthode ou de cadre pour bien organiser les fonctions.
- Les modifications d'une fonction entraînent d'autres modifications dans autres fonctions, etc.
- La portée d'une modification est trop grande et difficile à gérer.
- Redondance dans le code (la même chose est codé plusieurs fois)
- Propagation des erreurs – débogage difficile

# Est-ce qu'il faut oublier le C?



**NON!**

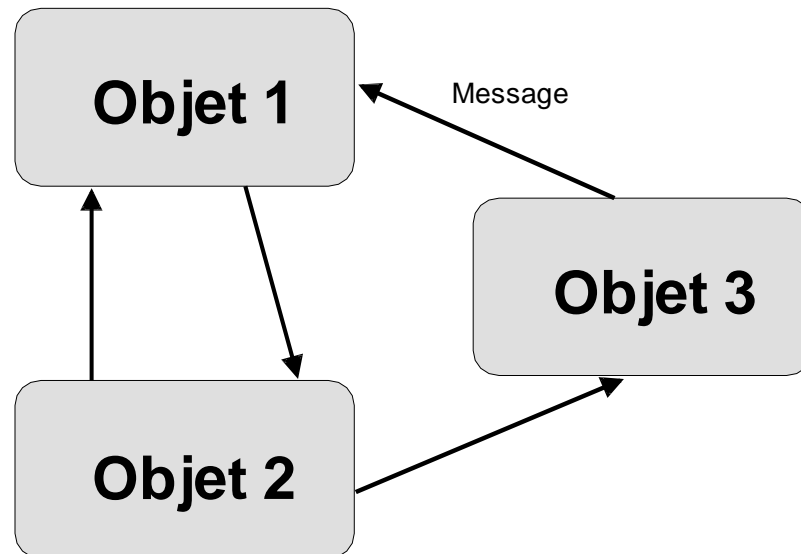
Vous allez avoir le  
choix parmi  
plusieurs  
méthodes de  
programmation



# Programmation orientée objet

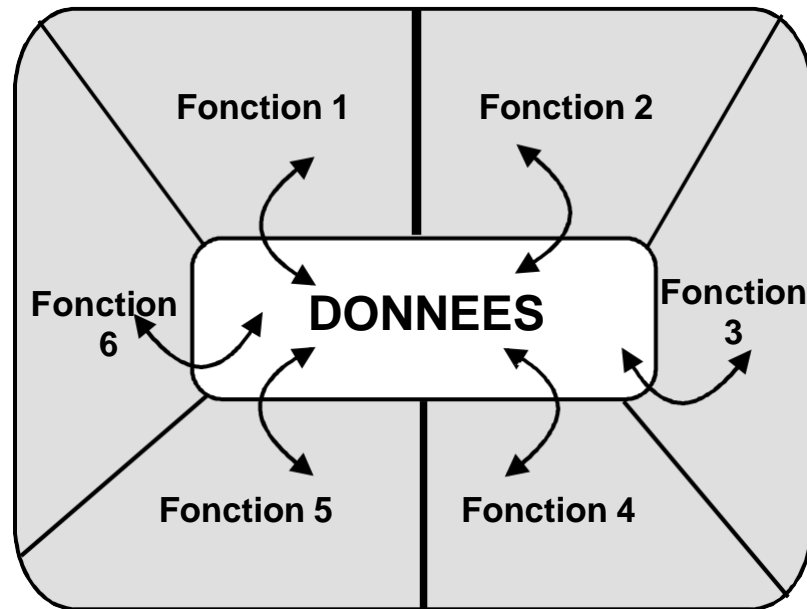
## POO

- Ensemble d'objets autonomes et responsables qui s'entraident pour résoudre un problème final en s'envoyant des messages.



# Qu'est ce qu'un objet ?

- Objet = données + méthodes (fonctions membres)



# POO

## Comment peut on y arriver?

- Introduction des nouvelles (?) notions
  - Objet
  - Classe
  - Instanciation
  - Hiérarchie des classes & héritage
  - Polymorphisme
  - Événement
- On va utiliser ces notions pour introduire le Concept de programmation orientée objet.



# Objet (exemple / fenêtre)

---

- Propriétés d'une fenêtre
  - ouverte/fermée
  - cassée/intacte
  - taille
  - sens d'ouverture
  - type de verre
  - coefficient de réflexion de chaleur
- **Pour une fenêtre concrète, ces propriétés ont des valeurs.**



# Objet (exemple / fenêtre)

---

- Opérations avec une fenêtre donnée
  - ouvrir
  - fermer
  - casser
  - réparer
  - changer la verre



# Objet (exemple / livre)

---

- Propriétés d'un livre dans une bibliothèque
  - état (emprunté / disponible / perdu)
  - date de la fin de l'emprunt
  - titre
  - auteur
  - nombre de pages
- **Pour un livre donné, ces propriétés ont des valeurs!**



# Objet (exemple / livre)

---

- Opérations sur un livre d'une bibliothèque
  - emprunter
  - rendre
  - perdre
  - voler



# Objet (définition)

---

- Un objet est un ensemble des propriétés ayant des valeurs et des actions (opérations ou méthodes) agissant sur les valeurs de ces propriétés.
- Remarque: c'est la tâche qui nous dira quelles sont les propriétés et les méthodes qu'on doit prendre en compte dans le programme.





# Classes

---

- La classification des choses se fait par abstraction.
- Il faut trouver les caractéristiques communes des éléments de la classe. En effet, ces sont ses caractéristiques qui définissent une classe.



# Classe des objets

- Les objets ayant des mêmes propriétés et les mêmes méthodes peuvent être mis dans une classe.



- **Une classe sera définie par les propriétés et les méthodes sur ses éléments.**



# Classe des livres

## ■ propriétés

- état (emprunté / disponible / perdu)
- date de la fin de l'emprunt
- titre
- auteur
- nombre de pages

## ■ méthodes

- emprunter
- rendre
- perdre
- voler



# Classe des fenêtres

## ■ propriétés

- état d'ouverture (ouverte/fermée)
- état (cassée/intacte)
- taille
- sens d'ouverture
- type de verre
- coef de réflexion de chaleur

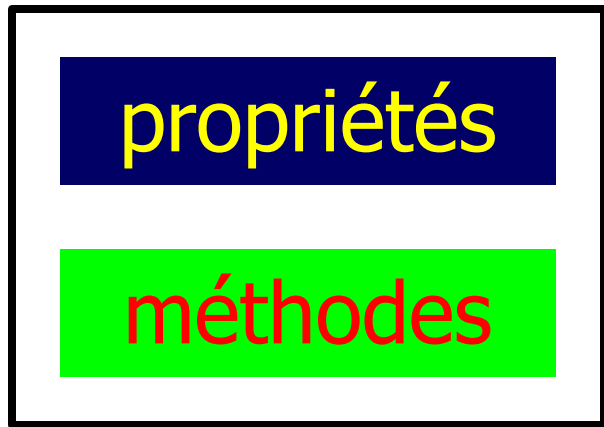
## ■ méthodes

- ouvrir
- fermer
- casser
- réparer
- changer la verre



# Classe

- Une classe est un ensemble des propriétés et des méthodes.



Dans le programme

—> variables

—> procédures (code)

## ENCAPSULATION

# Une classe dans un programme



---

- Une classe est une collection de données et des méthodes sur ces données.
- Une classe est une **encapsulation** des données et du code.
- La notion de l'encapsulation est la création des classes
  - donne une organisation naturelle des fonctions (méthodes)
  - permet d'effectuer la modification locale du code



# Masquage des informations

---

## ■ **Encapsulation des données**

- L'accès aux données des objets est réglementé
  - Données privées → accès uniquement par les fonctions membres
  - Données publiques → accès direct par l'instance de l'objet
- Conséquences
  - Un objet n'est vu que par ses spécifications
  - Une modification interne est sans effet pour le fonctionnement général du programme
  - Meilleure réutilisation de l'objet



# Un programme orienté objet

---

- Modélisation du domaine à l'aide des classes
- définition des classes
- Création des instances (peut être dynamique)
- Messages entre les objets (appel des méthodes)





# Interface vers l'utilisateur

---

- L'interface vers l'utilisateur est une collection des objets (boutons, champs de texte, menu déroulant, etc.)
- L'utilisateur dispose des actions sur les éléments de l'interface (sélectionner, cliquer, double cliquer, etc.)
- Une action correspond à un événement qui déclenche l'exécution d'une méthode.

# Un programme orienté objet

