

Programmation « orientée système »

APPROFONDISSEMENTS SEMAINE 5

Jean-Cédric Chappelier

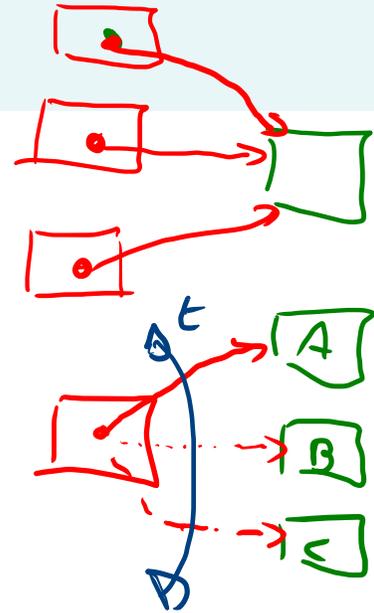
Laboratoire d'Intelligence Artificielle
Faculté I&C

Rappels des points clés

cette semaine

1) références

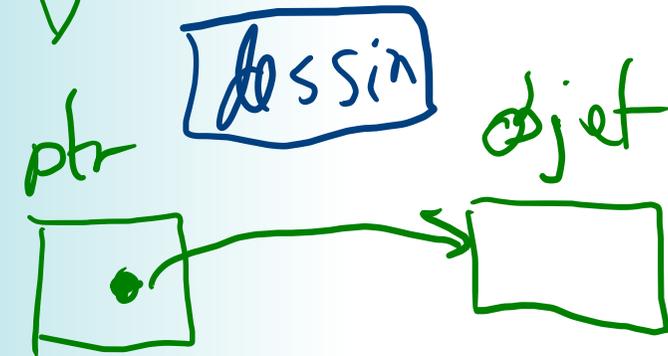
2) genericité



▶ les 3 cas d'utilisation des pointeurs

▶ toujours savoir « dans quel monde » on travaille (valeurs/adresses)
☞ ne pas confondre pointeur et objet pointé

▶ toujours initialiser vos pointeurs (comme toute variable, d'ailleurs !)



semaines suivantes

3) albc. dyn. : durée de vie > temps (partie)

FONCTIONS

affiche_tableau_personnes(
tab, &affiche_une_pers_FR);

↑

{

void (*f)(Personne...);

(*f)(p);

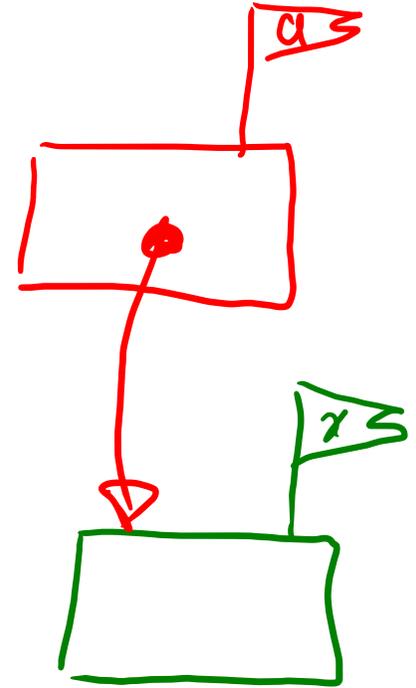
}

Etudes de cas

- ▶ reprendre en détail l'un ou l'autre exemple du cours ?
- ▶ (commentaire rapide sur les passages d'argument aux fonctions)
- ▶ dépendances cycliques : voiture avec chauffeur

void lit_double (double * a); ~~*~~

lit_double(&x);



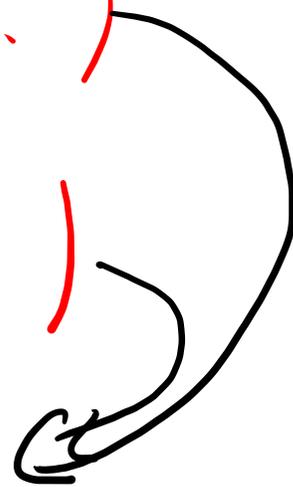
int tab [~~N~~];

f(tab, ...)

void f(int tab[N], ...)

void f(int tab[], ...)

void f(int *tab, ...)



Const double

on ne peut pas
modifier la
VALEUR pointée

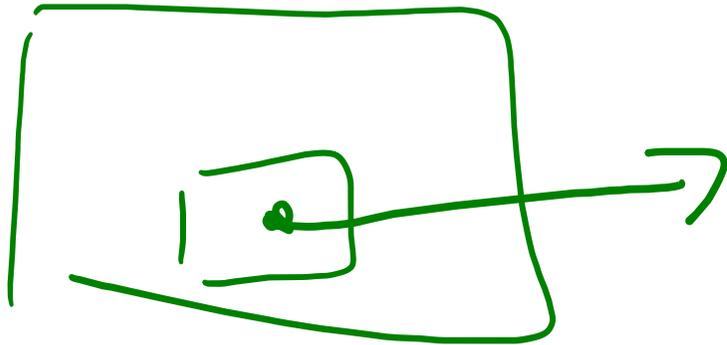
* const ptr =

On pointe
tjs au même
endroit

{ & x
malloc
(NULL) ;

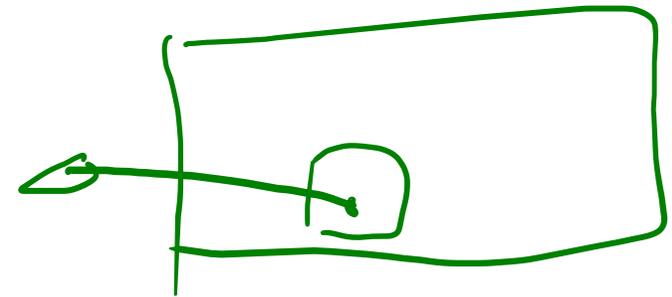
Personne

ont une
voiture
(propriétaire)

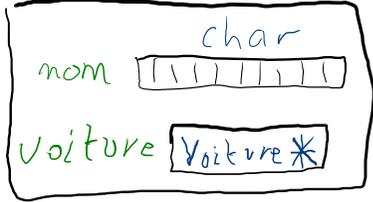


Voiture

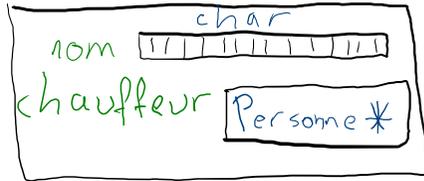
ont un
chauffeur



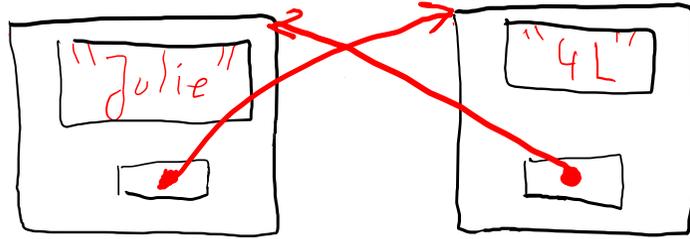
Personne:



Voiture:



Julie conduit sa 4L ^(proprio)



32

Pierre conduit la 4L de Julie

