



Gestion dynamique de la mémoire en C#

La bibliothèque alloc.h contient des opérateurs et fonctions qui permettent de gérer dynamiquement la mémoire.

Allouer une zone mémoire à un objet : l'opérateur new

nom_pointeur = new type_donnée ;

pointeur est le nom du pointeur qui contient l'adresse du bloc mémoire alloué
type_donnée désigne le type de l'objet qui doit être enregistré.

VAR i : POINTEUR D'ENTIER

int *i; // pointeur sur un entier

ALLOUER (i)

i = new int ;

ou

int *i = new int ;

TYPE NOEUD = ENREGISTREMENT

info : ENTIER

suivant : POINTEUR DE NOEUD

FIN-ENREGISTREMENT

Struct noeud

{

int info;

noeud *suivant;

};

VAR p : POINTEUR DE Nœud

noeud *p;

ALLOUER (p)

p = new noeud;

Accès à l'objet pointé par un pointeur

*nom_pointeur

i^

*i;

VAR p : POINTEUR DE Nœud

noeud *p ;

p^.info

(*p).info ;



Libérer une zone mémoire : l'opérateur delete	
<p>L'espace mémoire qui est alloué à l'aide de l'opérateur new demeure occupé tant qu'il n'a pas été explicitement libéré avec l'opérateur delete. Dans ce cas, un objet qui se trouvait dans cette zone mémoire est considéré comme étant effacé et sa valeur ainsi que l'objet lui-même ne sont désormais plus définis.</p> <p style="text-align: center;">delete nom_pointeur;</p> <p>nom_pointeur représente l'adresse d'une zone mémoire allouée avec et seulement avec new.</p> <p>La tentative d'effacer un objet dont la mémoire n'a pas été allouée avec new entraîne une erreur.</p> <p>La tentative d'effacer un objet qui a déjà été effacé se solde également par une erreur.</p>	
LIBERER (i)	delete i;
LIBERER (p)	delete p;
Le pointeur NULL	
i ← NIL	i = NULL;
Passage par valeur d'un pointeur	
PROCEDURE nom ((E) p : POINTEUR DE NOEUD)	void nom (noeud *p);
Passage par référence d'un pointeur	
PROCEDURE nom ((S) p : POINTEUR DE NOEUD)	void nom (noeud * &p);