

Chapitre 12 – Les pointeurs

Variable dynamique : peut être créée ou détruite au cours de l'exécution du bloc dans lequel elle est déclarée.

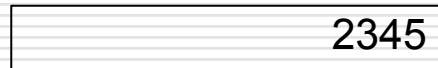
L'espace mémoire rendu libre peut être récupéré et l'accès à la valeur se fait par un pointeur.

Définition

- Un pointeur est une variable dont les valeurs sont des adresses. C'est une variable statique p appelé *variable pointeur*. Par contre, dans la plupart des cas, la variable pointée par p est dynamique.
- Etant donné un type quelconque T appelé *type de base*, une variable P de type pointeur est une variable scalaire qui peut prendre comme valeurs les adresses des variables de type T . Il existe donc autant de types pointeurs que de types de base.

$p = 43$

43



le type de base est Entier

- **Déclaration**
`Var p : ^Entier`

- **Valeur particulière**

Quel que soit son type de base, un pointeur peut prendre une valeur particulière *nil* (le pointeur ne pointe sur rien). Si $p = \text{nil}$ alors $p^{\wedge}$ renvoie toujours une erreur.

Opérations

Sorte : pointeur

Utilise : typeDeBase

Operations :

Allouer (TypeDeBase) $\rightarrow$ Pointeur {constructeur}

Recupérer (Pointeur) $\rightarrow$ {observateur}

$\wedge$ Pointeur $\rightarrow$ TypeDeBase {observateur}

« Allouer » réserve en mémoire principale un espace assez grand pour contenir un élément du type de base et renvoie un pointeur sur cet élément. La valeur de l'élément alloué est non définie (non nulle).

« Récupérer » rend la mémoire au système d'exploitation. p est alors inutilisable. $p^\wedge$ permet d'accéder à l'élément pointé par p .

Exemple

Var p,q : ^entier

p←allouer(entier)

q←allouer(entier)

p^←3

q^←p^

écrire(p^=q^) {VRAI}

écrire(p=q) {FAUX}

q←p

écrire(p^=q^) {VRAI}

écrire(p=q) {VRAI}

récupérer(p)

écrire(q^) {ERREUR ! ! ! car q est aussi récupéré}

{ERREUR ! ! ! Plus de moyen de récupérer la case pointée par q}