Objets et Classes

Idée très naive

Un objet est un enregistrement dont certains champs sont des pointeurs vers le code de procédures (méthodes)

(Une méthode est une procédure (ou fonction) particulière parce qu'elle a un paramètre caché, l'objet)

Inacceptable parce qu'il y a énormément de duplication inutile de pointeurs vers les méthodes

Traiter les appels de méthodes comme les autres appels?

C'est-à-dire compiler un appel comme un appel d'une procédure normale.

Quand une méthode d'une classe C est appelée le compilateur sait de quelle procédure il s'agit?

Pas toujours! dans le cas d'héritage avec possibilité de (re)définir une méthode connue au niveau supérieur, un pointeur qui est (syntaxiquement/statiquement) un pointeur vers la classe initiale peut être (dynamiquement) un pointeur vers un objet de la sous-classe. C'est la méthode de la sous-classe qu'il faut appeler.

Y compris le pointeur implicite qui est le paramètre caché.

Solution partielle: un tableau des méthodes

Chaque objet contient (dans son premier mot) un pointeur vers un tableau (constant) qui dépend du type de l'objet. Ce tableau contient les pointeurs vers les méthodes valables pour ce type. Le tableau d'une sous-classe contient des pointeurs vers les méthodes de la classe mère sauf quand ces méthodes ont une définition locale dans la sous-classe.

Coût supplémentaire mais pas très important:
espace: un petit tableau pour chaque classe et un pointeur pour chaque objet
temps: un niveau d'indirection pour chaque appel d'une méthode (sauf optimisation...)

Mais ce système ne marche toujours pas dans le cas d'héritage multiple!
ni pour les méthodes ni pour les champs

Une solution générale

Pour chaque méthode décider une position dans le tableau qui sera valable pour toutes les classes:

inconvénients

- des trous dans les tableaux
- impossible de générer du code indépendamment pour toutes les classes
- il faudra faire la même chose pour les champs: un niveau d'indirection
- algorithme nécessaire pour décider les positions

Une deuxième solution générale

Un objet et le tableau d'une classe sont divisé en sections, une pour chaque classe ancêtre

la section d'un objet contient un pointeur vers la section correspondante du tableau

Un pointeur vers un objet pointe vers la section du type du pointeur

Le tableau contient, pour chaque méthode, l'adresse de son code et un δ, qui sert à mettre à jour un pointeur