



Forum: WD 8.x

Topic: POO windev

Subject: Re: POO windev

Posté par: R&B

Contribution le : 7/6/2004 12:43:09

Bonjour Deedier.

[modérateur]Inutile de démultiplier vos posts dans tous les forums même si le sujet est générique, cela n'augmente pas les chances de réponses...[/modérateur]

Vous devez en premier lieu analyser avec précision la hiérarchie de vos composants (forme, ligne, cercles etc) afin d'éviter les bouclages d'héritage qui poseront sans doute des problèmes.

Je vous propose un objet dessin qui sera un conteneur et dont la fonction sera précisément de gérer les composants du dessin (instances des objets formes et cie).

Pour cela, je vous invite à étudier en détail les chapitres suivants de WinDev :

- Les objets dynamiques (allocation, libération etc)
- les tableaux dynamiques

ces deux éléments seront le noyau technique de votre classe :

```
cDessin est une classe
nDess est un entier = 0 // nombre d'objets
tDess est un tableau de
nDess objets dynamiques // tableau des objets
... PUBLIC CONSTANT
_FORME est un
entier = 0
_POLYGON est un entier = 1
_CERCLE est un entier = 2
... FIN
//
nouvel objet dans le dessin
PROCEDURE Ajoute(pnType::_FORME) oo est un objet dynamique //
ajoute une ligne au tableau
nDess=Dimension(tDess,nDess+1) // affecte le nouvel objet
SELECTION
pnType
CAS ::_FORME:
tdess[nDess] = allouer une forme(<params>)
CAS ::_POLYGON:
tdess[nDess] = allouer un polygone(<params>)
... FIN
RETOURNER nDess // nouvel objet dans
le dessin
PROCEDURE Ajoute(pnType::_FORME) oo est un objet dynamique // ajoute une ligne au
tableau
nDess=Dimension(tDess,nDess+1) // affecte le nouvel objet
SELECTION pnType
CAS
::_FORME:
tdess[nDess] = allouer une forme(<params>)
CAS ::_POLYGON:
tdess[nDess]
= allouer un polygone(<params>)
... FIN
RETOURNER nDess // libération des ressources
PROCEDURE Destructeur i est un entier
POUR i=1 à nDess
liberer tdess[i]
FIN Libérer tdess
```

Utilisation

```
oDess est un cDessin
nForme=oDess:Ajoute(cDessin::_FORME)
nPolygone=oDess:Ajoute(cDessin::_POLYGON) // manipule la forme
oDess:tdess[nForme]:
méthode_de_forme(<param>) // manipule le polygone
oDess:tdess[nPolygone]:
méthode_de_polygone(<param>)
```

Note : cDessin peut accessoirement servir à gérer les options de recouvrement en donnant une 'altitude' aux objets et donc un ordre d'affichage : le plus élevé étant affiché en dernier. Je vous laisse imaginer les extensions techniques possibles.

Merci éventuellement de nous retourner fruit de vos travaux...