

Méthode

UNIVERSITÉ 4D
ARLES 15, 16, 17 mai 2000

Programmation Générique et Modulaire
Niveau 2
Généricité et encapsulation

Thibaud Arguillère, Pierre Nayrolles
Matthieu Baudoux, Jean-Pierre Milliet

4D en profondeur
Méthode
Programmation
Communication
Déploiement
Interface
Internet
Plug-in
Questions réponses
Autour de 4D

4D 

Méthode

UNIVERSITÉ 4D
ARLES 15, 16, 17 mai 2000

Programmation Générique et Modulaire

Plan de l'atelier

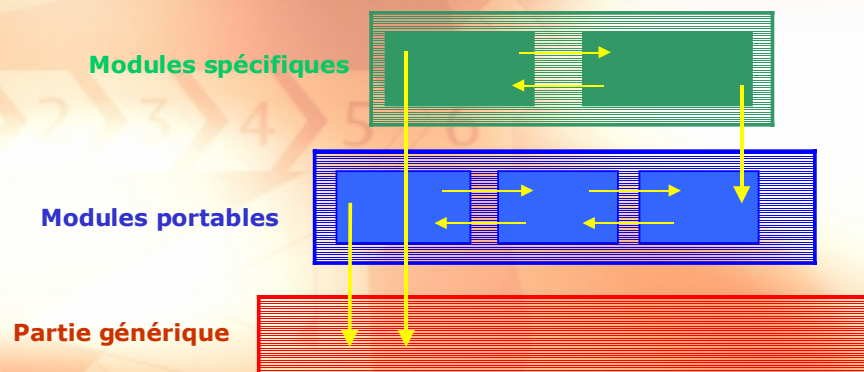
- Petit débat sur la nomenclature
- Niveaux de généricité, encapsulation
- Génériques complexes
- Exemple de module : iCp
- Conclusion

Nomenclature - petit débat

Niveaux de généricité Encapsulation

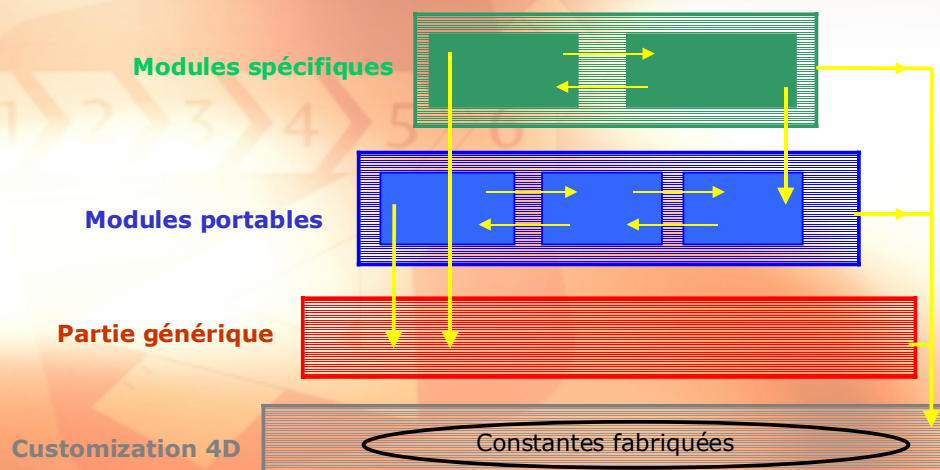
Méthode
Programmation Générique et Modulaire

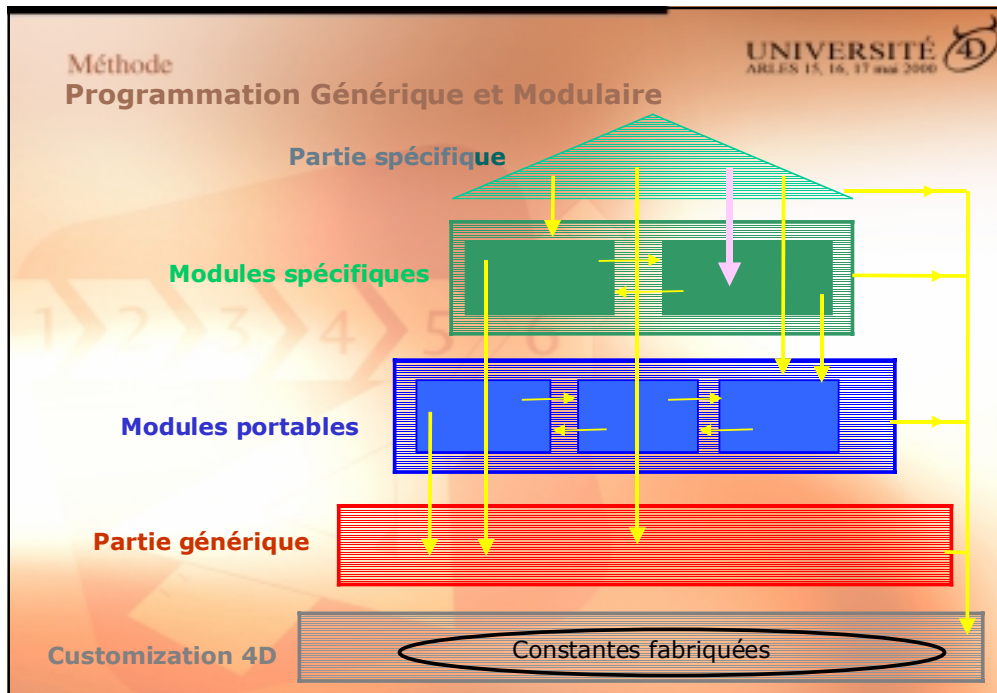
UNIVERSITÉ 4D
ARLES 15, 16, 17 mai 2000



Méthode
Programmation Générique et Modulaire

UNIVERSITÉ 4D
ARLES 15, 16, 17 mai 2000





Méthode
Programmation Générique et Modulaire

UNIVERSITÉ 4D
ARLES 15, 16, 17 mai 2000

Encapsulation

- Encapsulation notion venant de la POO

Encapsulation

- Encapsulation notion venant de la POO
- L 'encapsulation de M. Jourdain

Encapsulation

- Encapsulation notion venant de la POO
- L 'encapsulation de M. Jourdain
- Une seule façon d 'encapsuler dans 4D : L 'hygiène de programmation

Encapsulation

- Encapsulation notion venant de la POO
- L 'encapsulation de M. Jourdain
- Une seule façon d 'encapsuler dans 4D : L 'hygiène de programmation
- Notion de boîte noire

Encapsulation

- L 'accès aux informations contenues dans la boîte noire :

Encapsulation

- L'accès aux informations contenues dans la boîte noire :
« Points d'entrée »

Encapsulation

- L'accès aux informations contenues dans la boîte noire :
« Points d'entrée »
- La customization de la boîte noire :

Encapsulation

- L'accès aux informations contenues dans la boîte noire :
« Points d'entrée »
- La customization de la boîte noire :
« Points de sortie »

Génériques complexes

Génériques complexes

- Préambule : Toujours l'encapsulation

1 > 2 > 3 > 4 > 5 > 6

Génériques complexes

- Préambule : Toujours l'encapsulation

1 > 2 > 3 > 4 > 5 > 6

- 2 exemples de génériques qui encapsulent des objets :
 - t_Intersection
 - yβB_ExecuteFormula

Génériques complexes

T_Intersection

1 2 3 4 5 6

Génériques complexes

T_Intersection

- Générique wrappante de INTERSECTION, permettant de manipuler des ensembles de portée différentes

Génériques complexes

T_Intersection

- Générique wrappante de INTERSECTION, permettant de manipuler des ensembles de portée différentes
- Encapsule un ensemble local et un ensemble process, utilisés uniquement dans la méthode, détruits à la fin

Génériques complexes

yβB_ExecuteFormula

Génériques complexes

yBB_ExecuteFormula

- Générique wrappante de EXECUTER appliquée à une fonction renvoyant un booléen :
\$var:=yBB_ExecuteFormula(« Vrai »)
Plutôt que EXECUTER(« booleenGlobal:=Vrai »)

Génériques complexes

yBB_ExecuteFormula

- Générique wrappante de EXECUTER appliquée à une fonction renvoyant un booléen :
\$var:=yBB_ExecuteFormula(« Vrai »)
Plutôt que EXECUTER(« booleenGlobal:=Vrai »)
- Encapsule une variable process, détruite en sortie.

Développer un module : iCp

Module iCp

- But du module

Module iCp

- But du module

- Nomenclature : i - Cp

Module iCp

- Principe de fonctionnement : le compactage réorganise les N° d'enregistrements.

Module iCp

- Principe de fonctionnement : le compactage réorganise les N° d'enregistrements.
- La ruse du siècle : cf le code

Module iCp

- Principe de fonctionnement : le compactage réorganise les N° d'enregistrements.
- La ruse du siècle : cf le code
- Module simple, utilise :

Module iCp

- Principe de fonctionnement : le compactage réorganise les N° d'enregistrements.
- La ruse du siècle : cf le code
- Module simple, utilise :
 - Une table sans champ, [iCp]

Module iCp

- Principe de fonctionnement : le compactage réorganise les N° d'enregistrements.
- La ruse du siècle : cf le code
- Module simple, utilise :
 - Une table sans champ, [iCp]
 - une variable interprocess, `<>iCpßB_WasCompacted__C`

Conclusion

Conclusion

- Construction d 'une boîte à outils
 - Réutilisable
 - Toujours plus complète

Conclusion

- Construction d 'une boîte à outils
 - Réutilisable
 - Toujours plus complète
- Augmente la productivité

Conclusion

- Construction d 'une boîte à outils
 - Réutilisable
 - Toujours plus complète
- Augmente la productivité
- Facilite la maintenance

Conclusion

- Construction d 'une boîte à outils
 - Réutilisable
 - Toujours plus complète
- Augmente la productivité
- Facilite la maintenance
- Ratio « Spécifique/Générique ». Plus il est petit, plus le développement est rapide.