

Bibliographie

Références générales

Actes Conférences ECOOP, European Conference on Object-Oriented Programming. *Lecture Notes in Computer Science*, Springer Verlag. vol. 276 (1987), vol. 322 (1988), vol. 512 (1991).

Actes Conférences OOPSLA, Object-Oriented Programming, Systems, Languages, and Applications. *Special Issue SIGPLAN Notices*, ACM. vol. 21, n°11 (1986), vol. 22, n°12 (1987), vol. 23, n°11 (1988), vol. 24, n°10 (1989), vol. 25, n°10 (1991).

G. Agha. *Actors : a Model of Concurrent Computation in Distributed Systems*. MIT Press, Cambridge (Mass.), 1986.

B.J. Cox. *Object-Oriented Programming: an Evolutionary Approach*. Addison-Wesley, Reading (Mass.), 1986.

J. Ferber. *Conception et Programmation par Objets*. Collection Techniques de Pointe, Hermès, Paris, 1990.

G. Masini, A. Napoli, D. Colnet, D. Léonard, et K. Tombre. *Les Langages à Objets*. InterEditions, Paris, 1989.

B. Meyer. *Conception et Programmation par Objets*. InterEditions, Paris, 1989.

B. Shriver et P. Wegner, editors. *Research Directions in Object-Oriented Programming*. MIT Press, Cambridge (Mass.), 1987.

D. Tsichritzis, editor. Centre Universitaire d'Informatique, Université de Genève. *Objects and Things*, 1987. *Active Object Environments*. 1988. *Object Management*. 1990. *Object Composition*. 1991.

A. Yonezawa et M. Tokoro, editors. *Object-Oriented Concurrent Programming*. MIT Press, Cambridge (Mass.), 1987.

Articles sur des aspects particuliers

Les citations entre crochets font partie des références générales ci-dessus.

G. Agha et C.E. Hewitt. Actors: a Conceptual Foundation for Concurrent Object-Oriented Programming. In [Shriver et Wegner, 1987], pp. 49-74.

G. Agha et C.E. Hewitt. Concurrent Programming Using Actors. In [Yonezawa et Tokoro, 1987], pp. 37-53.

P. Cointe. *Implémentation et Interprétation des Langages Orientés Objets. Application aux Langages Smalltalk, Objvlisp et Formes*. Thèse de Doctorat d'État, Université de Paris VII, LITP 85.55, 1985.

L. Cardelli et P. Wegner. On Understanding Types, Data Abstraction, and Polymorphism. *ACM Computing Surveys*, vol. 17, n°4, pp. 471-522, 1985.

W.R. Cook, W.L. Hill, et P.S. Canning. Inheritance is not Subtyping. Actes *Principles of Programming Languages*, ACM, pp. 125-135, 1990.

S. Danforth et C. Tomlinson. Type Theories and Object-Oriented Programming. *ACM Computing Surveys*, vol. 20, n°1, pp. 29-72, 1988.

R. Ducourneau et M. Habib. La Multiplicité de l'Héritage dans les Langages à Objets. *Technique et Science Informatique*, vol. 8, n°1, pp. 41-62, 1989.

K. Gorlen. An Object-Oriented Class Library for C++ Programs. *Software Practice and Experience*, vol. 17, n°12, pp. 899-922, 1987.

H. Lieberman. Using Prototypical Objects to Implement Shared Behavior in Object-Oriented Systems. In [Actes OOPSLA, 1986], pp. 214-223.

B. Meyer. Genericity versus Inheritance. In [Actes OOPSLA, 1986], pp. 391-405.

D. Ungar, C. Chambers, B-W. Chang, et U. Hölzle. Organizing Programs without Classes. *International Journal of Lisp and Symbolic Computation*, vol. 4, n°3, 1991.

A. Yonezawa, E. Shibayama, T. Takada, et Y. Honda. Modelling and Programming in an Object-Oriented Concurrent Programming. In [Yonezawa et Tokoro, 1987], pp. 55-89.

Langages de programmation par objets

Les citations entre crochets font partie des références générales ci-dessus.

Byte Special Issue: the Smalltalk-80 System. *Byte*, vol. 6, n°8, 1981.

L. Cardelli, J. Donahue, L. Glassman, M. Jordan, B. Kalsow, et G. Nelson. *Modula-3 Report (revised)*. Research Report 52, DEC Systems Research Center, Palo Alto (Calif.), 1989.

P. Cointe. Metaclasses are First Class: the ObjVLisp Model. In [Actes OOPSLA, 1987], pp. 156-167.

O.J. Dahl et K. Nygaard. Simula, an Algol-based Simulation Language. *Comm. of the ACM*, vol. 9, n°9, pp. 671-678, 1966.

L.G. DeMichiel et R.P. Gabriel. The CommonLisp Object System: an Overview. In [Actes ECOOP, 1987], pp. 201-220.

Eiffel: the Language. Interactive Software Engineering, Inc., Goleta (Calif.), 1989.

M. Ellis et B. Stroustrup. *The Annotated C++ Reference Manual*. Addison-Wesley, Reading (Mass.), 1990.

A. Goldberg et D. Robson. *Smalltalk-80, the Language and its Implementation*. Addison-Wesley, Reading (Mass.), 1983.

S.E. Keene. *Object-Oriented Programming in CommonLisp. A Programmer's Guide to CLOS*. Addison-Wesley, Reading (Mass.), 1989.

H. Lieberman. Concurrent Object-Oriented Programming in Act1. In [Yonezawa et Tokoro, 1987].

S. Lippman. *A C++ Primer*. Addison-Wesley, Reading (Mass.), 1989.

D. Moon. Object-Oriented Programming with Flavors. In [Actes OOPSLA, 1986], pp. 1-8.

K. Schmucker. *Object-Oriented Programming for the Macintosh*. Hayden Book Company, Hasbrouck Heights, (New Jersey), 1986.

D. Ungar et R.B. Smith. Self: the Power of Simplicity. In [Actes OOPSLA, 1987], pp. 227-242.

A. Yonezawa, J-P. Briot, et E. Shibayama. Object-Oriented Concurrent Programming in ABCL/1. In [Actes OOPSLA, 1986], pp. 258-268.

Index

A

ABCL/1 104, 110
ACT1 104
acteur 11, 104
Ada 2, 4, 7, 15, 25
Afficheur 129
agglomération 125
Algol 2, 8
ami 54, 55, 126
arbre d'héritage 21, 28, 69,
120
Arc 37

B

Bloc 64, 77, 94, 119
Booléen 71, 74

C

C 2, 7, 9, 11, 15
C++ 4, 8, 9, 31, 35, 47, 50, 51,
53, 55, 56, 58, 59, 84,
96, 120, 124, 126
cache 73, 103
Carré 123
Cartésien 101
case 94

Ceyx 88

champ 13, 16, 65, 94
calculé 101
de classe 86, 127
classe 13, 14, 16, 33, 64, 82, 91,
97, 98, 118
abstraite 120, 123, 124, 132
active 119
agglomérée 125, 134
amie 54
atomique 119
composée 119, 125, 134
conteneur 58, 119
de base 21
dépendante 134
dérivée 21
générique 58
primitive 38
ClasseObjet 83, 85
ClassePile 83, 85
clonage 94, 96, 97, 116
CLOS 10, 88, 92
Clu 15
Collection 120, 122
combinaison 24
CommonLisp 2

- communication 110
- comportement exceptionnel 99
- composition 24, 125
- constructeur 56, 59, 84, 96
- continuation 105
- Create* 56
- D**
- destructeur 57, 59
- délégation 96, 104, 116
- Démo* 87
- démon 90
- dérivation parallèle 134
- dictionnaire des méthodes 28, 66, 69, 73
- domaine de visibilité 53, 128
- double distribution 129
- E**
- Eiffel 5, 8, 31, 35, 47, 50, 55, 56, 58, 59, 84, 126, 128, 135
- encapsulation 6, 15, 17, 52, 128
- enrichissement 21, 42, 69, 122, 123
- Entier* 38, 64, 78, 79
- envoi de message 13, 17, 34, 62, 104, 110, 116, 129
- F**
- facette 125
- factorielle 107
- Faux* 74, 100
- Fenêtre* 42, 44, 124, 129, 134
- Fenêtre3D* 134
- Flavors 88, 90
- Forme* 123
- Fortran 2
- G**
- gestion dynamique 59
- généricité 4, 7, 25, 58, 132
- GPile* 58
- graphe d'héritage 22, 27
- H**
- Hanoi* 111, 136
- HanoiG* 137
- héritage 14, 64, 69, 98, 121, 132
 - dynamique 102
 - multiple 22, 27, 44, 90, 124
 - privé 54, 124
 - répété 22, 124
 - simple 18, 22, 39
- HPile* 71, 86, 134
- I**
- Imprimante* 129
- initialisation 56, 84, 95
- instance 13, 14, 16, 64
- instanciation 16, 28, 33, 64, 66, 81, 84, 97
- Intervalle* 78
- invocation de méthode 17, 33, 51, 65, 110, 129
- itérateur 119, 121, 122
- J**
- jointure de continuations 109, 113
- L**
- langage
 - à objets 13, 111, 140
 - à typage dynamique 4
 - à typage statique 4
 - compilé 6, 8, 92
 - d'acteurs 4, 11, 104
 - de classes 94, 95, 96, 97, 116
 - de prototypes 11, 93, 116
 - faiblement typé 4, 135
 - fortement typé 4, 27
 - hybride 11
 - impératif 8
 - interprété 5, 61, 92

- non typé 5, 28, 61, 92, 103, 116, 128, 129
- parallèle 3, 104, 116
- semi-compilé 6, 10, 61
- séquentiel 3
- typé 5, 8, 26, 60, 92, 96, 128, 129, 131, 134
- Le_Lisp 2, 10, 88
- liaison 28
 - dynamique 27, 48, 50, 81, 92, 103, 116
 - statique 27, 48
- lien
 - d'héritage 14, 64, 93, 96
 - d'instanciation 14, 64, 65, 93, 96
- Lisp 2, 5, 9, 10, 15, 30, 65, 87, 116
- liste d'exportation 55, 126
- M**
- mandataire 104
- membre 53
- message 64, 104, 129
 - à mots clés 62
 - binaire 62, 72
 - express 110
 - unaire 62
- méta-circularité 10, 30, 84
- métaclasse 10, 28, 64, 81, 83, 90, 127, 135
- méthode 13, 16, 17, 34, 94, 125
 - amie 54
 - d'accès 125
 - de calcul 126
 - de classe 29, 84, 127
 - de construction 126
 - de contrôle 127
 - privée 118
 - publique 117
 - redéfinie 20
 - virtuelle 8, 49, 120
- ML 2, 15
- Modula3 4, 9, 31, 50, 53, 55, 59
- modularité 6, 7, 116, 140
- module 6, 15, 16
- moi 36
- O**
- Object Pascal 9
- Objective-C 11
- objet 1, 13, 21, 33, 64, 68, 69, 83, 91, 94, 111
- ObjVLisp 10, 88, 90
- OGraphique* 129, 134
- OGraphique3D* 134
- P**
- parent 96, 102
- Pascal 2, 4, 7, 13, 15, 25, 59, 78
- Pile* 15, 16, 19, 33, 39, 67, 71, 79, 83, 84, 94, 102
- PileAbstraite* 133, 139
- Plasma 104
- Polaire* 101
- Polygone* 123
- polymorphisme 24, 28
 - ad hoc 24, 26, 34, 80
 - d'héritage 26, 48, 125, 129
 - d'inclusion 25, 26
 - paramétrique 4, 25, 58
- privé 17, 54
- programmation
 - fonctionnelle 2, 18, 117
 - impérative 2, 18, 117
 - logique 2
 - modulaire 7
 - par objets 3, 7, 18, 116, 117
- Prolog 3
- protégé 54
- protocole 117, 122, 125, 132
- protoPile* 96, 98
- protoPoint* 101
- protoTour* 98
- prototypage 81, 92, 103, 128

- prototype 11, 94
- pseudo-variable
 - moi* 36
 - self* 72, 95
 - super* 72
- public 17, 54
- Q**
- Quadrilatère* 123
- R**
- ramasse-miettes 59, 60
- receveur 17, 34, 62, 105, 129
- Rectangle* 42, 123
- redéfinition de méthode 20, 26, 41, 69
- réification 30
- réutilisation 6, 7, 60, 128, 132, 134, 139, 140
- règles de visibilité 34, 52, 126, 128
- S**
- Scheme 2
- self* 72
- Self 6, 10, 72, 94, 95
- Simula 1, 4, 8, 31, 50
- Smalltalk 5, 6, 8, 9, 29, 61, 89, 91, 95, 100, 103, 118, 119, 135
- Sommet* 37
- sous-classe 21, 64, 71
- spécialisation 19, 21, 39, 122, 123
- super* 72
- superclasse 21
- surcharge 24, 131
- T**
- table virtuelle 51
- Tableau* 67
- Tour* 19, 39, 42, 44, 57, 98, 102, 111, 124, 136
- TourG* 42, 44, 137
- TourGM* 44, 124
- Tours de Hanoi 18, 98, 111, 119, 136
- typage 4, 8, 27, 28, 30, 49, 81, 92, 103, 135
- type 4, 16, 103
- V**
- variable d'instance 65
- variable de classe 86
- Voiture* 125
- Vrai* 74, 100
- vue 55