

Claude Delannoy

S'initier à la programmation

**Avec des exemples en
C, C++, C#, Java et PHP**

EYROLLES

Table des matières

Chapitre 1 : Ordinateur, programme et langage	1
1 - Le rôle de l'ordinateur	1
1.1 La multiplicité des applications	1
1.2 Le programme : source de diversité	2
1.3 Les données du programme, les résultats	2
1.4 Communication ou archivage	3
2 - Pour donner une forme à l'information : le codage	3
2.1 L'ordinateur code l'information	3
2.2 L'homme code l'information	4
2.3 Ce qui différencie l'homme de l'ordinateur	4
3 - Fonctionnement de l'ordinateur	5
3.1 À chacun son rôle	5
3.2 La mémoire centrale	6
3.3 L'unité centrale	7
3.4 Les périphériques	7
3.4.1 Les périphériques de communication	7
3.4.2 Les périphériques d'archivage	7
4 - Le langage de l'ordinateur	8
4.1 Langage machine ou langage de notre cru	8
4.2 En langage assembleur	9
4.3 En langage évolué	9
5 - Les concepts de base des langages évolués	11
6 - La programmation	12
7 - Notion de système d'exploitation et d'environnement de programmation	13

Chapitre 2 : Variables et instruction d'affectation	15
1 - La variable	15
1.1 Introduction	15
1.2 Choix des noms des variables	16
1.3 Attention aux habitudes de l'algèbre	16
2 - Type d'une variable	17
2.1 La notion de type est une conséquence du codage en binaire	17
2.2 Contraintes imposées par le type	17
2.3 Les types que nous utiliserons	18
2.4 Déclaration de type	18
3 - L'instruction d'affectation	19
3.1 Introduction	19
3.2 Notation	19
3.3 Rôle	20
3.4 Quelques précautions	21
3.5 Échanger les valeurs de deux variables	22
4 - Les expressions	23
4.1 Expressions de type entier	24
4.1.1 Constantes de type entier	24
4.1.2 Expressions de type entier	24
4.1.3 Les erreurs possibles	25
4.2 Expressions de type réel	26
4.2.1 Constantes réelles	26
4.2.2 Les expressions réelles	27
4.2.3 Les erreurs possibles	27
4.3 Expressions mixtes	28
4.4 Expressions de type caractère	30
4.5 Affectation et conversions	30
5 - Les variables non définies	31
6 - Initialisation de variables et constantes	32
6.1 Initialisation de variables	32
6.2 Constantes	32
6.3 Expressions constantes	33
7 - Les fonctions prédéfinies	33
Côté langages	34
Noms de variables	34
Types de base et codage	34
Déclaration de types	35
Instruction d'affectation	35
Opérateurs et expressions	36

Chapitre 3 : Instructions d'écriture et de lecture	39
1 - L'instruction d'écriture	39
1.1 Rôle	39
1.2 Présentation des résultats	40
1.2.1 Rien ne les identifie	40
1.2.2 Comment seront-ils présentés ?	41
1.2.3 Affichage de libellés	41
1.2.4 Cas des valeurs réelles	42
2 - L'instruction de lecture	43
2.1 Rôle	43
2.2 Intérêt de l'instruction de lecture	44
2.3 Présentation des données	45
2.4 Exemple	45
3 - Autres modes de communication avec l'utilisateur	46
3.1 Mode console ou programmation par événements	46
3.2 Mode batch	47
3.3 Programmation Internet	47
Exemples langages	48
En C	48
En C++	49
En C#	49
En Java	50
PHP	50
Chapitre 4 : La structure de choix	51
1 - Présentation de l'instruction de choix	52
1.1 Exemple introductif	52
1.2 Notion de bloc d'instructions	52
1.3 Un programme complet	54
2 - La condition du choix	54
2.1 Les conditions simples	54
2.2 Les conditions complexes	56
2.2.1 Présentation	56
2.2.2 Exemple	57
3 - Cas particulier : une partie du choix absente	58
4 - Les choix imbriqués	59
4.1 Exemple	59
4.2 En cas d'ambiguïté	60
4.3 Choix imbriqués ou succession de choix	61
5 - Un nouveau type de base : booléen	63
6 - Nos conventions d'écriture	64

Côté langages	64
Instruction de choix	64
Type booléen	65
Instruction de choix multiple	65
Exemples langages	66
C	66
C++	67
Java	67
C#	68
PHP	68
 Chapitre 5 : Les structures de répétition	69
1 - La répétition jusqu'à'	69
1.1 Exemple introductif	69
1.2 Nos conventions d'écriture	71
1.3 Exemples	72
1.3.1 Recherche de la première voyelle d'un mot	72
1.3.2 Doublement de capital	72
1.4 Faire des choix dans une boucle	74
2 - La répétition tant que	74
2.1 Exemple introductif	75
2.2 Conventions d'écriture	75
2.3 Lien entre répétition tant que et répétition jusqu'à	76
2.4 Exemple	77
3 - Comment réaliser des répétitions inconditionnelles	78
3.1 La notion de compteur de boucle	79
3.2 Introduire un compteur dans une répétition	79
3.2.1 Exemple 1	79
3.2.2 Exemple 2	80
3.3 Imposer un nombre de tours	81
3.3.1 Exemple 1	82
3.3.2 Exemple 2	82
3.3.3 Exemple 3	83
4 - La répétition inconditionnelle	84
4.1 Exemples d'introduction	84
4.1.1 Exemple 1	84
4.1.2 Exemple 2	85
4.2 Conventions d'écriture	86
4.3 Utiliser le compteur dans une répétition inconditionnelle	87
4.4 Éviter d'agir sur le compteur dans la boucle	87
4.5 Compteur et boucle pour	88
4.6 Un tour pour rien	89
4.7 Le compteur en dehors de la boucle	90

Côté langages	90
Les répétitions tant que et jusqu'à	90
La répétition pour	91
Exemples langages	92
C	92
C++	92
C#	93
PHP	94
 Chapitre 6 : Quelques techniques usuelles	95
1 - Le comptage d'une manière générale	95
1.1 Compter le nombre de lettres e d'un texte	96
1.2 Compter le pourcentage de lettres e d'un texte	96
2 - L'accumulation	98
2.1 Accumulation systématique	98
2.1.1 Un premier exemple	98
2.1.2 Un second exemple	99
2.2 Accumulation sélective	100
3 - Recherche de maximum	101
4 - Imbrication de répétitions	102
4.1 Exemple de boucle avec compteur dans une boucle conditionnelle	102
4.2 Exemple de boucle conditionnelle dans une boucle avec compteur	103
4.3 Exemple de boucle inconditionnelle dans une autre boucle inconditionnelle	104
4.3.1 Premier exemple	104
4.3.2 Second exemple	104
4.4 Une erreur à ne pas commettre	106
5 - L'itération	107
 Chapitre 7 : Les tableaux	109
1 - Notion de tableau à une dimension	110
1.1 Quand la notion de variable ne suffit plus	110
1.2 La solution : le tableau	110
2 - Utilisation d'un tableau à une dimension	111
2.1 Déclaration	111
2.2 Manipulation des éléments d'un tableau	112
2.3 Affectation de valeurs à des éléments d'un tableau	112
2.4 Lecture des éléments d'un tableau	113
2.5 Écriture des éléments d'un tableau	114
2.6 Utilisation de variables indicées dans des expressions	114
2.7 Initialisation d'un tableau à une dimension	116
3 - Quelques techniques classiques appliquées aux tableaux à une dimension	116
3.1 Somme et maximum des éléments d'un tableau	116
3.2 Test de présence d'une valeur dans un tableau	117

4 - Exemple d'utilisation d'un tableau	118
5 - Tri d'un tableau à une dimension	119
6 - Contraintes sur la dimension d'un tableau	120
7 - Débordement d'indice d'un tableau à une dimension	121
8 - Introduction aux tableaux à deux dimensions	122
9 - Utilisation d'un tableau à deux dimensions	124
9.1 Déclaration	124
9.2 Affectation de valeurs	124
9.3 Lecture des éléments	125
9.4 Écriture des éléments	126
10 - Quelques techniques classiques appliquées aux tableaux à deux dimensions	128
11 - Gestion de l'emplacement mémoire d'un tableau	128
12 - Notion de tableau associatif	129
Côté langages	130
C/C++	130
Java, C#	131
PHP	132
Exemples langages	133
C++	133
C	134
C#	134
Java	135
PHP	137
Chapitre 8 : Les fonctions	139
1 - Notion de fonction	140
1.1 Premier exemple	140
1.2 Notion de paramètre	141
1.3 Paramètres formels ou effectifs	143
1.4 Notion de variable locale	143
1.5 Notion de résultat	144
1.6 Exemple de fonctions à plusieurs paramètres	146
1.7 Indépendance entre fonction et programme	147
2 - Mode de transmission des paramètres	149
2.1 Introduction	149
2.2 Conséquences de la transmission par valeur	150
2.3 La transmission par référence	150
2.4 Nature des paramètres effectifs	152
2.5 Un autre exemple de transmission par référence	152
3 - Tableaux en paramètres	153
3.1 Cas des tableaux de taille déterminée	153

3.2 Cas des tableaux de taille indéterminée	154
3.3 Exemple	155
4 - Les fonctions en général	156
4.1 Propriétés des variables locales	156
4.1.1 <i>Les variables locales ne sont pas rémanentes</i>	156
4.1.2 <i>Initialisation des variables locales</i>	157
4.1.3 <i>Tableaux locaux</i>	158
4.1.4 <i>Imposer à une variable locale d'être rémanente</i>	158
4.2 Propriétés du résultat	159
4.3 Appels imbriqués	160
4.4 Variables globales	161
4.5 Concordance de type	161
4.6 Surdéfinition des fonctions	162
5 - Gestion de la mémoire des variables locales : notion de pile	162
6 - Programme principal et fonctions	163
7 - La récursivité	164
8 - Bibliothèques de fonctions	166
9 - Une autre présentation de la notion de fonction	167
Côté langages	168
Structure d'une fonction	168
Mode de transmission des paramètres	169
Programme principal	169
Séparation entre fonction et programme	170
Exemples langages	171
Fonction somme des éléments d'un tableau	171
Résultat	171
Variables globales	171
Fonction estVoyelle	173
Fonction tri d'un tableau avec fonction échange	175
 Chapitre 9 : Classes et objets	 179
1 - Introduction	179
2 - Un premier exemple : une classe Point	180
2.1 Utilisation de notre classe Point	181
2.1.1 <i>Le mécanisme déclaration, instanciation</i>	181
2.1.2 <i>Utilisation d'objets de type Point</i>	182
2.2 Définition de la classe Point	183
2.3 En définitive	184
2.4 Indépendance entre classe et programme	184
3 - L'encapsulation et ses conséquences	185
3.1 Méthodes d'accès et d'altération	185
3.2 Notions d'interface, de contrat et d'implémentation	186
3.3 Dérogations au principe d'encapsulation	187

4 - Méthode appelant une autre méthode	188
5 - Les constructeurs	188
5.1 Introduction	188
5.2 Exemple d'adaptation de notre classe Point	189
5.3 Surdéfinition du constructeur	190
5.4 Appel automatique du constructeur	191
5.5 Exemple : une classe Carré	192
6 - Mode des gestion des objets	195
Côté langages	196
Définition d'une classe	196
Utilisation d'une classe	197
Exemples langage	197
Java	197
C#	198
PHP	199
C++	200
 Chapitre 10 : Propriétés des objets et des méthodes	 203
1 - Affectation et comparaison d'objets	203
1.1 Premier exemple	203
1.2 Second exemple	205
1.3 Comparaison d'objets	206
1.4 Cas des langages gérant les objets par valeur	206
2 - Les objets locaux et leur durée de vie	207
3 - Cas des objets transmis en paramètre	208
3.1 Mode de transmission d'un objet en paramètre	209
3.2 L'unité d'encapsulation est la classe	210
3.3 Exemple	212
4 - Objet en résultat	214
5 - Attributs et méthodes de classe	215
5.1 Attributs de classe	215
5.1.1 Présentation	215
5.1.2 Exemple	217
5.2 Méthodes de classe	217
5.2.1 Généralités	217
5.2.2 Exemple	218
5.2.3 Autres utilisations des attributs et des méthodes de classe	219
6 - Tableaux d'objets	219
7 - Autoréférence	221
7.1 Généralités	221
7.2 Exemples d'utilisation de courant	221
8 - Classes standards et classe Chaîne	222

Côté langages	223
Affectation, transmission en paramètre et en résultat	223
Méthodes et attributs de classe	223
Autoréférence	224
Exemples langages	224
C#	224
Java	225
C++	226
PHP	227
 Chapitre 11 : Composition des objets	229
1 - Premier exemple : une classe Cercle	229
1.1 Droits d'accès	230
1.1.1 <i>Comment doter Cercle d'une méthode affiche</i>	230
1.1.2 <i>Doter Cercle d'une méthode déplace</i>	231
1.2 Relations établies à la construction	231
1.2.1 <i>Coordonnées en paramètres</i>	232
1.2.2 <i>Objet de type point en paramètre</i>	233
1.3 Cas de la gestion par valeur	233
2 - Deuxième exemple : une classe Segment	235
3 - Relations entre objets	238
4 - Copie profonde ou superficielle des objets	239
5 - Une classe « singleton »	240
Côté langages	242
Java, C# et PHP	242
C++	242
Java	243
C++	244
PHP	245
 Chapitre 12 : L'héritage	247
1 - La notion d'héritage	248
2 - Droits d'accès d'une classe dérivée à sa classe de base	250
2.1 Une classe dérivée n'accède pas aux membres privés de la classe de base	250
2.2 Une classe dérivée accède aux membres publics	251
2.3 Exemple de programme complet	252
3 - Héritage et constructeur	254
4 - Comparaison entre héritage et composition	256
5 - Dérivations successives	258
6 - Redéfinition de méthodes	259
6.1 Introduction	259
6.2 La notion de redéfinition de méthode	259

6.3 La redéfinition d'une manière générale	261
6.4 Redéfinition de méthode et dérivations successives	262
7 - Héritage et droits d'accès	263
Côté langages	264
Syntaxe de la dérivation et droits d'accès	264
Gestion des constructeurs	265
Redéfinition de méthodes	265
.....	266
Exemples langage	267
Java	267
C#	267
C++	268
PHP	269
 Chapitre 13 : Le polymorphisme	 271
1 - Les bases du polymorphisme	271
1.1 Compatibilité par affectation	272
1.2 La ligature dynamique	273
1.3 En résumé	273
1.4 Cas de la gestion par valeur	274
1.5 Exemple 1	274
1.6 Exemple 2	275
2 - Généralisation à plusieurs classes	276
3 - Autre situation où l'on exploite le polymorphisme	277
4 - Limites de l'héritage et du polymorphisme	280
4.1 Les limitations du polymorphisme	280
4.2 Valeurs de retour covariantes	281
Côté langages	283
Java et PHP	283
C#	283
C++	283
Exemples langages	284
Java	284
C#	285
PHP	286
C++	287
 Chapitre 14 : Classes abstraites, interfaces et héritage multiple	 289
1 - Classes abstraites et méthodes retardées	289
1.1 Les classes abstraites	289
1.2 Méthodes retardées (ou abstraites)	290
1.3 Intérêt des classes abstraites et des méthodes retardées	291
1.4 Exemple	292

2 - Les interfaces	293
2.1 Définition d'une interface	294
2.2 Implémentation d'une interface	294
2.3 Variables de type interface et polymorphisme	295
2.4 Exemple complet	296
2.5 Interface et classe dérivée	297
3 - L'héritage multiple	297
Côté langages	298
Classes abstraites et méthodes retardées	298
Exemples langage	299
Classes et méthodes abstraites	299
Interfaces	299
Interfaces	303
Chapitre 15 : Correction des exercices	307
Chapitre 2	307
Chapitre 3	310
Chapitre 4	312
Chapitre 5	314
Chapitre 6	318
Chapitre 7	321
Chapitre 8	325
Chapitre 9	328
Chapitre 10	333
Chapitre 11	335
Chapitre 12	339
Index	339