

Erratum (12/12/2008)

« S'initier à la programmation » de Claude Delannoy

Page 24, 6^{ème} ligne avant la fin, lire effectuer et non *effecteur*.

Page 25, fin de la 8^{ème} ligne, lire $(2 * n) + p$ et non $(2 * n) + c$.

Page 29, fin du 4^{ème} paragraphe, lire 14,3 et non 12,3.

Page 55, avant dernière ligne du tableau, le symbole correspondant à « supérieur ou égal » est \geq et non $=$.

Page 75, la 7^{ème} ligne de l'exemple de programme doit se lire :

écrire « capital un an plus tard : », cap

et non :

écrire « capital un an plus tard : »

Page 85, dans le dernier exemple de programme, il faut supprimer la 9^{ème} ligne :

```
i := i + 1
```

Page 90, 5^{ème} ligne, il faut lire :

```
répéter pour i := 1 à 5
```

et non :

```
répéter pour i := 1 à 10
```

Page 119, 11^{ème} ligne du programme de tri d'un tableau à une dimension, il faut lire :

```
si t[i] > t[j] alors
```

et non :

```
si t[i] < t[j] alors
```

Page 129, 2^{ème} ligne du 2^{ème} paragraphe, lire :

proches de celles que...

et non :

proches *ce* celles que...

Page 153, première ligne de l'exercice 8.4, lire :

dans l'exemple du paragraphe 2.3 p 151

et non :

dans l'exemple du paragraphe 1.6 p 146

Page 172, dans les deux instructions *System.out.println* de l'exemple Java, il manque le signe $=$ dans les deux libellés.

Page 215, dernier paragraphe, il faut lire :

ses propres attributs n et y

et non :

ses propres attributs n et x

Page 310, 5ème ligne de la correction de l'exercice 3.1, il manque l'affichage d'une virgule entre les deux noms des variables a et b. Il faut lire :

écrire «a = », a, «, b = », b

et non :

écrire «a = », a, «b = », b

Page 313, correction de l'exercice 4.5, la formule de calcul de la remise est incorrecte. Dans la quatrième ligne avant la fin du programme, il faut lire :

remise := prixTTC * tauxRemise

et non :

remise := prixTTC * (1. - tauxRemise)

Page 316, la correction traite de la même manière une valeur négative ou nulle et une valeur supérieure ou égale à 100. Pour être conforme à l'énoncé, il faudrait procéder ainsi dans le premier programme :

```
entier nombre := -1
écrire «donnez un entier positif inférieur à 100»
tant que nombre <=0 ou nombre >=100 répéter
{ lire nombre
  si nombre<=0 alors écrire «SVP, positif»
  si nombre>=100 alors écrire «SVP, inférieur à 100»
}
écrire «merci pour le nombre », nombre
```

De même, on pourrait procéder ainsi dans le second :

```
booléen OK := faux
entier nombre
écrire «donnez un entier positif inférieur à 100»
tant que non OK répéter
{ lire nombre
  si nombre>0 et nombre<100 alors ok := vrai
  si nombre<=0 alors écrire «SVP, positif»
  si nombre>=100 alors écrire «SVP, inférieur à 100»
}
écrire «merci pour le nombre », nombre
```

Page 323, 12^{ème} ligne du programme de l'exercice 7.6, il faut lire :

si t[i] > t[j] alors

et non :

si t[i] < t[j] alors

Page 324, en bas de page, il manque une accolade fermante (}) à la fin du programme.

Page 327, dans la fonction *maxmin* de l'exercice 8.5, il manque la déclaration de la variable locale *i*. Le bloc de la fonction doit se présenter ainsi :

```
{ entier i
  max := t[1]
  .....
```

et non :

```
{ max := t[1]
  .....
```

Page 327, dans la première ligne du programme de l'exercice 8.6, il faut lire :

```
fonction tri (tableau entier t[], entier nbElem)
```

et non :

```
fonction tri (tableau entier t|], int nbElem)
```

Page 339, dans l'exemple de programme de l'exercice 12.1, il manque une instruction de création d'un objet de type *PointA*. Après l'instruction :

```
p := Création Point
```

Il faut ajouter l'instruction :

```
pa := Création PointA
```

Page 339, dans la deuxième ligne de l'exemple d'utilisation, il faut lire :

```
PointNom pn1, pn2
```

et non :

```
PointNom pn1, pn3
```