

## TD : Exercice 1 : b)

### Énoncé :

#### 1. Exercice sur les structures de contrôle simple if-then-else à une condition

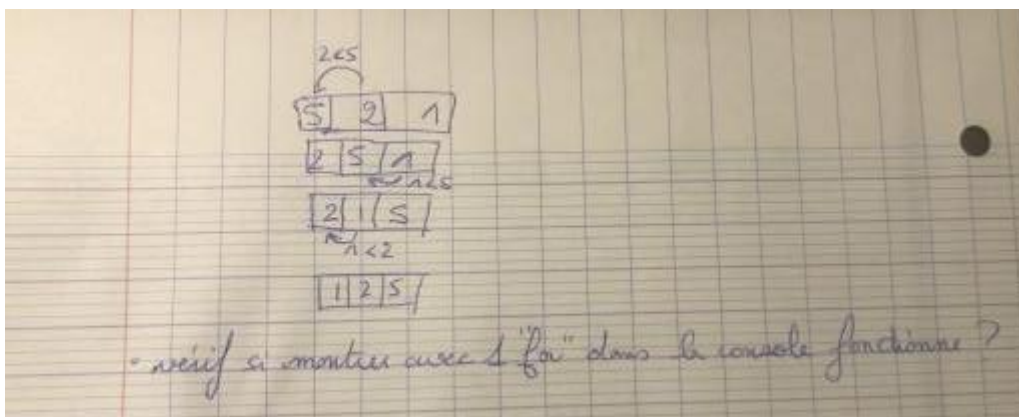
1.b) Ecrire le programme qui affiche à la console et dans l'ordre croissant les 3 valeurs flottantes initialisées, et qui colorie en bleu la fenêtre une fois le traitement terminé (alors qu'elle est blanche au démarrage).

2 tests :

i) float f1=3.0, f2=14.4, f3=7.18 ;

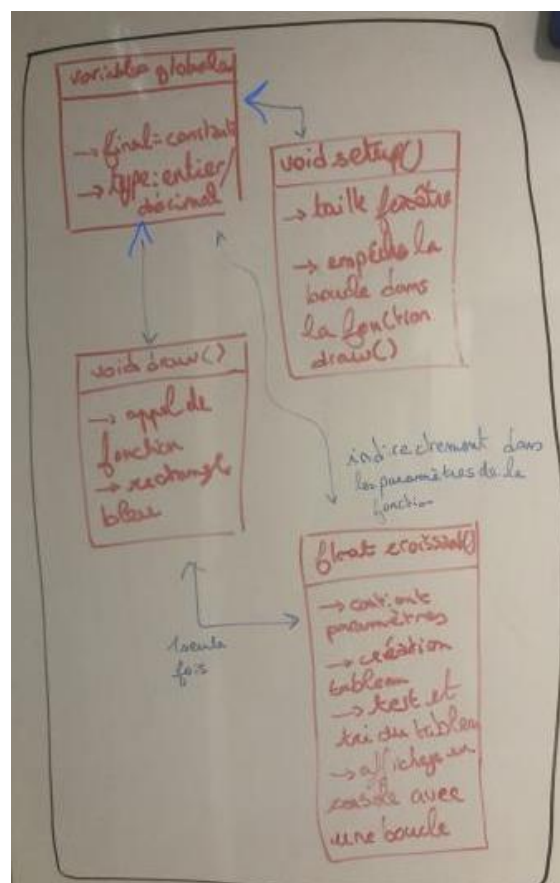
ii) double f1=-5.4, f2=-18.114, f3=0.0 ;

### Brouillon :



- J'ai pu garder plus ou moins le même fonctionnement dans mon programme final.

### Diagramme de classe :



**Exemples de ce qui se passe (graphique et console) :**

Si les valeurs d'entrées sont :

- 1<sup>er</sup> exemple : - `float f1=3.0, f2=14.4, f3=7.18 ;`

Dans le terminal après l'exécution du code, on aura :

La 1 valeur est : 3.0

La 2 valeur est : 7.18

La 3 valeur est : 14.4

➔ Une fenêtre de couleur bleu devrait aussi apparaître après avoir donné le résultat.

- 2<sup>ème</sup> exemple : - `double f1=-5.4, f2=-18.114, f3=0.0 ;`

Dans le terminal après l'exécution du code, on aura :

La 1 valeur est : -18.114

La 2 valeur est : -5.4

La 3 valeur est : 0.0

➔ Une fenêtre de couleur bleu devrait aussi apparaître après avoir donné le résultat.

⇒ On remarque après avoir réalisé le programme que ce dernier retourne les mêmes résultats.

**Capture d'écran du code final :** (programme avec les variables en float)

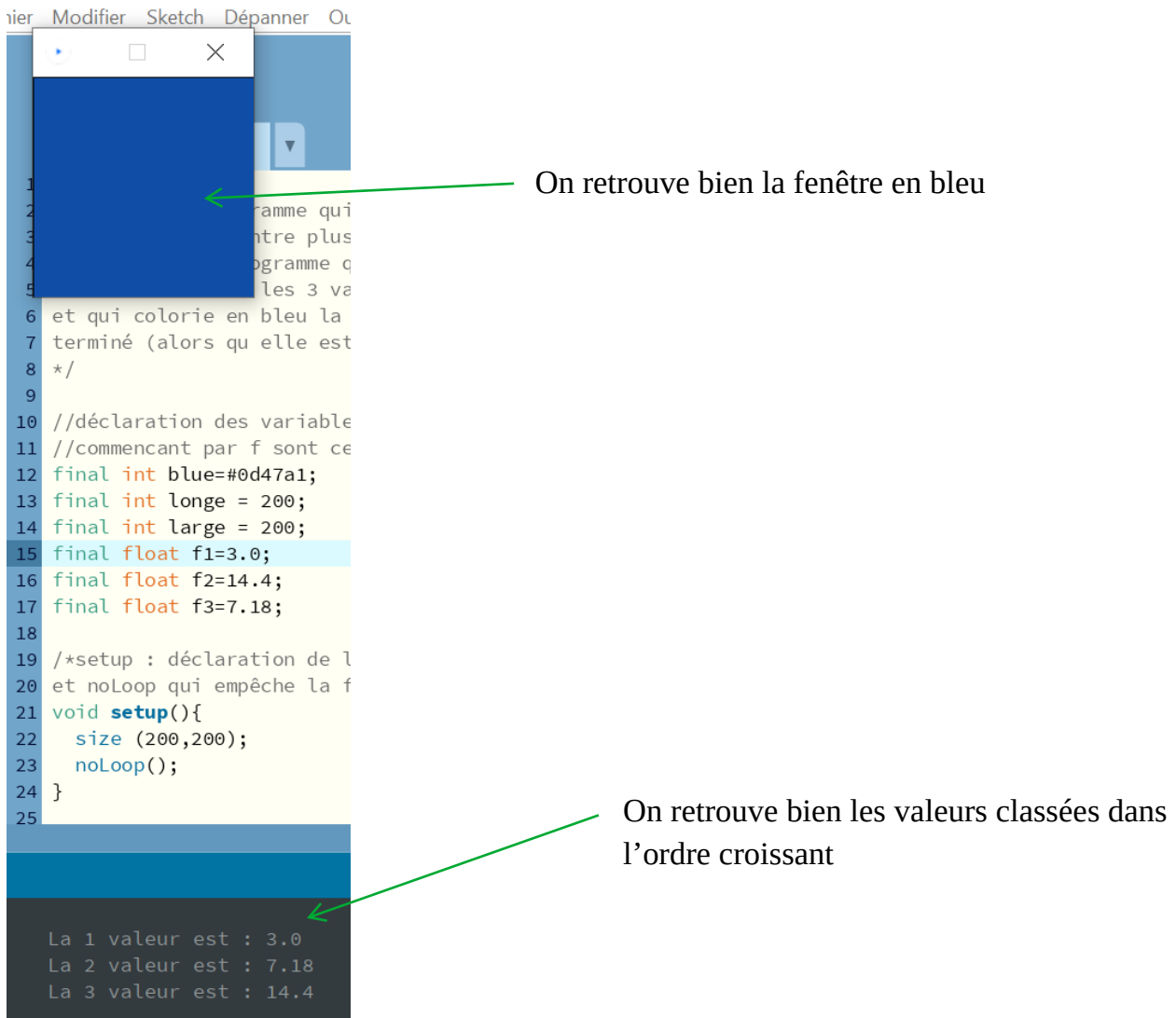
```

1  /*
2  exercice 1 : programme qui permet de trouver la
3  valeur maximale entre plusieurs variables
4  1.b) Ecrire le programme qui affiche à la console et dans
5  l'ordre croissant les 3 valeurs flottantes initialisées,
6  et qui colorie en bleu la fenêtre une fois le traitement
7  terminé (alors qu'elle est blanche au démarrage).
8  */
9
10 //déclaration des variables globales du programme les variables
11 //commencant par f sont celles à classer
12 final int blue=#0d47a1;
13 final int longe = 200;
14 final int large = 200;
15 final float f1=3.0;
16 final float f2=14.4;
17 final float f3=7.18;
18
19 /*setup : déclaration de la taille de la fenêtre
20 et noLoop qui empêche la fonction draw de se répéter*/
21 void setup(){
22     size (200,200);
23     noLoop();
24 }
25
26 //ouverture d'une fonction permettant de mettre le code "pratique"
27 void draw (){
28     background (255);
29     croissant(f1,f2,f3);
30     fill (blue);
31     rect (0, 0, longe, large);
32 }
33
34
35 /*création de notre fonction de type float permettant
36 de ranger les valeurs dans l'ordre croissant*/
37 float croissant (float _f1, float _f2, float _f3) {
38     float t = -1;
39     //on met les valeurs dans un tableau
40     float tab[]={_f1,_f2,_f3};
41     /*
42     tant que le tableau n'est pas rangé dans l'ordre croissant:
43     il y aura une comparaison grâce aux if entre les cases 0 et 1 puis
44     entre les cases 1 et 2. Selon les valeurs des cases, il peut y avoir
45     des inversions entre elles se qui rangera le tableau et fera sortir du while.
46     */
47     while (!(tab[0]<=tab[1] && tab[1]<=tab[2] && tab[0]<tab[2])){
48         if (tab[0]>tab[1]) {
49             float inter = tab[1];
50             tab[1]=tab[0];
51             tab[0]=inter;
52         }
53         //les if de comparaisons expliqués dans les commentaires au dessus
54         if(tab[1]>tab[2]){
55             float inter2 = tab[2];
56             tab[2]=tab[1];
57             tab[1]=inter2;
58         }
59     }
60
61     for (int i=0; i<tab.length;i++){
62         print("La ");
63         print(i+1);
64         print(" valeur est : ");
65         println(tab[i]);
66     }
67     //retourne une variable qui ne nous intéresse pas, sert juste à marquer la fin de la fonction
68     return t;
69 }

```

Remarque : le second fichier nommé question\_b\_double a le même code. Les seuls changements sont que le type des éléments à comparer est « double » au lieu de « float ». (Ils sont facilement identifiable avec un ctrl+f).

**Capture d'écran du l'exécution (terminal ...) :**



**Avis/difficultés/problèmes rencontrés/piste d'amélioration/remarques :**

- J'ai trouvé ce programme intéressant à réaliser et j'ai surtout bien aimé la réflexion et le travail en amont que j'ai dû réaliser.
- J'ai eu de mal à trouver un algorithme fiable à mettre dans ma fonction afin que ça ne soit pas trop compliqué et surtout compréhensible par mes paires.
- Je pense que j'aurais pu changer l'affichage afin que ça mette directement :

La première valeur est : 3.0  
La deuxième valeur est : 7.18  
La troisième valeur est : 14.4

- ⇒ pour cela j'aurais dû faire une boucle « for » en plus lors de l'affichage mais pendant de la création du programme je ne trouvais ça pas pertinent et trouvais l'affichage implémenté assez compréhensible.

## **Changement majeur :**

Ancienne version :

```
47     for (int i=0; i<tab.length;i++){
```

Nouvelle version :

```
47     while (!(tab[0]<=tab[1] && tab[1]<=tab[2] && tab[0]<tab[2])){
```

- Ce changement a été dur à voir pour moi, il me permet que le code soit plus fiable, compréhensible et rapide à exécuter. La boucle se répète tant que les chiffres ne sont pas classés dans l'ordre croissant donc on peut en sortir directement si l'ordre est déjà respecté.

- Alors que l'ancienne boucle tournait un nombre de fois fixe et faisait perdre un peu de temps lors de l'exécution du code.