**Les structure de contrôle**

**Le bloc d’instruction**

Le bloc d’instruction est un ensemble d’instructions délimité par 2 accolades {}

Il permet de regrouper les instructions.

NB : toute variable déclaré à l’intérieur d’un bloc d’instruction est illisible à l’extérieur du bloc

Toute variable déclaré a l’extérieur du bloc d’instruction est lisible dans le bloc

**La structure conditionnelle if else**

Les structure qu’on a vue s’exécute d’une manier séquentiel l’une après l’autre

Les bloc conditionnelle s’execute si la condition est respecter

L’operateur ternaire pour reduire le code de if else ; l’idé c’est de crée un boolean de lui affecter la condition et l’utiliser dans la candition if

EXP

int age3=20;  
boolean B4 =true ;  
boolean B5 = age3==20 && B4 ;  
System.*out*.println(B5?"l'age du capitaine est :" + age3 : "l'age du capitaine different de :"+age3);  
System.*out*.println(age3==20 && B4?"l'age du chaf est :" + age3 : "l'age du chaf different de :"+age3);

EXP if else if

int age4 = 43;  
  
if (age4 == 40){  
 System.*out*.println("l'age du capitain est égale à : 40" );  
}  
else if (age4>40){  
 System.*out*.println("l'age du capitain est superieur à : 40" );  
}  
else {  
 System.*out*.println("l'age du capitaine est inférieur à : 40" );  
}

**Résoudre les erreurs : le deboggeur**

On utilise la fonctionnalité de debugage offerte par les environement de développement comme intellij\_IDEA.

On exécute le projet avec le button debug a la place de l’execution normale ca permet d’executer le code line par line et de detecter si l ya quelque chose illogique méme si le code est correcte

**La structure conditionnelle switch - case**

La condition swich – case peut remplacer if else pour éviter la redondance

EXP

int age5 = 80 ;  
switch (age5){  
 case 20 :  
 System.*out*.println("le capitaine à 20 ans");  
 break;  
 case 10 :  
 System.*out*.println("le capitaine à 10 ans");  
 break;  
 default:  
 System.*out*.println("cette valeur ne correspond pas");

**Break** : pour arreter les la lecture des autre case le cas de realiser la condition

**Default** : pour donner une instruction le cas de ne pas realiser la condition dans tous les cases

**NB :** avant la version 7 de java on peut utiliser la structure switch-case seulement sur les type int et char, depuis la version java 7 on peut utiliser le type string

**La structure itérative for**

Permet de répéter un bloc d’instruction autant de fois souhaiter tant que ce qui est donnée en condition est vrai

for (int nbr7=0;nbr7<=3;nbr7++){  
 System.*out*.println("le nombre est : " + nbr7);  
  
}

resultat

le nombre est : 0

le nombre est : 1

le nombre est : 2

le nombre est : 3

Exp utilisation switch-case dans la boucle for

for (int nbr7=0;nbr7<=3;nbr7++){  
 switch (nbr7){  
 case 0 :  
 System.*out*.println("le nombre est : " + nbr7);  
 break;  
 case 1 :  
 System.*out*.println("le nombre est : " + nbr7);  
 break;  
 case 2 :  
 System.*out*.println("le nombre est : " + nbr7);  
 break;  
 case 3 :  
 System.*out*.println("le nombre est : " + nbr7);  
 break;  
 default :  
 System.*out*.println("aucun nombre ne correspond");  
 }

resulta est :

le nombre est : 0

le nombre est : 1

le nombre est : 2

le nombre est : 3

**La structure itérative while**

Le fonctionnement du mot clés **WHILE** est proche de **FOR**

Lors qu’on connais dès le départ le nombre d’iteration qu’on va réaliser on utilise FOR sinon on utilise **while** ca sera plus pratique.

boolean b9=true;  
 while (b9){  
 System.*out*.println("l'itération while active");  
 b9=false;  
 }

On peut utliser l’instruction inverse **do-while** c’est d’exécuter des instructions avant de commencer dans la boucle while

boolean b10=true;  
 do {  
 System.*out*.println("l'itération DO\_while n'es pas active");  
 b10=false;  
 }while (b10);

utiliser swich-case dans l’instruction WHILE

while (b<3){  
 switch (b){  
 case 0 :  
 System.*out*.println("le nombre b est : " + b);  
 break;  
 case 1 :  
 System.*out*.println("le nombre b est : " + b);  
 break;  
 case 2 :  
 System.*out*.println("le nombre b est : " + b);  
 break;  
 case 3 :  
 System.*out*.println("le nombre b est : " + b);  
 break;  
 default :  
 System.*out*.println("aucun nombre b ne correspond");  
  
 }  
 b++;  
}

Résulta

le nombre b est : 0

le nombre b est : 1

le nombre b est : 2

Utilisation du mot clés BREAK dans l’instruction WHILE permet d’éviter la redondance

Dans cette exemple si on utilise pas break ca donnera une boucle infinie.

boolean bo1 =true;  
while (bo1){  
 System.*out*.println("la boucle est active pour une autre seconde");  
 break;  
}