



STRUCTURES DE CONTROLE

Les Boucles : **La Boucle do while**

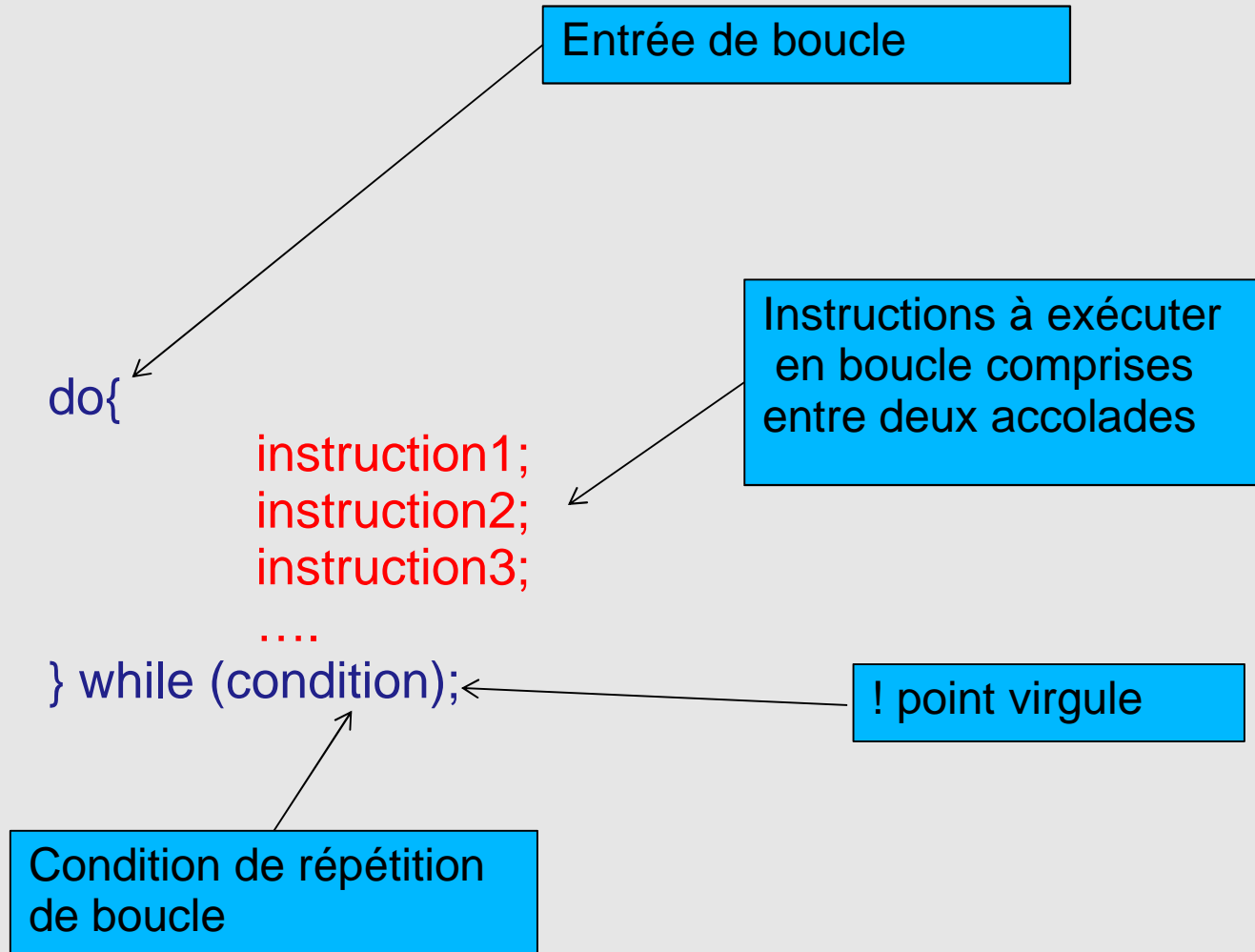
La boucle **do ... while** permet de répéter un nombre quelconque de fois (1 ou plus) l'exécution d'une instruction ou d'un bloc d'instructions.

Le nombre de répétitions n'est pas forcément connu au moment d'entrer dans la boucle.



STRUCTURES DE CONTROLE

Les Boucles : **La Boucle do while**





STRUCTURES DE CONTROLE

Les Boucles : **La Boucle do while**

```
int nombre ;
```

```
do{
```

```
    System.out.println("Taper un nombre positif");  
    nombre = In.readInt();
```

```
} while (nombre <=0);
```

```
.....
```

Le programme exécute une première fois la boucle.

Il évalue la condition.

Il recommence la boucle si la condition est vérifiée



STRUCTURES DE CONTROLE

Les Boucles : **La Boucle while**

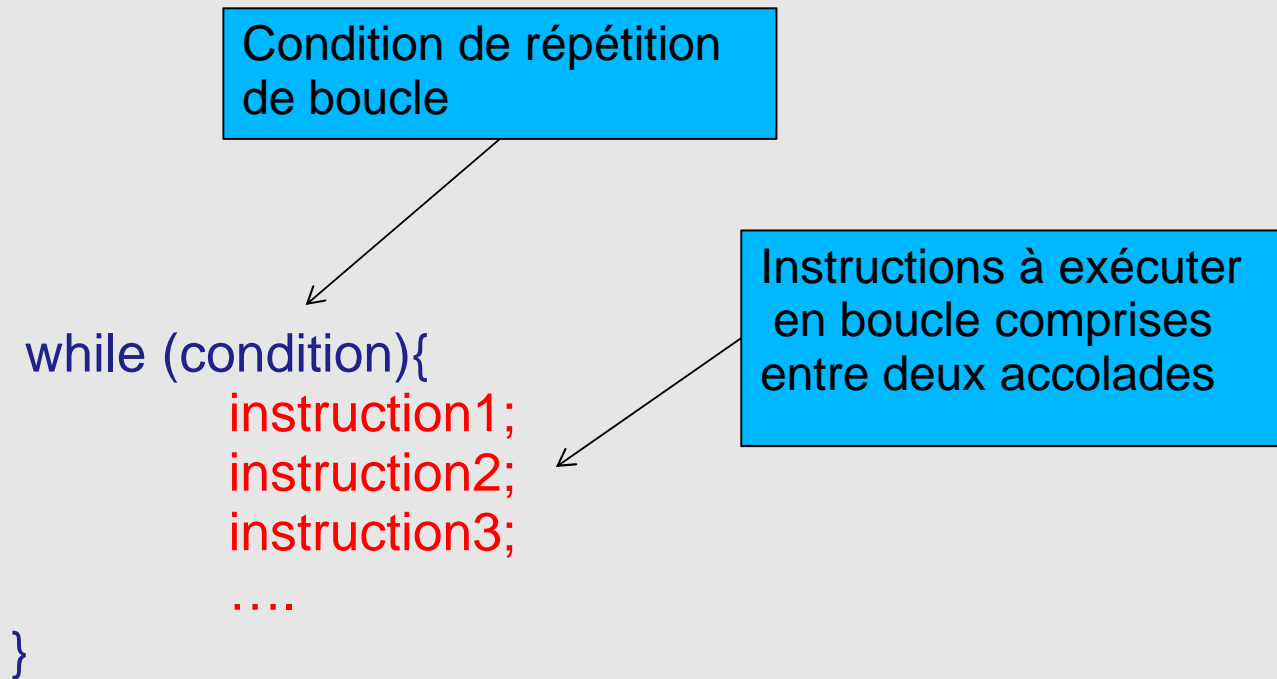
La boucle **while** permet de répéter un nombre quelconque de fois (0 ou plus) l'exécution d'une instruction ou d'un bloc d'instructions.

Le nombre de répétitions n'est pas forcément connu au moment d'entrer dans la boucle.



STRUCTURES DE CONTROLE

Les Boucles : **La Boucle while**





STRUCTURES DE CONTROLE

Les Boucles : **La Boucle while**

```
int nombre=0 ;  
  
while (nombre <=0) {  
    System.out.println("Taper un nombre positif");  
    nombre = In.readInt();  
  
}  
.....
```

Le programme évalue la condition.

Si la condition est vérifiée il exécute la boucle.

Il recommence ensuite jusqu'à ce que la condition soit fausse



STRUCTURES DE CONTROLE

Les Boucles : **while /do while**

- Exercice d'application

Réaliser une application qui permet de faire deviner à l'utilisateur, un nombre aléatoire compris entre 0 et 9 généré par le programme, en utilisant la boucle do-while puis la boucle while

Génération d'un nombre aléatoire:

```
Random rd = new Random();
```

```
int nombreAleatoire = rd.nextInt(10);
```

.