1. 循環結構的四要素

1 初始化條件

2 循環條件 --->是boolean類型

3 循環體

4 疊代條件  
說明:通常情況下，循環結束都是因為2中循環條件返回false了.

1. 三種循環結構

2.1.for循環的結構

for(1;2;4){

3

}

執行過程：1 - 2 - 3 - 4 - 2 - 3 - 4 - ... – 2

* 1. while循環的結構

1

while(2){

3;

4;

}

執行過程：執行過程：1 - 2 - 3 - 4 - 2 - 3 - 4 - ... – 2

說明：寫while循環千萬小心不要丟了疊代條件。一旦丟了，就可能導致死循環！

for和while循環總結：

1. 開發中,基本上我們都會從for、while中進行選擇,實現循環結構.
2. for循環和while循環是可以相互轉換的！

區別：for循環和while循環的初始化條件部分的作用範圍不同。  
(初始條件出了for之後沒作用,while可以)

3. 我們寫程序，要避免出現死循環。

2.3.do-while循環結構

1

do{

3;

4;

}while(2);

執行過程：1 - 3 - 4 - 2 - 3 - 4 - ... - 2

說明：

1.do-while循環至少會執行一次循環體！

2.開發中，使用for和while更多一些。較少使用do-while

1. “無限循環”結構：while(true) 或 for(;;)  
   總結:如何結束一個循環結構?  
   方式一:當循環條件是false時  
   方式二:在循環體中,執行break
2. 嵌套循環  
   1. 嵌套循環：將一個循環結構A聲明在另一個循環結構B的循環體中,就構成了嵌套循環  
    內層循環：循環結構A  
   　 外層循環：循環結構B

2. 說明

1 內層循環結構遍歷一遍，只相當於外層循環循環體執行了一次

2 假設外層循環需要執行m次，內層循環需要執行n次。此時內層循環的循環體一共執行了  
 m \* n次

3. 技巧：外層循環控制行數，內層循環控制列數

◎典型題目

練習一：

/\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*/

**for**(**int** j = 1;j <= 4;j++ ){

**for**(**int** i = 1;i <= 6;i++){

System.out.print('\*');

}

System.out.println();

}

練習二：

/\* i(行號) j(\*的個數)

\* 1 1

\*\* 2 2

\*\*\* 3 3

\*\*\*\* 4 4

\*\*\*\*\* 5 5

\*/

**for**(**int** i = 1;i <= 5;i++){//控制行數

**for**(**int** j = 1;j <= i;j++){//控制列數

System.out.print("\*");

}

System.out.println();

}

練習三：九九乘法表

練習四：100以內的質數

補充:衡量一個功能代碼的優劣  
1.正確性 2.可讀性 3.健壯性   
4.高效率與低存儲:時間複雜度、空間複雜度(衡量算法的好壞)

break和continue關鍵字的使用

使用範圍 循環中使用的作用(不同點) 相同點

break: switch-case

循環結構中 結束當前循環 關鍵字後面不能聲明執行語句

continue: 循環結構中 結束當次循環 關鍵字後面不能聲明執行語句

補充:帶標籤的break和continue的使用

return在方法中講