java_4

本节内容：

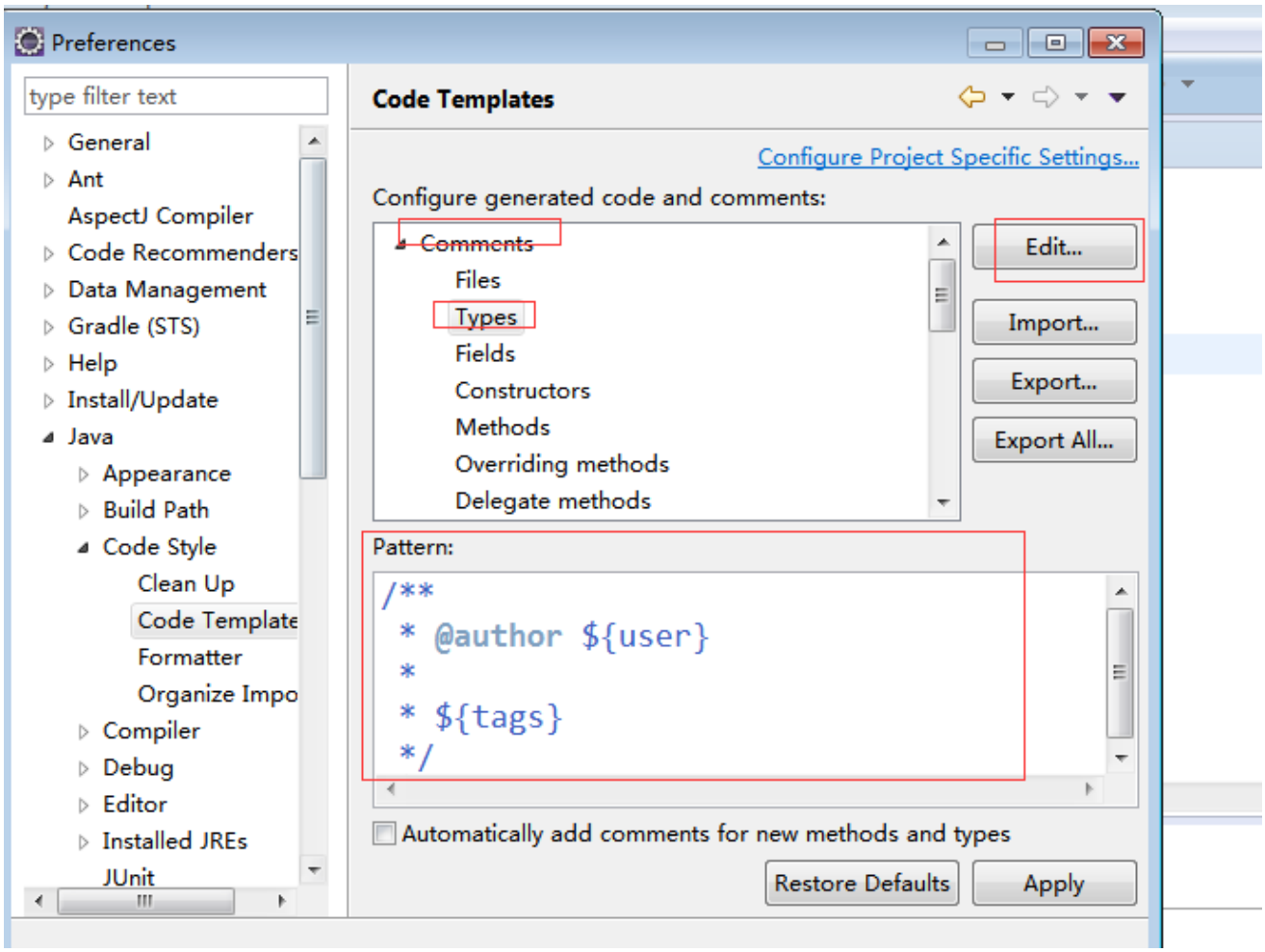
类注释模板设置

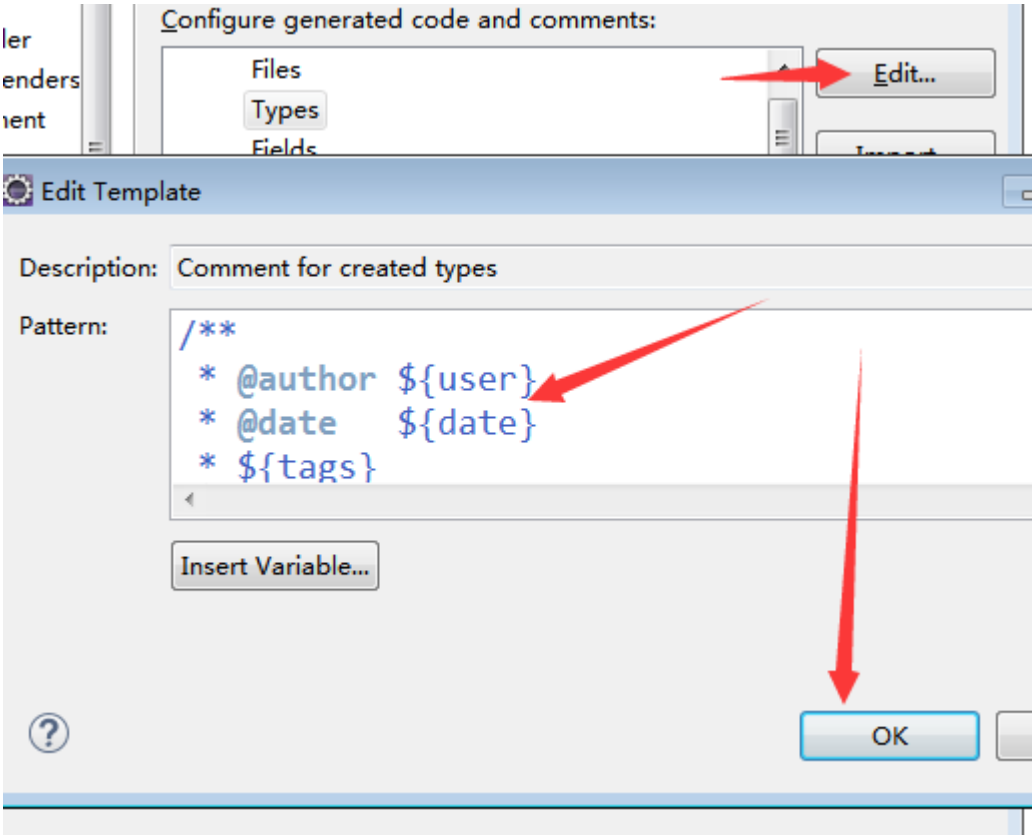
循环进阶

附类注释模板设置：

1).设置：点击菜单栏上的Window -->Preferences-->Java-->Code Style -->Code Templates

2).添加需要类注释的方式：






```
/**
 * 
 * @author Administrator
 * @date   2019年1月12日
 */
```

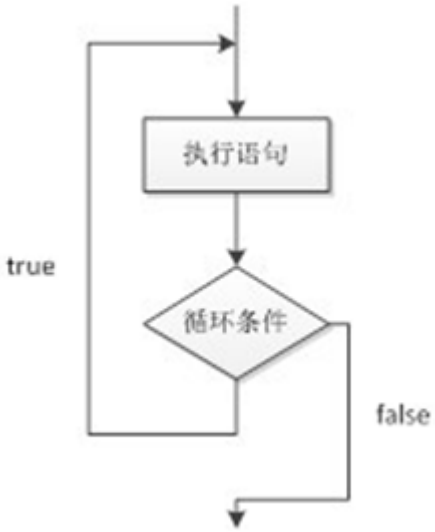
while循环语句

while循环语句和选择结构if语句相似，都是根据条件判断来决定是否执行大括号内的执行语句。区别在于，while语句会反复地进行条件判断，只要条件成立，大括号{}内的执行语句就会打印输出，直到条件不成立while循环才停止结束。while循环语句的语法结构如下：

```
while(循环条件){
    执行语句（条件成立执行多次）
    ......
}
```

上面的语法结构中，大括号{}中的执行语句被称作循环体（循环：一直执行，如同地球一样不停的圆行运动），循环体是否执行取决于循环条件。当循环条件为true时，循环体就会执行(死循环)。循环体执行完毕时会继续判断循环条件，如条件仍为true则会继续执行，直到循环条件为false时，整个循环过程才会结束。

while循环的执行流程如图：



while输出1到10的自然数,代码编写：

```
/**
 * while:输出1到10的自然数
 * @author dengjy
 * @date   2019年1月12日
 */
public class WhileNum {

    public static void main(String[] args) {
        showOneToTen();
    }

    public static void showOneToTen() {
        //定义变量名称为num，初始值为1
        int num =1;
        while(num <= 10){//循环条件
            System.out.println("输出数字为：" + num);
            //变量名称num的后自增
            num ++;
        }

    }

}
```

上述代码中，num初始值为1，在满足循环条件num <= 10的情况下，循环体不停的重复执行输出，并让变量名称num进行数字的自增，不断执行输出数据。

*：代码num++用于在每次循环时改变变量num的值，从而达到最终改变循环条件的目的。如果没有这行代码，整个循环会进入无限循环(俗称的死循环)的状态，永远不会结束。

```
    //操场跑步，停止次数传入值
    public static void runing(int runSize,String name){
        int i = 0;
        while(i < runSize){
            //System.out.println(i);
            System.out.println(name +"在学校的操场上跑第"+i+"圈");
            i++;
        }
        System.out.println(name + "总共跑了"+runSize+"圈");
    }
```

---

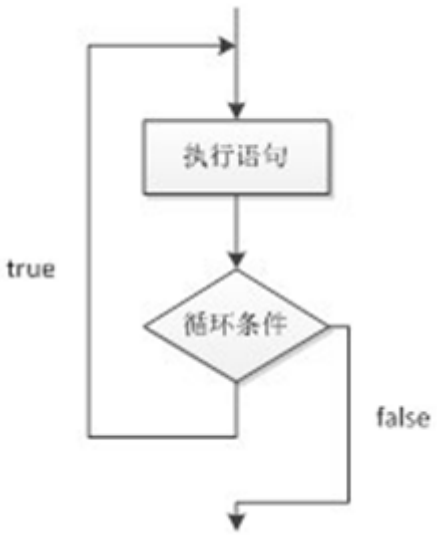
do...while循环语句

do...while循环语句和while循环语句功能几乎类似，语法结构如下：

```
do {
    执行语句
    ......
} while(循环条件);
```

上面的语法结构中，关键字do后面{}中的执行语句是循环体。do...while循环语句将循环条件放在了循环体的后面，意味着循环体会无条件执行一次，然后再根据循环条件来决定是否继续执行。

do...while循环的执行流程如图：



do...while循环输出操场跑步，停止次数传入值,代码编写：

```
/**
 * do...while循环
 * @author dengjy
 * @date   2019年1月12日
 */
```

```
public class DoWhileNum {

    public static void main(String[] args) {
        showOneToTen();
    }

    public static void showOneToTen() {
        //定义变量名称为num，初始值为1
        int num = 1;
        do{
            System.out.println("输出数字为：" + num);
            num++;//变量名称num的后自增
        }while(num <= 10);//循环条件
    }

}
```

运行结果与前面while方式一样。

发现WhileNum.java与DoWhileNum.java运行的结果一致。说明do ...while循环和while循环都是实现同样的功能。然而在程序运行过程中，这两种语句还是有差别的。如果循环条件在循环语句开始时就不成立，那么while循环的循环体一次都不会执行，而do...while循环的循环体最少会执行一次。如将DoWhileNum.java中的循环条件num <= 10改为x < 1，至少执行输出结果一次。

---

for循环语句(掌握)

for循环语句是最常用的循环语句（必须要求掌握，开发工作当中使用最多），一般用在循环次数已知的情况下，for循环语句的语法格式如下：

```
for（初始化表达式；循环条件；操作表达式）{
    执行语句
    ......
}
```

上面的语法结构中，for关键字后面()中包括了三部分内容：**初始化表达式、循环条件和操作表达式**，它们之间用";"分隔，大括号{}中的执行语句为循环体的执行输出结果。

接下来分别用①表示初始化表达式、②表示循环条件、③表示操作表达式、④表示循环体，通过序号来具体分析for循环的执行流程，如下：

```
for（① ; ② ; ③）{
   ④
}
```

```
第一步，执行①
第二步，执行②，判断结果为true，执行第三步，如判断结果为false，执行第五步
第三步，执行④
第四步，执行③，然后重复执行第二步
第五步，退出循环
```

*：可以使用断点执行方式来查看执行流程过程

for循环输出1~size的求和，代码编写：

```
    public static void showNum(int size) {
        int sum = 0;
        for(int i = 1; i <= size;i++){
            sum = sum +i;//sum + = i
        }
        System.out.println(sum);
    }
```

无限循环(也称之为死循环)

无限循环格式：

```
while(true){
    System.out.println("永远执行输出");
}
```

或

```
for(;;){//for(int i = 0;;i++){
    System.out.println("for永远执行不停止");
}
```

无限循环存在的原因是并不知道循环多少次，而是根据某些条件，来控制循环。

*：开发当中如果使用看需求是否需要使用，否则永远无限的循环执行输出（如服务器自然希望无限循环下去）。

循环嵌套

嵌套循环是指在一个循环语句的循环体中再定义一个循环语句的语法结构。while、do...while、for循环语句都可以进行嵌套[嵌套太多，维护就会麻烦]，可以互相嵌套语法语句，最常见的在for循环中嵌套for循环，格式如下：

```
for(初始化表达式; 循环条件; 操作表达式) {
    ......
    for(初始化表达式; 循环条件; 操作表达式) {
        执行语句
        ......
    }
    ......
}
```

双for循环输出“*”直角三角形，代码编写:

```
/**
 * 双重for循环(直角三角形)
 * @author djy
 * @date   2019年7月4日
 * @version
 */
public class TwoFor {

    public static void main(String[] args) {
        for(int i = 1 ;i <= 9;i++){//外层循环(行数)
            //System.out.println("第"+i+"*");
            for(int j =1 ;j <= i;j++){//内层循环(输出*)
                //System.out.print("第"+j+"#");
                System.out.print("*");
            }
            System.out.println();//换行
        }
    }

}
```

运行结果：

```
*
**
***
****
*****
******
*******
********
*********
```

上述代码定义两层for循环，分别为外层循环和内层循环，外层循环用于控制打印的行数，内层循环用于打印输出星，每一行的“*”个数逐行增加，最后输出一个直角三角形。执行流程，使用断点查看方式详，操作过程。

拓宽：
[思维题]
https://www.cnblogs.com/wzn520/p/11052098.html

[兔子猴子题]
https://blog.csdn.net/Alias_fa/article/details/52985112