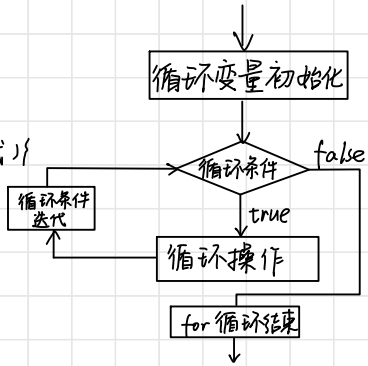
2021. 4. 26

循环控制 — for 循环

```
for (循环变量初始化;循环条件;循环变量迭代){
        循环操作(可多条语句);
}
```



for执行流程(内存分析法):

```
内存  i=1
    if (i<=2){
        执行
    }
      i++; // i=2
    if (i<=2){
        执行
    }
      i++, // i=3→跳出
```

for 循环注意事项和细节说明:

1) 循环条件是返回一个布尔值的表达式

2) for(;循环判断条件;)中初始化和变量迭代可以写到其他地方,但';'不能省. 例:

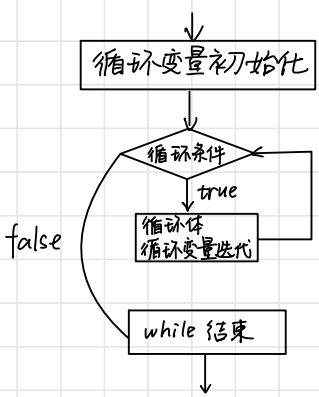
```
i=1;
for(;i<=10;){
    print("ok");
    i++;
}
```

好处:变量i的作用域扩大了.
若 for(int i=1; i<=10; i++){}
则i的作用域只在for循环体内.

3) 循环初始值可以有多条初始化语句,但要求类型一样,并且中间用逗号隔开,循环变量迭代也可以有多条变量迭代语句,中间用逗号隔开。

循环控制 — while 循环

```
循环变量初始化;
while(循环条件){
        循环体(语句);
        循环变量迭代;
}
```



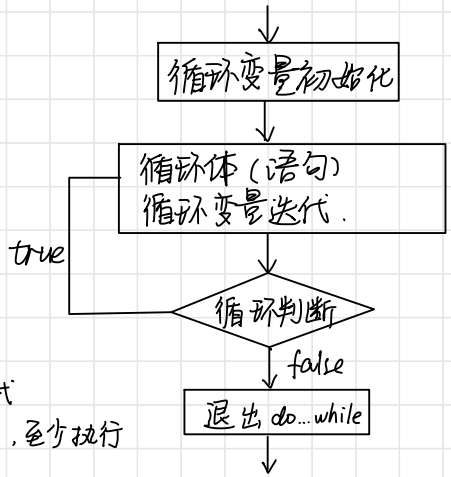
while 循环注意事项和细节

1) 循环条件是返回布尔值的表达式

2) while循环是先判断再执行语句

循环控制 — do while

```
循环变量初始化;
do {
    循环体(语句);
    循环变量迭代;
} while (循环条件);
```



do while 注意事项和细节说明

1) 循环条件是返回一个布尔值的表达式

2) do... while 循环是先执行,再判断,至少执行1次

多重循环控制(难点!重点!)

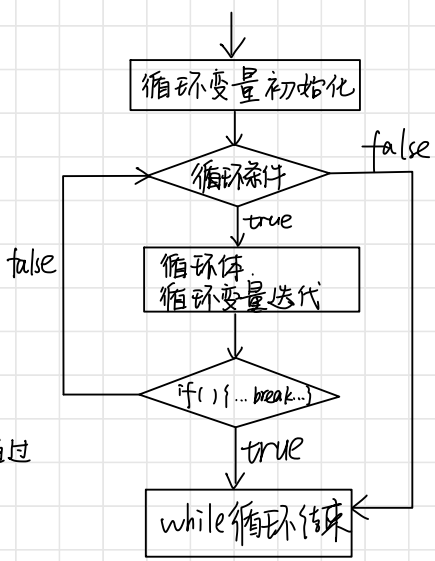
嵌套循环.

跳转控制语句 - break

在 switch、for、while 和 do while 中均可使用

```
{ ...
    break;
...
}
```

以 while 为例画出流程图



break 注意事项和细节说明:

1) break 语句出现在多层嵌套循环中时,可以通过标签指明要终止的是哪一层语句块.

2) 标签的基本使用

```
label 1:{ ...
label 2:    { ...
label 3:        {...
                    break label2;
                    ...
                }
            }
        }
```

实际开发中,尽量不要使用标签;如果没有指定标签,则默认退出最近的循环.

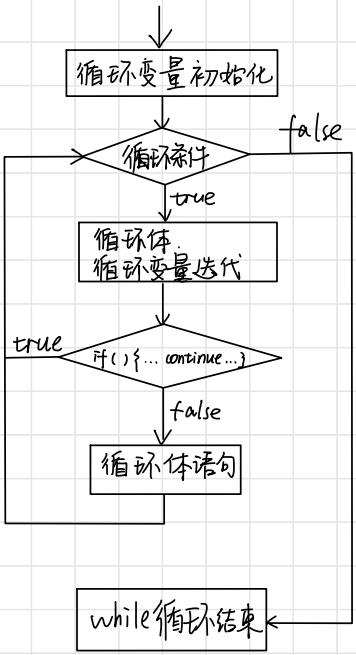
跳转控制语句 - continue

1) continue 用于结束本次循环，继续执行下一次循环

2) continue 语句也可使用标签达到跳转指定循环的效果

```
{ ...
  continue;
  ...
}
```

以 while 使用 continue 为例，画出示意图



跳转控制语句 - return

return 语句多用在自定义方法中，表示退出该方法，若 return 写在 main 方法中，则退出程序.