

循环控制

素数

- 只能被1和自己整除的数，不包括1
- 2, 3, 5, 7, 11, 13, 17, 19....

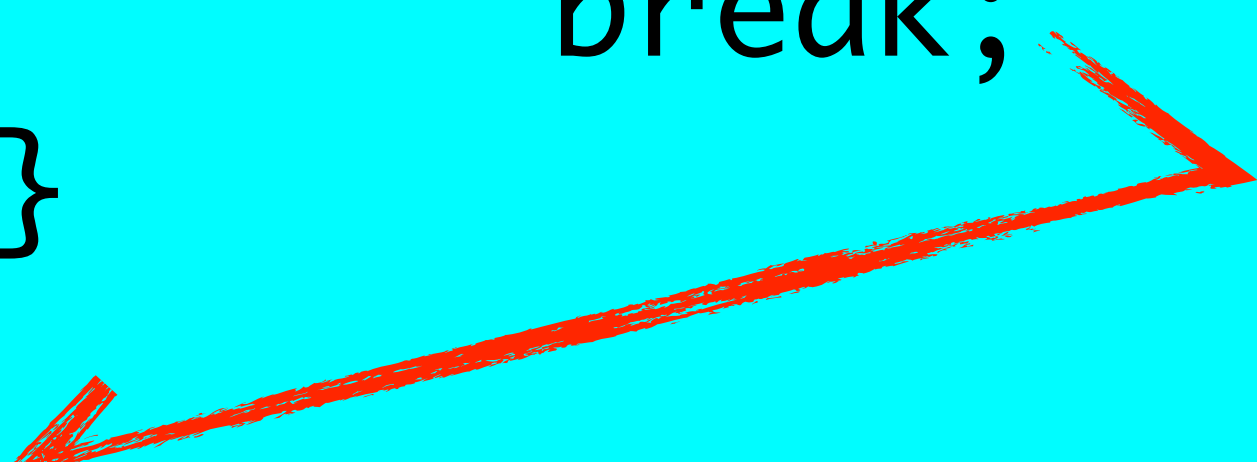
```
int isPrime = 1;
for ( int i = 2; i<n; i++ )
{
    if ( n % i == 0 )
    {
        isPrime = 0;
        System.out.println(n+"不是素数,i="+i);
        break;
    }
}
if ( isPrime == 1 )
{
    System.out.println(n+"是素数");
}
else
{
    System.out.println(n+"不是素数");
}
}
```

```
int sum = 0;
int number = 0;

while (number < 20) {
    number++;
    sum += number;
    if (sum >= 100)
        break;
}
```

```
System.out.println("The number is " +
number);
System.out.println("The sum is " + sum);
```

```
int sum = 0;  
int number = 0;  
  
while (number < 20) {  
    number++;  
    sum += number;  
    if (sum >= 100)  
        break;  
}
```

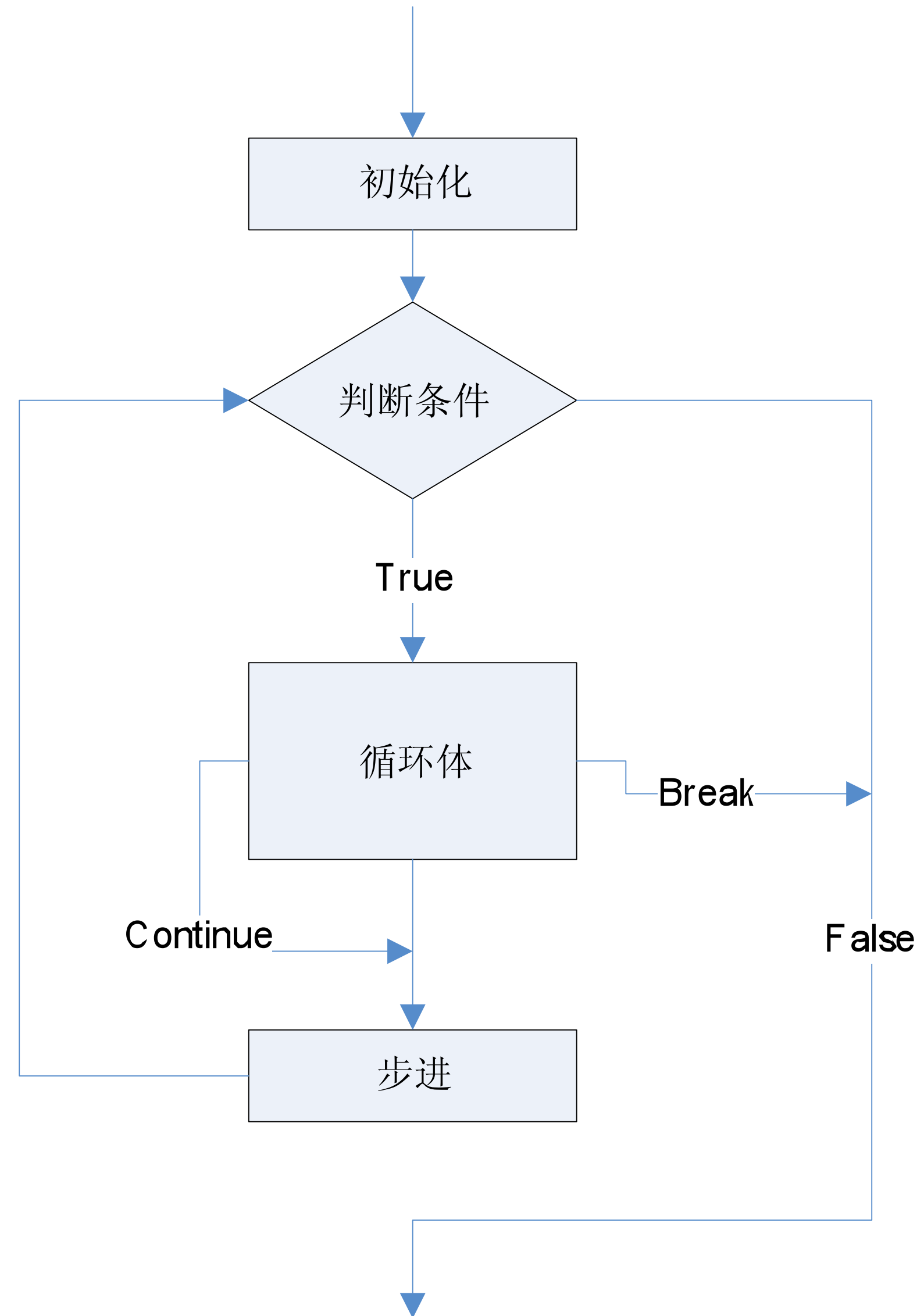


```
System.out.println("The number is " +  
number);  
System.out.println("The sum is " + sum);
```

break vs continue

- break: 跳出循环
- continue: 跳过循环这一轮剩下的语句进入下一轮

break vs continue

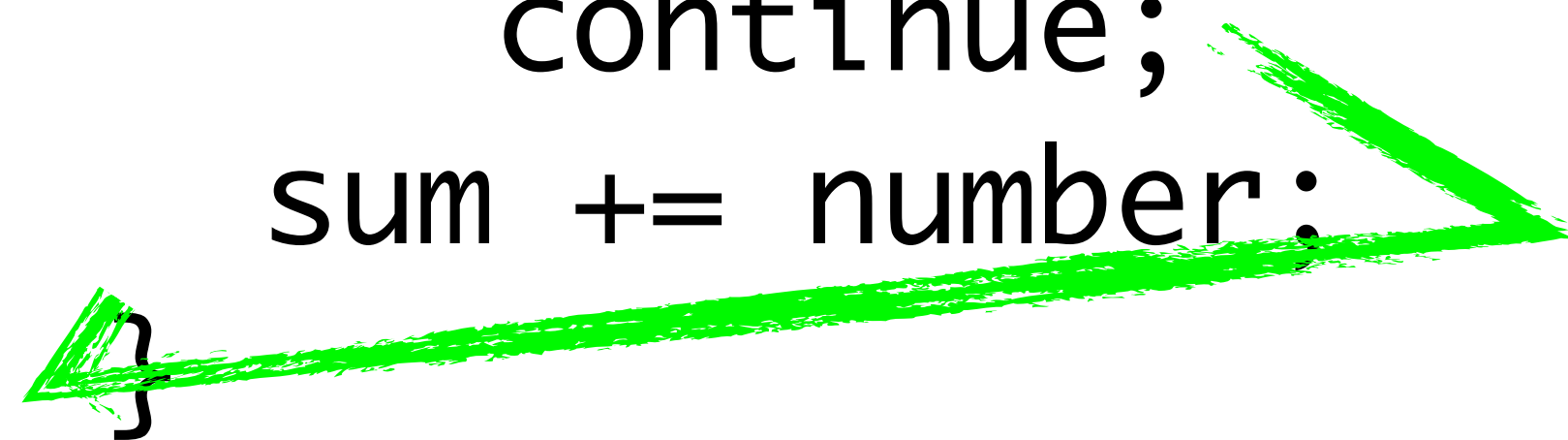


```
while (number < 20) {  
    number++;  
    if (number == 10 || number == 11)  
        continue;  
    sum += number;  
}
```

```
System.out.println("The sum is " + sum);
```



```
while (number < 20) {  
    number++;  
    if (number == 10 || number == 11)  
        continue;  
    sum += number;  
}
```



```
System.out.println("The sum is " + sum);
```

嵌套的循环

100以内的素数

- 如何写程序输出100以内的素数？

嵌套的循环

- 循环里面还是循环

```
for ( int n = 2; n<100; n++ )
{
    int isPrime = 1;
    for ( int i = 2; i<n; i++ )
    {
        if ( n % i == 0 )
        {
            isPrime = 0;
            break;
        }
    }
    if ( isPrime == 1 )
    {
        System.out.print(n+" ");
    }
    else
    {
        //      System.out.println(n+"不是素数");
    }
}
```

多重循环

```
for ( int n1=0; n1<10; ++n1 )  
{  
    for ( int n2=0; n2<50; ++n2 )  
    {  
        System.out.println(n1+":"+n2);  
    }  
}
```

九九乘法表

```
for (int i = 1; i <= 9; i++)  
{  
    for (int j = 1; j <= 9; j++)  
    {  
        System.out.println("\t" + (i * j));  
    }  
    System.out.println();  
}
```

前50个素数

- 如何写程序输出前50个素数

嵌套循环时的break

凑硬币

- 如何用1角、2角和5角的硬币凑出10元以下的金额呢？

```

Scanner in = new Scanner(System.in);
int amount = in.nextInt();
for ( int one = 0; one <=amount; ++one )
{
    for ( int five = 0; five <= amount/5; ++five )
    {
        for ( int ten = 0; ten <= amount/10; ++ten )
        {
            for ( int twenty = 0; twenty <= amount/20; ++twenty )
            {
                if ( one+five*5+ten*10+twenty*20 == amount )
                {
                    System.out.println(one+"张1元,"+five+"张5元,"+ten+"张10元,"+twenty+"张20元-
>" +amount);
                }
            }
        }
    }
}

```

从嵌套中break?

```
for (;;) {  
    for (;;) {  
        if ( ... ) break;  
    }  
}
```

break和continue

- 只能对它所在的那层循环做

接力break

```
boolean exit=false;
for (;;) {
    for (;;) {
        if ( ... ) { exit=true; break; }
    }
    if ( exit ) break;
}
```

逻辑类型

- 关系运算的结果是一个逻辑值，true或false。这个值可以保存在一个对应的逻辑类型的变量中，这样的变量类型是boolean
- 布尔（Boolean）是为了纪念George Boole对逻辑计算的贡献
 - `boolean flag = true;`
 - `boolean tooHigh, tooSmall, tooRough;`
 - `boolean done = false;`

goto

```
for (;;) {  
    for (;;) {  
        if ( ... ) goto OUT;  
    }  
}  
  
OUT: ...
```


Break和continue

- 在循环前可以放一个标号来标示循环：
- label:
- 带标号的break和continue对那个循环起作用

```
public class Cash2 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int amount;
        do {
            System.out.println("请输入金额(1-100): ");
            amount = scan.nextInt();
        } while ( amount <1 || amount > 100 );
        Outer:
        for ( int one = 0; one <=amount; ++one )
            for ( int five = 0; five <= amount/5; ++five )
                for ( int ten = 0; ten <= amount/10; ++ten )
                    for ( int twenty = 0; twenty <= amount/20; ++twenty )
                        if ( one+five*5+ten*10+twenty*20 == amount ) {
                            System.out.println(one+"张1元,"+five+"张5元,"+ten+"张10
元,"+twenty+"张20元->" +amount);
                            break Outer;
                        }
                }
            }
    }
```

逻辑运算

```
int isPrime = 1;
for ( int i = 2; i<n; i++ )
{
    if ( n % i == 0 )
    {
        isPrime = 0;
        System.out.println(n+"不是素数,i="+i);
        break;
    }
}
if ( isPrime == 1 )
{
    System.out.println(n+"是素数");
}
else
{
    System.out.println(n+"不是素数");
}
}
```

逻辑类型

- 关系运算的结果是一个逻辑值，true或false。这个值可以保存在一个对应的逻辑类型的变量中，这样的变量类型是boolean
- 布尔（Boolean）是为了纪念George Boole对逻辑计算的贡献
 - `boolean flag = true;`
 - `boolean tooHigh, tooSmall, tooRough;`
 - `boolean done = false;`

逻辑运算

- 逻辑运算是对逻辑量进行的运算，只有逻辑量可以参与运算

运算符	描述	示例	结果
!	逻辑非	!a	如果a是true结果就是false， 如果a是false结果就是true
&&	逻辑与	a && b	如果a和b都是true，结果就是true； 否则就是false
	逻辑或	a b	如果a和b有一个是true，结果为true； 两个都是false，结果为false

TRY

- 如果要表达数学中的区间，如： $x \in (4,6)$ 或 $x \in [4,6]$ ，应该如何写Java的表达式？

TRY

- 如果要表达数学中的区间，如： $x \in (4,6)$ 或 $x \in [4,6]$ ，应该如何写？

像 $4 < x < 6$ 这样的式子，不是Java能接受的式子，因为 $4 < x$ 的结果是一个逻辑值，逻辑值是不能和数值6做关系计算的

TRY

`x > 4 && x < 6`

- 如果要表达数学中的区间，如： $x \in (4,6)$ 或 $x \in [4,6]$ ，应该如何写？

像 $4 < x < 6$ 这样的式子，不是Java能接受的式子，因为 $4 < x$ 的结果是一个逻辑值，逻辑值是不能和数值6做关系计算的

try

- `age > 20 && age < 30`
- `index < 0 || index > 99`
- `! age < 20`

优先级

- ! > && > ||
 - !done && (count > MAX)

优先级

优先级	运算符	结合性
1	()	从左到右
2	! + - ++ --	从右到左（单目的+和-）
3	* / %	从左到右
4	+ -	从左到右
5	< <= > >=	从左到右
6	== !=	从左到右
7	&&	从左到右
8		从左到右
9	= += -= *= /= %=	从右到左

try

- 下面哪个表达式的值是true?
 - `! (4<5)`
 - `! false`
 - `(2>2) || ((4==4) && (1<0))`
 - `(2>2) || (4==4) && (1<0)`
 - `(34!=33) && !false`