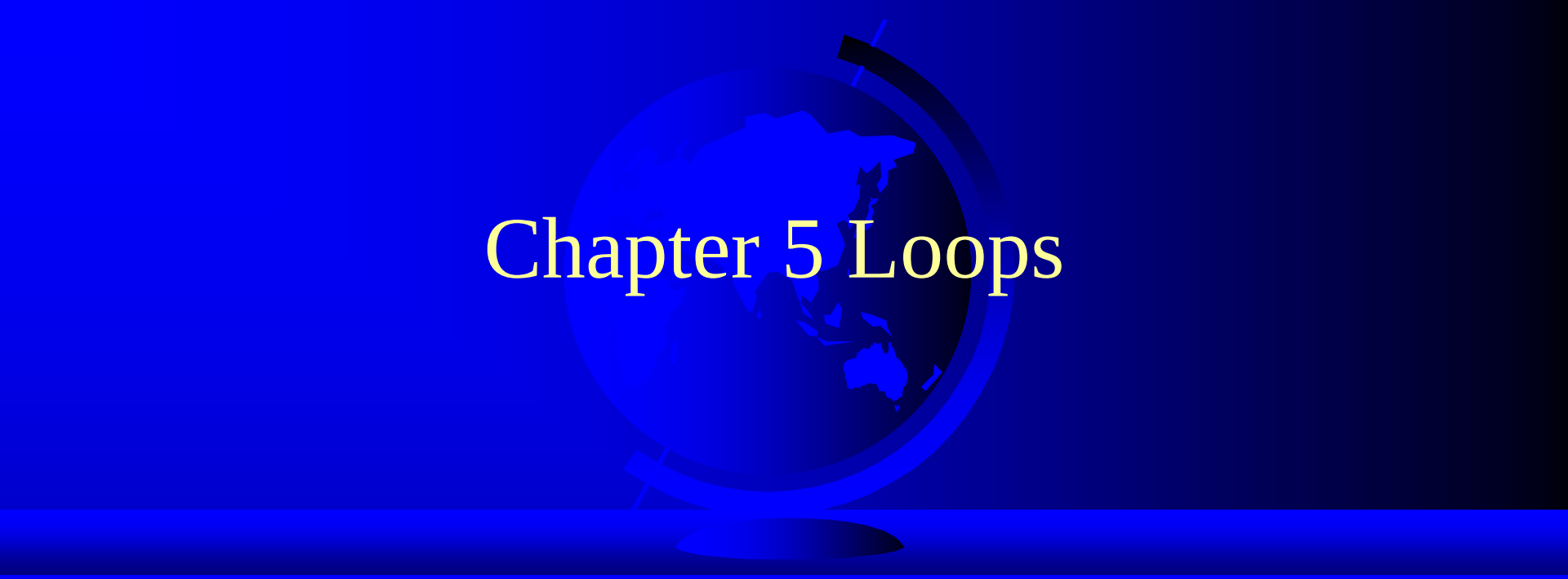
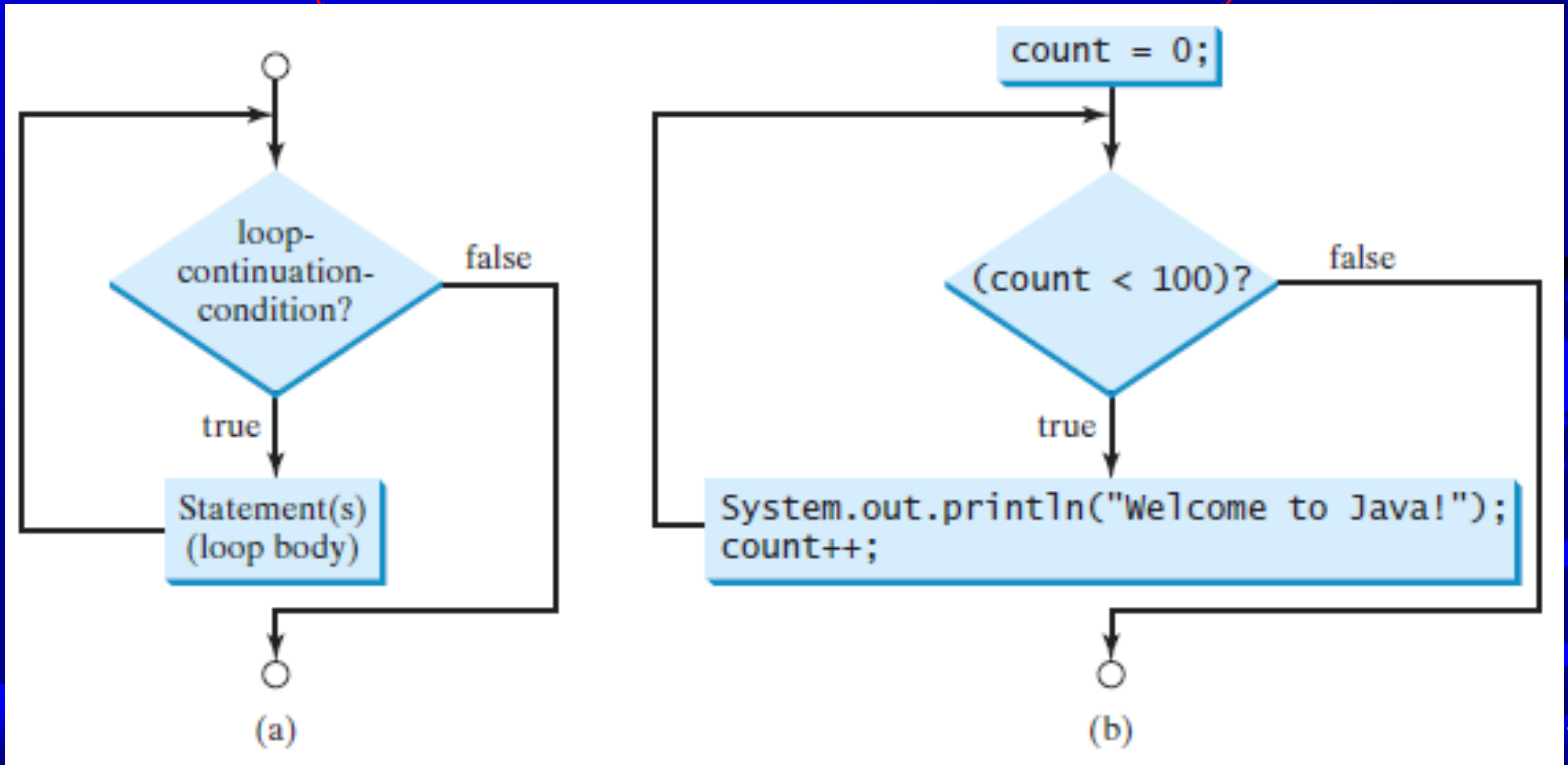


Chapter 5 Loops

While循环

```
while (loop-continuation-condition) {
  // loop-body;
  Statement(s);
}
```

```
int count = 0;
while (count < 100) {
  System.out.println("Welcome to Java!");
  count++;
}
```



单步执行一下

Initialize count

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

单步执行一下

(count < 2) is true

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

Print Welcome to Java

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

Increase count by 1
count is 1 now

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

(count < 2) is still true since count is 1

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

Print Welcome to Java

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

Increase count by 1
count is 2 now

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

(count < 2) is false since count is 2 now

单步执行一下

```
int count = 0;
while (count < 2) {
  System.out.println("Welcome to Java!");
  count++;
}
```

The loop exits. Execute the next statement after the loop.

猜数游戏

计算机随机生成0..100之间的整数，每次提示用户输入一个整数，直到猜对这个随机数为止。

提示：随机数生成可以用Math.random()，该方法可以生成一个[0,1)之间的double。所以0..100的随机数可以这样生成：

```
int number = (int)(Math.random() * 101);
```

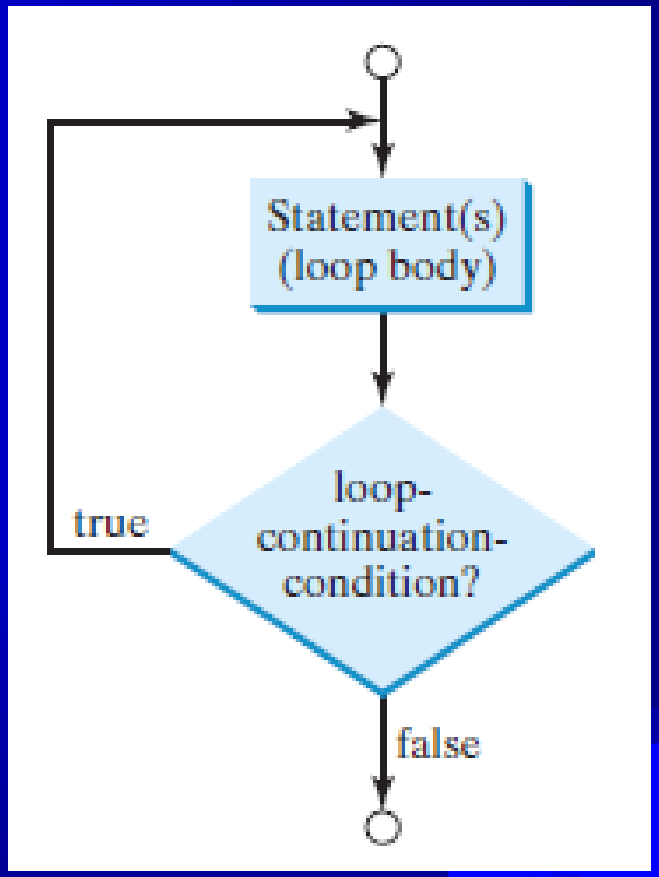
代码1/2

```
import java.util.Scanner;

public class GuessNumber {
  public static void main(String[] args) {
  int number = (int) (Math.random() * 101);

  Scanner input = new Scanner(System.in);
  System.out.println("Guess a magic number between 0
and 100");

  int guess = -1; //初始猜测值
```

代码2/2

```
  while (guess != number) {
    System.out.print("\nEnter your guess: ");
    guess = input.nextInt();

    if (guess == number)
      System.out.println("Yes, it is " + number);
    else if (guess > number)
      System.out.println("Your guess is too high");
    else
      System.out.println("Your guess is too low");
  }
 }
}
```

do-while循环

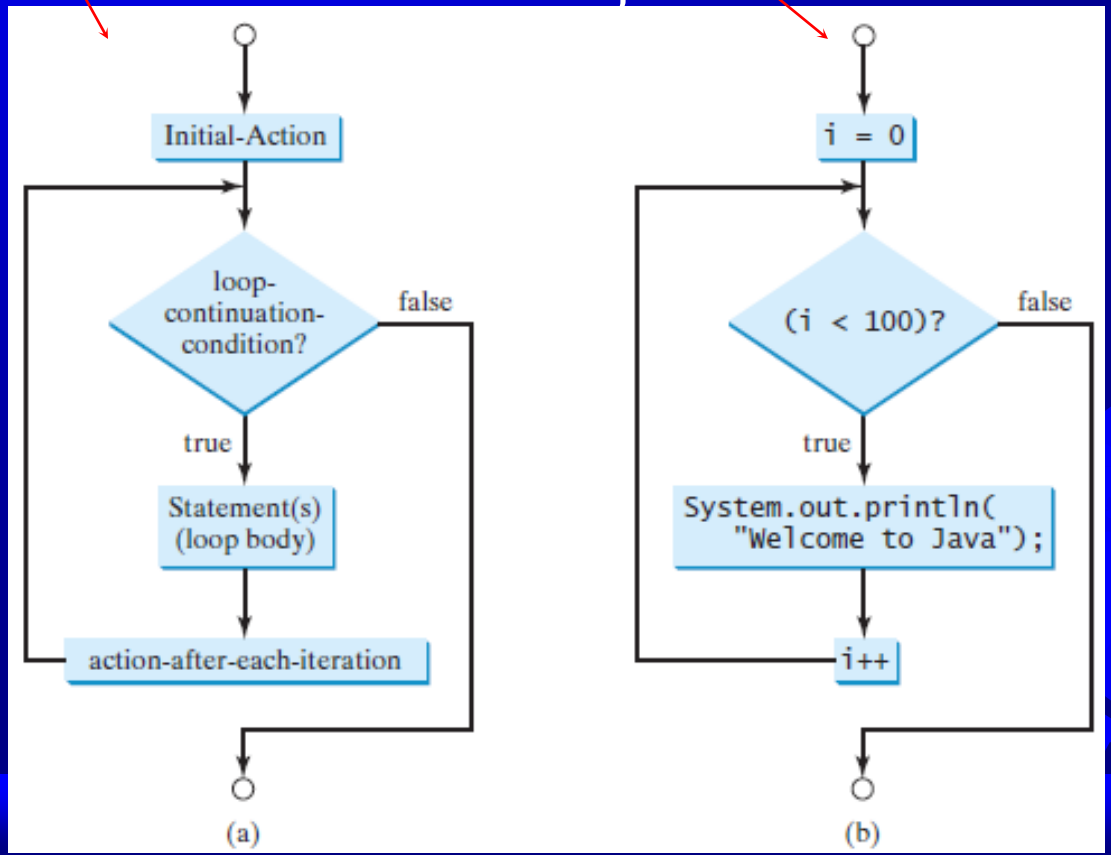



```
do {
  // Loop body;
  Statement(s);
} while (loop-continuation-condition);
```

For循环

```
for (initial-action; loop-
    continuation-condition;
    action-after-each-iteration) {
  // loop body;
  Statement(s);
}
```

```
int i;
for (i = 0; i < 100; i++) {
  System.out.println(
    "Welcome to Java!");
}
```



单步执行一下

声明i

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println(
    "Welcome to Java!");
}
```

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println(
    "Welcome to Java!");
}
```

设置i初值为 0

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println( "Welcome to Java!");
}
```

(i < 2) 为true
因为此时i为 0

单步执行一下

打印Welcome to Java

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

执行i++，现在i为 1

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

(i < 2)仍然为 true
因为i是 1

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

打印Welcome to Java

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

执行i++，现在i是 2

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java!");
}
```

(i < 2) is false
since i is 2

单步执行一下

```
int i;
for (i = 0; i < 2; i++) {
  System.out.println("Welcome to Java");
}
```

退出循环，执行for的下一个语句。

说明

for循环的表达式1和表达式3可以是任何语句，不一定和循环相关。如果有多个语句，可以使用逗号分隔。所以下面两个for语句都是正确的，尽管不太常见：

```
for (int i = 1; i < 100; System.out.println(i++));


for (int i = 0, j = 0; (i + j < 10); i++, j++) {
  // Do something
}
```

说明

for语句的表达式二是循环是否继续的条件，如果放空则表示true，相当于一个无限循环。所以下面两个写法都是无限循环。不过(b)的写法更加容易理解。

```
for ( ; ; ) {
  // Do something
}
```

(a)

Equivalent

```
while (true) {
  // Do something
}
```

(b)

注意

没事干不要在for的后面乱加分号，否则会导致for语句的循环体为空。例如下面这个例子，大括号中的语句已经不属于for循环了。

Logic Error

```
for (int i=0; i<10; i++);
{
  System.out.println("i is " + i);
}
```

另一个例子

同样的，下面这个例子也是错误的:

```
int i=0;
while (i < 10);
{
  System.out.println("i is " + i);
  i++;
}
```

Logic Error

不过对于do…while循环来说，while后的分号是一定要加的，注意和while循环的区别。

```
int i=0;
do {
  System.out.println("i is " + i);
  i++;
} while (i<10);
```

Correct

问题：找最大公约数

这是一个很简单的穷举法，从1开始往上挨个试：

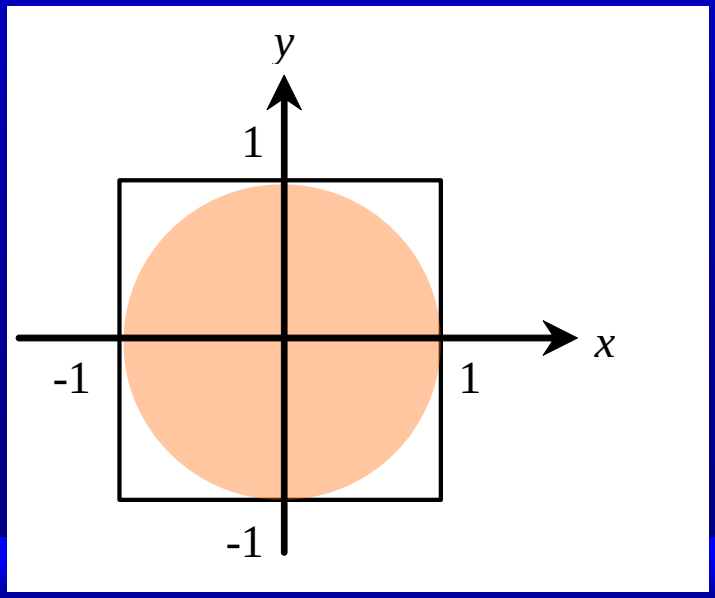
```
int gcd = 1; // Initial gcd is 1
int k = 2; // Possible gcd
while (k <= n1 && k <= n2) {
    if (n1 % k == 0 && n2 % k == 0)
        gcd = k; // Update gcd
    k++; // Next possible gcd
}
```

当然也可以选择从n1和n2中较小的那个数往下挨个试，或者专业一点，用辗转相除法。

问题：Monte Carlo模拟

Monte Carlo模拟是随机数和概率相结合的一种技术。这里我们试着用这个技术来估算π的近似值。

假设我们往正方形区域中扔一个点，那么这个点落在圆中（称为击中）的概率＝圆面积/正方形面积。显然概率对于大样本才有意义，因此我们模拟扔1000000个点，统计一下击中次数，然后估算π 的近似值。



由于 圆面积/ 正方形面积 = π / 4

因此 π 的近似值就是：

4 * 击中次数 / 1000000

实现代码

```
public class MonteCarloSimulation {
  public static void main(String[] args) {
    final int NUMBER_OF_TRIALS = 10000000;
    int numberOfHits = 0;

    for (int i = 0; i < NUMBER_OF_TRIALS; i++) {
      double x = Math.random() * 2.0 - 1;
      double y = Math.random() * 2.0 - 1;
      if (x * x + y * y <= 1)
        numberOfHits++;
    }

    double pi = 4.0 * numberOfHits / NUMBER_OF_TRIALS;
    System.out.println("PI is " + pi);
  }
}
```

break语句

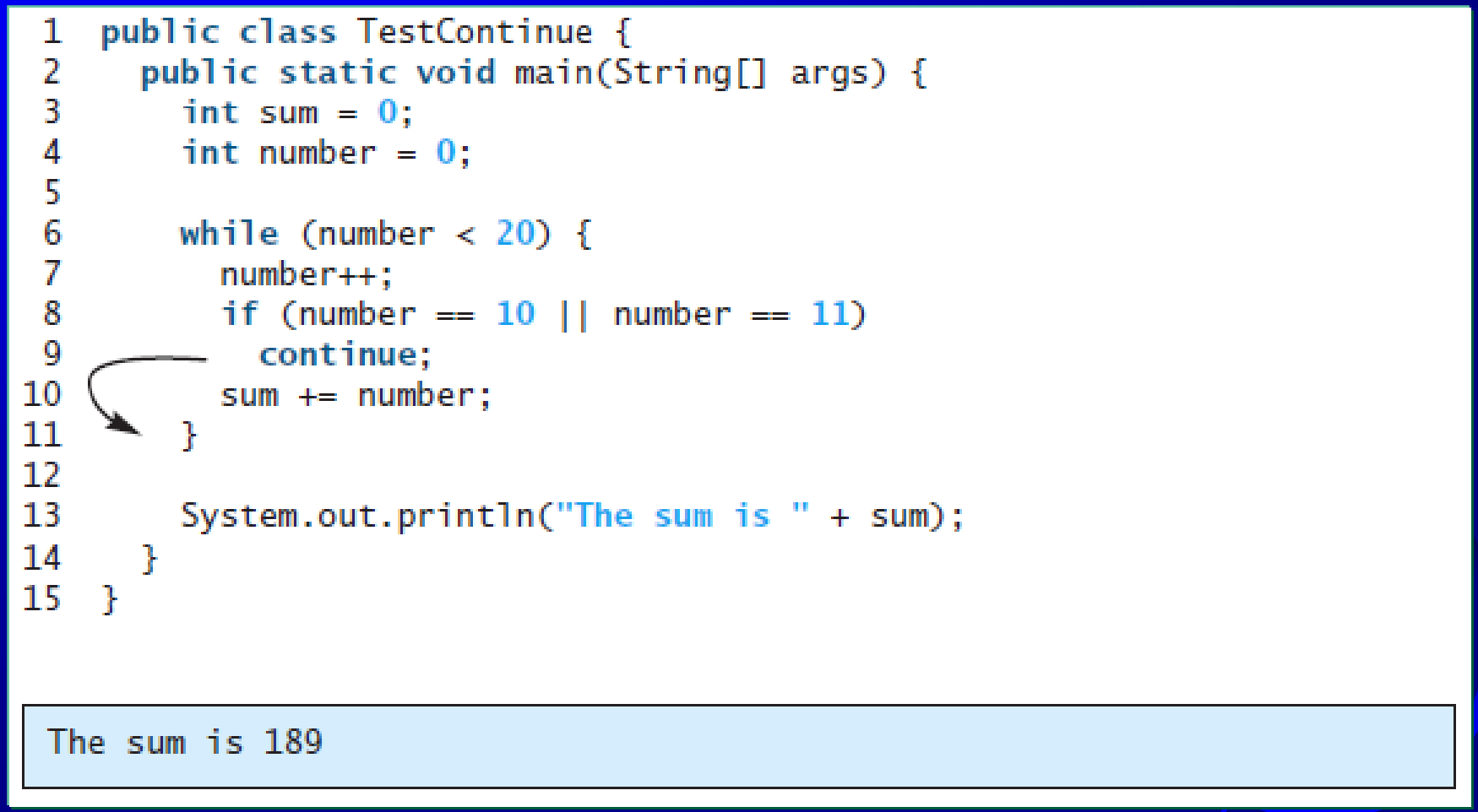
break的作用是跳出循环，如箭头所示：

```
1  public class TestBreak {
2    public static void main(String[] args) {
3      int sum = 0;
4      int number = 0;
5
6      while (number < 20) {
7        number++;
8        sum += number;
9        if (sum >= 100)
10         break;
11     }
12
13     System.out.println("The number is " + number);
14     System.out.println("The sum is " + sum);
15   }
16 }
```

```
The number is 14
The sum is 105
```

continue语句

continue的作用是结束一次循环，注意那个箭头：



```
1  public class TestContinue {
2    public static void main(String[] args) {
3      int sum = 0;
4      int number = 0;
5
6      while (number < 20) {
7        number++;
8        if (number == 10 || number == 11)
9          continue;
10       sum += number;
11     }
12
13     System.out.println("The sum is " + sum);
14   }
15 }
```

```
The sum is 189
```

问题：显示素数

编程求出前50个素数，并且按照每行10个输出到屏幕上。

解题思路：整个问题可以分为四个大步骤：

- 穷举，从2, 3, 4, 5, 6, ...挨个往上找，直到求满50个为止；
- 测试给定的数是否素数；
- 计算当前的素数个数；
- 打印素数，如果已经满10个，还要输出一个换行。

main函数关键代码—1

```
final int NUMBER_OF_PRIMES = 50; // Number of primes to display
final int NUMBER_OF_PRIMES_PER_LINE = 10; // Display 10 per line
int count = 0; // Count the number of prime numbers
int number = 2; // A number to be tested for primeness

System.out.println("The first 50 prime numbers are \n");
```

main函数关键代码—2

```
while (count < NUMBER_OF_PRIMES) {
  // Assume the number is prime
  boolean isPrime = true; // Is the current number prime?

  // Test whether number is prime
  for (int divisor = 2; divisor <= number / 2; divisor++) {
    if (number % divisor == 0) { // If true, number is not
prime
      isPrime = false; // Set isPrime to false
      break; // Exit the for loop
    }
  }
```

main函数关键代码—3

```
// Display the prime number and increase the count
if (isPrime) {
  count++; // Increase the count

  if (count % NUMBER_OF_PRIMES_PER_LINE == 0) {
// Display the number and advance to the new line
  System.out.println(number);
  } else
  System.out.print(number + " ");
}
// Check if the next number is prime
number++;
}
```

THE END