Java核心_流程控制之循环结构

课程概要

学习目标

理解循环结构的概念和分类

能够根据for循环的基本格式定义并正确使用for循环

能够根据while循环的基本格式定义并使用while循环

能够根据do...while循环的基本格式定义并正确使用while循环

能够说出for、while、do...while三种循环的特点和区别

能够区分使用break语句和continue语句进行循环的跳转

循环结构的概念和分类

现实生活中有哪些循环呢?


DELIVERY
DELIVERY
DELIVERY
DELIVERY


PRECIPITATION
SNOWMELT
TRANSPIRATION
EVAPORATION
RIVER
LAKE
UNDERGROUND WATER
OCEAN

循环结构的概念

循环，即事物周而复始的变化。 循环结构，使一部分代码按照次数或一定的条件反复执行的一种代码结构。

循环结构的分类

for循环

while循环

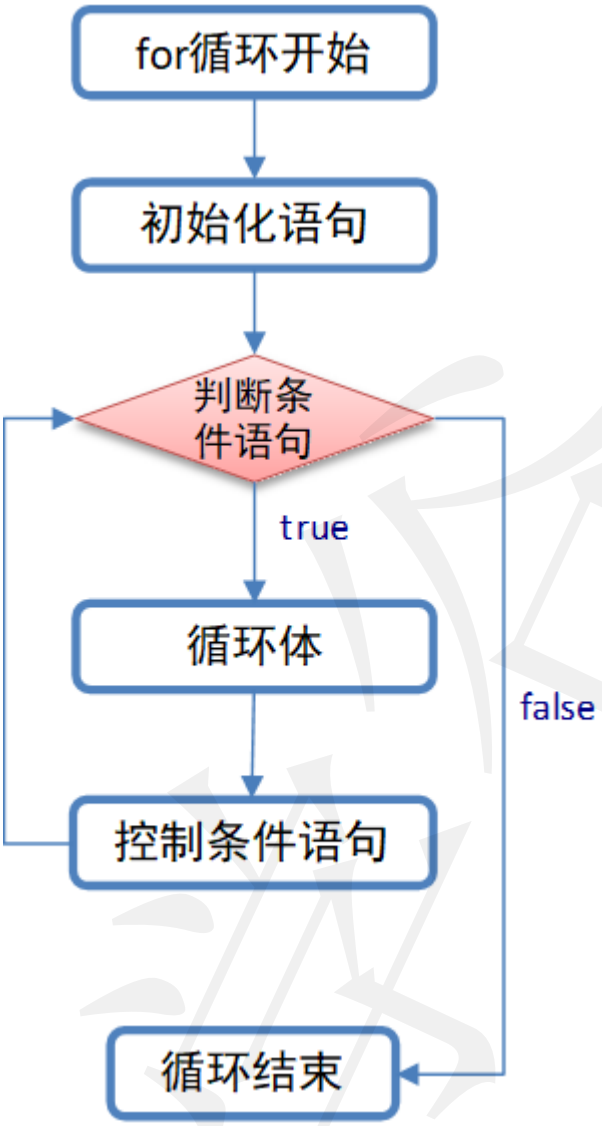
do...while循环

for循环

格式：

```
for(初始化语句； 判断条件语句； 控制条件语句) {
    // 循环体
}
```

执行流程：



for循环案例

需求：在控制台输出5次HelloWorld

代码演示

```
public static void main(String[] args) {
    // 需求：在控制台输出5次HelloWorld

    // 原始做法
    System.out.println("HelloWorld");
    System.out.println("HelloWorld");
    System.out.println("HelloWorld");
    System.out.println("HelloWorld");
    System.out.println("HelloWorld");
    System.out.println("----------");

    // 用for循环改进
    for(int x = 1; x <= 5; x++) {//  x = 6
        System.out.println("HelloWorld");
    } // 循环结束
}
```

需求：在控制台输出1-5/在控制台输出5-1

分析：

A：用原始方式实现输出1-5

B：用for循环实现输出1-5

C：用for循环实现输出5-1

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
1
2
3
4
5
----------
1
2
3
4
5
----------
5
4
3
2
1
----------

Process finished with exit code 0
```

需求：输出1-5数据之和

分析：

A：定义一个求和变量sum，初始化值是0

B：用for循环获取1-5的数据

C：把每一次获取到的数据累加到求和变量sum

sum = sum + x; 或者 sum += x;

D：输出求和变量

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
sum:15

Process finished with exit code 0
```

需求：求出1-100之间偶数和

分析：

A：定义一个求和变量sum，初始化值是0

B：获取1-100之间的数，用for循环实现

C：判断每一个数是否为偶数，是就累加，否则不做操作对2取余等于0，则为偶数： x % 2 == 0

D：for循环结束，输出求和变量的值

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
sum: 2550

Process finished with exit code 0
```

需求：在控制台输出所有的"水仙花数"

分析：

水仙花数：所谓的水仙花数是指一个三位数，其各位数字的立方和等于该数本身 举例：153是一个水仙花数： 111 + 555 + 333 = 1 + 125 + 27 = 153

步骤：

A：获取所有的三位数，即100-1000之间的数 B：获取每一个三位数的个位，十位，百位 个位：153 % 10 = 3 十位：153/10%10 = 5 百位：153/10/10%10 = 1 C：拿个位，十位，百位的立方和与该数本身进行比较，如果相等，则在控制台打印该数

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
153
370
371
407

Process finished with exit code 0
```

需求：统计所有的”水仙花数”的个数

分析：

A：定义统计变量count，即计数器，初始化值为0 B：获取所有的三位数，即100-1000之间的数 C：判断每一个三位数是否为水仙花数，是则count自增1 count ++；D：循环结束，输出计数器count的值

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
水仙花数的个数是：4

Process finished with exit code 0
```

while循环

while循环语句格式

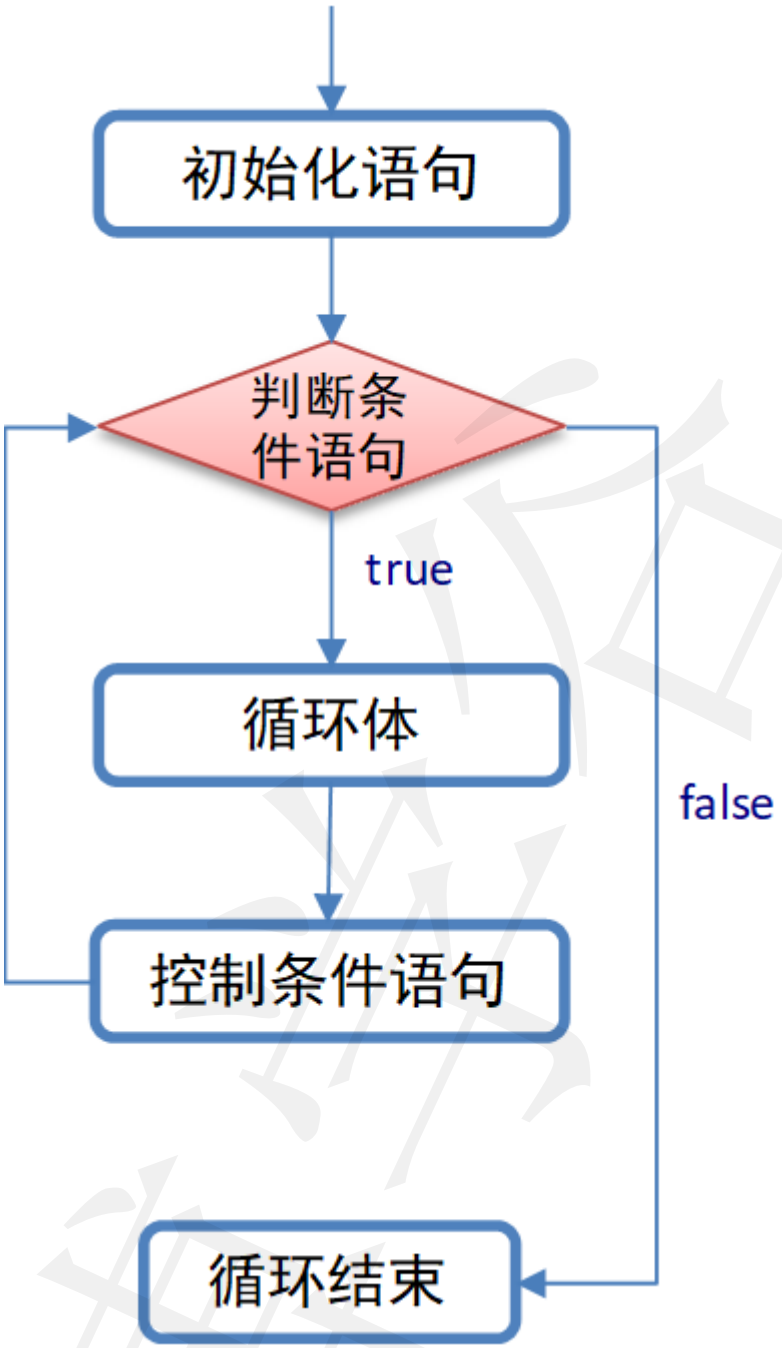
格式：

```
初始化语句；
while(判断条件语句) {
    循环体语句；
    控制条件语句；
}
```

注意事项：

初始化语句可以省略 控制条件语句可以省略

执行流程：



while循环案例

需求：使用while循环在控制台输出5次“HelloWorld”

分析：

A：用for循环实现输出“HelloWorld”

B：用while循环实现输出“HelloWorld”

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld
----------
HelloWorld
HelloWorld
HelloWorld
HelloWorld
HelloWorld
----------

Process finished with exit code 0
```

需求：求1-100之间的数据和

分析：

A：定义求和变量sum，初始化值为0

B：使用while循环获取1-100之间的数据

C：将每个数据累加到sum变量上

D：循环结束，输出sum的值

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
sum:5050


Process finished with exit code 0
```

do...while循环

do...while循环语句格式 格式：

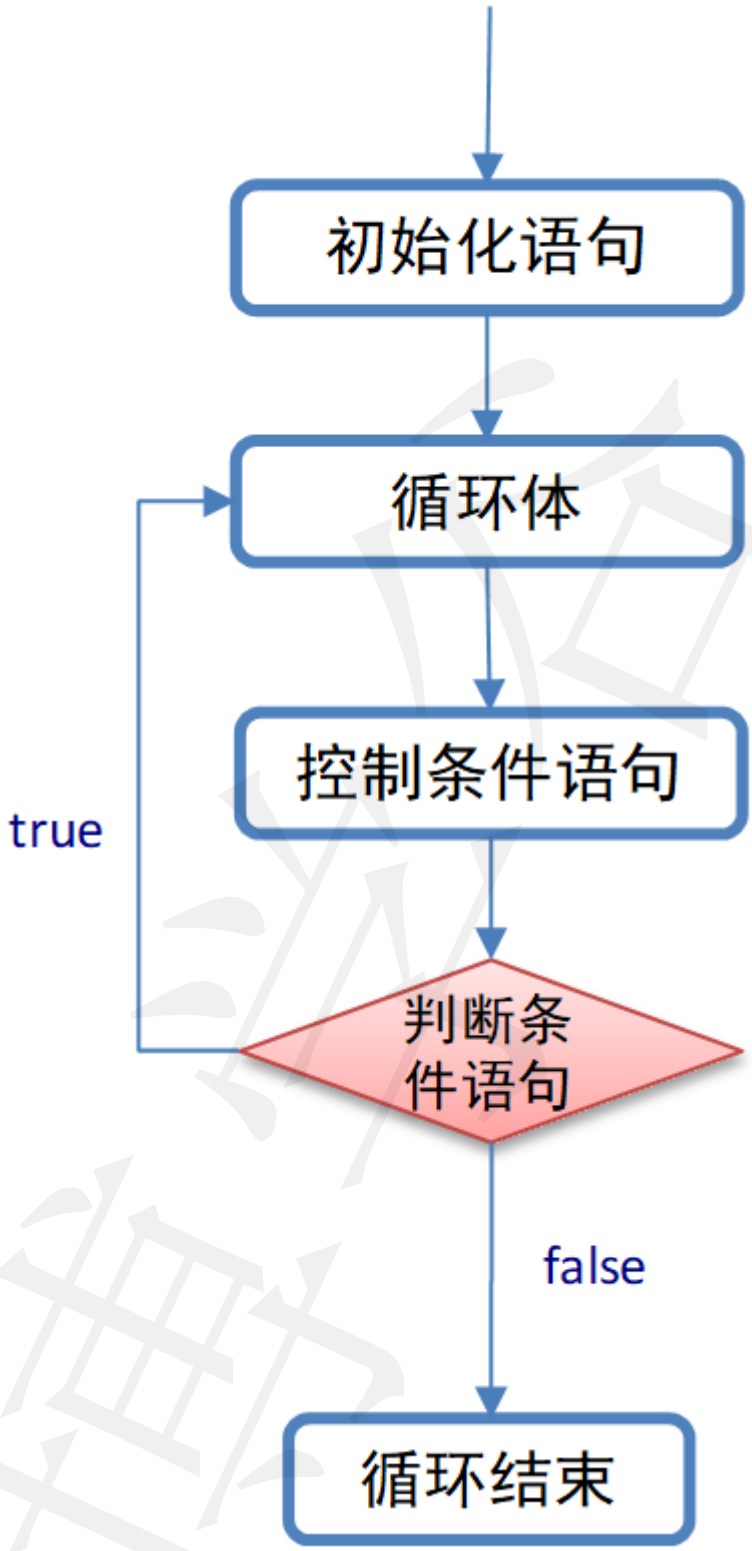
```
初始化语句;
    do {
        循环体语句;
        控制条件语句;
    } while(判断条件语句);
```

注意事项：

while小括号后的分号不可省略

do...while循环的循环体语句至少执行1遍

执行流程：



do...while循环案例

需求：用do...while循环模拟：学完一个知识，至少练习1次

分析：

A：定义int型变量count，即练习的次数，初始化值为1

B：定义boolean型变量isOK，作为一个标记，表示是否学会，初始化值为false

C：在do...while循环体中：

输出正在练习的次数

判断当练习次数不少于3时，表示已学会：isOK=true

每练习一次，次数自增1：count ++;

D：while判断是否学会

```
int count = 1; // 练习次数
boolean isOK = false; // 是否学会
do{
    System.out.println("正在进行第" + count + "次练习");
    if(count >= 3) { // 要求练习三次
        isOK = true; // 练习三次就学会了
    }
    count ++;
} while(!isOK); // 是否学会
```

三种循环的区别

死循环

两种简单的死循环

for循环

```
for(;;){
    // 循环体，根据实际需求结束循环
}
```

while循环

```
while(true){
    // 循环体，根据实际需求结束循环
}
```

break和continue

break

中断，用于switch语句和循环语句： 在switch语句中，表示结束switch代码块 在循环语句中，表示结束循环

break案例

需求：查找班级编号为3的同学（假设班级中有15位同学）

分析：

A：使用for循环先遍历班级每一个同学 B：在班级循环体中，判断同学编号是否为3 若该同学编号为3，则打印该同学编号，结束循环 若该同学编号不为3，不做任何操作

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
已经找到了编号为3的同学

Process finished with exit code 0
```

continue

继续，用于循环语句，表示结束本次循环，继续下次循环

continue案例

需求：一起来玩逢7必过小游戏

游戏规则：多人围坐在一起，依次快速说出从1-100的数字，所有含7或7的倍数的数不能说，否则就失败受到惩罚

分析：

A：使用for循环遍历1-100的数 B：在循环体中，判断数中是否含7或是否为7的倍数 是否含7：个位含7（模以10等于7），十位含7（70-79） 是否为7的倍数：对7取模，余数为0 C：跳过所有含7和7的倍数的数：continue D：打印其它数

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
1
2
...
5
6
8
...
13
15
...
69
80
...
98
99
100

Process finished with exit code 0
```

循环嵌套概述

在一个循环体语句中包含另一个循环语句时，称为循环嵌套

需求：按班级获取所有同学（3个班级，每班15个同学）

分析：

A：先使用for循环遍历每一个班级 B：在班级循环体中，再使用for循环遍历每个班级的同学 C：打印正在获取的第几个班级的第几位同学

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
正在获取第1个班级的第1个同学
正在获取第1个班级的第2个同学
正在获取第1个班级的第3个同学
...
正在获取第2个班级的第1个同学
正在获取第2个班级的第2个同学
正在获取第2个班级的第3个同学
...
正在获取第3个班级的第1个同学
正在获取第3个班级的第2个同学
正在获取第3个班级的第3个同学
...
正在获取第3个班级的第13个同学
正在获取第3个班级的第14个同学
正在获取第3个班级的第15个同学
Process finished with exit code 0
```

标号

概念：即循环的名称。给循环定义一个标号，就可以根据需要结束或跳转到指定循环，常用于多层嵌套循环中

语法

```
标号: for() {}  // while和do…while举例略
break 标号;     // 结束指定标号的循环
continue 标号;   // 跳转到指定标号的循环继续执行
```

需求：标号案例: break 标号;

程序猿同学受邀加入A公司，现按班级查找程序猿同学。现有3个班级，每班15个同学，假设第3个班级的第10位同学名叫程序猿，找到该同学后则停止查找。

分析:

A：先使用for循环遍历每一个班级，定义标号：

label_class: for () { }

B：在班级循环体中，再使用for循环遍历每个同学

C：判断如果班级编号为3，同学编号为10，则停止查找：break label_class;

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
正在获取第1个班级的第1个同学
正在获取第1个班级的第2个同学
正在获取第1个班级的第3个同学
...
正在获取第2个班级的第1个同学
正在获取第2个班级的第2个同学
正在获取第2个班级的第3个同学
...
正在获取第3个班级的第1个同学
正在获取第3个班级的第2个同学
正在获取第3个班级的第3个同学
...
正在获取第3个班级的第9个同学
正在获取第3个班级的第10个同学

Process finished with exit code 0
```

需求：标号案例：continue 标号;

按批次检测商品的次品量。现有3个批次，每个批次有10件商品，如果某批次商品中包含任意一个次品，则该批次商品不合格，跳过该批次剩余商品的检测并记录，继续下个批次。假设查找到第2个批次的第5件商品为次品。

分析:

A：先使用for循环遍历每一个批次，定义标号： label_batch: for (){} B：在批次循环体中，再使用for循环遍历每个商品 C：判断如果批次编号为2，商品编号为5，则结束当前批次的检测，继续下个批次： continue label_batch;

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
正在检测第1个批次的第1个商品
正在检测第1个批次的第2个商品
正在检测第1个批次的第3个商品
...
正在检测第2个批次的第1个商品
正在检测第2个批次的第2个商品
正在检测第2个批次的第3个商品
正在检测第2个批次的第4个商品
正在检测第2个批次的第5个商品
记录：第2个批次的第5个商品为次品
正在检测第3个批次的第1个商品
正在检测第3个批次的第2个商品
正在检测第3个批次的第3个商品
...
正在检测第3个批次的第9个商品
正在检测第3个批次的第10个商品

Process finished with exit code 0
```

循环案例

1024程序员节，小黑带你发橙子

需求：

*1024**程序员节*，是传智播客发起的中国程序员共同的节日。每到10月24日，小黑都会按班级给每位同学发橙子。假设有3个班级，每个班级有35个同学，现在要将100个橙子分别发放给每位同学，每人只能拿一个。

条件：如果该同学已经有了橙子，则不再发给该同学；如果橙子发完了，则发放活动终止。

分析：

A： 模拟发橙子的过程：循环每一个班级，然后遍历班级的每个同学，所以需要双层循环

B：假设编号为5的倍数的同学都已经有了橙子，则发放到该同学时，使用continue语句结束该次循环

C：橙子的数量为0时，使用[break + 标号]语句结束外层循环，发放活动终止。

步骤：

A：实现给每位同学发橙子的功能 B：添加判断条件：跳过编号为5的倍数的同学 C：添加判断条件：橙子数目为0，则终止发放

技术点：

循环嵌套 break continue 标号

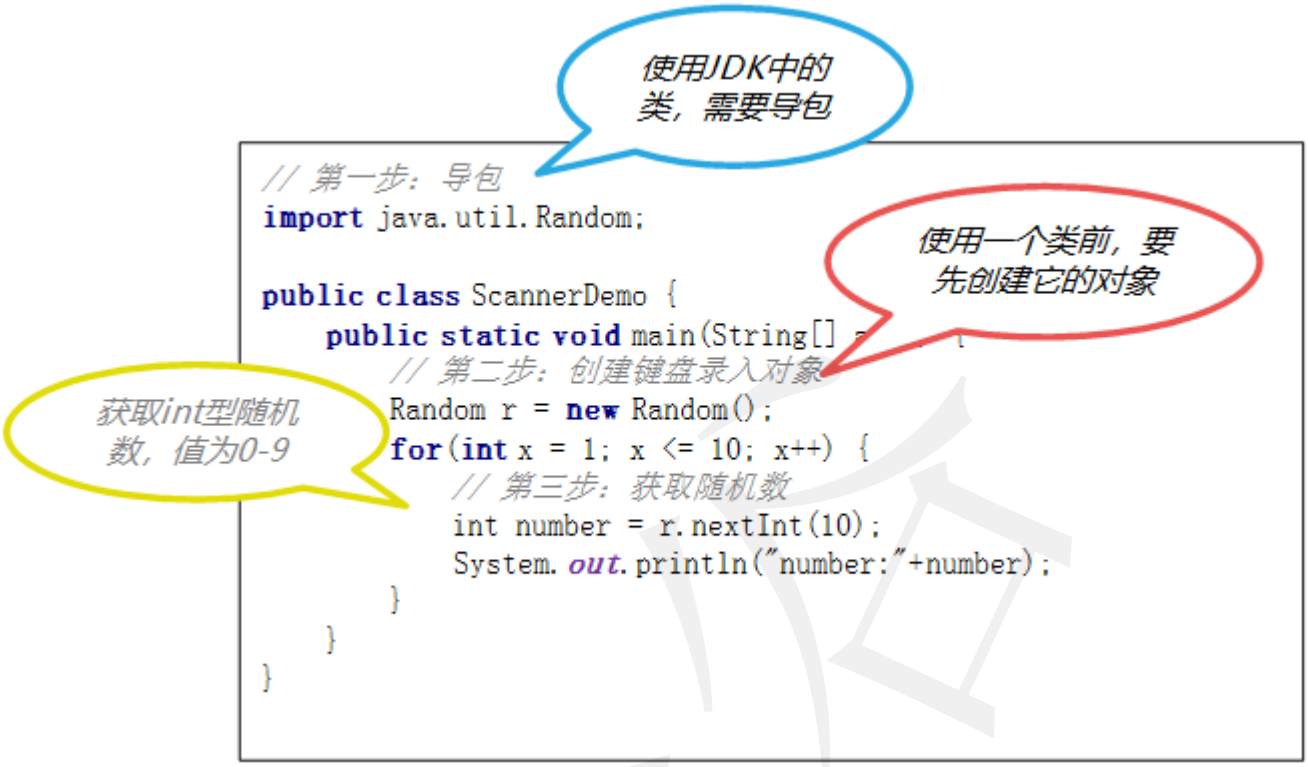
```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
正在给第1个班级的第1个同学发放橙子。
正在给第1个班级的第2个同学发放橙子。
正在给第1个班级的第3个同学发放橙子。
正在给第1个班级的第4个同学发放橙子。
正在给第1个班级的第6个同学发放橙子。
...
正在给第2个班级的第23个同学发放橙子。
正在给第2个班级的第24个同学发放橙子。
正在给第2个班级的第26个同学发放橙子。
...
正在给第3个班级的第29个同学发放橙子。
正在给第3个班级的第31个同学发放橙子。
正在给第3个班级的第32个同学发放橙子。
正在给第3个班级的第33个同学发放橙子。
正在给第3个班级的第34个同学发放橙子。
共发放了84个橙子

Process finished with exit code 0
```

Random类的简单使用

Random，即随机数。用于产生随机数的类。

Random类的使用步骤

1. 导包
2. 创建键盘录入对象
3. 获取随机数



猜数字小游戏

需求：

系统产生一个1-100之间的随机数，请猜出这个数是多少。

分析：

A：随机产生一个1-100之间的数 int number = r.nextInt(100) + 1; B：键盘录入我们要猜的数据 C：比较这两个数据，看我们猜的是否正确： 如果大了，提示：你猜的数据大了 如果小了，提示：你猜的数据小了 如果相等，提示：恭喜你，猜中了 D：加入循环实现多次猜数据，不知道猜的次数，怎么办呢?
使用死循环： while(true) {...} for(;;) {...}

步骤：

A：实现随机产生1-100之间的随机数的功能

B：实现键盘录入一个int型的数的功能

C：实现比较是否相等，并给出提示的功能

D：改进代码，使用死循环实现多次录入功能

E：改进代码，当猜中时结束循环

技术点：

Random类产生随机数

Scanner类实现键盘录入

if语句的第三种格式

死循环

```
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" ...
请输入你要猜的整数(1-100)：
50
你猜的数据50大了
请输入你要猜的整数(1-100)：
25
你猜的数据25小了
请输入你要猜的整数(1-100)：
37
你猜的数据37大了
请输入你要猜的整数(1-100)：
31
你猜的数据31小了
请输入你要猜的整数(1-100)：
34
你猜的数据34大了
请输入你要猜的整数(1-100)：
32
恭喜你，猜中了

Process finished with exit code 0
```