

目录

[Day04. Java](#)

[1 方法的返回值](#)

[2 变量](#)

[2.1 局部变量](#)

[2.2 成员变量](#)

[3 基本类型](#)

[3.1 byte](#)

[3.2 二进制](#)

[3.3 基本类型类型转换](#)

[4 运算符](#)

[4.1 位运算](#)

[4.2 运算符优先级](#)

[5 作业](#)

Day04. Java

商品录入查询

names

["A", "B", "C"]

price

[1, 2, 3]

numbers

[10, 50, 20]

项目: day0401_商品录入查询

类: day0401.Test1

```
package day0401;
```

```
import java.util.Scanner;
```

```
public class Test1 {
```

```
    static String[] names = {"iPhoneXS", "华为Mate20Pro", "荣耀Magic2", "小米Mix3", "vivo NEX"};
```

```
    static double[] price = {7200, 5999, 4799, 3599, 4299};
```

```
    static int[] numbers = {200, 90, 120, 50, 230};
```

```
    public static void main(String[] args) {
```

```
        /*
```

```
        * -----
```

```
        * 1. 商品列表
```

```
        * 2. 商品录入
```

```
        * 3. 商品查询
```

```

* 4. 统计信息
*
* -----
* 1
* ....
*/
outer: //循环命名
while(true) {
    menu();
    int c = new Scanner(System.in).nextInt();

    switch(c) {
        case 1: f1(); break;
        case 2: f2(); break;
        case 3: f3(); break;
        case 4: f4(); break;
        case 5: break outer; //中断名字叫outer的循环
    }
}

}

private static void menu() {
    System.out.println("-----");
    System.out.println("1. 商品列表");
    System.out.println("2. 商品录入");
    System.out.println("3. 商品查询");
    System.out.println("4. 统计信息");
    System.out.println("5. 退出");
    System.out.println("-----");
    System.out.print("选择: ");
}

private static void f1() {
    /*
        names
        ["A", "B", "C"]

        price
        [1,    2,    3]

        numbers
        [10,   50,  20]

        i

        1. 名称: A, 价格: 1, 数量: 10
        2.
    */
    //遍历数组
    for(int i=0; i<names.length; i++) {
        String n = names[i];
        double p = price[i];
        int b = numbers[i];
        System.out.println(
            (i+1)+" . 名称: "+n+", 价格: "+p+", 数量: "+b);
    }
}

private static void f2() {
    /*
        names

```

```
        ["A", "B", "C"]

        price
        [1,    2,    3]

        numbers
        [10,   50,  20]

        i
    */
    //遍历数组
    for(int i=0;i<names.length;i++) {
        System.out.println("录入第"+(i+1)+"件商品: ");
        System.out.print("名称: ");
        String n = new Scanner(System.in).nextLine();
        System.out.print("价格: ");
        double p = new Scanner(System.in).nextDouble();
        System.out.print("数量: ");
        int b = new Scanner(System.in).nextInt();
        names[i] = n;
        price[i] = p;
        numbers[i] = b;
    }
    //重新显示列表
    f1();
}

private static void f3() {
    /*
        B
        names
        ["A", "B", "C"]

        price
        [1,    2,    3]

        numbers
        [10,   50,  20]

        i
    */
    System.out.println("输入查询的商品名称: ");
    String t = new Scanner(System.in).nextLine();
    for(int i=0;i<names.length;i++) {
        // t 和 names[i]是否相等
        // 比较字符串是否相等, 用equals()方法
        if(t.equals(names[i])) {
            String n = names[i];
            double p = price[i];
            int b = numbers[i];
            System.out.println(
                (i+1)+" . 名称: "+n+", 价格: "+p+", 数量: "+b);
            return;
        }
    }
    System.out.println("找不到此商品");
}

private static void f4() {
    /*
        * 商品总价
        * 单价均价
    */
}
```

```
    * 最高总价
    *
    */
    double spzj = 0; //商品总价
    double djzj = 0; //单价总价
    double zgzj = 0; //最高总价
    double zgdj = 0; //最高单价

    for(int i=0;i<names.length;i++) {
        spzj += price[i]*numbers[i];
        djzj += price[i];
        if(price[i]*numbers[i]>zgzj) {
            zgzj = price[i]*numbers[i];
        }
        if(price[i]>zgdj) {
            zgdj = price[i];
        }
    }
    System.out.println("商品总价: "+spzj);
    System.out.println("单价均价: "+(djzj/names.length));
    System.out.println("最高总价: "+zgzj);
    System.out.println("最高单价: "+zgdj);
}
}
```

1 方法的返回值

- 方法中的计算结果，可以返回到调用位置

```
String f() {
}
```

- void 空，没有返回值
- 有返回值，定义返回值的数据类型

方法返回值

项目: day0402_方法返回值
类: day0402.Test1

```
package day0402;

import java.util.Arrays;
import java.util.Random;

public class Test1 {
    public static void main(String[] args) {
        System.out.println("\n\n-----");
        int a = f1();
        System.out.println(a);

        System.out.println("\n\n-----");
        char b = f2();
        System.out.println(b);

        System.out.println("\n\n-----");
    }
}
```

```
    int[] c = f3();
    System.out.println(Arrays.toString(c));
}

static int f1() {
    return new Random().nextInt(5);
}

static char f2() {
    // 'a'+[0, 26)
    return (char)
        ('a'+new Random().nextInt(26));
}

static int[] f3() {
    int[] a = new int[5];
    for(int i=0; i<a.length; i++) {
        a[i] = new Random().nextInt(100);
    }
    return a;
}
}
```

双色球

红球33选6

蓝球16选1

项目: day0403_双色球

类: day0403.Test1

```
package day0403;

import java.util.Arrays;
import java.util.Random;

public class Test1 {
    public static void main(String[] args) {
        //准备红球和蓝球号码数组
        int[] r = zbsz(33); //准备33个红球号码数组
        int[] b = zbsz(16); //准备16个红球号码数组
        //System.out.println(Arrays.toString(r));
        //System.out.println(Arrays.toString(b));

        //从红球数组r, 选出6个号码
        int[] red = xuanRed(r);
        //从蓝球数组b, 选出1个号码
        int blue = xuanBlue(b);

        System.out.println(Arrays.toString(red));
        System.out.println(blue);
    }

    private static int[] xuanRed(int[] r) {
        /*
         * r
         *
         * j
         */
    }
}
```

```

    *
    [3,14,9,4,2,6,7,8,1,10,11,12,13,5,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33]
    *
    */
    //i循环访问前6个位置
    for(int i=0;i<6;i++) {
        //j位置随机定位[0, 33)
        int j = new Random().nextInt(33);
        //i和j位置的值交换
        int t = r[i];
        r[i] = r[j];
        r[j] = t;
    }
    //截取前6个位置, 生成一个新的数组
    return Arrays.copyOf(r, 6);
}

private static int xuanBlue(int[] b) {
    return b[new Random().nextInt(16)];
}

private static int[] zbsz(int n) {
    int[] a = new int[n];
    for(int i=0;i<a.length;i++) {
        a[i] = i+1;
    }
    return a;
}
}
```

2 变量

- 局部变量
- 成员变量

2.1 局部变量

- 定义在方法中, 或局部代码块中
- 需要手动初始化
 - 初始化, 第一次赋值时, 分配内存空间

```
int a;
print(a); //错, 没有初始化
a = 5; //初始化
print(a);
```

- 局部变量作用域, 作用范围, 到它定义的代码块结束
- 作用域内, 不能重复定义

```

void a() {
    int i=5;
    if(...) {
        //int i = 7;
        print(i);
        int j = 6;
    }
    print(j); //错, j作用范围外
    int j = 8;
}

```

2.2 成员变量

- 定义在类中
- 自动初始化成默认值
- 访问范围受访问控制符的控制

3 基本类型

- byte 1, short 2, int 4, long 8
- float 4, double 8
- char 2
- boolean 1
- 字面值规则 (5条)
 - 34345, int
 - byte, short, char
 - byte a = 127; byte
 - byte a = 128; int
 - 3.14, double
 - L F D
 - 0x 0 \u
- 运算规则 (5条)
 - 3/2, 1
 - byte, short, char
 - byte a = 3;
 - a = a+a; int+int
 - Integer.MAX_VALUE+1
 - 溢出变成最小值
 - 2-1.9, 0.1000000000000009
 - 4.35*100, 434.999999999999994
 - Infinity 3.14/0
 - NaN Math.sqrt(-4)

3.1 byte

```

10000000 -128
...
11111011 -5
11111100 -4
11111101 -3
11111110 -2
11111111 -1
00000000 0
00000001 1

```

00000010 2
00000011 3
00000100 4
00000101 5
00000110 6
00000111 7
00001000 8
...
01111111 127

3.2 二进制

83709 8*10 ⁴ 3*10 ³ 7*10 ² 9*10 ⁰	11011 2 ⁴ 2 ³ 2 ¹ 2 ⁰
640 6400 64000	110 1100 11000
	1 1 10 2 100 4 1000 8 10000 16 100000 32 1000000 64 10000000 128 100000000 256 1000000000 512 10000000000 1024
	219 11011011 219/2, 109, 1 109/2, 54, 1 54/2, 27, 0 27/2, 13, 1 13/2, 6, 1 6/2, 3, 0 3/2, 1, 1 1/2, 0, 1
	1000个苹果，分装在几个箱子，不管可用要多少苹果，都可以整箱交付 1 2 4 8 16 32 64 128...

3.3 基本类型类型转换

- byte a = 127;
int b = a;

01111111
00000000 00000000 00000000 01111111
- byte a = -1;
int b = a;


```
11111111
11111111 11111111 11111111 11111111

● int a = 356;
  byte b = (byte) a;

00000000 00000000 00000001 01100100
01100100
```

4 运算符

- + - * /
- %
- == != > >= < <=
- && || !
- ++/--
- ? :
- =
- +=
- ()

4.1 位运算

&	位与，上面是1并且下面是1，结果是1
	位或，上面是1，或者下面是1，结果都是1
^	异或，不同是1
~	求反，0变1,1变0
>>	带符号右移位 符号位是0，补0 符号位是1，补1
>>>	不带符号右移位 不管正负，都补0
<<	左移位

```
00000000000000000000000001110101
0000000000000000000000000101101 &
-----
0000000000000000000000000100101

00000000000000000000000001110101
0000000000000000000000000101101 |
-----
0000000000000000000000000111101

00000000000000000000000001110101
0000000000000000000000000101101 ^
-----
00000000000000000000000001011000

0000000000000000000000000101101 ~
-----
1111111111111111111111111010010

0000000000000000000000000101101 >>2
00000000000000000000000001011
```

[illegible]

打印int的32位

项目: day0404_int的32位

类: day0404.Test1

[illegible]

Test2

```
package day0404;
```



```

package day0405;

import java.util.Arrays;
import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        System.out.print("输入整数: ");
        int n = new Scanner(System.in).nextInt();

        //把n的值, 传递到chaiFen()方法
        //拆分成4个byte值, 存入一个数组
        byte[] a = chaiFen(n);
        System.out.println(Arrays.toString(a));

        System.out.println("\n-----");
        //把a数组中的4个字节, 合并成int整数
        int t = heBing(a);
        System.out.println(t);
    }

    private static int heBing(byte[] a) {
        /*
            11100110
            10001101
            10000110
            10111010

            r
            11100110 10001101 10000110 00000000

            00000000 00000000 00000000 10111010

            0000 0000 0000 0000 0000 0000 1111 1111
            0   0   0   0   0   0   f   f
        */
        //保存结果的变量 r
        int r = 0;
        //数组中的字节值,
        //与0x000000ff 求位与, 把左侧24位抹成0
        //再向左移动24,16,8,0
        //再与r求位或, 结果存到r

        r = r | ((a[0]&0x000000ff) << 24);
        r = r | ((a[1]&0x000000ff) << 16);
        r = r | ((a[2]&0x000000ff) << 8);
        r = r | ((a[3]&0x000000ff) << 0);

        return r;
    }

    private static byte[] chaiFen(int n) {
        //新建byte[]数组, 用来存放4个字节
        byte[] a = new byte[4];
        //n右移24,16,8,0位, 再强转成byte
        a[0] = (byte) (n>>24);
        a[1] = (byte) (n>>16);
        a[2] = (byte) (n>>8);
        a[3] = (byte) (n>>0);

        return a;
    }
}

```

```
}  
}
```

4.2 运算符优先级

$(1+(2-3))*4/5$

- 优先级高低，不用背，多加小括号

5 作业

- 重写
 - day0403_双色球
 - day0405_int拆分4字节
- 数组随机打乱顺序
 - 创建10个长度的int[]数组
 - 顺序放入1到10
 - 随机打乱顺序
 - ◆ i从头到尾递增
 - ◆ j在所有位置中随机定位
 - ◆ i, j位置的值交换

$$\begin{array}{cccccccccc} & & & & & & & & j & \\ [5, 9, 4, 3, 1, 6, 7, 8, 2, 10] \\ i & & & & & & & & & \end{array}$$