

目录

- Day02. Java
 - 1 基本类型 (续)
 - 1.1 char
 - 1.2 基本类型的字面值 (5条)
 - 1.3 基本类型运算规则 (5条)
 - 1.4 基本类型的类型转换
 - 1.4.1 小 --> 大
 - 1.4.2 大 --> 小
 - 2 运算符
 - 3 方法
 - 4 流程控制
 - 4.1 if-else if-else
 - 5 作业

Day02. Java

1 基本类型 (续)

byte	1	-2^7 到 2^7-1	
short	2	-2^{15} 到 $2^{15}-1$	
int	4	-2^{31} 到 $2^{31}-1$	
long	8	-2^{63} 到 $2^{63}-1$	
float	4	单精度	IEEE-754 美国电子电气工程师协会 浮点数表示规范
double	8	双精度	
char	2	0到65535, 0到 $2^{16}-1$	
boolean	1	真	true
		假	false

变量交换

```
graph TD
    A["a = 4"] -- 1 --> T["t = 3"]
    T -- 3 --> B["b = 3"]
    B -- 2 --> A
```

项目: day0201_变量交换
类: day0201.Test1

```
package day0201;

import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        System.out.print("输入整数a: ");
        //先获得输入的值,
        //再存到变量 a
        int a = new Scanner(System.in).nextInt();
        System.out.print("输入整数b: ");
        int b = new Scanner(System.in).nextInt();

        int t = a;
        a = b;
        b = t;

        System.out.println("a = " + a);
    }
}
```

```
        System.out.println("b = " + b);
    }
}
```

浮点数

项目: day0202_基本类型
类: day0202.Test1

```
package day0202;

public class Test1 {
    public static void main(String[] args) {
        /*
         * 两种浮点数的最小值和最大值,
         * 保存到abcd四个变量
         */
        float a = Float.MIN_VALUE;
        float b = Float.MAX_VALUE;
        double c = Double.MIN_VALUE;
        double d = Double.MAX_VALUE;

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
    }
}
```

自由落体距离

项目: day0203_自由落体距离
类: day0203.Test1

```
package day0203;

import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        System.out.print("输入降落时间 (秒) : ");
        //先获得输入的 double 值
        //再存到变量t
        double t = new Scanner(System.in).nextDouble();

        //求降落距离,
        //存到变量 d
        double d = 0.5 * 9.8 * t * t;
        System.out.println(t+"秒降落了"+d+"米");
    }
}
```

1.1 char

- 字符，或字符的整数编码
- Unicode编码表

0	?
1	?
2	?
3	
....	
97	a
98	b
....	
20013	中

....	
65535	

- char c1 = 'a';
char c2 = 97;

char

day0202_基本类型
新建 Test2 类

```
package day0202;

public class Test2 {
    public static void main(String[] args) {
        char c1 = 'a';
        char c2 = 97;
        char c3 = 'b';
        char c4 = 98;
        char c5 = '中';
        char c6 = 20013;

        System.out.println(c1);
        System.out.println(c2);
        System.out.println(c3);
        System.out.println(c4);
        System.out.println(c5);
        System.out.println(c6);
    }
}
```

1.2 基本类型的字面值（5条）

1. 整数字面值，是int类型

```
int a = 63454;
//错，右侧是 int 类型，但超出范围
long a = 999999999;
```

2. byte,short,char，三种比int小的类型，允许用范围内的值直接赋值

```
byte a = 127;//右侧是byte类型
byte a = 128;//右侧是int类型
```

3. 浮点数字面值是 double 类型

```
double a = 3.14;
//错，右侧是double
float a = 3.14;
```

4. 字面值的后缀

```
L - long
    long a = 9999999999L;

F - float
    float a = 3.14F;

D - double
    double a = 3.0;
    double a = 3d;
```

5. 进制前缀

```
0x - 16进制
    两个16进制数字，可以表示二进制的8位，一个字节
    0xff 255

0 - 8进制
```

1个8进制数字，可以表示二进制的 3 位

\u - char类型16进制

必须完整表示char类型的两个字节值

```
char c1 = '\u0061'  
char c2 = '\u4e2d'
```

1.3 基本类型运算规则（5条）

1. 结果的类型，与最大的类型一致

3/2 得1
int/int, 结果int

3d/2 得1.5
double/int, 结果是double

2. byte, short, char, 三种比int小的类型，运算时，会先自动转成int

```
byte a = 3;  
byte b = 4;  
//错，右侧是int+int, 结果是int  
byte c = a + b;
```

3. 整数运算溢出



最大值+1，得负数最小值

300000000*60*60*24*365 错，溢出
300000000L*60*60*24*365 对

4. 浮点数运算不精确

使用IEEE-754表示格式，
做某些数字运算时，会产生不精确结果

2-1.9
得 0.100000000000009

4.35*100
得 434.99999999999994

Java中提供了方法，来做精确运算，后面会讲

5. 浮点数特殊值

■ Infinity
无穷大

3.14 / 0

■ NaN
Not a Number

Math.sqrt(-2)

运算规则测试

day0202_基本类型
新建测试类 Test3

```
package day0202;
```

```
public class Test3 {  
    public static void main(String[] args) {  
        System.out.println(3/2);  
        System.out.println(3d/2);  
        System.out.println("\n\n-----");  
  
        System.out.println('s' + 'b');  
        System.out.println("\n\n-----");  
  
        System.out.println(Integer.MAX_VALUE+1);  
        System.out.println(300000000*60*60*24*365);  
        System.out.println(300000000L*60*60*24*365);  
        System.out.println("\n\n-----");  
  
        System.out.println(2-1.9);  
        System.out.println(2-1.8);  
        System.out.println(2-1.7);  
        System.out.println(2-1.6);  
        System.out.println(2-1.5);  
        System.out.println(4.35*100);  
        System.out.println(4.36*100);  
        System.out.println(4.37*100);  
        System.out.println(4.38*100);  
        System.out.println(4.39*100);  
        System.out.println("\n\n-----");  
  
        System.out.println(3.14/0);  
        System.out.println(Math.sqrt(-4));  
  
    }  
}
```

1.4 基本类型的类型转换

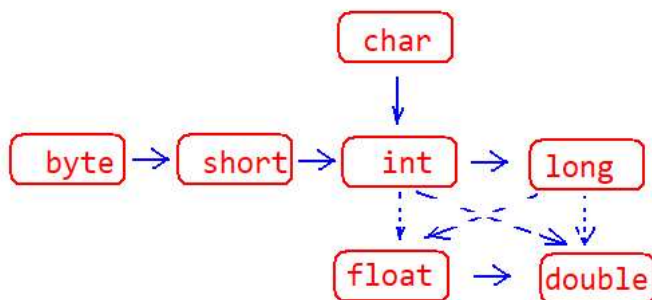
- 数字类型之间，可以互相转换

1.4.1 小 --> 大

- 可以自动转换

```
byte a = 3;  
int b = a;
```

- 整形转成浮点数，可能会有精度损失



1.4.2 大 --> 小

- 必须强制转型

```
int a = 257;  
byte b = (byte) a; // 1
```

- 浮点数转成整数，直接舍弃小数

牛郎织女

牛郎织女星相距16.4光年

光速 299792458 米/秒
喜鹊身长 0.46 米
牛郎织女相会需要多少只喜鹊

项目：day024_牛郎织女
类：day0204.Test1

```
package day0204;

public class Test1 {
    public static void main(String[] args) {
        System.out.println("牛郎织女星相距16.4光年");
        System.out.println("光速 299792458 米/秒");
        System.out.println("喜鹊身长 0.46 米");
        System.out.println("牛郎织女相会需要多少只喜鹊");

        //1光年是多少米
        long ly = 299792458L*60*60*24*365;
        //16.4光年是多少米
        double d = ly*16.4;
        //求喜鹊数量
        double r = d/0.46;
        //向上取整
        //Math.ceil(r)
        //返回的取整结果是double类型
        // 34534524525234.0
        //再转成 long 类型
        long n = (long) Math.ceil(r);
        System.out.println(n);
    }
}
```

2 运算符

+ - *	
+	字符串连接运算
%	取余、求模
==	相等比较
!=	不等比较
> >= < <=	大小比较
&&	逻辑与，左边是真，并且右边是真，结果是真 短路与，左侧是假，右侧被忽略不执行
	逻辑或，左边是真，或者右边是真，结果是真 短路或，左侧是真，右侧被忽略不执行
!	非真是假，非假是真 用来做相反的条件判断 如果“非a”是真，即a是假 if(!a) { }
++ --	自增自减 a = 5; print(a++);先取值再自增 1. 取a的原值等待打印 2. a自增变成6 3. 打印原值5 a = 5; print(++a);先自增再取值 1. a自增变成6 2. 取出a的新值等待打印 3. 打印新值6
? :	三项运算 1 ? 2 : 3 1是真，取2 1是假，取3

=	赋值
+= *=	复合的赋值运算 a += 2 //a = a+2 会自动转型 byte a = 3; a = (byte) (a+2); a += 2; a += 3.14;
()	转型运算

平年闰年

闰年条件:

- *) 能被4整除, 并且不能被100整除
- *) 能被400整除

2000 闰年
1000 平年
2016 闰年
2018 平年

项目: day0205_平年闰年
类: day0205.Test1

```
package day0205;

import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        System.out.print("输入年号: ");
        //先得到输入的年号,
        //再存到变量 y
        int y = new Scanner(System.in).nextInt();

        //保存结果的变量
        //后面判断满足闰年条件, 再改成闰年
        String r = "平年";
        //如果y能被4整除
        /*if(y % 4 == 0) {
            //能被4整除的情况下
            //进一步判断如果不能被100整除
            if(y % 100 != 0) {
                r = "闰年";
            }
        }
        //如果y能被400整除
        if(y % 400 == 0) {
            r = "闰年";
        }*/

        //能被4整除, 并且不能被100整除
        //或者能被400整除
        if((y%4==0 && y%100!=0) || y%400==0) {
            r = "闰年";
        }

        System.out.println(y+"年是"+r);
    }
}
```

自增自减

项目: day0206_自增自减
类: day0206.Test1

```
package day0206;

public class Test1 {
    public static void main(String[] args) {
        int a;

        a = 5;
```

```
System.out.println(a++);

a = 5;
System.out.println(++a);

a = 5;
int b = a++;
System.out.println(a);
System.out.println(b);
}
}
```

三个整数最大值

项目: day0207_三个整数最大值

类: day0207.Test1

```
package day0207;

import java.util.Scanner;

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

        //a,b的最大值, 存到 max
        //int max = a>b?a:b;
        //max,c的最大值, 存在 max
        //max = max>c?max:c;

        int max = a>b ? (a>c?a:c) : (b>c?b:c);

        System.out.println("最大值: "+max);
    }
}
```

3 方法

- 命名的代码块

```
static void f() {
    ...
}
```

- 方法参数

```
static void f(int x) {
    ...
}
```

```
f(2)
f(5)
```

方法

项目: day0208_方法

类: day0208.Test1

```
package day0208;

public class Test1 {

    public static void main(String[] args) {
        System.out.println(1);
        f();
    }
}
```



```
        System.out.println(2);
        f();
        System.out.println(3);
        f();
    }

    static void f() {
        System.out.println(4);
    }
}
```

Test2

```
package day0208;

public class Test2 {
    public static void main(String[] args) {
        System.out.println(1);
        f(2);
        System.out.println(3);
        f(4);
        System.out.println(5);
        f(6);
    }

    static void f(int x) {
        System.out.println(x*x);
    }
}
```

4 流程控制

- if
- switch
- for
- while
- do-while

4.1 if-else if-else

```
if(条件) {
    代码
}

if(条件) {
    代码1
} else {
    代码2
}

if(条件1) {
    代码1
} else if(条件2){
    代码2
} else if(条件3){
    代码3
} else {
    代码4
}
```

个人所得税

项目: day0209_个人所得税
类: day0209.Test1

级数	全月应纳税所得额	税率(%)	速算扣除数
1	不超过1,500元	3	0
2	超过1,500元至4,500元的部分	10	105
3	超过4,500元至9,000元的部分	20	555
4	超过9,000元至35,000元的部分	25	1,005
5	超过35,000元至55,000元的部分	30	2,755
6	超过55,000元至80,000元的部分	35	5,505
7	超过80,000元的部分	45	13,505

```
package day0209;

import java.util.Scanner;

public class Test1 {
    public static void main(String[] args) {
        System.out.print("工资: ");
        double s =
            new Scanner(System.in).nextDouble();

        //把工资s的值, 传递到 f() 方法进行计算
        f(s);
    }

    static void f(double salary) {
        //如果salary没有到起征点, 不交税
        if(salary<=3500) {
            System.out.println("不需要缴税");
            //返回到上面调用位置继续
            return;
        }

        //salary刨掉3500
        salary -= 3500; //salary = salary-3500;

        //定义税率和速算扣除数变量
        double r = 0;
        int k = 0;

        //根据salary的范围, 来确定r和k的值
        if(salary<=1500) {
            r = 0.03;
            k = 0;
        } else if(salary<=4500) {
            r = 0.1;
            k = 105;
        } else if(salary<=9000) {
            r = 0.2;
            k = 555;
        } else if(salary<=35000) {
            r = 0.25;
            k = 1005;
        } else if(salary<=55000) {
            r = 0.3;
            k = 2755;
        } else if(salary<=80000) {
            r = 0.35;
            k = 5505;
        } else {
            r = 0.45;
            k = 13505;
        }
    }
}
```

```
double tax = salary * r - k;  
System.out.println("个人所得税: "+tax);  
}  
}
```

5 作业

- 重写，写到能默写
 - day0201_变量交换
 - day0205_平年闰年
 - day0209_个人所得税
- 商品打折
 - 输入商品价格，求折扣价
 - 满500, 9折
 - 满1000, 85折
 - 满2000, 8折
 - 满5000, 7折
- 预习
 - 流程控制