

如有问题和建议，请与董老师联系！QQ:1050968899 或加微信：QingYunJiao 备注：学习
或加群：365676266 java培训二群 备注：入门
本博客地址：

1 | <https://blog.csdn.net/atgfg>

其它小站：

1 | <http://www.gyun.org>

JavaSE变量与数据类型

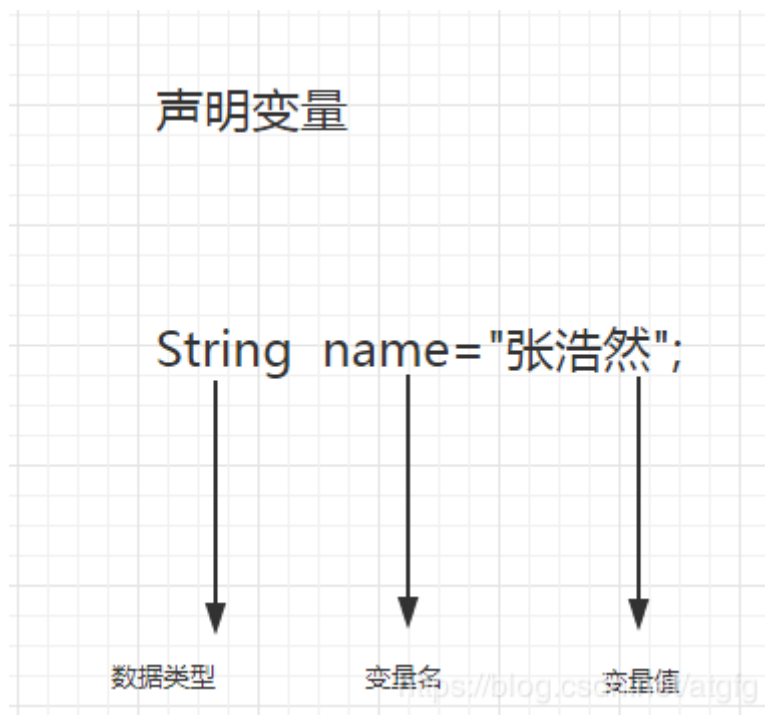
1. 变量:就是一个数据储存空间的表示

◆ 内存地址不好记，怎么办？

房间	变量
房间名字	变量名
房间类型	变量类型
入住的客人	变量值

◆ 通过变量名可以简单快速地找到数据

2. 变量图：



2.0.1. 在线画图工具

1 | <https://www.processon.com/i/5d244b71e4b0ae3317dddac2>

3. 变量的使用步骤:

3.1. 第一步：声明变量，根据数据类型在内存申请空间

如: `int num;`

3.2. 第二步：赋值，即“将数据存储至对应的内存空间”

如: `num=10;`

注意: 第一步和第二步可以合并 为: 声明赋值 `int num=10;`

3.3. 第三步：使用变量，即“取出数据使用”

4. 变量的规则:

4.1. 字下美人数骆驼

4.2. 不能用关键字: 如: public class int double

5. java常用的数据类型:

5.1. 八大基本数据类型: char byte short int long float double boolean

5.2. 引用数据类型: 数组 类(String) 接口

5.3. 数值 : byte short int long float double

5.4. 非数值: boolean String char

6. 数据类型说明

数据类型说明

数据类型	大小	取值范围
byte	1字节8位	-128 ~ +127
int	4字节32位	-2147483648 (-2^{31}) ~ + 2147483647 ($2^{31}-1$)
short	2字节16位	-32768 (-2^{15}) ~ + 32767 ($+2^{15}-1$)
long	8字节64位	-2^{63} ~ + $2^{63}-1$
float	4字节32位浮点数	1.4E-45 ~ 3.4E+38 , -1.4E-45 ~ -3.4E+38
double	8字节64位浮点数	4.9E-324 ~ 1.7E+308, -4.9E-324 ~ -1.7E+308

<https://blog.csdn.net/a10g>

6.0.0.1. 节点小知识:

int 为 4 个字节 32 位

- 1 “1位”表示为 1 bit
- 2 “1个字节”表示为 1 Byte
- 3 “1个字节”=“8位” 即 1 Bytes=8 bit
- 4 因此“4个字节”=4*8位=32位
- 5 此外, 字母用1个字节(即8位)进行表示和存储, 而一个汉字则使用两个字节(即16位, 请看后面说明)进行表示和存储。
- 6 说明:
- 7 一个汉字占用的存储空间大小取决于使用的编码, 一个汉字占2字节, 是在说国内的中文编码标准 GB2312、GBK、BG18030。

7. 变量的使用案例演示

```
1 package org.gyun.test;
2
3 /**
4  * 输出变量名
5  *
6  * @author DongGaoYun
7  *
8  */
9 public class variableTest {
10
11     /**
12      * 主方法 实现需求:
13      */
14     public static void main(String[] args) {
15         //使用变量分二步:
16         // 1.声明赋值
17         String name = "张卓";
18         String sex = "男";
19         char sex1 = '女';
20         double score = 98.5;
21         // 2.使用变量
22         System.out.println("输出Java课考试最高分: " + score);
23         System.out.println("输出最高分学员姓名: " + name);
24         System.out.println("输出最高分学员性别: " + sex1);
25
26     }
```

8. java中float和double的取值范围是什么？(了解)

一、两者的取值范围如下：

1、float：

最小值：Float.MIN_VALUE=1.4E-45（2的-149次方）。

最大值：Float.MAX_VALUE=3.4028235E38（2的128次方-1）。

2、double：

最小值：Double.MIN_VALUE=4.9E-324（2的-1074次方）。

最大值：Double.MAX_VALUE=1.7976931348623157E308（2的1024次方-1）。

二、IEEE754不是一个具体值，是IEEE二进位浮点数算术标准（IEEE Standard for Floating-Point Arithmetic）的标准编号，它规定了浮点数在计算机当中的存储方式以及算术标准等。

（一）IEEE754的规定：

1、[单精度浮点数](#)字长32位，尾数长度23，指数长度8,指数偏移量127；[双精度浮点数](#)字长64位，尾数长度52，指数长度11，指数偏移量1023。

2、约定小数点左边隐含有一位，通常这位数是1，所以上述单精度尾数长度实际为24(默认省略小数点左边的1则为23)，双精度尾数长度实际为53（默认省略小数点左边的1则问53）。

（二）IEEE754有三部分组成：

符号位、指数位、尾数位。三种精度的浮点数各个部分位数如下：

精度	符号	阶/指数	尾数	总位数
单精度	1	8	23	32
双精度	1	11	52	64
长双精度	1	15	64	80

扩展资料：

IEEE754标准的特殊值：

标准规定了一些特殊值(零，无穷，非数字)的存储方式：

- 1、如果 指数 是0 并且 小数部分 是0, 这个数 ± 0 (和符号位相关,这就是上文说的原因)。
- 2、如果 指数 = $2^e - 1$ 并且 小数部分 是0, 这个数是 \pm 无穷大 (同样和符号位相关)。
- 3、如果 指数 = $2^e - 1$ 并且 小数部分 非0, 这个数表示为不是一个数(NaN)。

具体如下：

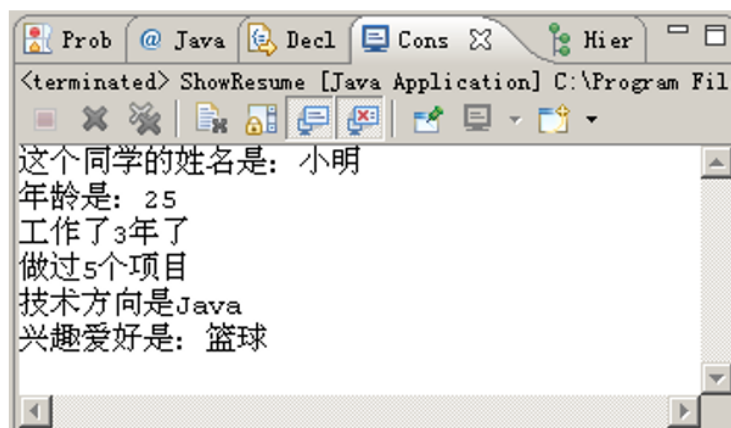
form	biased exponent	fraction
zero	0	0
denormalized	0	not 0
normalized	$0 < b_exp < 2^e - 1$	
\pm infinity ($\pm\infty$)	$2^e - 1$	0
NaN	$2^e - 1$	not 0

9. 练习

练习1：输出个人简历

◆需求说明

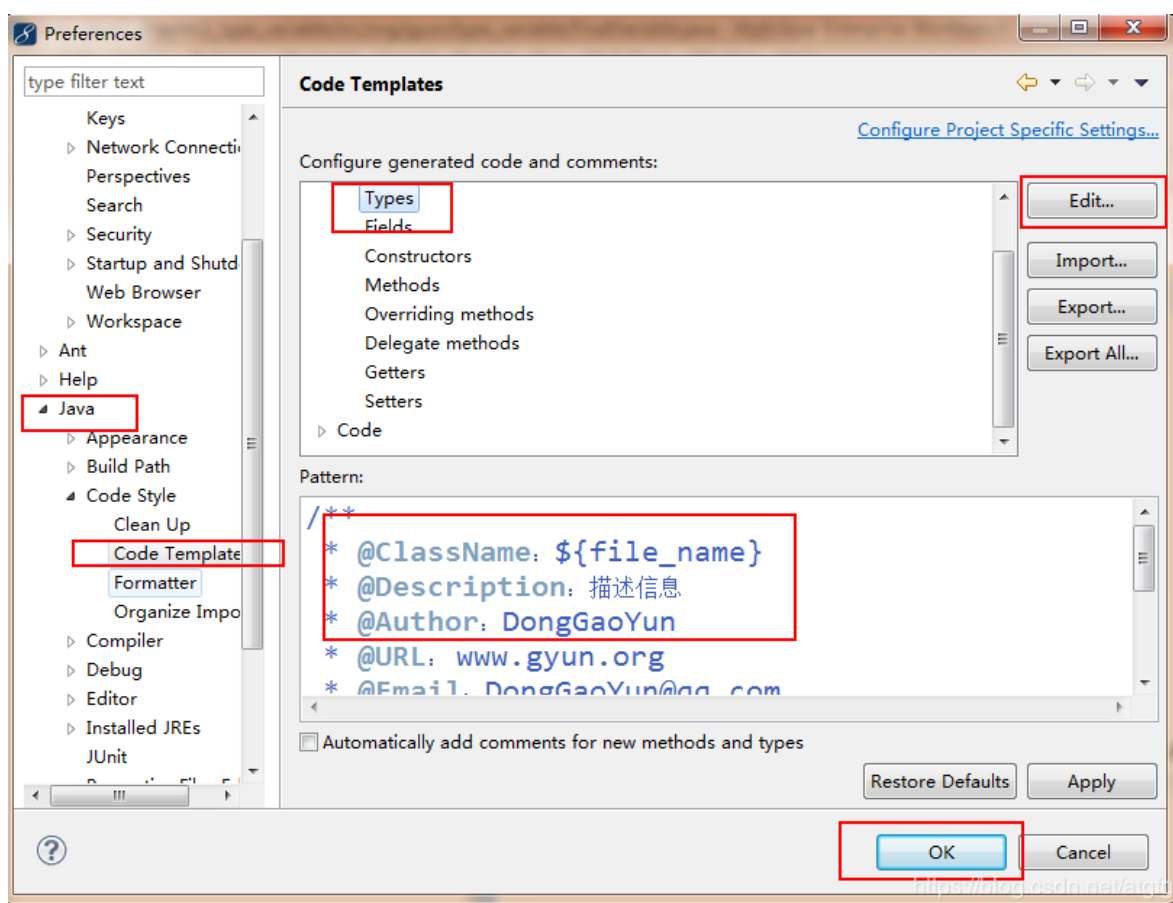
- 使用变量存储数据，实现个人简历信息的输出



<https://blog.csdn.net/taigfg>

10. 设置个人的文档注释

10.1. 设置个人文档注释的步骤:



<https://blog.csdn.net/taigfg>

10.2. 注释信息:

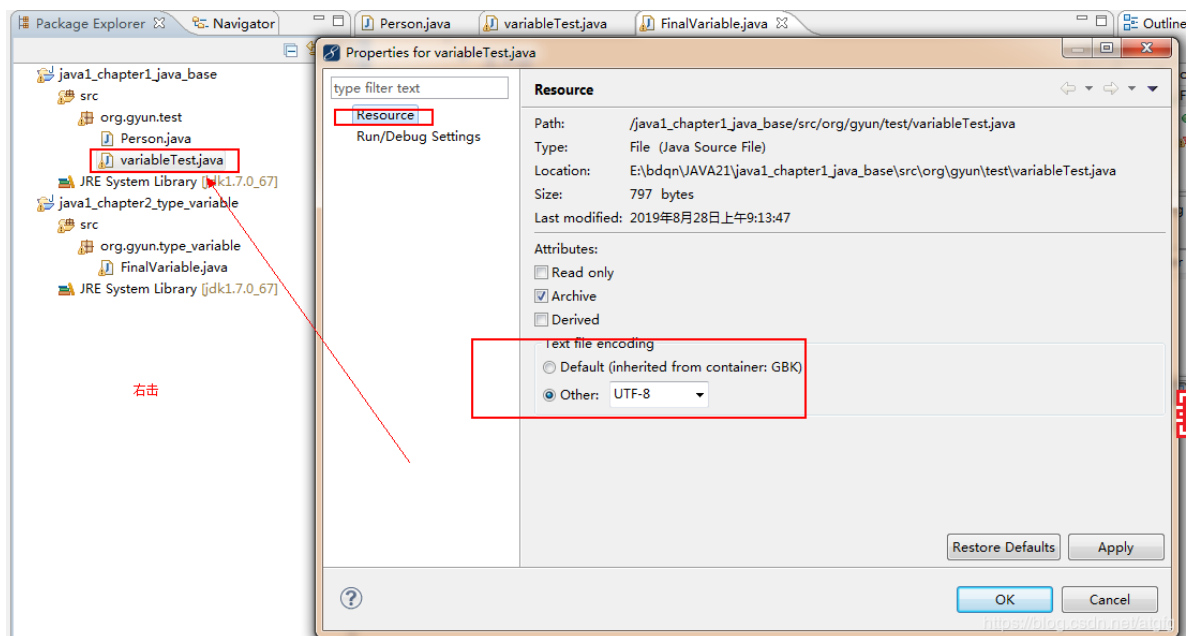
/**

- @Description : 描述信息
- @Author : DongGaoYun
- @Date : \${date} \${time}
- */

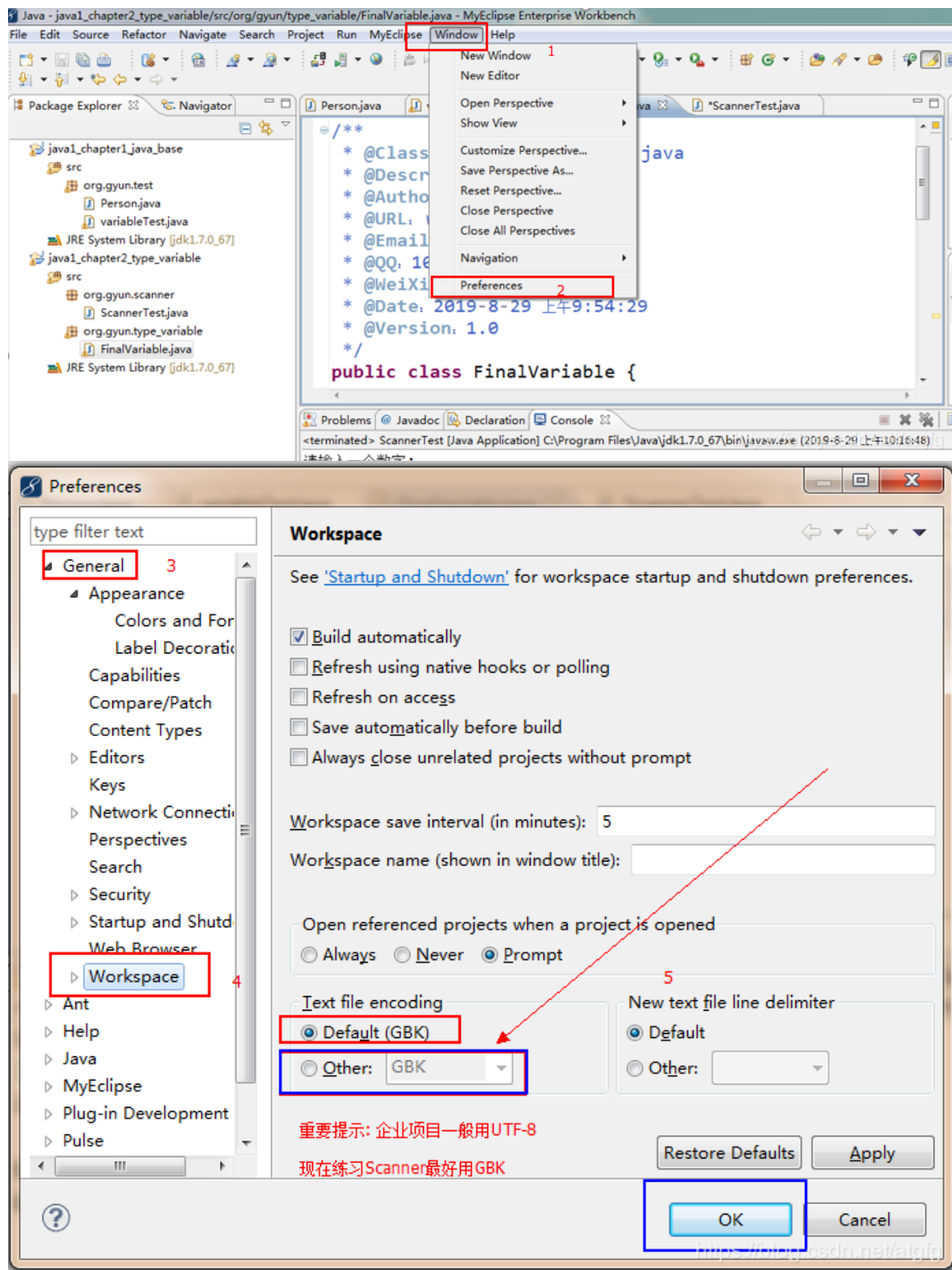
11. 变量和常量的应用演示案例:

```
1 package org.gyun.type_variable;
2
3 /**
4  * @ClassName: FinalVariable.java
5  * @Description: 使用变量和常量
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
11 * @Date: 2019-8-29 上午9:54:29
12 * @Version: 1.0
13 */
14 public class FinalVariable {
15
16     /**
17      * 主方法
18      */
19     public static void main(String[] args) {
20         //variableTest();
21         //常量
22         // 1.声明赋值
23         //变量 字下美人数骆驼
24         final String NAME_DONG_GAO_YUN = "董老师";
25         // 2.使用
26         System.out.println(NAME_DONG_GAO_YUN);
27
28     }
29
30     private static void variableTest() {
31         // 变量
32         // 1.声明赋值
33         int num = 10;
34         // 2.使用
35         System.out.println(num);
36     }
37
38 }
```

12. 设置项目编码:



13. 设置全局编码:



13.0.1. Scanner对象 最好使用GBK

14. Scanner对象的使用

Scanner的使用



问题

在程序运行中，如何接收用户键盘输入的值？

◆ 使用Scanner

- 第一步：导入Scanner类

```
import java.util.*;
```

- 第二步：创建Scanner对象

```
Scanner input = new Scanner(System.in);
```

- 第三步：获得键盘输入的数据

```
int now = input.nextInt();
```

<https://blog.csdn.net/atgfg>

15. Scanner演示案例:

```
1 package org.gyun.scanner;
2
3 import java.util.*;
4
5 /**
6  * @ClassName: ScannerTest.java
7  * @Description: Scanner的应用
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org
10 * @Email: DongGaoYun@qq.com
11 * @QQ: 1050968899
12 * @Weixin: QingYunJiao
13 * @Date: 2019-8-29 上午10:11:00
14 * @Version: 1.0
15 */
16 public class ScannerTest {
17
18     /**
19      * 分析需求：1. 导入包 2. 声明(创建)对象 3. 使用
20      */
21     public static void main(String[] args) {
22         // 使用Scanner 1. 导包 2. 创建对象 3. 使用
23         Scanner input = new Scanner(System.in);
24         System.out.println("请输入一个数字: ");
25         int num = input.nextInt();
26         System.out.println("请输入您的姓名: ");
27         String name = input.next();
28         // 打印
29         System.out.println("输入的数字为: " + num);
30         System.out.println("输入的姓名为: " + name);
```

```
31     }
32 }
33
```

16. 自动转换的规则:

自动类型转换规则

◆ **规则1：如果一个操作数为double型，则整个表达式可提升为double型**

◆ **规则2：满足自动类型转换的条件**

- 两种类型要兼容
 - 数值类型（整型和浮点型）互相兼容
- 目标类型大于源类型：
 - 例如：double 型大于 int 型

<https://blog.csdn.net/atgig>

17. 自动转换演示案例:

```
1 package org.gyun.scanner;
2
3 import java.util.*;
4
5 /**
6  * @ClassName: ScannerTest.java
7  * @Description: Scanner的应用
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org
10 * @Email: DongGaoYun@qq.com
11 * @QQ: 1050968899
12 * @Weixin: QingYunJiao
13 * @Date: 2019-8-29 上午10:11:00
14 * @Version: 1.0
15 */
16 //@SuppressWarnings("all")
17 public class ScannerTest {
18
19     /**
20      * 分析需求：1. 导入包 2. 声明(创建)对象 3. 使用
21      */
22     public static void main(String[] args) {
23         //scannerTest();
24         //自动转换有两个条件：
25         //1. 都是数据类型,可以相互兼容
26         //2. 目标类型必须大于等于源类型
27
28         int a=10;
```

```

29     short b=(short) a;//强转
30     double c=a;//自动转换
31
32 }
33
34
35 private static void scannerTest() {
36     // 使用Scanner 1.导包 2.创建对象 3.使用
37     Scanner input = new Scanner(System.in);
38     System.out.println("请输入您的姓名: ");
39     String names=input.nextLine();
40     System.out.println("请输入您的姓名: ");
41     String name=input.next();
42     System.out.println("请输入一个数字: ");
43     int num= input.nextInt();
44     // 打印
45     System.out.println("输入的数字为: " + num);
46     System.out.println("输入的姓名为: " + name);
47     System.out.println("输入的姓名为: " + names);
48 }
49 }
50

```

18. 案例没有强转,会报错:

◆ 去年Apple笔记本所占市场份额是20，今年增长市场份额是9.8，求今年所占份额？

```

int before = 20; //apple笔记本市场份额
double rise = 9.8; //增长的份额
int now = before + rise; //现在的份额

```

18.1. bug原因:

目标类型小于源类型!

18.2. 规则:

```

1 //自动转换有两个条件:
2 //1. 都是数据类型,可以相互兼容
3 //2. 目标类型必须大于等于源类型

```

19. 运算符:

20. 赋值运算符

赋值运算符

- ◆ 学员王浩的Java成绩是80分，学员张萌的Java成绩与王浩的相同，输出张萌的成绩

变量名 = 表达式;

表达式就是符号（如加号、减号）与操作数（如b, 3等）的组合

<https://blog.csdn.net/atguigu>

20.1. "=" java赋值号

21. 两个数据的交换演示案例:

```
1 package org.gyun.num;
2
3 /**
4  * @ClassName: NumTest.java
5  * @Description: 两个数据的交换
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
11 * @Date: 2019-8-29 上午11:22:41
12 * @Version: 1.0
13 */
14 public class NumTest {
15
16     /**
17      * 主方法 分析需求:
18      * 1. 已知条件num1=6; num2=8;源
19      * 2. 交换成num1=8; num2=6;
20      * 目标 我怎么去做?
21      * 学会借力: 借一个容器
22      * 3. 声明 num3=0; 赋值运算符 num3=num1; //num3=6 num1=num2; //num1=8
23      * num2=num3; //num2=6;
24      *
25      */
26     public static void main(String[] args) {
27         // 声明赋值
28         int num1 = 6;
29         int num2 = 8;
30         // 声明临时容器
31         int temp = 0;
32         // 使用
33         temp = num1; // num3=6
34         num1 = num2; // num1=8
35         num2 = temp; // num2=6;
```

```
36      //打印
37      System.out.println("num1=" + num1);
38      System.out.println("num2=" + num2);
39  }
40 }
```

22. 算术运算符

◆ 基本的算术运算

+、-、*、/、%、++、--

◆ 复合算术运算符

+=、-=、*=、/=、%=

<http://blog.csdn.net/wang>

22.1. / % ++ --

22.2. 算术运算的演示案例:

```
1  package org.gyun.num;
2
3  /**
4   * @ClassName: Test3.java
5   * @Description: 使用/ % ++ --的应用
6   * @Author: DongGaoYun
7   * @URL: www.gyun.org
8   * @Email: DongGaoYun@qq.com
9   * @QQ: 1050968899
10  * @Weixin: QingYunJiao
11  * @Date: 2019-8-29 上午11:35:48
12  * @Version: 1.0
13  */
14  public class Test3 {
15      //主方法
16      // ++ 自身加1
17      // -- 自身减1
18
19      //++i 自身先加1,再赋值
20      //i++ 先赋值,再加1;
21
22      //--i 自身先减1,再赋值
23      //i-- 先赋值,再减1;
```

```

24
25     public static void main(String[] args) {
26         // "/"
27         //取整数的商 //2
28         int num1=5/2;
29         System.out.println(num1);
30         // "%"
31         //取整数的余
32         int num2=5%2;
33         //打印需求:
34         System.out.println(num2); //1
35         System.out.println(++num2); //2
36         System.out.println(num2++); //3
37         System.out.println(num2--); //3 2
38         System.out.println(num2--); //2 1
39         // 1 2 2 3
40         num1=num2++ +num2++ +--num2+ ++num2;
41         System.out.println(num1);
42         //作业
43         //num1= --num1+num2+++++num2+--num2+++num1;
44     }
45 }
46
47

```

22.3. 求四位会员卡号之和演示案例:

```

1  package org.gyun.scanner;
2
3  import java.util.Scanner;
4
5  /**
6   * @ClassName: ScannerCardTest.java
7   * @Description 求四位会员卡号之和
8   * @Author: DongGaoYun
9   * @URL: www.gyun.org
10  * @Email: DongGaoYun@qq.com
11  * @QQ: 1050968899
12  * @Weixin: QingYunJiao
13  * @Date: 2019-8-29 下午12:06:51
14  * @Version: 1.0
15  */
16  public class ScannerCardTest {
17
18      /**
19       * 主方法
20       * 分析需求:
21       * 1. 技术点:Scanner对象 算术运算符 / % 赋值运算符 = 打印
22       * 2. 变量名: ones 个位 tens 十位 hundreds 百位 thousands 千位
23       * 3. 细化:Scanner 1.导包 2. 创建对象 3. 使用
24       *
25       */
26      public static void main(String[] args) {
27          //输入四位会员卡号:
28          //Scanner 1.导包 2. 创建对象 3. 使用

```

```

29 Scanner input =new Scanner(System.in);
30 System.out.println("请输入四位会员卡号： ");
31 int card = input.nextInt();
32 //得到个位：
33 int ones=card%10;
34 System.out.println("个位的数为："+ones);
35 //得到十位：
36 int tens=card/10%10;
37 System.out.println("十位的数为："+tens);
38 //得到百位：
39 int hundreds=card/100%10;
40 System.out.println("百位的数为："+hundreds);
41 //得到千位：
42 int thousands=card/1000;
43 System.out.println("千位的数为："+thousands);
44 System.out.println("四位的会员卡号之和为： "+
    (ones+tens+hundreds+thousands));
45 }
46
47 }
48

```

23. 关系运算符

关系运算符

运算符	含义	范例	结果
==	等于	5==6	false
!=	不等于	5!=6	true
>	大于	5>6	false
<	小于	5<6	true
>=	大于等于	5>=6	false
<=	小于等于	5<=6	true



注意

=为赋值运算符，==为等于运算符

关系表达式的结果为布尔值

>、<、>=、<= 运算符的优先级别高于==、!=

>、<、>=、<=运算符只支持数值类型的数据比较

<https://blog.csdn.net/ato1g>

24. 逻辑运算符

逻辑运算符

运算符	含义	运算规则
&&	短路与	两个操作数都是true，结果才是true
	短路或	两个操作数一个是true，结果就是true
!	非	操作数为true，结果为false
		操作数为false，结果为true

如果第一个操作数已经能确定表达式的值，第二个操作数就不必计算了；
优先级 ! > && > ||

<https://blog.csdn.net/atgfg>

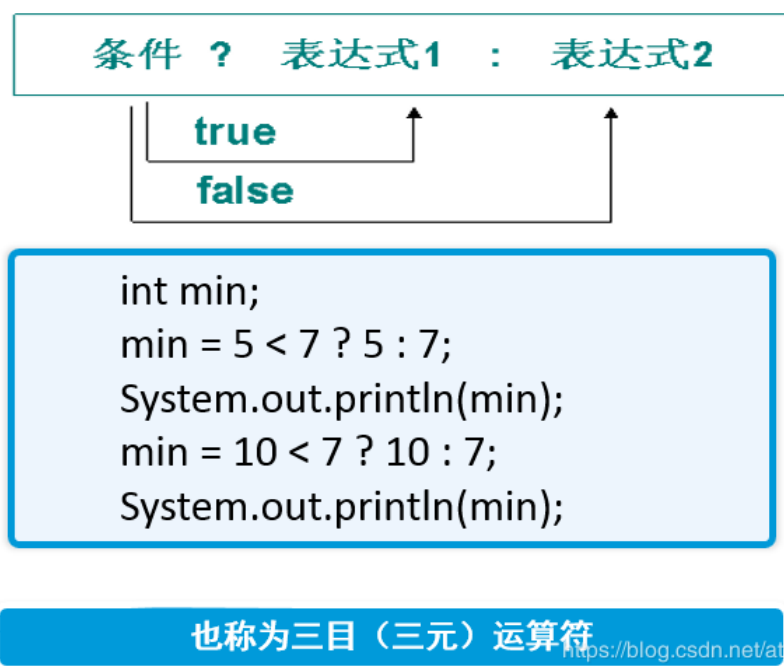
24.1. 逻辑运算符演示案例:

```
1 package org.gyun.scanner;
2
3 /**
4  * @ClassName: TestAnd_Or.java
5  * @Description: 逻辑运算符 && || !
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
11 * @Date: 2019-8-29 下午12:23:13
12 * @Version: 1.0
13 *
14 * && 短路与(且) 左边的为true 才去看右边 否则不用看右边 左边和右边同时成立,结果
   才是true
15 * || 短路或 左边为true 右边就不用看了,只要有一边是true,结果就是true !
16 * 取非 是true取非,结果就是false 是false取非,结果就是true
17 */
18 public class TestAnd_Or {
19     public static void main(String[] args) {
20         // 声明赋值
21         int num = 5;
22         int num1 = 6;
23         int num2 = 7;
24         String str = "李自成";
25         // "!" 取非
26         if (!str.equals("李自成")) {
27             System.out.println(1);
28         } else {
29             System.out.println(2);
30         }
31         // "&&" 短路与(且)
32         if (num > num1 && num2 > num1) {
33             System.out.println(num2);
34         } else {
35             System.out.println(num1);
36         }
37         // "||" 短路或
```

```
38         if (num > num1 || num2 > num1) {
39             System.out.println(num2);
40         } else {
41             System.out.println(num1);
42         }
43     }
44 }
```

25. 条件(三目)运算符

条件运算符



26. 经验:

运算符优先级

- ◆ 单目运算符包括! ++ -- , 优先级别高
- ◆ 优先级别最低的是赋值运算符
- ◆ 可以通过()控制表达式的运算顺序, ()优先级最高
- ◆ 从右向左结合性的只有赋值运算符、三目运算符和单目运算符
- ◆ 算术运算符 > 关系运算符 > 逻辑运算符

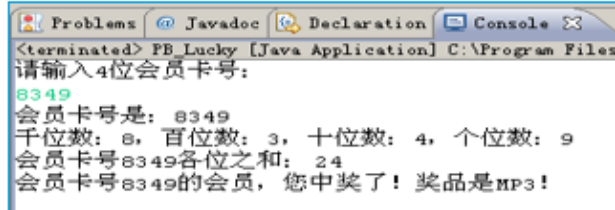
<https://blog.csdn.net/atgfg>

27. 练习:

练习4：判断是否中奖

◆ 需求说明

- 根据分解后的数字之和，判断用户是否中奖。如果数字之和大于20则中奖



```
<terminated> PB_Lucky [Java Application] C:\Program Files
请输入4位会员卡号:
8349
会员卡号是: 8349
千位数: 8, 百位数: 3, 十位数: 4, 个位数: 9
会员卡号8349各位之和: 24
会员卡号8349的会员, 您中奖了! 奖品是MP3!
```

<https://blog.csdn.net/atgfg>

如有问题和建议，请与董老师联系！QQ:1050968899 或加微信: QingYunJiao 备注: 学习
或加群: 365676266 java培训二群 备注: 入门