

● 键盘录入的基本格式讲解(掌握)

- A:为什么要使用键盘录入数据
 - a:为了让程序的数据更符合开发的数据
 - b:让程序更灵活一下
- B:如何实现键盘录入呢?
 - 先照格式来。
 - a:导包
 - 格式:
 - `import java.util.Scanner;`
 - 位置:
 - 在 class 上面。
 - b:创建键盘录入对象
 - 格式:
 - `Scanner sc = new Scanner(System.in);`
 - c:通过对象获取数据
 - 格式:
 - `int x = sc.nextInt();`
- C:案例演示
 - 键盘录入 1 个整数，并输出到控制台。

```
import java.util.Scanner; //引入包
public class Demo5 {
    public static void main(String[] args) {
        //创建了一个可以接受键盘输入的对象
        Scanner sc = new Scanner(System.in);
        //写一个用户提示
        System.out.println("请输入一个整数: ");
        int a = sc.nextInt();//从键盘接收一个整数
    }
}
```

```
        System.out.println("接收到: "+a);
    }
}
```

- 键盘录入 2 个整数，并输出到控制台。

```
import java.util.Scanner; //引入包
public class Demo5 {
    public static void main(String[] args) {
        // 创建了一个可以接受键盘输入的对象
        Scanner sc = new Scanner(System.in);
        // 写一个用户提示
        System.out.println("请输入一个整数: ");
        int a = sc.nextInt(); // 从键盘接收一个整数，会阻塞，等待用户的输入数据
        System.out.println("接收到: "+a);

        System.out.println("请输入第二个整数: ");
        int b = sc.nextInt(); // 每到这一行就会等待输入
        System.out.println("收到: "+b);
    }
}
```

● 键盘录入的练习 1(掌握)

- A: 案例演示
 - 键盘录入练习：键盘录入两个数据，并对这两个数据求和，输出其结果

```
import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入第一个整数: ");
        int a = sc.nextInt();
        System.out.println("请输入第二个整数: ");
        int b = sc.nextInt();
        int result = a+b;
        System.out.println("结果是: "+result);
    }
}
```

- B:案例演示

- 键盘录入练习：键盘录入两个数据，获取这两个数据中的最大值

```
import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("请输入第一个整数: ");
        int a = sc.nextInt();
        System.out.println("请输入第二个整数: ");
        int b = sc.nextInt();

        int max = a>b?a:b;//获得了a 与 b 中教大的
        System.out.println("两个数中教大的是: "+max);
    }
}
```

- 键盘录入的练习 2(掌握)

- A:案例演示

- 键盘录入练习：键盘录入两个数据，比较这两个数据是否相等

```
import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("请输入第一个整数: ");
        int a = sc.nextInt();
        System.out.println("请输入第二个整数: ");
        int b = sc.nextInt();

        String result = a == b ? "相等":"不相等";
        System.out.println(result);
    }
}
```

- B:案例演示

- 键盘录入练习：键盘录入三个数据，获取这三个数据中的最大值

●三元运算符的基本用法(掌握)

- A:三元（三目）运算符的格式
 - (关系表达式) ? 表达式 1 : 表达式 2;
- B:三元运算符的执行流程
- C:案例演示
 - 获取两个数中的最大值

```
public class Demo4 {  
    public static void main(String[] args) {  
        int a = 30;  
        int b = 20;  
        String result = a > b ? "a 大于 b" : "a 小于 b"; //三目运算符，或者叫三元运算符  
        System.out.println(result);  
        //三目运算符运算之后会返回给我们一个值（常量）  
        System.out.println(a <= b ? "a<=b" : "a>b");  
        System.out.println(a >= b ? '大' : '小');  
    }  
}
```

●三元运算符的练习(掌握)

- A:案例演示
 - 比较两个整数是否相同
- B:案例演示
 - 获取三个整数中的最大值

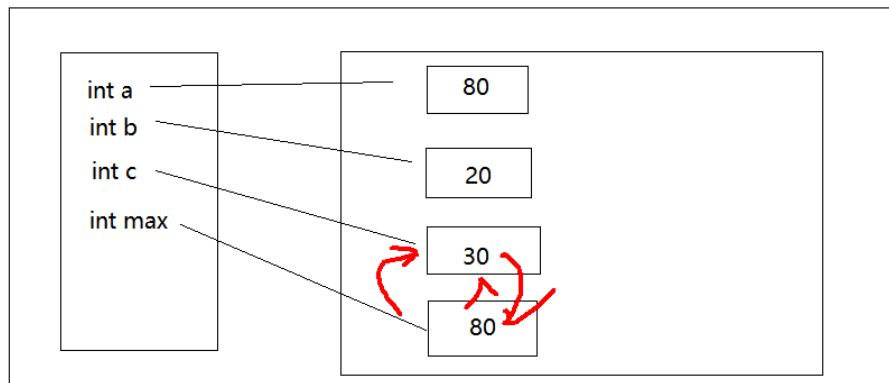
```
public class Demo4 {  
    public static void main(String[] args) {  
        int a = 80;
```

```
int b = 20;
String result = a == b ? "相等":"不相等";//关系运算符==的执行优先于赋值运算符=
System.out.println(result);
```

//比较三个数中的最大的那个

```
int c = 30;
int max = a > b ? a:b;//取到了a 于 b 中教大的那个
max = max > c ? max : c;
System.out.println("三个数中最大的是: "+max);
}
```

```
int a = 80;
int b = 20;
//比较三个数中的最大的那个
int c = 30;
int max = a > b ? a:b;//取到了a 于 b 中教大的那个
max = max > c ? max : c;
System.out.println("三个数中最大的是: "+max);
```

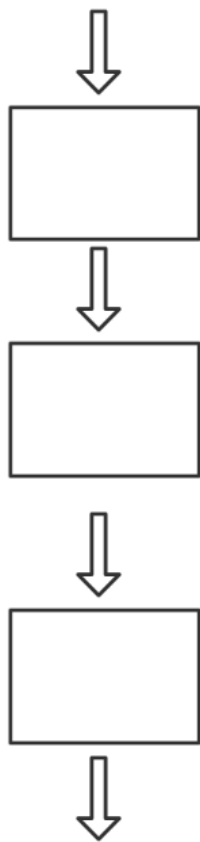


● 顺序结构语句(了解)

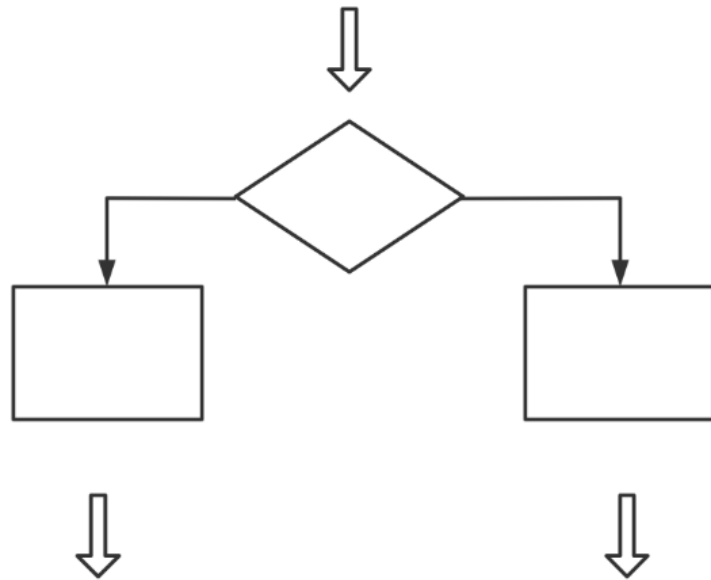
- A: 什么是流程控制语句
 - 流程控制语句：可以控制程序的执行流程。
- B: 流程控制语句的分类
 - 顺序结构
 - 选择结构

- 循环结构

顺序结构



选择结构



- C: 执行流程:
 - 从上往下，依次执行。
- D: 案例演示
 - 输出几句话看效果即可

● 选择结构 if 语句格式 1 及其使用(掌握)

- A: 选择结构的分类
 - if 语句
 - switch 语句
- B: if 语句有几种格式
 - 格式 1
 - 格式 2

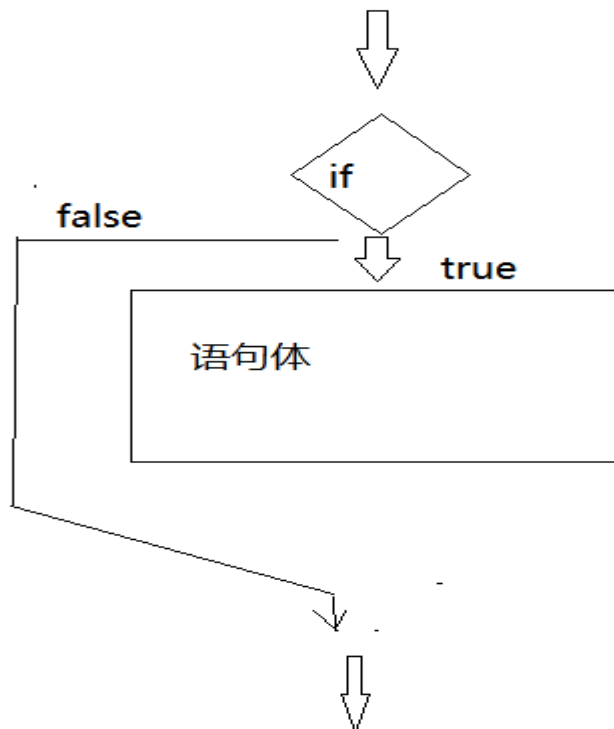
- 格式 3

- C:if 语句的格式 1

```
if(比较表达式) {  
    语句体;  
}
```

- D:执行流程:

- 先计算比较表达式的值，看其返回值是 true 还是 false。
- 如果是 true，就执行语句体；
- 如果是 false，就不执行语句体；



```
public class Demo6 {  
    public static void main(String[] args) {  
        int age = 17;  
        if(age>=18){  
            System.out.println("成年了，可以进网吧，开黑");  
        }  
    }  
}
```

```
        System.out.println("if 之后的程序");
    }
}
```

```
public class Demo6 {
    public static void main(String[] args) {
        int age = 19;
        if(age >=18 && age <=65){ //java 中不接受这种写法 18<= age <=65
            System.out.println("可以去企业上班");
        }
    }
}
```

● 选择结构 if 语句注意事项(掌握)

■ A: 案例演示

- a: 比较表达式无论简单还是复杂，结果必须是 boolean 类型
- b: if 语句控制的语句体如果是一条语句，大括号可以省略；
 - 如果是多条语句，就不能省略。建议永远不要省略。
- c: 一般来说：有左大括号就没有分号，有分号就没有左大括号

● if—else(掌握)

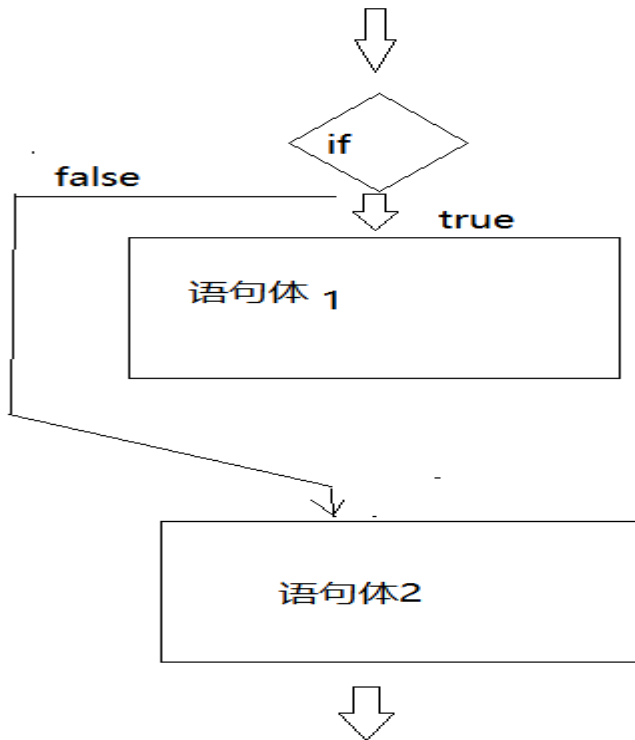
■ A: if—else 语句的格式

■

```
if(比较表达式) {
    语句体 1;
}else {
    语句体 2;
}
```

■ B: 执行流程：

- 首先计算比较表达式的值，看其返回值是 true 还是 false。
- 如果是 true，就执行语句体 1；
- 如果是 false，就执行 else 后面的语句体 2；



```

public class Demo6 {
    public static void main(String[] args) {
        int age = 19;
        if(age >=18 && age <=65) {
            System.out.println("可以去企业上班");
        }else{
            System.out.println(age+"岁，你还是个娃，好好听你妈妈的话");
        }
        System.out.println("if-else 之后的程序");
    }
}
  
```

- C:案例演示
 - a:获取两个数据中较大的值

```

public class Demo6 {
    public static void main(String[] args) {
        int a = 50;
        int b = 39;
        int temp; // 声明变量
        if(a>b){ // 如果
            temp = a;
        }else{ // 否则的意思
  
```

```

        temp = b;
    }
    System.out.println("较大的是: "+temp);
}
}

```

- b:判断一个数据是奇数还是偶数,并输出是奇数还是偶数

```

import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        int a = 51;

        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个整数: ");
        int b = sc.nextInt();
        if(b % 2 == 0){
            System.out.println("是偶数");
        }else{
            System.out.println("是奇数");
        }
    }
}

```

- 注意事项: else 后面是没有比较表达式的,只有 if 后面有。

● if-else if-else 和三元的相互转换问题(掌握)

- A:案例演示
 - if 语句和三元运算符完成同一个效果
- B:案例演示
 - if 语句和三元运算符的区别
 - 三元运算符实现的,都可以采用 if 语句实现。反之不成立。

- 什么时候 if 语句实现不能用三元改进呢？
 - 当 if 语句控制的操作是一个输出语句的时候就不能。
 - 为什么呢?因为三元运算符是一个运算符，运算符操作完毕就应该有一个结果，而不是一个输出。

```
import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        int a = 51;
        if(a % 2 == 0){
            System.out.println("是偶数");
        }else{
            System.out.println("是奇数");
        }

        //三元运算符
        String result = a%2==0 ? "偶数":"奇数";//复制一行 Ctrl+D
        System.out.println("result:"+result);
        // String result2 = a%2==0 ? System.out.println("偶数"):System.out.println("奇数");
        //? 后面只能是常量，在三目运算符中不能有打印语句

    }
}
```

●选择结构 if 语句格式 3 及其使用(掌握)

- A:if 语句的格式 3:

```
if(比较表达式 1) {
    语句体 1;
}else if(比较表达式 2) {
    语句体 2;
}else if(比较表达式 3) {
    语句体 3;
}
...
```

```
else {  
    语句体 n+1;  
}
```

■ B: 执行流程:

- 首先计算比较表达式 1 看其返回值是 true 还是 false,
- 如果是 true, 就执行语句体 1, if 语句结束。
- 如果是 false, 接着计算比较表达式 2 看其返回值是 true 还是 false,
- 如果是 true, 就执行语句体 2, if 语句结束。
- 如果是 false, 接着计算比较表达式 3 看其返回值是 true 还是 false,
- 如果都是 false, 就执行语句体 n+1。

■ C: 注意事项: 最后一个 else 可以省略, 但是建议不要省略, 可以对范围外的错误值提示

```
import java.util.Scanner;  
  
public class Demo6 {  
    public static void main(String[] args) {  
        int floor = 6;  
        if(floor ==1){ //if 如果  
            System.out.println("到了第一层");  
        }else if(floor ==2){ //else if 再如果  
            System.out.println("到了第二层");  
        }else if (floor ==3){  
            System.out.println("到了第三层");  
        }else{  
            System.out.println("没有你想到楼层了");  
        }  
    }  
}
```

● 选择结构 if 语句格式 3 练习 (掌握)

■ A: 练习 1

■

需求：键盘录入一个成绩，判断并输出成绩的等级。

90-100 优

80-89 良

70-79 中

60-69 及

0-59 差

```
import java.util.Scanner;

public class Demo6 {
    public static void main(String[] args) {
        /*
            需求：键盘录入一个成绩，判断并输出成绩的等级。
            90-100 优
            80-89 良
            70-79 中
            60-69 及
            0-59 差
        */
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入成绩：");
        int score = sc.nextInt();
        if(score >=90 && score <=100){
            System.out.println("优");
        }else if(score >=80 && score <= 89){
            System.out.println("良");
        }else if(score >= 70 && score <=79){
            System.out.println("中");
        }else if(score >= 60 && score <=69){
            System.out.println("及格");
        }else if(score >=0 && score <= 59){
            System.out.println("差");
        }else{
            System.out.println("你输入的有误！");
        }
    }
}
```

■ B:练习 2

■ 需求：

- 键盘录入 x 的值，计算出 y 的值，并输出。

- $x \geq 3$ $y = 2 * x + 1;$
- $-1 < x < 3$ $y = 2 * x;$
- $x \leq -1$ $y = 2 * x - 1;$

```
import java.util.Scanner;

public class Demo6 {
    static int y; //这个在方法外，类中的变量称为全局变量，成员变量，它的有效范围是整个类中，底层会自动给它赋予默认值int: 0, float\double:0.0;boolean:false,String:null
    public static void main(String[] args) {
        /*
        ▪ x>=3      y = 2 * x + 1;
        ▪ -1<x<3    y = 2 * x;
        ▪ x<=-1     y = 2 * x - 1;
        */
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个整数: ");
        int x = sc.nextInt();
        //      int y = 0; //在方法内声明的变量称为局部变量，局部变量在使用之前，必须赋值，它的有效范围在方法内
        if(x >= 3){
            y = 2*x+1;
        }else if(x>-1 && x<3){
            y = 2*x;
        }else if(x<=-1){
            y = 2*x-1;
        }
        System.out.println("y="+y);
    }
}
```

●选择结构 if 语句的嵌套使用(掌握)

- A:案例演示
 - 需求：获取三个数据中的最大值
 - if 语句的嵌套使用。

●选择结构 switch 语句的格式及其解释(掌握)

- A:switch 语句的格式

```
switch(表达式) {  
    case 值 1: //case: 如果  
        语句体 1;  
        break;  
    case 值 2:  
        语句体 2;  
        break;  
    ...  
    default: //缺省等同于 else  
        语句体 n+1;  
        break;  
}
```

- B:switch 语句的格式解释
- C:面试题
 - byte 可以作为 switch 的表达式吗?
 - long 可以作为 switch 的表达式吗?
 - String 可以作为 switch 的表达式吗?
- C:执行流程
 - 先计算表达式的值
 - 然后和 case 后面的匹配, 如果有就执行对应的语句, 否则执行 default 控制的语句

```
public class Demo7 {  
    public static void main(String[] args) {  
        int floor=7;  
        switch (floor){  
            case 1:  
                System.out.println("到了第一层");  
                break;//中止  
            case 2:  
                System.out.println("到了第二层");  
                break;  
            case 3:  
                System.out.println("到了第三层");  
                break;  
            case 4:
```

```

        System.out.println("到了第四层");
        break;
    default: //处理以上都不满足的情况
        System.out.println("没有你想去的楼层");
        break;
    }
}
}

```

```

public class Demo7 {
    public static void main(String[] args) {
        int floor=3;
        switch (floor){
            case 1:
                System.out.println("到了第一层");
                break; //中止, 如果没有break, 那么会发生穿透的现象, 这个break 一定不能丢
            case 2:
                System.out.println("到了第二层");
                break;
            case 3:
                System.out.println("到了第三层");
                break;
            case 4:
                System.out.println("到了第四层");
                break;
            default: //处理以上都不满足的情况, 一般情况下 default 放在最后写, default 的执行顺序是
                所有的 case 之后
                System.out.println("没有你想去的楼层");
                break; //如果 default 在最后, 这个break 可以去掉
        }
    }
}

```

● 选择结构 switch 语句的练习 (掌握)

■ A: 案例演示 1

整数 (给定一个值, 输出对应星期几)

```

import java.util.Scanner;

public class Demo7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个星期数: ");
    }
}

```



```

int week = sc.nextInt();
switch (week){
    case 1:
        System.out.println("周一");
        break;
    case 2:
        System.out.println("周二");
        break;
    case 3:
        System.out.println("周三");
        break;
    case 4:
        System.out.println("周四");
        break;
    case 5:
        System.out.println("周五");
        break;
    case 6:
        System.out.println("周六");
        break;
    case 7:
        System.out.println("周日");
        break;
    default:
        System.out.println("输入有误，请重新输入！");
}
}
}

```

```

import java.util.Scanner;

public class Demo7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一个月份数：");
        int week = sc.nextInt();
        switch (week){ //switch 的括号内可以接受的数据类型：byte\short\int\long\char\String\枚
            case 3:
            case 4:
            case 5:
                System.out.println("春暖花开");
                break;
            case 6:
            case 7:
            case 8:

```

```
        System.out.println("夏日清爽");
        break;
    case 9:
    case 10:
    case 11:
        System.out.println("秋高气爽");
        break;
    case 12:
    case 1:
    case 2:
        System.out.println("冻死个人");
    default:
        System.out.println("输入有误, 请重新输入! ");
    }
}
```

switch 使用予非范围判断的控制条件中
如果是在一个范围内, 优先选用 if_ else if_ else

● 选择结构 switch 语句的注意事项(掌握)

▪ A: 案例演示 2

- a: case 后面只能是常量, 不能是变量, 而且, 多个 case 后面的值不能出现相同的
- b: default 可以省略吗?
 - 可以省略, 但是不建议, 因为它的作用是对不正确的情况给出提示。
 - 特殊情况:
 - case 就可以把值固定。
 - A, B, C, D
- c: break 可以省略吗?
 - 最后一个可以省略, 其他最好不要省略
 - 会出现一个现象: case 穿透。
 - 最终我们建议不要省略
- d: default 一定要在最后吗?
 - 不是, 可以在任意位置。但是建议在最后。

- e:switch 语句的结束条件
 - a:遇到 break 就结束了
 - b:执行到 switch 的右大括号就结束了

• 举例：皇帝翻牌子

```
import java.util.Scanner;

public class Demo7 {
    public static void main(String[] args) {
        String feizi = "李嬷嬷";
        switch (feizi){
            case "张贵妃":
                System.out.println("她老爸是将军，好好努力！");
                break;
            case "李嬷嬷":
                System.out.println("李嬷嬷比较凶，会扎针，小心点");
                break;
            case "陈皇后":
                System.out.println("琵琶弹的好，温柔体贴");
                break;
            case "张爱妃":
                System.out.println("做的糕点好吃");
                break;
            default:
                System.out.println("没有抽到，太高兴了，终于可以休息一天了");
        }
    }
}
```

● 选择结构 switch 语句练习 (掌握)

- A:看程序写结果:

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

        int x =2;
        int y =3;
        switch(x){
            default:
                y++;
                break;
        }
    }
}
```

```
        case 3:
            y++;
        case 4:
            y++;
    }
    System.out.println("y="+y);
}
}
```

B:看程序写结果:

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

        int x =2;
        int y =3;
        switch(x){
            default:
                y++;
            case 3:
                y++;
            case 4:
                y++;
        }
        System.out.println("y="+y);
    }
}
```

●选择结构 if 语句和 switch 语句的区别(掌握)

- A:总结 switch 语句和 if 语句的各自使用场景
- switch 建议判断固定值的时候用
- if 建议判断区间或范围的时候用
- B:案例演示
 - 分别用 switch 语句和 if 语句实现下列需求:
 - 键盘录入月份, 输出对应的季节