

Java核心_流程控制之选择结构

课程概要

[Scanner的基本使用](#)

[流程控制结构概述](#)

[流程控制结构之选择结构](#)

[选择结构if语句的三种格式](#)

[选择结构switch语句](#)

学习目标

能够根据步骤使用Scanner实现键盘录入功能

理解流程控制结构的概念和分类

能够使用if语句的三种格式控制代码的执行流程

能够按照基本格式正确定义switch语句

能够使用switch语句基本格式控制代码的执行流程

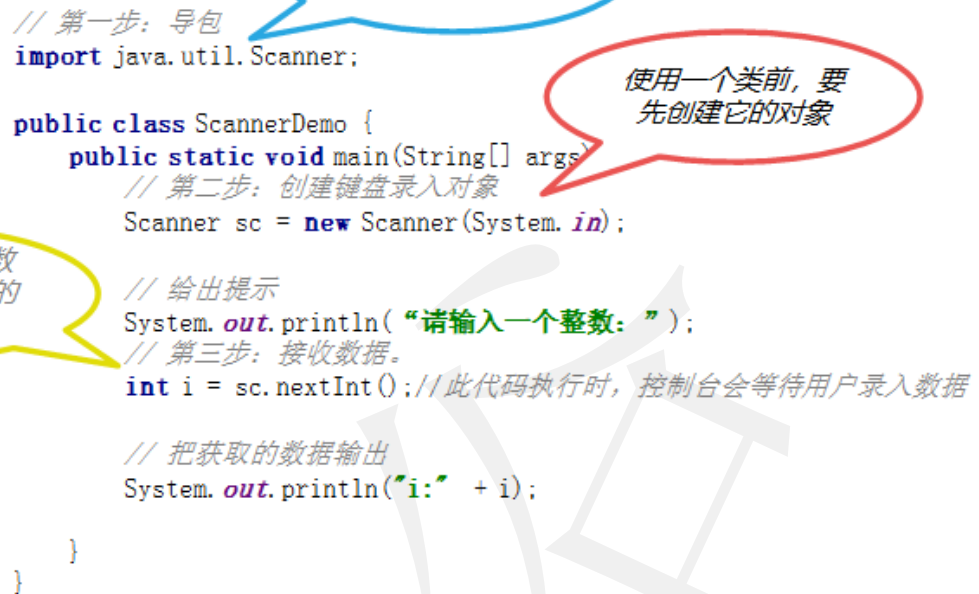
Scanner的基本使用

Scanner的概念

扫描器。即可以通过Scanner类扫描用户在控制台录入的数据

键盘录入数据的步骤

1. 导包
2. 创建键盘录入对象
3. 接收数据



```
// 第一步：导包
import java.util.Scanner;

public class ScannerDemo {
    public static void main(String[] args) {
        // 第二步：创建键盘录入对象
        Scanner sc = new Scanner(System.in);

        // 给出提示
        System.out.println("请输入一个整数：");
        // 第三步：接收数据。
        int i = sc.nextInt(); // 此代码执行时，控制台会等待用户录入数据

        // 把获取的数据输出
        System.out.println("i:" + i);
    }
}
```

使用Scanner类，需要先导包

使用一个类前，要先创建它的对象

接收用户输入的数据并赋值给int型的变量i

需求：键盘录入两个整数，求它们的和

实现键盘录入功能的步骤：

第一步：导包

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

第二步：创建对象

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

第三步：接收数据

```
int a = sc.nextInt();
```

流程控制结构概述

流程控制结构的概念

程序的结果跟代码的执行顺序紧密相关，通过使用一些特定的语句控制代码的执行顺序从而完成特定的功能，这就是流程控制结构。

流程控制结构的分类

- 顺序结构
- 选择（判断）结构

- 循环结构

顺序结构的概念

即程序按照代码的顺序从上往下、从左往右依次执行的代码结构。

顺序结构是最简单的流程控制结构，不需要特定的语句，程序的大多数代码都是这样执行的。

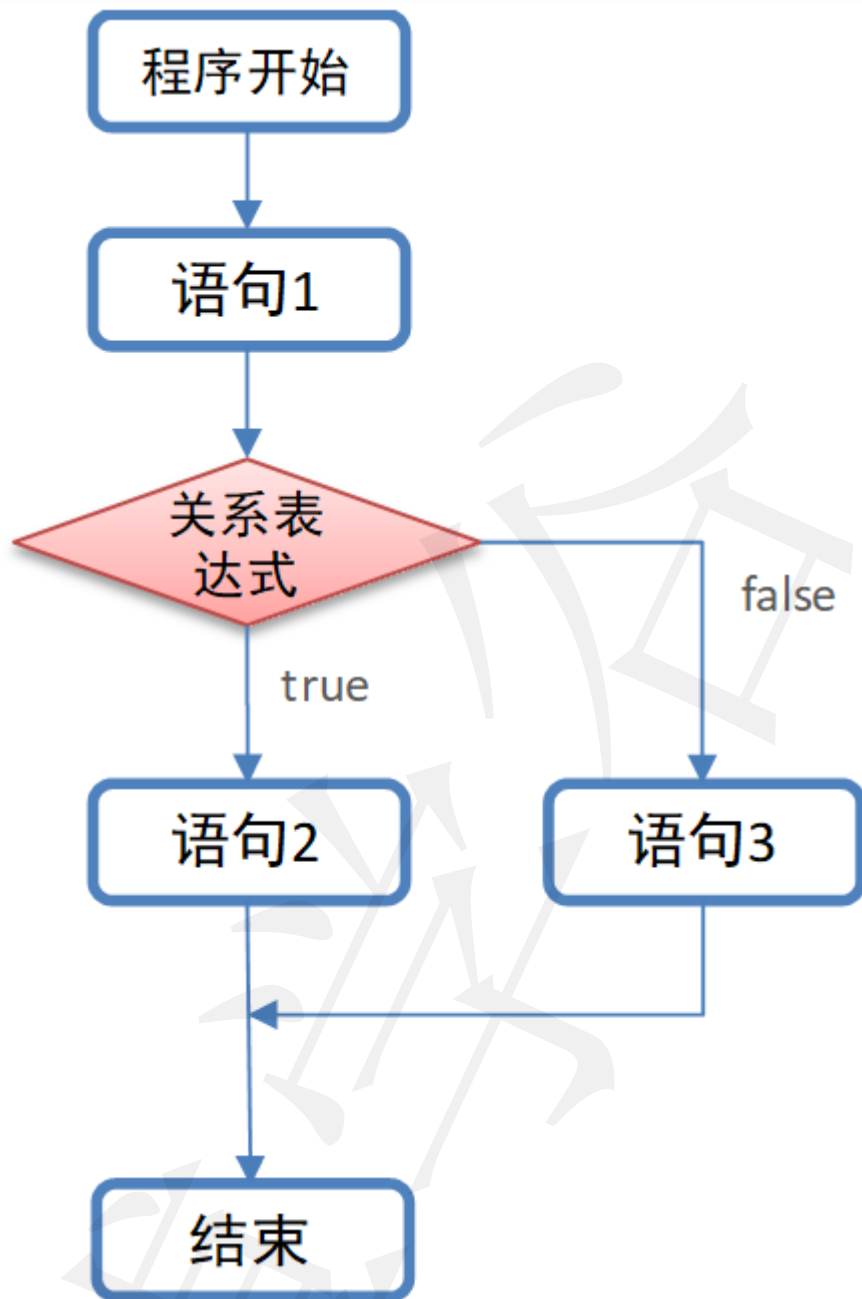


流程控制结构之选择结构

选择结构的概念

根据不同的条件选择不同的代码来执行的代码结构。

选择结构要先判断条件是否成立，条件成立则执行一部分代码，不成立则执行另外一部分或结束代码，所以选择结构也叫**判断结构**。



选择结构的分类

if语句（共有三种格式）：如果

if一般用做区间值的判断，比如 当前时间为0~8点，小黑和你说“早上好”； 当前时间为9~12点，小黑和你说“中午好”；

switch语句：转换，切换

switch一般用做固定值的判断，比如 你说“星期一”，小黑说“Monday”； 你说“星期五”，小黑说“Friday”；

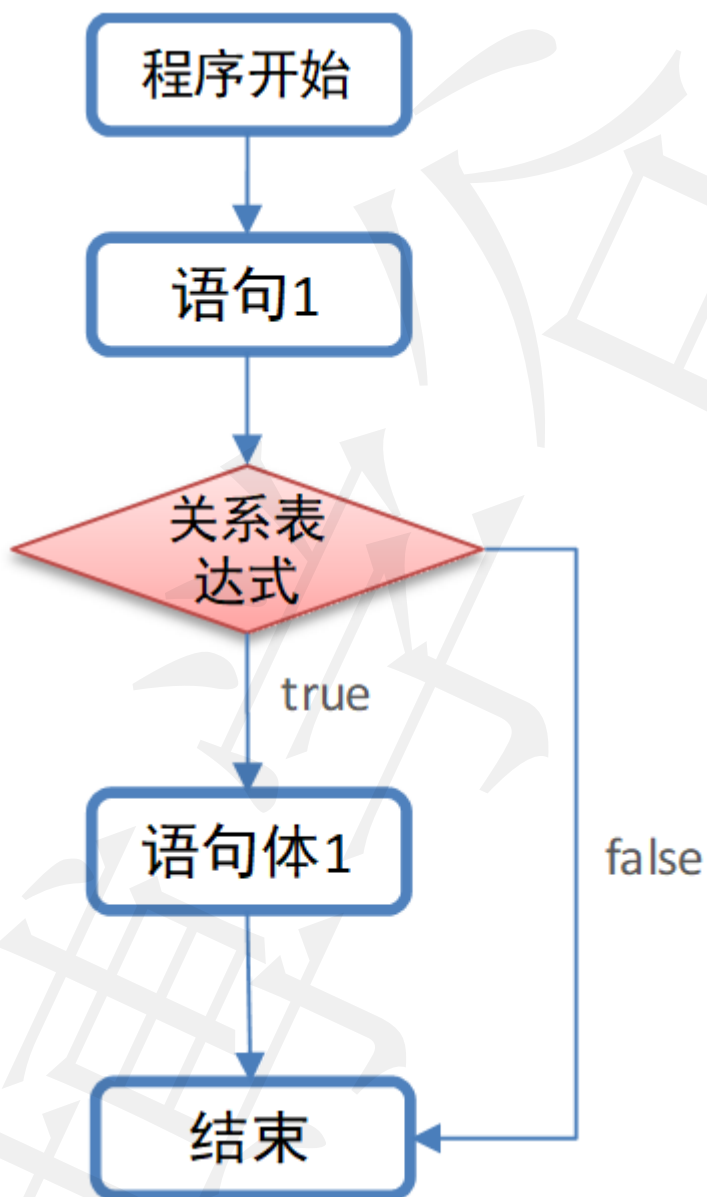
选择结构if语句的三种格式

选择结构if的第一种格式

if语句（共有三种格式）：如果

```
// 第一种格式：  
if(关系表达式) {  
    // 语句体  
}
```

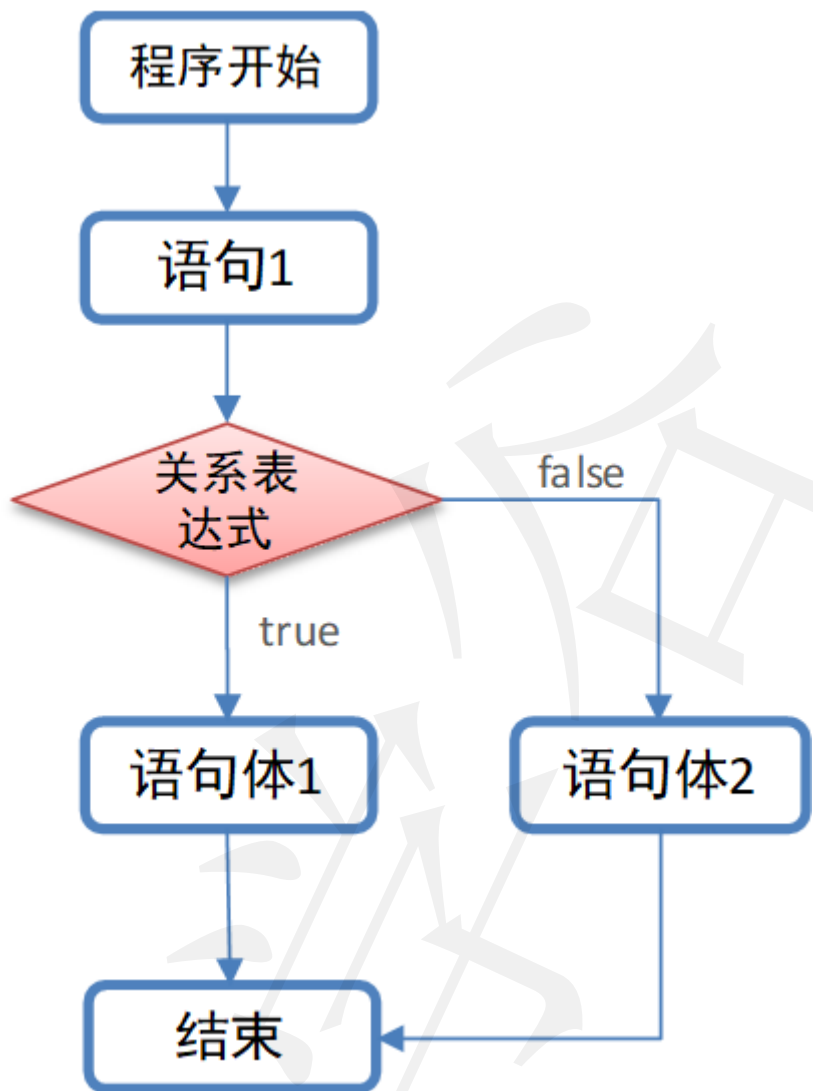
执行流程：



选择结构if的第二种格式

```
// 第二种格式：  
if(关系表达式) {  
    // 语句体1  
} else {  
    // 语句体2  
}
```

执行流程：

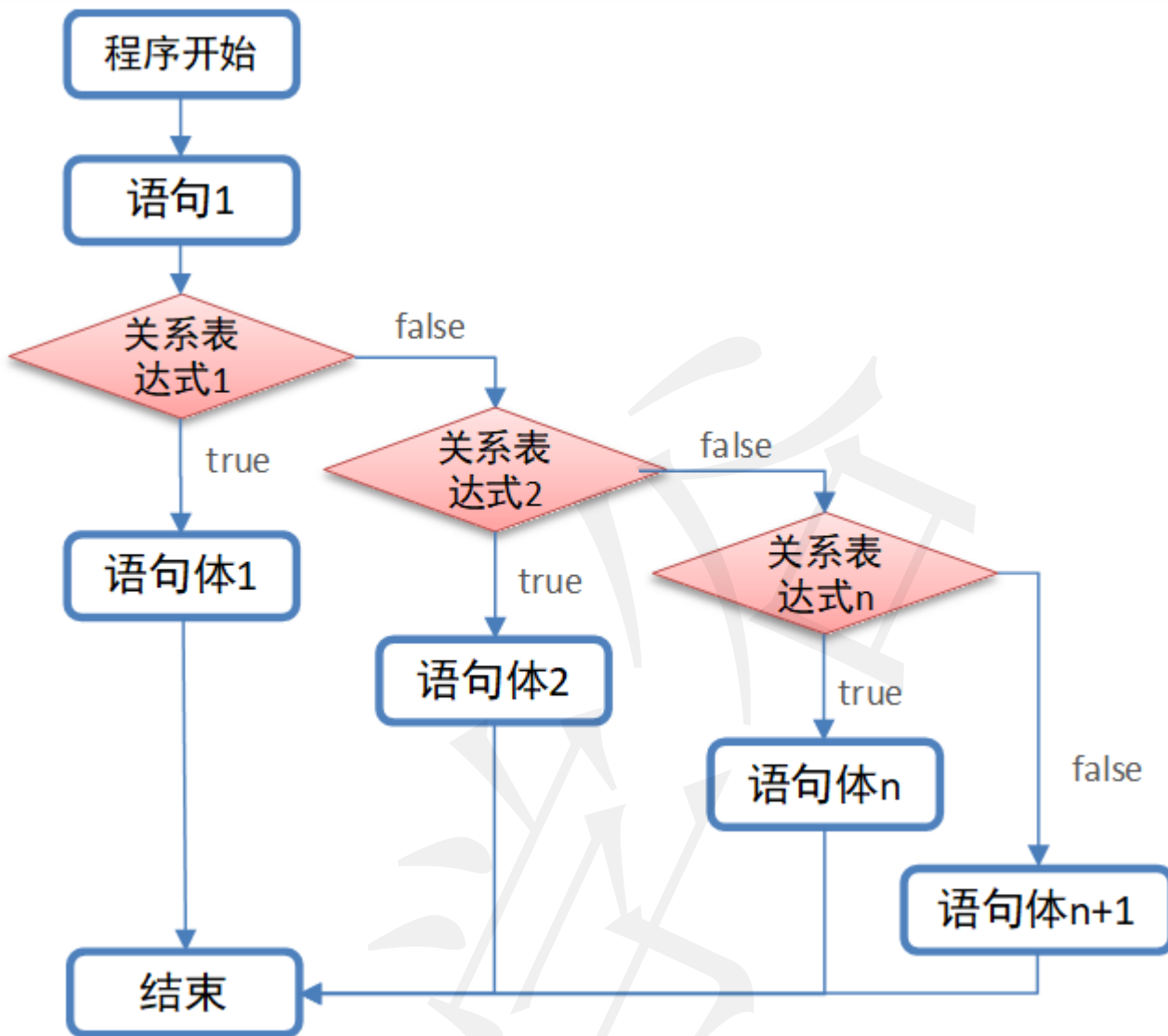


选择结构if的第三种格式

```

// 第三种格式:
if(关系表达式1) {
    // 语句体1
} else if(关系表达式2) {
    // 语句体2
} // ...
else {
    // 语句体n+1
}
  
```

执行流程：



if语句案例之获取两个整数的较大值

需求：键盘录入两个数据，获取这两个数据的较大值

分析：

A：键盘录入的三个步骤：

导包；创建键盘录入对象；获取数据；

B：获取两个数据的较大值：if语句的格式2实现

C：把较大的结果输出

三种导包方式：

A：手动导包

B：点击鼠标自动生成

C：快捷键：Ctrl + Shift + O

```
//定义一个变量用于保存较大的值
int max;
if(a > b) {
    max = a; // 变量a是第一个录入变量
} else {
    max = b; // 变量b是第二个录入变量
}
System.out.println("较大的值是：" + max);
```

if语句案例之根据学生成绩输出对应级别

需求：键盘录入学生考试成绩，请根据成绩判断该学生属于哪个级别

- * 90-100 皇帝
- * 80-90 宰相
- * 70-80 大臣
- * 60-70 县官
- * 60以下 草民

分析：

- A：回顾键盘录入数据的步骤
- B：分数段较多，该使用if语句的格式3进行判断
- C：根据判断结果输出对应的级别

小贴士：

写程序的时候，做数据测试，应该测试这样的几种情况：

正确数据

错误数据

边界数据

选择结构switch语句

switch语句的格式


```
switch(表达式) {  
    case 值1:  
        语句体1;  
        break;  
    case 值2:  
        语句体2;  
        break;  
    // ...  
    default:  
        语句体n+1;  
        break;  
}
```

格式含义解释

```
switch(表达式) {  
    case 值1:  
        语句体1;  
        break;  
    case 值2:  
        语句体2;  
        break;  
    ...  
    default:  
        语句体n+1;  
        break;  
}
```

switch(表达式)

表达式的取值类型: byte, short, int, char
JDK5以后可以是枚举
JDK7以后可以是String

case 值1, case 值2

case后面跟的是要和表达式进行比较的值

语句体1, 语句体2, 语句体n+1

语句体部分可以是一条或多条语句

break

表示中断, 结束的意思, 可以结束switch语句

default

表示所有情况都不匹配的时候, 就执行该处的内容, 和if语句的else相似

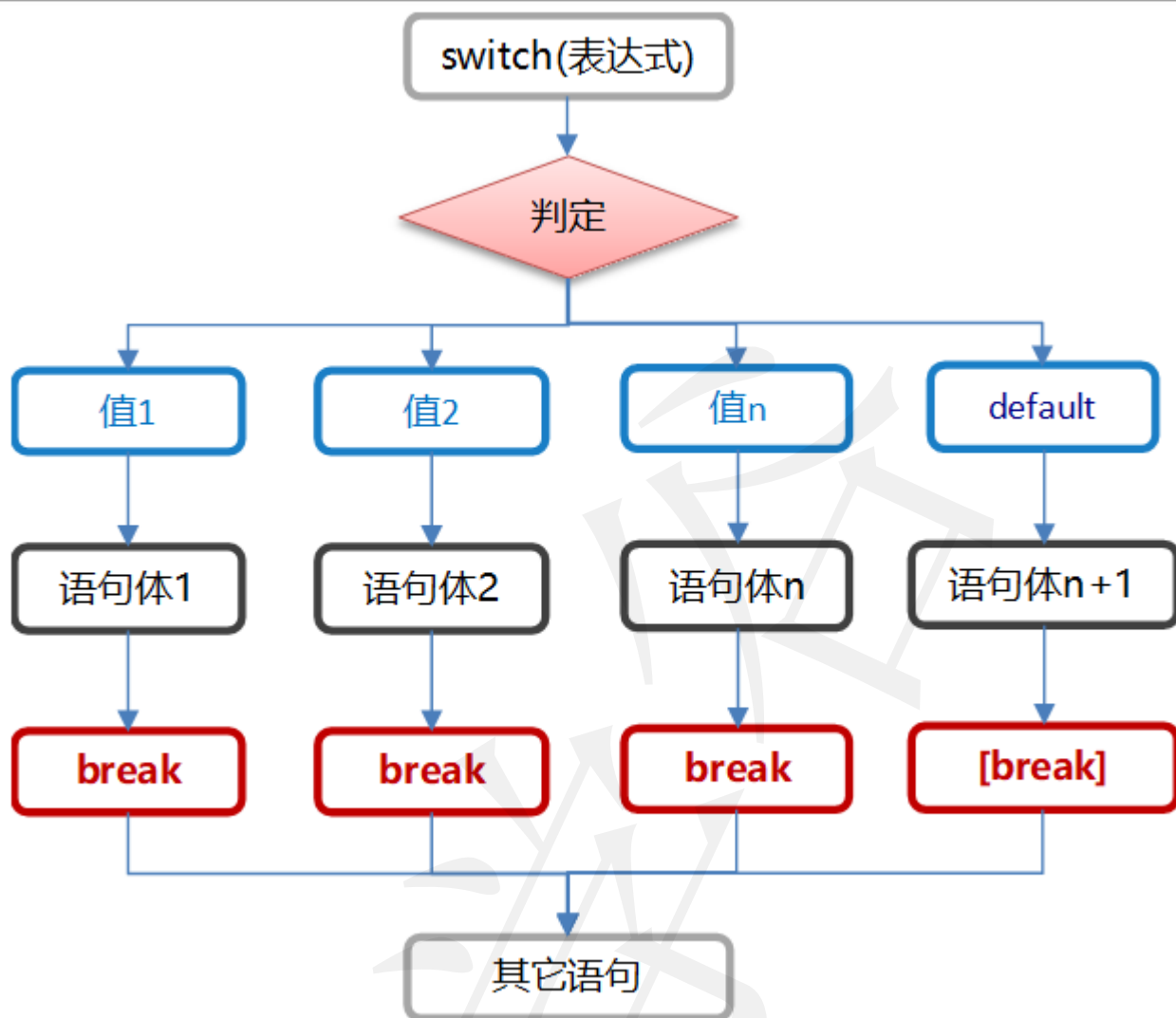
小贴士:

枚举指的是一系列数目可数的数据。比如每周的星期, 每年的月份。

小贴士:

String即字符串。双引号中的内容都是String类型。

switch语句的执行流程



switch语句的案例

需求：根据键盘录入的数字（1-7），输出对应的星期

快捷键：代码格式化 Ctrl + Alt + L

分析：

- A：使用键盘录入功能获取用户输入的数字
- B：switch的表达式为接收用户录入数据的变量
- C：分别用1-7作为case的值，并输出对应的星期
- D：当输入数据有误时，在default中输出提示

效果图如下：

1. 根据输入数字判断星期

1



星期一



2. 如果输入数字不是1-7之间，则提示不知道是星期几

8



不知道

